國 立 交 通 大 學

電信工程研究所

博 士 論 文

超大型積體電路的熱分析技術
Thermal Simulation Techniques for Very Large Scale Integration (VLSI) Circuits

研究生：黃培育

指導教授：李育民

中 華 民 國 一百 年 十 月

超大型積體電路的熱傳分析技術

Thermal Simulation Techniques for Very Large Scale Integration (VLSI) Circuits

研究生：黃培育　　　　　Student: Pei-Yu Huang

指導教授：李育民　博士　　　　Advisor: Dr. Yu-Min Lee

國立交通大學

電信工程研究所

博士論文

A Dissertation
Submitted to Institute of Communication Engineering
College of Electrical and Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in
Communication Engineering
Hsinchu, Taiwan

中　華　民　國 100 年 10 月

超大型積體電路的熱傳分析技術

學生: 黃培育　　　　　　　　　　　指導教授:李育民

國立交通大學電信工程所博士班

摘　　要

持續縮小元件大小的互補式金氧半導體製程技術造成了在晶片上的高功率密度。這個事實導致在超大型積體電路上有很高的晶片溫度。晶片溫度將會影響到電路效能以及可靠度。此外，亦會增加電路的消耗功率。因此，許多研究者已致力於發展將溫度視為導向之一的電路最佳化以及效能分析技術。由於將熱傳視為導向的最佳化引擎需要在最佳化過程中執行許多次的熱傳分析，因此在以熱傳為導向之一的設計流程中需要一個準確且快速的熱分析器。為了提供前端設計流程的熱傳分析，此博士論文中發展了三個準確且快速的熱傳分析器。

給定了晶片上的功率分布之後，第一個分析器首先利用一組基底來表示晶片上的溫度。得到晶片溫度的表示式之後，我們發展了一個基於快速傅立葉轉換的演算法來計算出晶片的溫度分佈。基於以上的分析架構，第一個分析器也提供了堆疊晶片(stacked package)以及無接觸連線(contactless interconnection)架構的三維度積體電路之熱傳分析功能。

為了考慮製程變異以及溫度對於漏電功率的影響，第二個分析器提供了兩種熱電分析架構以快速且準確地估計晶片溫度的擾動。此外，為了提供更有意義的熱傳丈量尺度給前端設計之最佳化引擎，第二個分析器也準確且快速地提供了晶片上熱傳可靠度分布圖(thermal yield profile)。在此，熱傳可靠度分布圖為晶片溫度小於一個使用者給定之臨界溫度的機率分布圖。

為了提供以矽穿孔(through silicon via)技術為架構的三維度積體電路之熱傳分析，第三個分析器提供了一個基於查表法的分析架構。利用此查表法的分析架構，耗時的熱傳電導矩陣處理過程將可以被避免。

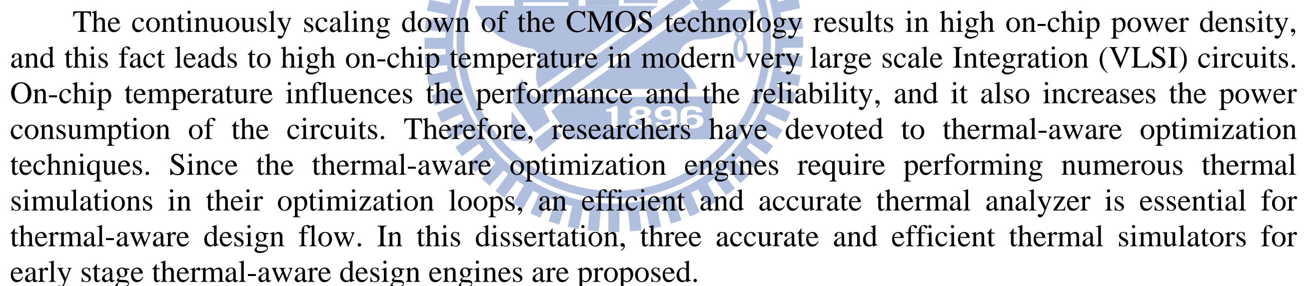
我們的實驗已驗證了以上三個熱傳分析器具有高度的估計準確率以及分析效能。

Thermal Simulation Techniques for Very Large Scale Integration (VLSI) Circuits

Student: Pei-Yu Huang   Advisor: Yu-Min Lee

Institute of Communication Engineering
National Chiao Tung University

ABSTRACT



The continuously scaling down of the CMOS technology results in high on-chip power density, and this fact leads to high on-chip temperature in modern very large scale Integration (VLSI) circuits. On-chip temperature influences the performance and the reliability, and it also increases the power consumption of the circuits. Therefore, researchers have devoted to thermal-aware optimization techniques. Since the thermal-aware optimization engines require performing numerous thermal simulations in their optimization loops, an efficient and accurate thermal analyzer is essential for thermal-aware design flow. In this dissertation, three accurate and efficient thermal simulators for early stage thermal-aware design engines are proposed.

Given the deterministic on-chip power profile, the first simulator represents the on-chip temperature profile by a set of bases. Then, a fast Fourier transform based algorithm is developed to obtain the on-chip temperature profile. Based on the above simulation framework, the first proposed simulator also provides the thermal simulation for the stacked-chip or the contactless interconnection based three-dimensional integrated circuits (3-D ICs).

To take into account the impacts of the process variation and the temperature to leakage powers, the second simulator provides two electro-thermal simulation frameworks to accurately and efficiently predict the fluctuation of on-chip temperature profile. Moreover, to ensure the on-chip thermal reliability and provide more meaningful thermal costs for thermal-aware design engines, the second simulator can efficiently deliver the thermal yield profile, which is the probability profile of the temperature being less or equal to a user specified threshold temperature.

To provide the thermal estimation for early stage thermal-aware design engines for the through silicon via based 3-D ICs, the third proposed simulator provides a look-up table based thermal simulation framework. With the look-up table based framework, the time consumed dealing processes for the thermal conductance matrix of the equivalent thermal circuit can be avoided.

The experimental results have demonstrated the high-accuracy and high-efficiency of all the three proposed thermal simulators.

誌　謝

這篇論文能夠順利地完成，首先要感謝我的指導教授 李育民博士，當我遭遇困難時，老師總會適時指引我方向，讓我能夠繼續前進。在攻讀博士期間，老師給我對於專業上的訓練指導以及在論文寫作方面夙夜匪懈的幫助，讓我確實感受到老師對於自己的用心。我在此對你致上最高的謝意。

論文中實驗的部份，特別感謝李義明老師提供我們台積電65奈米製程的製程參數，標準元件檔以及標準元件的電路檔。除了李義明老師，在此特別感謝工業技術研究院蒯定明技術組長以及林昌賜技術副理。兩位所提供的測試電路以及三維度晶片相關的工業規格，讓此論文中對於三維度晶片相關的實驗能夠更符合工業標準。沒有你們的幫助，要完成論文中實驗的部分是相當困難的。

在實驗室成員裡，感謝麒文、佳鴻、懷中、以及啟平對於部分實驗的幫助。沒有你們，要獨力完成整個實驗是相當困難的。

特別感謝好友羿辰，給予我許多支持與鼓勵，陪伴我度過最低潮的日子。最後要深深地感謝我的父母，你們對於我在學業上衝刺的全力支持，讓我能夠順利完成博士學業。僅在此將本論文獻給你們，共享這份喜悅與榮耀。

Contents

1 Introduction **1**
1.1 Thermal Related Issues of Modern VLSI Design . . . 1
1.1.1 Timing and Reliability Issues . . . 1
1.2 The Effects of Temperature and Process Variations on the Leakage Powers . . . 4
1.2.1 Subthreshold Leakage Current . . . 4
1.2.2 Gate Tunneling Leakage Current . . . 6
1.2.3 Leakage Powers Inducing Electro-Thermal (Temperature-Power) Coupling Effect . . . 7
1.2.4 Variations of Physical Device Parameters . . . 8
1.3 Three-Dimensional Integrated Circuit and Its Thermal Issues . . . 9
1.4 Thermal-Aware Design Flow . . . 12
1.5 Review of On-Chip Thermal Simulation Methods . . . 14
1.5.1 Simulation Methods of Deterministic On-Chip Temperature Profile . . 14
1.5.2 Simulation Methods of Statistical On-Chip Temperature Profile . . . 16
1.5.3 Simulation Methods of the Temperature Profile for 3-D ICs . . . 18
1.6 Contributions of this Dissertation . . . 19
1.7 Organization of this Dissertation . . . 21

2 Simulation Method I – ***Full-Chip Thermal Analysis for Early Design Stages via Generalized Integral Transforms*** **22**
2.1 Thermal Modeling for Early Design Stages . . . 22
2.2 Full-Chip Thermal Simulation . . . 28
2.2.1 Auxiliary Problem for Generating Appropriate Spatial Bases . . . 29
2.2.2 System Transformation for Time-Varying Coefficients . . . 30
2.2.3 Average Rising Temperature Evaluation of Grid Cells . . . 31
2.3 Thermal Simulation for Stacked-Layer 3-D ICs . . . 41
2.4 Approach to Handle the Temperature Dependent Issue of Leakage Powers . . . 44
2.5 Experimental Results . . . 45
2.5.1 Accuracy and Fast Convergence of the GIT Based Thermal Simulator . 45
2.5.2 Thermal Simulation for the Full-Chip Containing Lots of Functional Blocks . . . 46
2.5.3 Accuracy and Efficiency of the GIT Based Thermal Simulator for the 3-D IC Thermal Analysis . . . 48

3 Simulation Method II – ***An Efficient Method for Analyzing the Process Variations Considered On-Chip Thermal Reliability*** **52**
3.1 Motivation Illustrations . . . 52
3.1.1 Electro-Thermal Coupling Issue under Process Variations . . . 52
3.1.2 Concept of On-Chip Thermal Yield Profile . . . 54

3.2 Preliminaries . . . 55
3.2.1 Leakage Power Modeling . . . 55
3.2.2 Modeling of Variations for Physical Device Parameters . . . 59
3.2.3 Problem Formulation . . . 61
3.3 Statistical Electro-Thermal Analyzer . . . 64
3.3.1 Stochastic Projection Based Statistical Expression Generator . . . 67
3.3.2 Stochastic Collocation Based Statistical Expression Generator . . . 78
3.3.3 Implementation of the Deterministic Electro-Thermal Simulation . . . 84
3.3.4 On-Chip Thermal Yield Computation . . . 85
3.3.5 Mixed-Mesh Thermal Yield Estimation . . . 89
3.4 Experimental Results . . . 91
3.4.1 Statistical Thermal Simulations With/Without Considering Electro-Thermal Effects . . . 93
3.4.2 Accuracy and Efficiency . . . 93

4 Simulation Method III – *LUTSim: A Look-Up Table Based Thermal Simulator for 3-D ICs* **103**
4.1 Thermal Model for Early Design Stages of TSVs based 3-D IC Structures . . . 104
4.2 LUTSim . . . 107
4.2.1 Overview of LUTSim . . . 107
4.2.2 Recursive Look-Up Table based Full-Chip Thermal Simulation Framework . . . 109
4.2.3 Fine-Mesh Table Establishment . . . 111
4.2.4 Double-Mesh Table Establishment . . . 116
4.3 Experimental Results . . . 118
4.3.1 Experimental Settings . . . 118
4.3.2 Validation . . . 119
4.3.3 Robustness Verification . . . 119

5 Conclusion **123**
5.1 Summary of Current Research Results . . . 123
5.2 Future Research Directions . . . 125
5.2.1 Statistical Thermal Simulation of 3-D ICs . . . 125
5.2.2 Thermal-aware Timing Analysis . . . 125
5.2.3 Thermal-aware Design Engines . . . 125

A Derivation of the time-varying coefficients for GIT based thermal simulation method and error bound analysis of GIT based steady state temperature formulae **126**
A.1 Derivation of the Analytical Expression of Time-Varying Coefficients for the Approximated Temperature . . . 126
A.2 Error Bound Analysis of GIT Based Steady State Temperature Formulation . . . 127

B Derivation of the Projection Coefficients of Subthreshold and Gate Tunneling Leakage Powers **130**
B.1 Derivation of the Evaluating Algorithm for the Projection Coefficient of Gate Tunneling Leakage Power . . . 130
B.1.1 Derivation of Evaluating Algorithm for the Projection Coefficient of Subthreshold Leakage Power . . . 133

C Proof of Proposition 1 **134**

List of Figures

1.1 The temperature dependences for the delays of gates and interconnects. (a) The temperature dependency of a NAND gate at the 65nm technology node. (b) The temperature dependences for the delay of interconnect corresponding to different wire length L (reprinted from [1]). 3
1.2 An example for illustrating the non-uniform on-chip temperature profile inducing clock skew. 4
1.3 The temperature and process variation dependencies of subthreshold and gate tunneling leakage currents of a NAND gate at 65nm technology node. (a) The temperature and process variation dependencies of subthreshold leakage current. (b) The temperature and process variation dependencies of gate tunneling leakage current. Here, L is the device channel length, and its unit is nm. t_{ox} is the oxide thickness, and its unit is $\mathring{A}m$. 5
1.4 The mechanism of electro-thermal coupling. 8
1.5 Parameter variations impacts on the leakage currents (reprinted from [2]). Here, the y-coordinate indicates the normalized occurrence frequency of the value of the device channel length, and the x-coordinate indicates the normalize value for the subthreshold leakage currents (I_{sb}). 9
1.6 Implementation categories of 3-D ICs (reprinted from [3]). (a) Wire-bonded structure. (b) Microbump-3D package structure. (c) Face-to-face structure. (e) Contactless-capacitive with buried bumps structure. (f) Contactless-inductive structure. (f) Through silicon vias (TSVs) based structure with silicon substrates on bulk. (g) TSVs based structure with silicon substrates on insulator (SOI). . 10
1.7 The wafer bonding process of the TSVs based 3-D IC (reprinted from [4]). . . . 11
1.8 Temperature-Aware Design Flow 13

2.1 Compact thermal model for early design stages. 23
2.2 Energy conservation law and the heat conduction equation. 24
2.3 The 1-D thermal model for estimating the roughly steady state average temperature of die. The modeled thermal resistance network is shown in the right hand side. 26
2.4 The executing flow of the proposed GIT based thermal simulation method. . . . 28
2.5 The overview of using *2D-SLT-FFT* and *2D-LTS-FFT* to evaluate the average rising temperature of grid cells. 34
2.6 Procedure of *1D-STL-FFT*. 35
2.7 The sketch of the computational flow for *1D-STL-FFT* with $\widetilde{M} = 8$ and $M = 16$. 35
2.8 Procedure of *1D-LTS-FFT*. 36
2.9 The sketch of the computational flow for *1D-LTS-FFT* with $\widetilde{M} = 8$ and $M = 16$. (a) The *1D-LTS-FFT*. (b) The *1D-LTS-FFT* for negative frequencies. 37
2.10 Procedure of *2D-STL-FFT*. 38

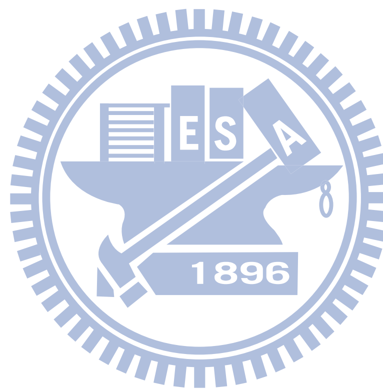
2.11 Simulating algorithm of the proposed steady state thermal simulator. 40
2.12 The schematic diagram of a 3-D IC with N_l chip layers. 41
2.13 Temperature-power iterative framework for dealing with the temperature dependence issue of leakage power. 44
2.14 Accuracy and the maximum error trend of a test chip. (a) Floorplan, (b) geometries of the test chip, (c) power distribution, (d) the rising temperature distribution of the top surface of the die, (e) the relative error distribution, and (f) the maximum relative error versus truncation point. 47
2.15 The power density and temperature distribution of a 1 cm $\times$ 1 cm chip with one million functional blocks. (a) The power density distribution, and (b) the rising temperature distribution. 49
2.16 Power density and temperature distribution of a test 3-D chip. Figures (a), (c) and (e) are the power density profiles on the top surface of the top, middle and bottom silicon layers, respectively. Figures (b), (d) and (f) are the temperature distribution on the top surface of the top, middle and bottom silicon layers, respectively. 51

3.1 An example for the electro-thermal coupling mechanism under process variations. 53
3.2 PDFs of on-chip temperature values at two different positions (B and R) of a die for indicating which one is the *statistically hot-spot* location. 54
3.3 Compact thermal model of physical design stages under process variations. . . 62
3.4 An iterative scheme for computing the appropriate thermal conductivity of die. μ_T is a roughly average mean temperature of die, and μ_P is the mean of total on-chip power consumption after executing an iteration. μ_P can be obtained by the zeroth order of H-PC projected power of gates proposed in Figure 3.7 and Figure 3.9 of section 3.3.1. 63
3.5 Overview of the developed statistical electro-thermal analyzer. 65
3.6 The electro-thermal updating scheme of the stochastic projection based statistical expression generator. 70
3.7 The evaluating algorithm of projection coefficients of gate tunneling leakage power up to second order of HPs. 73
3.8 The function that evaluates the related vectors for calculating the leakage powers. 74
3.9 Subthreshold leakage power projection algorithm. 75
3.10 Stochastic projection based electro-thermal analysis algorithm. *NumGateType* in *Line* 13 is the number of gate types given from the industrial library file. . . . 77
3.11 The number of sampling random variables comparison between the Monte Carlo method and the Smolyak sparse grid formulation. Here, the samples of Smolyak sparse grid are adopted for achieving a level two approximation. 79
3.12 Deterministic electro-thermal analysis for each sampling point, $\boldsymbol{\xi}^j$, in sparse grid. p_{leak}, p_d and p are the leakage, dynamic and total power density profiles for each sampling point, respectively. 81
3.13 Stochastic Collocation Based Statistical Expression Generating Algorithm. . . 83
3.14 Implementation of solving the deterministic heat transfer equations. 84
3.15 Weighted sum of two independent non-central chi-square random variables. *Case1*: the skewness of the PDF decreases because a left-skewed distribution and a right-skewed distribution are moving integrated. *Case2*: the skewness of the PDF increases because two right-skewed distributions are moving integrated. 87
3.16 The executing sketch of the mixed-mesh thermal yield estimation. 89

3.17 Floorplan of the test die, geometries of the test chip and package, and mean and standard deviation profiles of the power density on the test chip. (a) Floorplan of the test die. (b) Geometries of the test chip and package. (c) The mean profile of power density. (d) The standard deviation profile of power density. Here, L_x and L_y are the width and length of the test chip, respectively. 92
3.18 Results of the Monte Carlo method with or without considering electro-thermal effects. (a) The mean temperature profile with considering the electro-thermal effect. (b) The mean temperature profile without considering the electro-thermal effect. (c) The standard deviation profile of temperature distribution with considering the electro-thermal effect. (d) The standard deviation profile of temperature distribution without considering the electro-thermal effect. 94
3.19 Simulation results of the developed methods. (a) and (b) the mean and standard deviation profiles of the estimated temperature distribution got by the stochastic projection method, respectively. (c) and (d) the mean and standard deviation profiles of the estimated temperature distribution obtained by the stochastic collocation method, respectively. (e) and (f) the error distributions of the mean and standard deviation of the estimated temperature distribution got by the stochastic projection method, respectively. 96
3.20 Thermal yield profiles of the test chip with $\left(\frac{WID}{WID+D2D}, \frac{D2D}{WID+D2D}\right) = (60\%, 40\%)$. (a) Profile obtained by the Monte Carlo method. (b) Profile obtained by the proposed skew-normal based method. (c) Profile obtained by APEX. 99
3.21 The error distributions of the skew-normal based method and APEX. (a) Distribution of the skew-normal based method comparing with the Monte Carlo method. (b) Distribution of APEX comparing with the Monte Carlo method. . 100
3.22 The temperature CDF curve at position **A** in Figure 3.20(a) got by the Monte Carlo method, and its estimated CDF curves obtained by the skew-normal model based method, APEX with the 4-th oredr and the 9-th order for the PDF/CDF shifting process. 100
3.23 The estimated thermal yield profile and the error distribution of the mixed-grid thermal estimation strategy. 102

4.1 Key points of LUTSim for the early physical design stages in 3-D ICs. 103
4.2 Thermal model for the early design stage of a 3-D IC with three tiers. 105
4.3 The flowchart of LUTSim. 108
4.4 Examples for the *lateral locality* and *local similarity* of the temperature response induced by a unit power source. Each unit power source is inserted to a grid on the top-surface of the tier adjacent to the secondary heat flow path. (a)–(f) are the temperature responses with inserting a unit power source on grids $(0,0)$, $(1,1)$, $(32,0)$, $(33,0)$, $(32,32)$ and $(33,33)$, respectively. 112
4.5 The table establishing process of TR-UPS of a specific grid in S_g. 113
4.6 An example of the selected representative grids in S_g of a specific tier. Gray color grids are the representative grids. 114
4.7 The table shifting and interpolation processes for the grid having no pre-built unit power temperature response table. 115
4.8 The error accumulation phenomenon of the fine-mesh look-up table strategy. . . 116
4.9 The double-mesh table establishment of the unit power temperature response. . 117
4.10 The calculating process of double-mesh look-up table technique. 118

4.11 Placement, power profiles, estimated temperature profiles of a two-tier industrial chip by ANSYS and R-LUTSim. (a) Placement. (b) Power profiles. (c) Estimated temperature profiles by ANSYS. (d) Estimated temperature profiles by LUTSim. 120
4.12 Error distribution of LUTSim compared with ANSYS. 120
4.13 The distribution of insertion numbers of TSVs for the test chip, “g-chip3”, stated in Table 4.1. 121
4.14 Power profiles, estimated temperature profile of fast MNA solver, estimated temperature profile of LUTSim and the error distribution between fast MNA solver and LUTSim of the test chip “g-Chip3”. a two-tier industrial chip by ANSYS and R-LUTSim. (a) Power profiles. (b) Estimated temperature profile of fast MNA solver. (c) Estimated temperature profile of LUTSim (d) Error distribution between fast MNA solver and LUTSim. 122



List of Tables

2.1 Accuracy and Runtime Comparison of the proposed GIT based method and the Algorithm II of [5]. 48

3.1 Accuracy comparison of leakage power models for an NAND gate under $65nm$ technology node. The results of HSPICE simulation with TSMC model card are employed to be the reference solution. The second column represents the fitting components of $f_g(L, t_{ox}, T)$ and $f_s(L, t_{ox}, T)$ adopted by the models proposed by [6–8] and our proposed models. 56

3.2 Accuracy comparison of leakage current models in [9] for an NAND gate under $65nm$ technology node. 58

3.3 Parameters and Truncation Points for the Channel Length and the Oxide Thickness. 91

3.4 Equivalent Thermal Parameters. 92

3.5 Accuracy and Efficiency of the Developed Statistical Expression Generators. . . 93

3.6 Accuracy and Efficiency Comparison of the Skew Normal Model and APEX for Estimating Thermal Yield Profiles. The results are compared with the Monte Carlo method with 2×10^5 samples. 97

4.1 Comparison between LUTSim and the fast MNA solver [10]. 121

Chapter 1

Introduction

In this chapter, several major thermal related issues of modern VLSI design will be summarized in sections 1.1–1.3. Then, the description of thermal-aware design flow, the essentiality of the thermal simulation and the review of existing thermal simulation methods are given in sections 1.4 and 1.5. Finally, the contribution and organization of this dissertation are summarized in sections 1.6 and 1.7.

1.1 Thermal Related Issues of Modern VLSI Design

1.1.1 Timing and Reliability Issues

Since the threshold voltage and carrier mobility are temperature-dependent, the drain current is affected by temperature fluctuations. Therefore, the propagation delay of gate will drift while the gate operating at different temperature. The propagation delay of a metal-oxide-semiconductor field-effect transistor (MOSFET) can be approximated as follow [11, 12].

$$t_d(T) = \frac{C_L V_{\text{dd}}}{I_d(T)} = \frac{\beta \times \mu_{\text{eff}}(T)}{\left(V_g - V_{\text{th}}(T)\right)^{\alpha}}, \tag{1.1}$$

where

$$\beta = \frac{L_{\text{eff}} C_L V_{\text{dd}}}{W C_{\text{ox}}}, \tag{1.2}$$

and C_L is the load capacitance, V_{dd} is the supply voltage, I_d is the drain current in the saturation region, C_{ox} is the gate oxide capacitance, L_{eff} is the effective channel length, W is the channel width, V_g is the gate voltage, α is the velocity saturation index, T is the operating temperature, $\mu_{\text{eff}}(T)$ is the effective carrier mobility and $V_{\text{th}}(T)$ is the threshold voltage. The temperature

dependences of $\mu_{\text{eff}}(T)$ and $V_{\text{th}}(T)$ can be illustrated by following equations [13].

$$\text{nMOS}: V_{\text{th}}(T) = V_{\text{th}}(T_0) + \left(\text{KT1} + \frac{\text{KT1L}}{L_{\text{eff}}} + V_{\text{bseff}}\text{KT2}\right)\left(\frac{T}{T_0} - 1\right), \quad (1.3)$$

$$\text{pMOS}: V_{\text{th}}(T) = V_{\text{th}}(T_0) - \left(\text{KT1} + \frac{\text{KT1L}}{L_{\text{eff}}} + V_{\text{bseff}}\text{KT2}\right)\left(\frac{T}{T_0} - 1\right), \quad (1.4)$$

$$\mu_{\text{eff}}(T) = U_0\left(\frac{T}{T_0}\right)^{U_{\text{te}}}\left\{1 + (U_c(T)\,V_{\text{bseff}} + U_a(T))\,\theta(T) + U_b(T)\,\theta^2(T)\right\}^{-1}, \quad (1.5)$$

where

$$\theta(T) = \left(\frac{V_{\text{gst}} + 2V_{\text{th}}(T)}{T_{\text{OXE}}}\right), \quad (1.6)$$

and T_0 is the reference temperature, KT1 is the temperature coefficient of threshold voltage, KT1L is the channel length corresponding temperature coefficient of threshold voltage, KT2 is the temperature inducing body-bias coefficient of the threshold voltage, V_{bseff} is the effective substrate to bias voltage, U_0 is the mobility at the reference temperature, U_{te} is a fitting coefficient, T_{OXE} is the electrical gate-oxide thickness, U_a is the first-order mobility degradation coefficient, U_b is the second-order mobility degradation coefficient and U_c is the coefficient of the mobility degradation inducing body effect.

According to equations (1.1)–(1.6), the temperature fluctuation drifts the propagation delay of a MOSFET. The above phenomenon is illustrated in Figure 1.1(a). As shown in Figure 1.1(a), the gate delay is increased by the operating temperature. Besides the above phenomenon of a MOSFET/gate, Khan et. al. [12] have addressed that the on-chip temperature profile induces considerable variations of the full-chip circuit performance beyond the 90*nm* technology nodes.

In addition to the effect of the gate delay, the temperature dependence of the wire resistance r can be written as

$$r = \rho_0(1 + \alpha T), \quad (1.7)$$

where ρ_0 is the resistance per unit length at 0 °C and α is the temperature coefficient of the resistance per unit length (1/°C). Since the delay of wire is proportion to the resistance, it will be impacted by the temperature. To illustrate the above phenomenon, the temperature dependences of the wire delay corresponding to different lengths are shown in Figure 1.1(b).

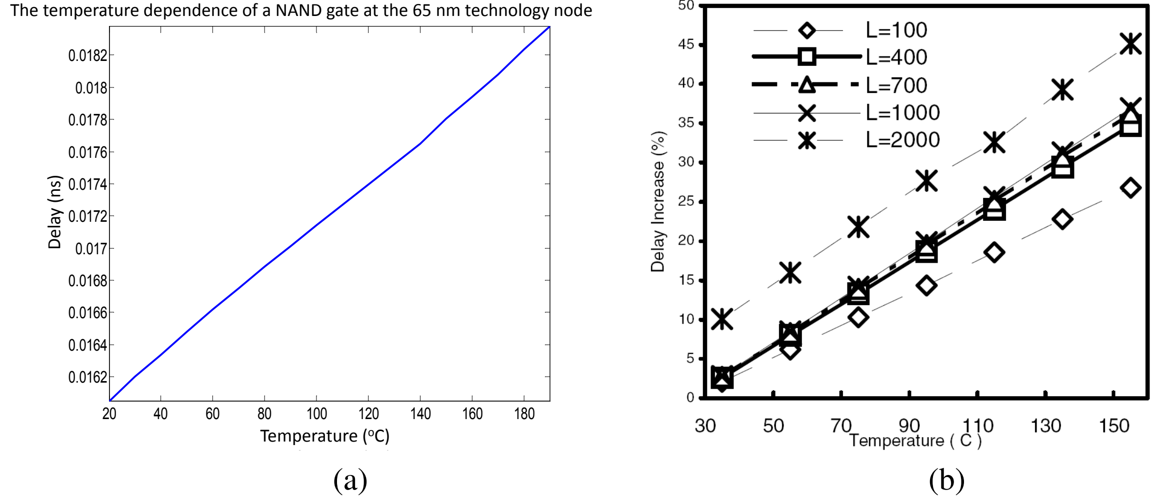


Figure 1.1: The temperature dependences for the delays of gates and interconnects. (a) The temperature dependency of a NAND gate at the 65*nm* technology node. (b) The temperature dependences for the delay of interconnect corresponding to different wire length L (reprinted from [1]).

As shown in Figure 1.1(b), the wire delay will be increased by the operating temperature. And the longer the wire length is, the larger the temperature inducing wire delay is.

The temperature dependences of gates and wires result in several design issues. For example, the non-uniform on-chip temperature profile may induce the timing fault of the design [14]. As shown in Figure 1.2, the non-uniform on-chip temperature profile results in different delays of the wires and the registers/gates at different positions of the chip. Therefore, the clock skew occurs even though the clock tree is designed as a symmetric H-shape topology with equal distances to the registers. Moreover, based on Black's equation [15], the median-time-to-failure (MTF) of wire can be written as

$$MTF = Aj^{-n} \exp\left(\frac{Q}{k_B T_m}\right), \tag{1.8}$$

where A is a geometry-dependent constant, j is the average current density, n is an empirical constant with its value being 2 for the normal condition, k_B is the Boltzmann's constant and T_m is the temperature of the wire. As shown in equation 1.8, the MTF of wire negatively and exponentially depends on the operating temperature. Therefore, high on-chip temperature degrades the lifetime reliability of the circuit.

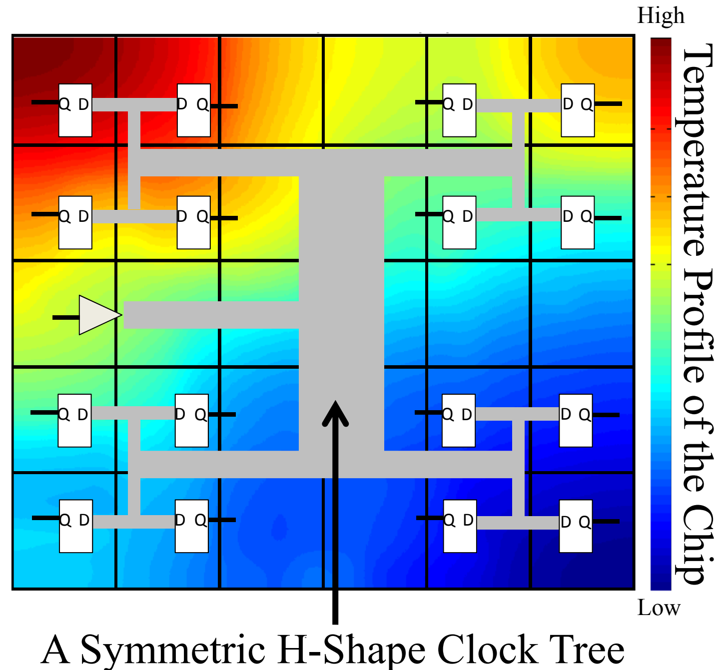


Figure 1.2: An example for illustrating the non-uniform on-chip temperature profile inducing clock skew.

1.2 The Effects of Temperature and Process Variations on the Leakage Powers

Due to the shrinking of device geometries, it is more difficult to control the physical device parameters. The growing variability of physical device parameters, such as the effective channel length and the gate oxide thickness, can induce considerable leakage power fluctuations. Since the on-chip temperature is transformed from the on-chip power, the fluctuations of on-chip leakage powers lead to the thermal simulation with nominal leakage powers is no longer effective to predict the on-chip temperature. In addition, since the leakage powers depend on the operating temperature, the temperature-power coupling effect occurs. This leads the electro-thermal analysis to be more concerned for ensuring the thermal reliability. In this chapter, we will introduce the temperature and process variations issues of two major leakage powers, gate tunneling and subthreshold leakage powers, and highlight its impacts on the on-chip temperature.

1.2.1 Subthreshold Leakage Current

The subthreshold leakage current of a MOSFET is defined as the conduction current between source and drain in “OFF” state. The subthreshold leakage current of the MOSFET can be

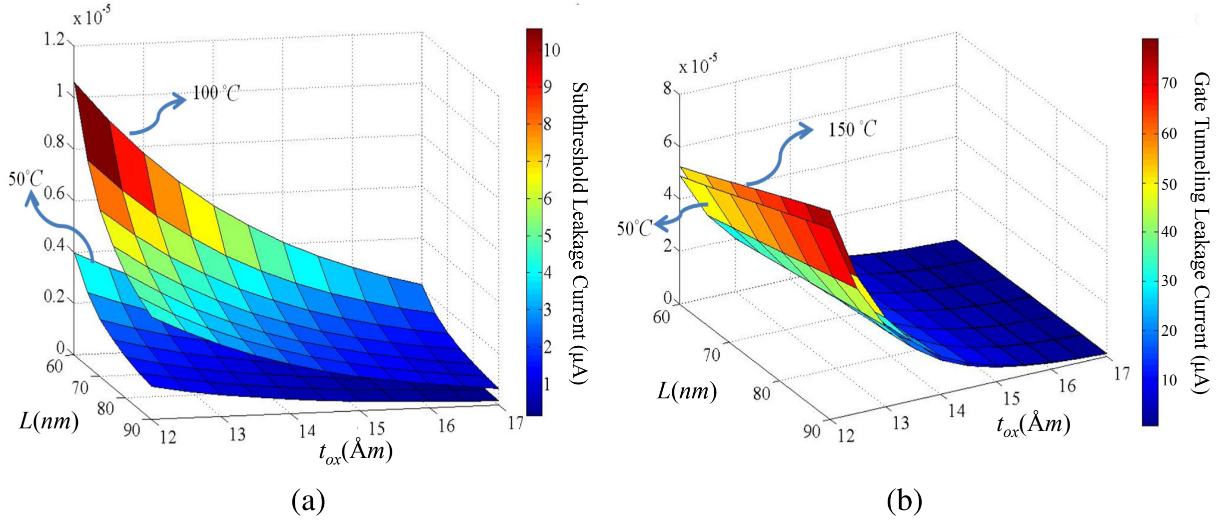


Figure 1.3: The temperature and process variation dependencies of subthreshold and gate tunneling leakage currents of a NAND gate at 65*nm* technology node. (a) The temperature and process variation dependencies of subthreshold leakage current. (b) The temperature and process variation dependencies of gate tunneling leakage current. Here, L is the device channel length, and its unit is *nm*. t_{ox} is the oxide thickness, and its unit is $\mathring{A}m$.

written as [16, 17]

$$I_s = I_0 \exp\left(\frac{V_{gs} - V_{\text{th}}}{nkT/q}\right)\left(1 - \exp\left(\frac{-V_{ds}}{kT/q}\right)\right), \tag{1.9}$$

where

$$I_0 = \mu_0 C_{\text{ox}} \left(\frac{W}{L}\right)\left(\frac{kT}{q}\right)^2 e^{1.8}, \tag{1.10}$$

and V_{gs} is the gate-to-source voltage, V_{th} is the threshold voltage, n is the subthreshold swing factor, k is the Boltzmann's constant, T is the operating temperature, q is the charge of an electron, V_{ds} is the drain-to-source voltage, μ_0 is the low field carrier mobility, C_{ox} is the gate oxide capacitance, W is the channel width and L is the channel length.

According to the above model, the subthreshold leakage current exponentially depends on the operating temperature. Although the exponential dependencies for channel length L and oxide thickness t_{ox} are not shown in equation (1.10), the subthreshold leakage current exponentially depends on L and t_{ox} because V_{th} is a function of these physical device parameters [18]. To illustrate the dependencies of the temperature, the channel length and the oxide thickness for the subthreshold leakage current, the HSPICE simulation result for a NAND gate at 65*nm* technology node is shown in Figure 1.3(a).

1896

1.2.2 Gate Tunneling Leakage Current

According to quantum mechanics, carriers have a finite probability to tunnel through the gate oxide. The current generated by these carriers is so-called the gate tunneling leakage current and have been characterized by BSIM4 gate tunneling model [19]. For describing the major parameters that affect the gate tunneling leakage current, BSIM4 model for the gate tunneling leakage current is simplified as [16, 20]

$$I_g = (A \cdot C)(W \cdot L) \exp\left(-B \cdot \frac{t_{ox}}{V_{gs}}\alpha\right), \tag{1.11}$$

where $A = q^3/8\pi h\phi_b$, $C = (V_{gs}/t_{ox})^2$, W is the channel width, L is the channel length, $B = 8\pi\sqrt{2m_{ox}}\phi_b^{3/2}/3hq$, t_{ox} is the oxide thickness, V_{gs} is the gate-to-source voltage, α is a parameter with range from 0.1 to 1 depending on the voltage drop across the oxide (a typical value is 0.22867), h is the Planck's constant, m_{ox} is the effective mass of electron/hole, q is the charge of an electron, and ϕ_b is the barrier height for electrons/holes in the conduction/valance band. The value of ϕ_b is $3.1eV$ for electron and $4.5eV$ for hole.

According to the above empirical model, the gate tunneling leakage current negatively and exponentially depends on the oxide thickness. When the value of the oxide thickness is larger than $20\mathring{A}m$, the gate tunneling leakage current is relatively small comparing with other leakage currents, such as the subthreshold leakage current. However, the gate tunneling leakage current increases 2.5× for $1\mathring{A}m$ decrease of oxide thickness. This results in over 30× increase of the gate tunneling leakage current per technology generation [21]. Therefore, the gate tunneling leakage current becomes an important factor in the advanced technology node, e.g the sub-$100nm$ technology node [22]. Furthermore, although the dependencies of the temperature are not explicitly shown in equation (1.11), based on the SPICE simulation with BSIM4 model [23], the gate tunneling leakage current weakly depends on the temperature. To illustrate dependencies of the temperature, the channel length and the oxide thickness for the gate tunneling leakage current, the HSPICE simulation result for a NAND gate at $65nm$ technology node is shown in Figure 1.3(b).

1.2.3 Leakage Powers Inducing Electro-Thermal (Temperature-Power) Coupling Effect

Generally, the on-chip power consumption P_{chip} consists of dynamic power and leakage power, and it can be calculated by [24]

$$P_{\text{chip}} = S_{\text{act}} C_{\text{total}} V_{\text{dd}}^2 f + V_{\text{dd}} I_{\text{leak}}, \tag{1.12}$$

where S_{act} is the average switching activity of gates, C_{total} is the total load capacitance of gates, V_{dd} is the supply voltage, f is the operating frequency and I_{leak} is the total leakage current of gates.

In the right hand side of equation (1.12), the first term is the dynamic power induced by the charging and discharging currents to the load capacitance of gates, and the second term is the leakage power induced by the leakage currents of gates. As mentioned in sections 1.2.1 and 1.2.2, the leakage powers exponentially depend on the operating temperature, and the on-chip leakage power will catch up with the on-chip dynamic power beyond the 90*nm* technology node [24,25]. On the other hand, the on-chip temperature is transformed from the on-chip power consumption. Therefore, electro-thermal (temperature-power) coupling occurs, and it induces the thermal reliability issues in the modern VLSI designs. For example, *thermal runaway* [26] may happen if the elector-thermal coupling is not well concerned during the package and the cooling system design.

The mechanism of electro-thermal coupling is exhibited in Figure 1.4. First, with an initial on-chip temperature, the initial on-chip power consumption is obtained. Based on the zeroth law of thermodynamics [27], surplus powers, which can not be dissipated by the package and the cooling system, will transform into heat for achieving the equilibrium of the generating and dissipated powers; therefore, the on-chip temperature increases. On the contrary, as the power dissipation capacity of the package and the cooling system is larger than the generating power of the chip, the on-chip temperature decreases. Because the leakage power consumption depends on the temperature, the total power consumption will change after the temperature is updated. With repeating the above mechanism, if the equilibrium of the generating and dissipated powers of the chip can be achieved, stable on-chip temperature and power consumption

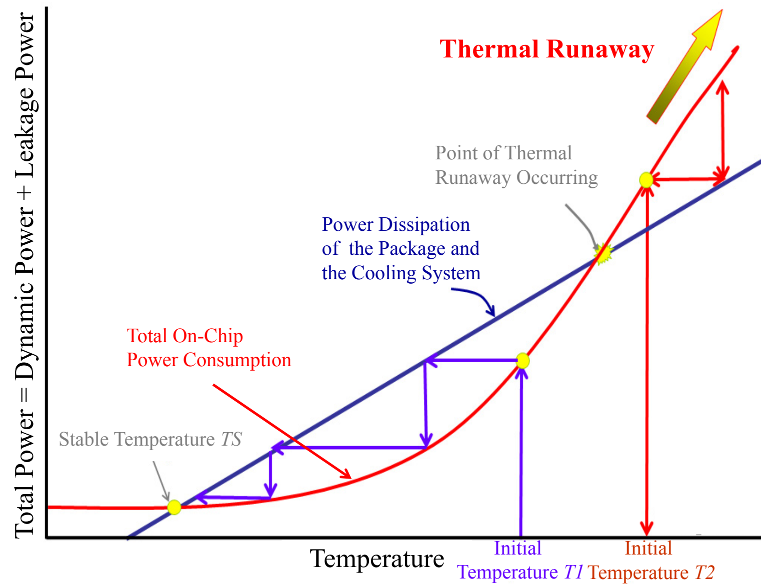


Figure 1.4: The mechanism of electro-thermal coupling.

are accomplished. On the contrary, if the *thermal equilibrium* can not be achieved[1], the chip thermally runs away. For example, as shown in Fig 1.4, the red curve indicates the generating power of the chip operating at different temperatures. The straight line indicates the maximum power that can be dissipated by the package and cooling system at different operating temperatures. Given an initial temperature $T1$, the stable operating temperature TS can be achieved after electro-thermal coupling is proceeded. On the other hand, if the initial temperature is $T2$, the thermal runaway occurs.

As the example illustrating, it is important to consider the electro-thermal coupling to ensure the thermal reliability of the circuit.

1.2.4 Variations of Physical Device Parameters

The shrinking of device geometries has leaded to considerable variations of physical device parameters. As mentioned in sections 1.2.1 and 1.2.2, leakage powers are sensitive to the physical device parameters, such as the channel length and the oxide thickness. Therefore, the variations of physical device parameters will induce considerable fluctuations of the leakage powers. As shown in Figure 1.5, Borkar et. al. [2] have pointed out that 30% process variations can cause about 20× leakage power fluctuations. Because of the electro-thermal coupling, this will result

[1]The curves of the generating power of the chip and dissipated power of the package and the cooling system do not have intersection points, or the initial operating temperature is not well chosen [26]

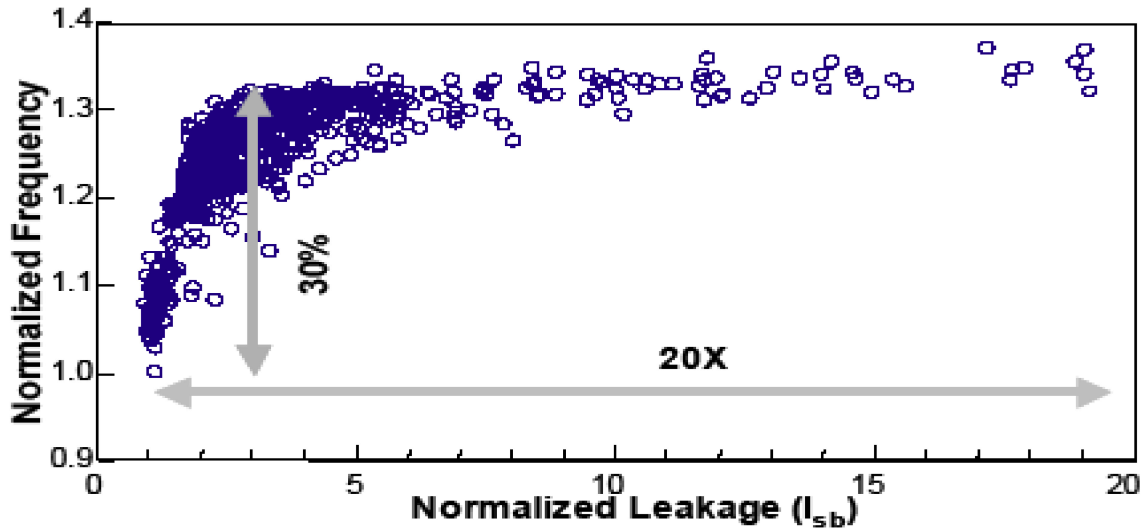


Figure 1.5: Parameter variations impacts on the leakage currents (reprinted from [2]). Here, the y-coordinate indicates the normalized occurrence frequency of the value of the device channel length, and the x-coordinate indicates the normalize value for the subthreshold leakage currents (I_{sb}).

in considerable fluctuations of the on-chip temperature and the temperature inducing leakage power fluctuations. Therefore, under process variations, the on-chip temperature and leakage power should be treated statistically, especially for the leakage power dominated technology.

1.3 Three-Dimensional Integrated Circuit and Its Thermal Issues

The rapid growth of the functionalities and performance requirements of the computer and information technology industry leads to the continually scaling down of the technology. This fact degrades the routability and induces longer interconnects, and limits the performance of planar (two-dimensional) integrated circuits (2-D ICs). For example, the longer interconnect increases the delay of the signal transmission, raises the on-chip power consumption [28], and results in the issues such as signal integrity and routing congestion. With the vertical interconnect strategy, three-dimensional integrated circuit (3-D IC) can reduce the wire length, the transmission delay, the interconnect power and the chip area. Therefore, in recent years, 3-D IC has been regarded as an effective design strategy to overcome the performance bottlenecks of 2-D ICs [3, 29–33].

As shown in Figure 1.6, there are several implementation categories of 3-D IC [3]. The

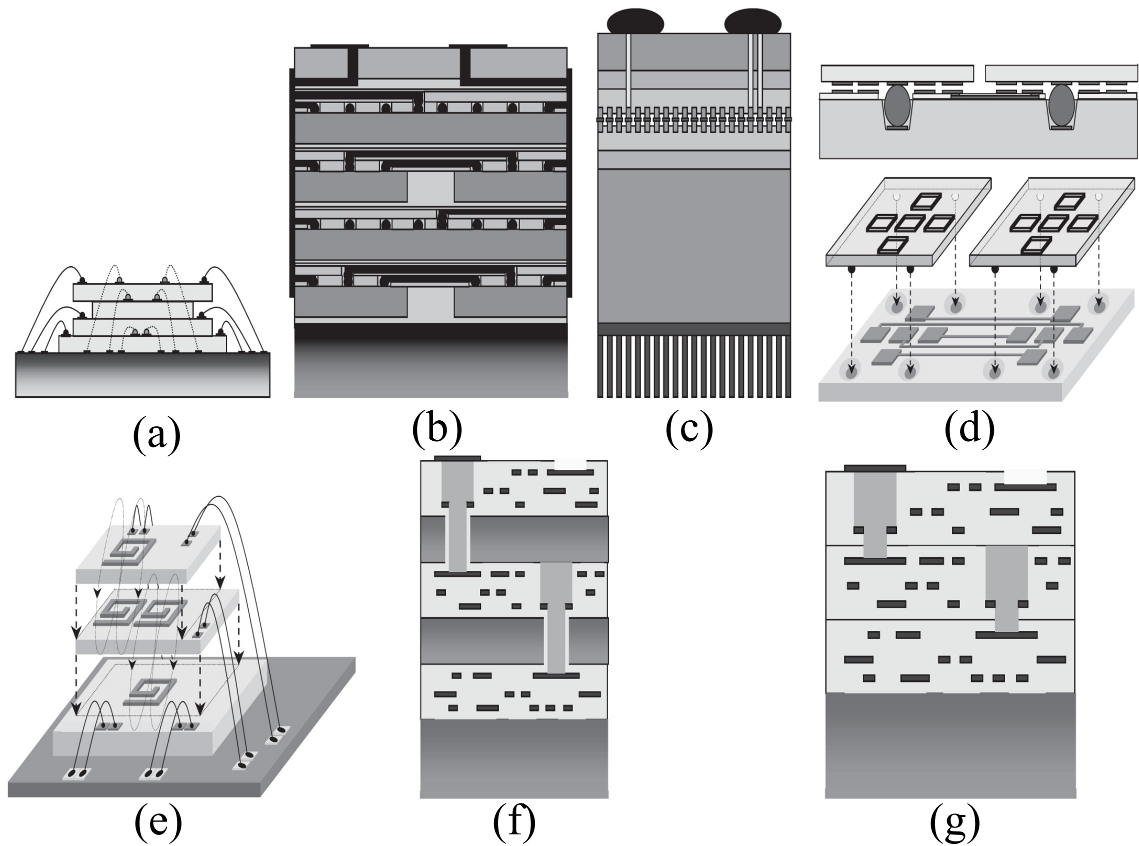


Figure 1.6: Implementation categories of 3-D ICs (reprinted from [3]). (a) Wire-bonded structure. (b) Microbump-3D package structure. (c) Face-to-face structure. (e) Contactless-capacitive with buried bumps structure. (f) Contactless-inductive structure. (f) Through silicon vias (TSVs) based structure with silicon substrates on bulk. (g) TSVs based structure with silicon substrates on insulator (SOI).

wire-bonded structure stacks tiers, and transmits the signal between tiers by using the wires connecting to the board. This category suffers from the limitation on the resolution (for example, $35\mu m$ pitch between $15\mu m$ wires) of wire bonders on the board. Therefore, it is only practical for the design with small amount of inputs/outputs (I/Os) between stacked dies.

With the package technology, which can assemble fabricated tiers into a set of carrier wafers with a fixed size, the microbump-3D package structure connects signals between tiers by employing the solder bumps on top surfaces of tiers and the interconnects connecting to peripheries of tiers. This structure offers a much greater vertical interconnect density than that of the wire-bonded structure. However, because it still requires routing signals to the tier periphery before sending them back to the destination inside the tiers, this structure does not significantly reduce the transmission delay.

The face-to-face structure flips the top tier, and connects top and bottom tiers by using the

interconnects of the metal layers or the through-via approach. With this structure, the interconnect length can be reduced. However, this structure is restricted to two tiers.

Different from the wire bonded, the microbump-3D package and face-to-face structures, as shown in Figure 1.6 (e) and (f), the contactless-capacitive or the contactless-inductive structure employs the capacitive or inductive coupling to communicate signals between tiers. However, this structure suffers from the challenge for supplying DC power to tiers. Typically, engineers use solder bumping to provide the DC power connectivity between tiers or between a tier and a substrate. Since the distance between the two tiers will be resulted by solder bumps, there are implementation difficulties to combine solder bumping DC connectivity and AC-coupled interconnection. For example, for ensuring the functionality of the capacitive coupling interconnection, the distance between tiers must be small enough to allow sufficient capacitive coupling.

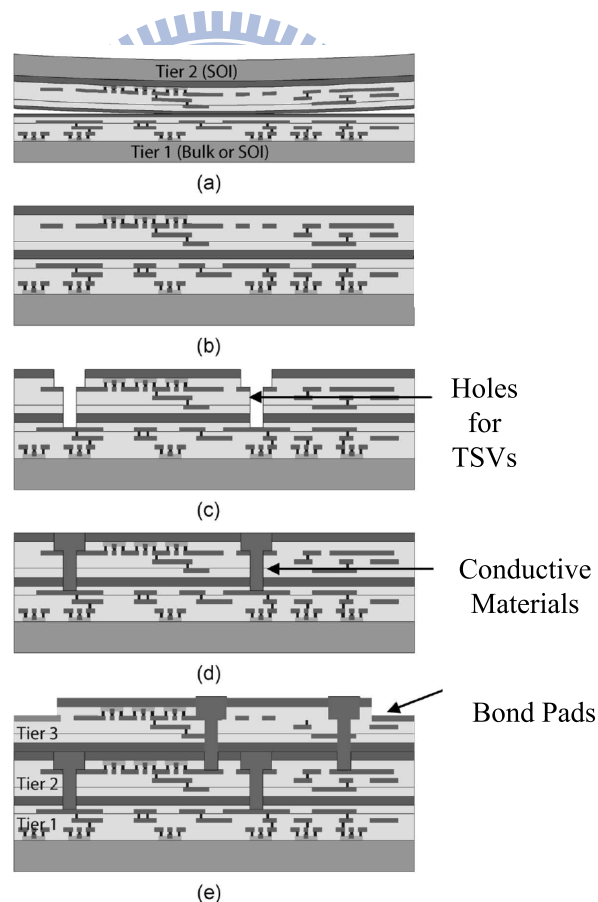


Figure 1.7: The wafer bonding process of the TSVs based 3-D IC (reprinted from [4]).

Finally, as shown in Figure 1.6 (g) and (h), the structures of TSVs based 3-D ICs with silicon on bulk and on insulator etch holes passing through silicon substrates and fill the holes

with conducting materials to provide connectivity between tiers. Typically, these categories can be implemented by the wafer bonding process [4]. The wafer bonding process of a 3-D ICs consisting of three tiers are illustrated in Figure 1.7. First, as shown in Figure 1.7 (a) and (b), the two tiers with completed circuits are planarized, aligned, and bonded face to face. Then, the top handle substrate is removed. After that, as show in Figure 1.7 (c)–(d), holes for TSVs are etched through the top tier. Then, the conducing material are formed to generate TSVs. Finally, as shown in Figure 1.7 (d), the same process is repeated to generate the third stacked tier, and bond pads are etched to generate the I/Os. Due to the ability of current technology, the physical size of TSVs can be small (less than $50\mu m$). Thereby, the TSVs based structures of 3-D IC have the potential to offer the greatest interconnect density; currently, they are the most popular implementation categories of 3-D ICs.

With the stacked tiers structure, 3-D ICs can also provide the flexibility for the mixed signal design, the suitability for the circuit operating on different supply voltages, and the capability for the heterogeneous integration. However, due to the higher power density and the ill of heat dissipation capability, the operating temperature of 3-D ICs will be higher than that of 2-D ICs. The tradeoff between the circuit performance and the thermal issue of 3-D ICs has been studied. As indicated by [3, 32, 33], the expected performance and design reliability of 3-D ICs are degraded because of the high temperature of 3-D ICs. Therefore, researchers have devoted to deal with the thermal issues in different stages of 3-D IC design flow [34–38].

1.4 Thermal-Aware Design Flow

To ensure design qualities such as performance, power consumption and thermal reliability, researchers have devoted to thermal-aware design techniques for dealing with the thermal issues. The execution flow of thermal-aware design is shown in Figure 1.8. With the initial package/cooling system and the designed circuit, the power density profile, the thermal conductivity profiles, and the thermal model of the chip are established by the thermal model establishment process shown in D3 of 1.8. Then, the on-chip temperature profile is obtained by the thermal simulator shown in D4 of Figure 1.8. After the on-chip temperature profile is obtained, the circuit performances, such as the timing of the circuit, MTF of the interconnect and the power

consumption, can be evaluated. Then, the circuit performances are checked if they meet the design requirement.

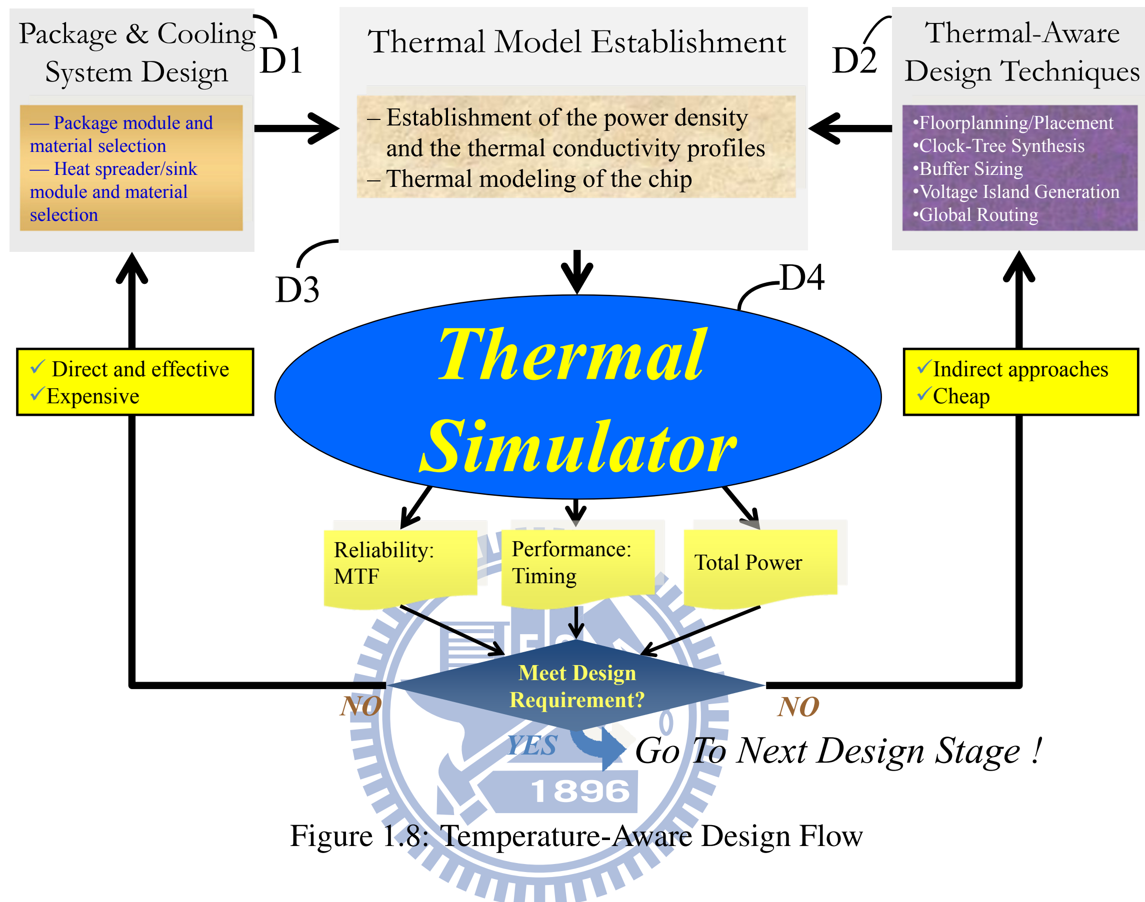


Figure 1.8: Temperature-Aware Design Flow

If the design requirement is not satisfied, one branch for reducing the on-chip temperature is to re-design the package and cooling system, and re-perform the processes shown in D3 and D4 of Figure 1.8 until the circuit performances meet the design requirement. Although the above thermal-aware design strategy is direct and effective, advanced package and cooling systems are expensive for reducing on-chip temperature [39].

Another design branch is to utilize circuit design techniques, such as thermal-aware floorplanning and placement [34–38,40–42], thermal-aware clock-tree synthesis [43], thermal-aware buffer sizing [44], thermal-aware voltage island generation [45, 46] and thermal-aware global routing [47, 48]. In this branch, the adjustment of the circuit, the thermal model establishment and the thermal simulation are performed until the circuit performances meet the design requirement. The above strategy is prevalent in modern VLSI design because it can meet the design requirement with cheaper package and cooling systems.

1.5 Review of On-Chip Thermal Simulation Methods

1.5.1 Simulation Methods of Deterministic On-Chip Temperature Profile

As mentioned in section 1.4, a thermal simulator is essential to obtain the temperature profile for providing the thermal cost to different design techniques. In general, after the power and the thermal conductivity profiles of the chip have been obtained, the temperature profile of the 2-D or 3-D chip can be accurately obtained by solving the equivalent SPICE-compatible thermal circuit generated by the finite difference method (FDM) [49, 50]. However, there are numerous nodes in the equivalent SPICE-compatible thermal circuit. This leads to a highly computational complexity for solving the on-chip temperature profile by employing the direct methods, such as HSPICE simulation or the LU decomposition based solver of the modified nodal analysis (MNA) system. Therefore, researchers have devoted to develop efficient yet accurate methods to speed up the runtime of the thermal analysis [5, 51–59].

These advanced thermal simulators can be categorized into two classes, numerical and analytical methods. The numerical methods [51–59] firstly apply the FDM to generate the equivalent SPICE-compatible thermal circuit, and then developed advanced numerical techniques to solve the large scale MNA system. Wang et. al. [51] utilized the alternating-direction-implicit (ADI) method to split the equivalent thermal circuit into different alternating directions, and alternately performed the line smooth scheme in each direction. In [52, 53], based on the moment matching and the Krylov subspace projection techniques, the model order reduction techniques were employed to improve the efficiency of transient simulation for the on-chip temperature profile. Li et. al. [54] applied the multi-grid method to speed up the convergence rate of basic iterative methods, and proposed a model order reduction scheme to further save the runtime of the transient simulation. Based on the framework of the multi-grid method, Yong et. al. [55] proposed the adaptive volume meshing and time-step selecting approaches to further reduce the complexity of the thermal analysis. Due to the flexibility of dealing with the complex structure, the numerical methods are the main stream in back-end design stages, such as the post layout thermal verification.

As pointed out in [40–42, 57–59], temperature-aware design should be brought to early design stages, such as floorplanning and placement stages. Since the detail layout of inter-

connects is not available in early design stages, an appropriate thermal model is required for the pre-layout interconnect layers. Therefore, Huang and Skadron el. al. [56–59] proposed compact thermal modeling techniques for the package, the cooling system and the pre-layout interconnect layers. With the compact thermal model, they proposed macro and grid based thermal circuit modeling techniques for coarse and fine granularities thermal simulation of the micro-architecture level design, respectively. Since their proposed thermal analysis method also requires solving a SPICE-compatible thermal circuit, the advanced numerical methods stated in [51–55] can also be adopted to further reduce the complexity.

The other category of thermal simulators, which is suitable for early design stages, is the analytical method. The primary advantage of analytical approaches is that they avoid the volume meshing procedure of entire substrate, and have closed-form representations of the on-chip temperature profile. Hence, they are flexible to obtain the temperature profile of certain user-specified regions without performing the full-chip thermal simulation. Furthermore, based on the closed-form representations, the on-chip temperature profile can be fast evaluated without solving the equivalent MNA system of the thermal circuit.

One existing analytical based full-chip thermal simulation technique is the Green's function based method [5]. The simulation framework proposed in [5] is executed as follows. First, the closed-form of the steady state Green's function corresponding to an impulse power source locating at any arbitrary location of die is firstly obtained. After that, the steady state on-chip temperature profile can be got by performing a table look-up method for obtaining the convolution of Green's function and the on-chip power density profile. To deal with the low efficiency of the look-up table method for the huge number of power sources, they approximated the on-chip power density profile by cosine waveforms. After that, they casted the on-chip temperature profile into the form of discrete cosine transform (DCT). With the fast Fourier transform (FFT), their complexity for obtaining the on-chip temperature profile can be in $O(MN \log_2 MN)$. Here, M and N are numbers of divisions for representing the on-chip power density and temperature profiles along x- and y-directions of die, respectively.

However, as their results exhibiting, a large truncation number of their basis functions is required to achieve an accurate estimation of the on-chip temperature profile. Moreover, the

numbers M and N are restricted to be the same with the truncation numbers of their basis functions along x-, and y-directions of die, respectively. This results in the restriction of the mesh size in their framework. Further, their formulation can only provide the steady state on-chip temperature profile. Although the steady state on-chip temperature profile is more concerned in thermal-aware physical design engines [34–38, 42], as indicated by [56], the temporary characteristics of the on-chip temperature are also important for the real-time dynamic thermal management with transient workloads. Therefore, although the existing analytical based on-chip thermal analyzer [5] takes the advantage of their closed-form representation for the on-chip temperature profile, it still requires improving strategies to extend their application scopes.

1.5.2 Simulation Methods of Statistical On-Chip Temperature Profile

As mentioned in section 1.2.4, the process variations induce leakage power fluctuations. For the same designed circuit, this fact will lead to different temperature profile of different fabricated chip. However, the thermal simulation methods mentioned in section 1.5.1 did not take into account the process variations issues in their leakage power models. Thus, under the process variations being considered, they are inadequate to precisely provide the on-chip temperature profile, the hot-spot locations and the thermal related costs of thermal-aware design engines. To provide the statistical characteristics of the on-chip temperature profile, one direct strategy is to apply the Monte Carlo (MC) method. However, the MC method requires performing a large amount of thermal simulations corresponding to the sampling points for the device parameters. This leads the MC method to be inefficient for the practical application.

Instead of the MC method, Jaffari et. al. [9] proposed a recursive log-normal approximation algorithm to obtain the mean and standard deviation profiles of the statistical on-chip temperature distribution. Compared with the MC method, they have successfully demonstrated its efficiency and accuracy for estimating the mean and standard deviation profiles of the on-chip temperature distribution in the macro-architectural level. However, instead of constructing the leakage power models for each type of macro/gate, their proposed leakage power models were built for each bin (grid) of die. Hence, their bin based leakage power models need to be rebuilt after the macros/gates are exchanged by the optimization engines such as floorplanners or placers. Since the establishment of their leakage power models require time-consuming HSPICE

simulation and curve fitting, the re-establishing of the bin based leakage power models will degrade their efficiency for providing the thermal simulation in thermal-aware design flow. Moreover, their recursive log-normal approximation algorithm is restricted to their proposed leakage power models. As a matter of fact, the leakage power model is going to be more complicated for maintaining an acceptable accuracy level while the technology continuously scales down. Thus, their simulation framework still requires to be reformed for dealing with more complex leakage power models, which is required for more advanced technology generations.

Besides the issues of leakage power models, although they can provide the mean and variance profiles of on-chip temperature distribution, the figure of merit for identifying statistical hot-spot locations is still ambiguous if only the mean and variance profiles are reported. For example, if only the mean profile of the on-chip temperature distribution is employed as the figure of merit, it is very likely (about 50%) to incorrectly indicate hot-spot locations. Furthermore, if only the mean profile $\mu_T(\mathbf{r})$ and standard deviation profile $\sigma_T(\mathbf{r})$ of on-chip temperature distribution are provided, by utilizing the Chebyshev inequality, a large temperature value is estimated to ensure the 90% lower bound of thermal reliability, i.e. T_{ref} needs to be $\mu_T(\mathbf{r}) + 3\sigma_T(\mathbf{r})$ to ensure $Prob\left(T(\mathbf{r}) \leq T_{ref}\right) \geq 0.9$. Here, $T(\mathbf{r})$ is the statistical on-chip temperature profile. Since the Chebyshev inequality does not always get a tight lower bound for any type of random variable[2], its estimating reference temperature $\mu_T(\mathbf{r}) + 3\sigma_T(\mathbf{r})$ might be an immoderately conservative constrain for the thermal reliability. This undesirable phenomenon can result in the immoderate guard-banding for the circuit design.

As pointed by [13, 14, 60, 61] and mentioned in section 1.1.1, the gate delay is influenced by the variations of device parameters and the operating temperature. Although Jaffari's log-normal random variable model [9] can compactly express the temperatures in bins, they are still random variables correlating with the random variables for modeling the device parameters. Therefore, elegant strategies are still necessary to incorporate the log-normal random variable model into the statistical performance analysis engines such as statistical static timing analysis (SSTA) [60, 61].

[2]For example, suppose that x is a standard normal random variable, $Prob\,(x \leq 1.28\sigma_x) = 0.9$. However, the Chebyshev inequality requires a larger reference value, $3\sigma_x$, to obtain the same probability as the lower bound, i.e. $Prob\,(x \leq 3\sigma_x) \geq 0.9$.

1896

1.5.3 Simulation Methods of the Temperature Profile for 3-D ICs

Although the numerical thermal simulation methods [51–59] of 2-D ICs provide adequate flexibility for the full-chip thermal analysis of TSV based 3-D IC, elegant incremental temperature updating strategies are still required to be developed for updating the temperature profile in thermal-aware design flow of the TSV based 3-D IC. Generally, the compact thermal model of early physical design stages in 2-D ICs can be reasonably built as the one composed of the die with homogeneous material and the effective thermal models for the package, the cooling system and interconnect layers [40–42, 56–59]. Hence, the equivalent thermal circuit will not be changed during the design procedure, e.g. optimization loops of the floorplanning and placement [40–42]. Therefore, the handling process of the thermal conductance matrix[3] can be performed before executing the optimization loop and does not be re-performed during the optimization loop.

However, for the early physical design stages of TSV based 3-D ICs, it is impractical to model the silicon substrates as homogeneous material layers because the positions of TSVs and macros/gates are simultaneously considered in the floorplanning and placement stages [34–38]. In the other words, the equivalent thermal circuit will be altered after each optimization loop is executed. Therefore, the handling process of the thermal conductance matrix needs to be re-performed for each optimization loop, and this decreases the efficiency of the thermal simulation methods [51–59]. Moreover, although the analytical based thermal simulation method [5] can obtain the closed-form expression for the on-chip temperature profile of 2-D ICs, its simulation framework cannot directly be applied to 3-D ICs because the closed-form expression of the temperature can only be obtained on the homogeneous material structures.

To avoid performing the time-consuming detail thermal simulation for updating the operating temperature in each optimization loop, Cong et. al. [62] simplified the SPICE-compatible thermal circuit of the TSV based 3-D chip [50] by independent stacked tile one-dimensional (1-D) thermal circuits. With the simplification, they employed the stacked tile 1-D thermal circuits to update the temperature profile in the optimization loops. The simplified stacked tile

[3]The handling processes of the thermal conductance matrix in [51–59] are the LU decomposition of a tri-diagonal matrix in each alternative direction [51], the multilevel restriction-interpolation construction [54, 55] and the LU decomposition of the thermal conductance matrix [52, 53, 56–59].

1-D thermal modeling technique is widely adopted by recent thermal-aware design techniques of TSV based 3-D such as [35, 37, 63, 64]. Although the stacked tile 1-D thermal model can fast update the temperature profile, the lateral thermal spread is ignored. This fact can lead to inaccurate result of the on-chip temperature estimation, and the decisions of optimization engines might be misled. To improve the accuracy of the stacked tile 1-D thermal model that ignores the lateral thermal spread and the efficiency of the thermal conductance matrix re-handling process [52, 53, 56–59], an accurate thermal simulation method with the incremental temperature updating ability is essential for thermal-aware design flow of TSV based 3-D ICs.

1.6 Contributions of this Dissertation

The targets of this dissertation are on providing accurate and efficient simulation methods to estimate deterministic and statistical temperature profiles for thermal-aware design flow. The contributions are summarized as follows.

Deterministic Thermal Simulation:

1. Compared with the Green's function based method [5], a generalized integral transforms (GIT) [65–67] based thermal simulation method is proposed to speed up the error decaying rate of the analytical framework. The proposed method can accurately estimate the full-chip temperature profile with very small truncation points (N_x and N_y) of the representing spatial bases. Compared with [5], the experimental results show that N_xN_y can be far less than MN without sacrificing any accuracy. Here, M and N are the mesh sizes of the power density and temperature profiles along x- and y-directions of a die, respectively. N_x, and N_y are truncation points of bases along x- and y-directions, respectively. Besides the steady state thermal simulation, the proposed method also provides the transient thermal simulation.

2. A FFT based evaluating algorithm, which avoids the zero-padding of the standard FFT algorithm while the sizes of input and output data are different, is developed to efficiently evaluate the on-chip temperature profile, and its complexity is in the order of $O(MN \log_2 N_xN_y)$.

3. An efficient thermal simulation method, which is a hybrid scheme combining the GIT and numerical simulation frameworks, is proposed for the thermal simulation of the wire-bound, the face-to-face, the contactless-capacitive interconnection and the contactless-inductive interconnection based 3-D ICs. Moreover, the hybrid scheme can be directly applied to get more accurate on-chip temperature distribution while components in primary and secondary heat flow paths are modeled as stacked layers with different thermal conductivities.

Statistical Thermal Simulation:

1. Comparing with the bin based model [9], a cell based model is adopted for characterizing the leakage powers. With its pre-characterizing property, the re-establishing process of the leakage powers can be avoided while the macros/gates are exchanged by the optimization engines such as floorplanners or placers.

2. Two techniques, stochastic projection and collocation based methods, are proposed to generate the polynomial expressions of the statistical on-chip temperature distribution. Comparing with [9], both of these techniques are more flexible for complex fitting models of leakage powers. Moreover, the generating polynomial expressions can be easily casted into the framework of SSTA with the first or the second order polynomial form [60, 61].

3. Instead of only providing the mean and standard deviation profiles of the on-chip temperature distribution, the concept of the thermal yield profile is introduced for characterizing the statistical on-chip temperature distribution more precisely. And an efficient technique is proposed for estimating this figure of merit.

4. A mixed-mesh strategy is developed to enhance the efficiency of the proposed on-chip thermal yield profile estimator. As the results demonstrating, the proposed mixed-mesh strategy enables the efficiency of the thermal yield profile estimation to catch up with that of deterministic thermal simulation (hot-spot estimators) without sacrificing the accuracy.

Thermal Simulation of 3-D ICs:

1. A look-up table based method is proposed to estimate the steady state temperature profile of 3-D ICs. With utilizing the pre-built tables of the temperature response induced by a unit power source, a recursive table look-up technique is proposed to estimate the temperature profile of 3-D ICs.

2. For the full-chip thermal simulation, this simulation method can efficiently calculate the on-chip temperature without dealing with the large scale thermal conductance matrix, which is the major computation effort of prior arts [50–59], of the equivalent thermal circuit.

3. After TSVs are moved by the optimization engines, this simulation method can update the on-chip temperature profile by recursively table looking-up without re-performing the dealing process of the thermal conductance matrix.

1.7 Organization of this Dissertation

The rest of the dissertation is organized as follows. In chapter 2, the problem formulation, the simulation algorithm and the experimental results of the proposed deterministic thermal simulation method are detailed. In chapter 3, the problem formulation, the models of leakage powers, the simulation algorithm and the experimental results of the proposed statistical thermal simulation method are detailed. In chapter 4, the problem formulation, the simulation algorithm and the experimental results of the proposed look-up table based thermal simulation method for early design stages of 3-D ICs are detailed. Finally, the conclusion of this dissertation is presented in chapter 5.

Chapter 2

Simulation Method I – *Full-Chip Thermal Analysis for Early Design Stages via Generalized Integral Transforms*

As addressed by [40–42, 56–59], temperature-aware design should be brought to early design stages such as thermal-aware floor-planning and placement. Therefore, the proposed simulation method in this chapter is on efficiently proving the on-chip temperature profile for early design stages. This chapter is organized as follows. First, the thermal model for early design stages is presented in section 2.1. The generalized integral transform (GIT) based computational formula of the on-chip temperature profile and the proposed evaluating algorithms are described in section 2.2. After that, in section 2.3, a method with a hybrid scheme of the GIT based analytical and the FDM based numerical formulations will be addressed for simulating the temperature profile of the stacked dies and package structure. Finally, the experimental results are given in sections 2.5.

2.1 Thermal Modeling for Early Design Stages

To estimate temperature with a reasonable accuracy and a little computational effort, instead of employing a detail thermal model for the post-layout thermal verification, a compact thermal model is essential for fast temperature simulation in early design stages. A popular compact thermal model of the chip for early design stages is illustrated in Figure 2.1 [56–59]. This model consists of three portions: the primary heat flow path, the secondary heat flow path, and the heat transfer characteristic of each macro/block on the silicon die. The primary heat flow path is composed of thermal interface material, heat spreader and heat sink. The secondary

heat flow path contains interconnect layers, I/O pads and the print circuit board (PCB). The functional blocks are modeled as many power generating sources attached to a thin layer close to the top surface of die with the thickness being equal to the junction depth[1]. The major concerns

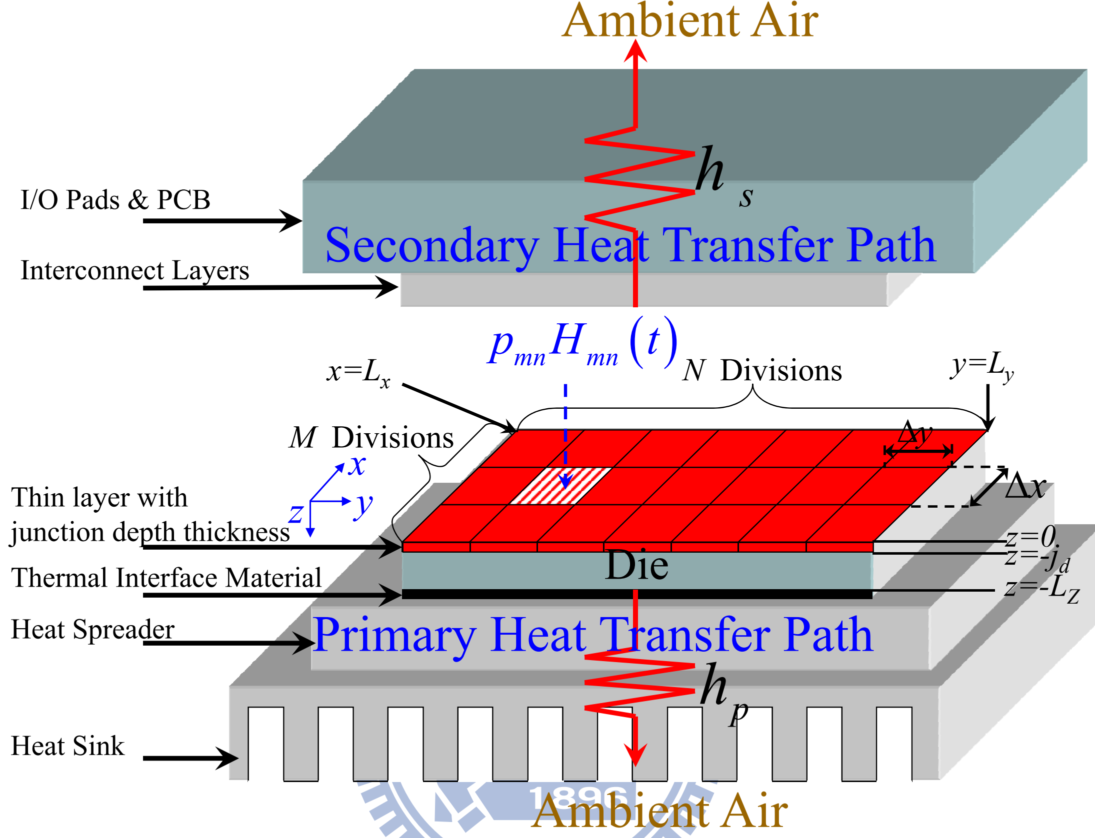


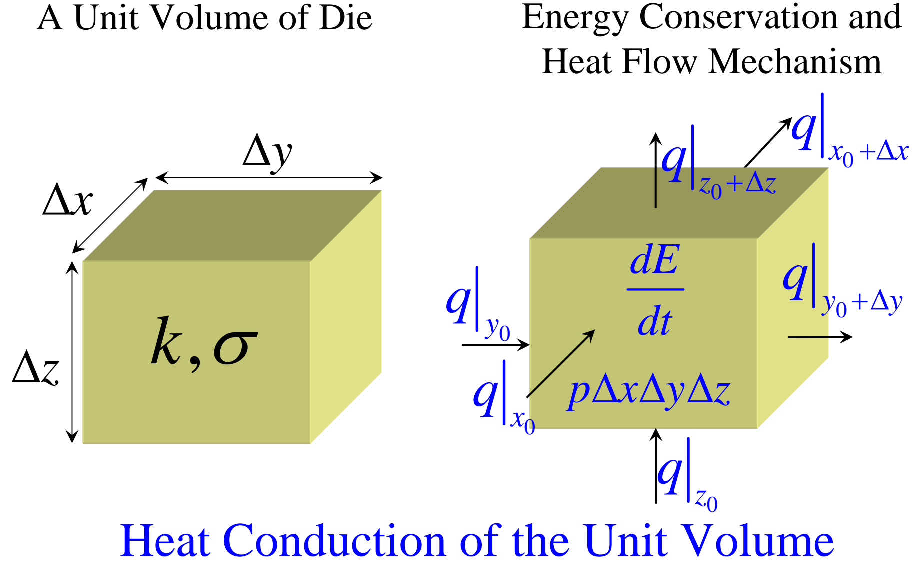
Figure 2.1: Compact thermal model for early design stages.

of early-stage temperature-aware optimization procedure are to reduce the temperature or the thermal gradient of die. Thus, we focus on estimating the temperature profile of die.

According to energy conservation law, the changing rate of energy in a unit volume of substrate equals to the conduction heat through the unit volume [65]. Figure 2.2 illustrates this heat conduction mechanism. In Figure 2.2, dE/dt is the energy change rate for the unit volume and is equal to $\sigma\Delta x\Delta y\Delta z\partial T/\partial t$. The conduction heat flowing into the unit volume is equal to the sum of $q|_{x_0} = -\kappa\Delta y\Delta z\, \partial T/\partial x|_{x_0}$, $q|_{y_0} = -\kappa\Delta x\Delta z\, \partial T/\partial y|_{y_0}$ and $q|_{z_0} = -\kappa\Delta x\Delta y\, \partial T/\partial z|_{z_0}$. The conduction heat flowing outward the unit volume is the sum of $q|_{x_0+\Delta x} = -\kappa\Delta y\Delta z\, \partial T/\partial x|_{x_0+\Delta x}$, $q|_{y_0+\Delta y} = -\kappa\Delta x\Delta z\, \partial T/\partial y|_{y_0+\Delta y}$ and $q|_{z_0+\Delta z} = -\kappa\Delta x\, \Delta y\partial T/\partial z\, |_{z_0+\Delta z}$. $p\Delta x\Delta y\Delta z$ is the energy generation rate of that unit volume. κ and σ are the thermal conductivity, and the product of the

[1]Because major part of currents only passes through the channel, this approximation is more reasonable than setting the power generating sources distributed to the entire die.

material density and the heat in the unit volume, respectively. p is the density of power consumption in the unit volume.



$$dE/dt = \left(q|_{y_0+\Delta y} - q|_{y_0}\right) + \left(q|_{x_0+\Delta x} - q|_{x_0}\right) + \left(q|_{z_0+\Delta z} - q|_{z_0}\right) + p\Delta x\Delta y\Delta z$$

Figure 2.2: Energy conservation law and the heat conduction equation.

Based on the heat conduction mechanism, the temperature $T_d(\mathbf{r}, t)$ of die can be governed by the following heat transfer equations [51–59]

$$\sigma(T_d)\frac{\partial T_d(\mathbf{r}, t)}{\partial t} = \nabla \cdot (\kappa(T_d)\nabla T_d(\mathbf{r}, t)) + p(\mathbf{r}, t); \mathbf{r} \in D \quad (2.1)$$

$$\kappa(T_d)\frac{\partial T_d(\mathbf{r}, t)}{\partial n_{b_s}} + h_{b_s}T_d(\mathbf{r}, t) = f_{b_s}(\mathbf{r}). \quad (2.2)$$

Here, $\mathbf{r} = (x, y, z)$ that is the position at die, $\kappa(T_d)$ is the thermal conductivity (W/m·°C) of die, $\sigma(T_d)$ is the product of the material density and the specific heat (J/m^3·°C) of die, $p(\mathbf{r}, t)$ is the power density of heat source (W/m^3), ∇ is the diverge operator, $D = (0, L_x) \times (0, L_y) \times (-L_z, 0)$ is the dimension of die, L_x and L_y are the lateral lengths of die, L_z is the thickness of die, b_s is any specific boundary surface of the die, h_{b_s} is the heat transfer coefficient on b_s, $f_{b_s}(\mathbf{r})$ is an arbitrary function on b_s, and $\partial/\partial n_{b_s}$ is the differentiation along the outward direction which is normalized to b_s.

To provide reasonable accuracy of the temperature estimation with a little computational effort during executing early-stage temperature-aware optimization procedures, heat transfer

coefficients on the boundary surfaces of are suggested to be appropriately modeled [56–59]. Based on the model proposed by [56–59], the thermal model of the primary heat flow path can be modeled as an effective heat transfer coefficient h_p by combining the effect of each component on the primary heat flow path. Since the detailed layout of interconnects is not available in early design stages, interconnect layer is modeled as an equivalent thermal resistance based on the densities and the regularity structure assumption of metal and dielectric material [56–59]. Furthermore, the I/O pads and print circuit board (PCB) can also be modeled as an effective thermal resistance by using the technique proposed by [68]. Then, the equivalent heat transfer coefficient h_s of these successively connected thermal resistors can be calculated by the technique shown in [49]. After h_p and h_s have been obtained, $f_{b_s}(\mathbf{r})$'s for the top and bottom surfaces are set to h_sT_a and $-h_pT_a$, respectively [5, 49, 51–59]. Here, T_a is the ambient air temperature. Because of the chip and package structures, the area of vertical surface is strictly less than the area of horizontal surface, and the thermal conductivity of air is much less than the thermal conductivities of primary and secondary heat flow paths. Therefore, the boundary condition of each vertical surface can be reasonably set to be adiabatic [5].

Generally, the values of $\kappa(T_d)$ and $\sigma(T_d)$ are temperature dependent. The difference of peak temperature is about 5 °C between the result with temperature-dependent thermal parameters and the result with constant thermal parameters at 25 °C [55]. In current VLSI design, the on-die temperature can be in the degree of 100 °C. Under this situation, this difference may lead to about 5% error for the peak temperature of die. However, the effort to amend this error is relatively high because several iterations of the thermal simulation have to be executed for correcting the difference caused by the temperature dependences of $\kappa(T_d)$ and $\sigma(T_d)$. For practical purposes, these thermal parameters are usually treated as appropriate constants while performing temperature-aware floor-planning and placement [40–42].

The value of each thermal parameter can be found by applying a 1-D thermal circuit shown in Figure 2.3 to estimate the roughly average steady state temperature of the die. In Figure 2.3, the values of thermal resistors are $R_s = 1/A_{dz}h_s$, $R_p = 1/A_{dz}h_p$ and $R_{die} = \mathrm{D_T}/\kappa A_{dz}$. $T_{avg}(z)$ is the average steady state temperature on the lateral planes at arbitrary z position of die. Here, R_{die} can be viewed as a variable resistor when obtaining $T_{avg}(z)$ at certain z position. P_T is the

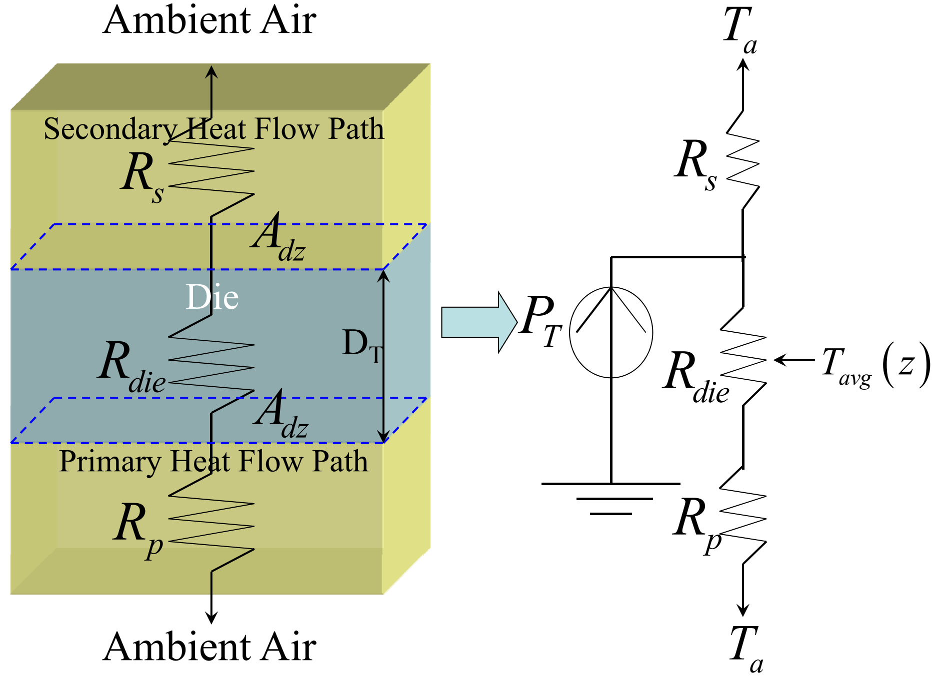


Figure 2.3: The 1-D thermal model for estimating the roughly steady state average temperature of die. The modeled thermal resistance network is shown in the right hand side.

total average steady state power consumption of die. A_{dz} is the cross area of die normal to the z-direction, and D_T is the thickness of die. The computation flow is processed as follows. In the beginning, R_{die} is calculated using the thermal conductivity of die at the room temperature. After thermal resistances R_s, R_p and R_{die} are obtained, the average rising temperature T_{avg} of die is equal to

$$T_{avg} = \frac{T_{avg}(0) + T_{avg}(-L_z)}{2}, \tag{2.3}$$

where $T_{avg}(0)$ and $T_{avg}(-L_z)$ can be obtained by solving the temperatures in the 1-D thermal circuit shown in Figure 2.3.

Once T_{avg} is calculated, R_{die} is re-calculated using the thermal conductivity of die at T_{avg}. This calculating procedure is repeated until T_{avg} converges. After that, the thermal parameters are calculated at the average temperature. With these estimated thermal parameters, the error of the peak temperature between the simulated temperature profiles for with and without considering the temperature dependence of the thermal parameters can be reduced.

With the above models, the heat diffusion equations for the rising temperature profile of die,

$T(\mathbf{r},t) = T_d(\mathbf{r},t) - T_a$, in early design stages can be written as

$$\sigma \frac{\partial T(\mathbf{r},t)}{\partial t} = \kappa \nabla^2 T(\mathbf{r},t) + p(\mathbf{r},t); \mathbf{r} \in D, \quad (2.4)$$

$$\left.\frac{\partial T(\mathbf{r},t)}{\partial x}\right|_{x=0,L_x} = \left.\frac{\partial T(\mathbf{r},t)}{\partial y}\right|_{y=0,L_y} = 0, \quad (2.5)$$

$$\kappa \left.\frac{\partial T(\mathbf{r},t)}{\partial z}\right|_{z=-L_z} = h_p T(x,y,-L_z,t), \quad (2.6)$$

$$\kappa \left.\frac{\partial T(\mathbf{r},t)}{\partial z}\right|_{z=0} = -h_s T(x,y,0,t). \quad (2.7)$$

Here, κ and σ are the thermal conductivity, and the product of the material density and the specific heat of die got by using the roughly steady state average temperature, respectively, and the initial condition $T(\mathbf{r},0) = 0$.

As shown in Figure 2.1, after dividing the layer with power generating sources into MN grids with M and N are numbers of divisions in x- and y- directions, respectively, the power density profile $p(\mathbf{r},t)$ stated in equation (2.4) can be written as

$$p(\mathbf{r},t) = \begin{cases} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} p_{mn}(t) \Pi_{mn}(x,y) & , \mathbf{r} \in D_p; \\ 0 & , \mathbf{r} \in D \backslash D_p. \end{cases} \quad (2.8)$$

Here, $D_p = (0,L_x) \times (0,L_y) \times (-j_d, 0)$, j_d is the junction depth of device, $\Pi_{mn}(x,y)$ is an indicative function with nonzero value being 1 only when (x,y) is in $[m\Delta x, (m+1)\Delta x] \times [n\Delta y, (n+1)\Delta y]$, $\Delta x = L_x/M$, $\Delta y = L_y/N$, m and n are indices of divisions, and $p_{mn}(t)$ is the power density waveform of grid cell (m,n) in the thin layer with thickness j_d.

For the transient (dynamic) thermal simulation, $p_{mn}(t)$ is a time-interval function with the magnitude of each interval being equal to the average power density of each time interval. We should note that the thermal time-constant of heat conduction is much larger than the clock period of circuit [51, 56]. As indicated by [56], the temperature takes at least 100K cycles to rise 0.1 °C. Practically, the time interval specified by the user can be much larger than the clock period of circuit. For the steady state thermal simulation, the input power profile is usually set to the steady power profile (the average power profile for a very long time period estimation) [51, 56]. Therefore, $p_{mn}(t)$ can be reasonably viewed as a step function with the magnitude being equal to its average power density for a long time period.

With the above discussion and governing equations (2.4)–(2.7), our goal is to get the rising temperature distribution of the die corresponding to the ambient temperature.

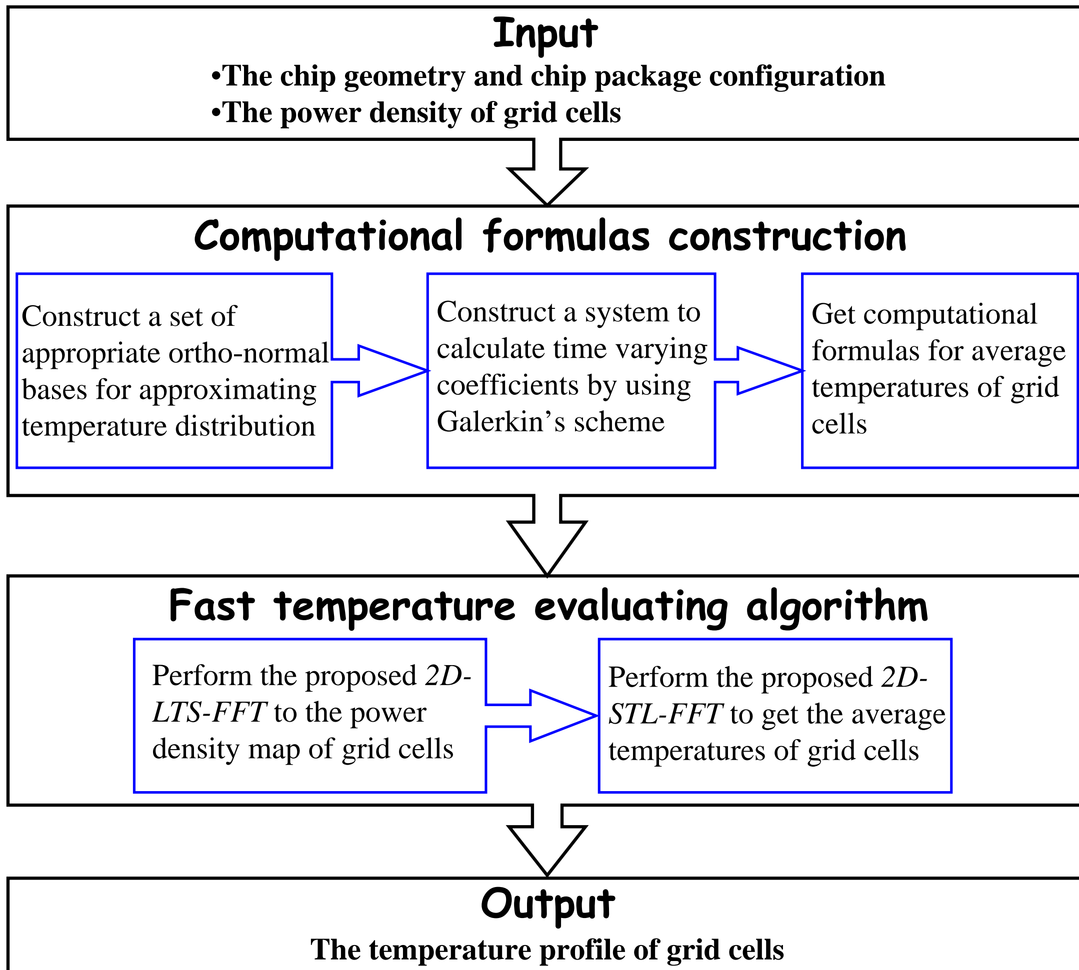


Figure 2.4: The executing flow of the proposed GIT based thermal simulation method.

2.2 Full-Chip Thermal Simulation

The executing flow of our GIT based thermal simulation method is summarized in Figure 2.4. After the chip geometry, package configuration and power density of grid cells are given, the compact thermal model described in section 2.1 is built. Then, the GIT based computational formulas of the on-chip temperature distribution are derived. As shown in the first major block (Computational formulas construction) of Figure 2.4, three steps are involved to construct the formulas. In the beginning, a set of appropriate bases is generated by a system-compatible auxiliary problem. After that, the temperature distribution can be expressed by these bases with suitable time-varying coefficients. With the Galerkin's scheme [65, 67], those time-varying coefficients can be found by an un-coupled system for estimating the temperature in the sense of least square residual approximation. Finally, the calculating formula of the average temperature

for each specific grid cell is obtained by averaging the temperatures in that grid area. After the temperature computational formulas are derived, we develop two efficient FFT like evaluating algorithms, *2D-LTS-FFT* and *2D-STL-FFT*, as shown in the second major block (Fast temperature evaluating algorithm) of Figure 2.4, to get the transformed coefficients for the power density map of grid cells and the desired temperature distribution, respectively[2].

In the rest of section 2.2, each sub-block of the two major blocks shown in Figure 2.4, the bounds of error decaying rates for [5] and our GIT based formula of calculating the average steady state temperature distribution, and the dynamic thermal simulation are discussed.

2.2.1 Auxiliary Problem for Generating Appropriate Spatial Bases

Several guidelines [65–67] need to be followed for choosing this auxiliary problem.

1. The auxiliary problem should be as similar as possible to the original problem.
2. The generated bases have to be completely ortho-normalized to ensure the property of convergence in mean for the approximated temperature distribution.
3. The ortho-normal bases should be time independent for the efficiency consideration.

The auxiliary problem can be introduced by considering the homogeneous problem which the temperature distribution satisfies equations (2.4)-(2.7) with $p(\mathbf{r}, t) = 0$. As stated in [65–67], the auxiliary problem can be set to be the following Sturm-Liouville problem with specific boundary conditions.

$$\nabla^2 \phi_{ilq}(\mathbf{r}) + \lambda_{ilq}^2 \phi_{ilq}(\mathbf{r}) = 0; \; \mathbf{r} = (x, y, z) \in D, \tag{2.9}$$

$$\left.\frac{\partial \phi_{ilq}(\mathbf{r})}{\partial x}\right|_{x=0,L_x} = \left.\frac{\partial \phi_{ilq}(\mathbf{r})}{\partial y}\right|_{y=0,L_y} = 0, \tag{2.10}$$

$$\kappa \left.\frac{\partial \phi_{ilq}(\mathbf{r})}{\partial z}\right|_{z=-L_z} = h_p \phi_{ilq}(x, y, -L_z), \tag{2.11}$$

$$\kappa \left.\frac{\partial \phi_{ilq}(\mathbf{r})}{\partial z}\right|_{z=0} = -h_s \phi_{ilq}(x, y, 0). \tag{2.12}$$

[2]In general, the leakage powers of gates are temperature dependent. Although the approach for solving this issue does not should in the executing flow of the proposed GIT based thermal simulation method, it can be easily handled using the temperature-power iterative framework presented in section 2.4.

The solutions of Sturm-Liouville problem form a set of completely ortho-normal spatial bases on the die, and the general forms of $\phi_{ilq}(\mathbf{r})$ and λ_{ilq}^2 can be obtained as follows [65].

$$\phi_{ilq}(\mathbf{r}) = \frac{\cos(\frac{i\pi x}{L_x})\cos(\frac{l\pi y}{L_y})\phi_q(z)}{\sqrt{N_{ilq}}}, \tag{2.13}$$

$$\lambda_{ilq}^2 = \lambda_{x_i}^2 + \lambda_{y_l}^2 + \lambda_{z_q}^2, \tag{2.14}$$

where i, l and q are non-negative integers, N_{ilq} is the normalized value being equal to $\theta_{il}L_xL_yN_{z_q}$, $\theta_{00} = 1/2$, $\theta_{i0} = \theta_{0l} = 1/4$, $\theta_{il} = 1/8$ with $i \neq 0$ and $l \neq 0$, $\lambda_{x_i}^2 = (i\pi/L_x)^2$, $\lambda_{y_l}^2 = \left(l\pi/L_y\right)^2$,

$$N_{z_q} = \frac{\left(h_p^2 + \kappa^2\lambda_{z_q}^2\right)\left(\frac{\kappa h_s}{h_s^2+\kappa^2\lambda_{z_q}^2} + L_z\right) + \kappa h_p}{\lambda_{z_q}^2}, \tag{2.15}$$

and

$$\phi_q(z) = \kappa\cos\left(\lambda_{z_q}(z + L_z)\right) + \frac{h_p}{\lambda_{z_q}}\sin\left(\lambda_{z_q}(z + L_z)\right). \tag{2.16}$$

Here, each λ_{z_q} is a positive value satisfying

$$\frac{\kappa^2\lambda_{z_q}^2 - h_p h_s}{\kappa\lambda_{z_q}(h_p + h_s)} = \cot(\lambda_{z_q}L_z). \tag{2.17}$$

To obtain each λ_{z_q}, we apply Newton-Raphson method [69] to equation (2.17) with the initial guess of each q being $\pi q/L_z$+$0.2\pi/L_z$ because the period of the right hand side in equation (2.17) is equal to π/L_z.

Each $\phi_{ilq}(\mathbf{r})$ is called as an eigenfunction, λ_{ilq}^2 is its eigenvalue, and $\lambda_{x_i}^2$, $\lambda_{y_l}^2$ and $\lambda_{z_q}^2$ are eigenvalues in x-, y- and z-directions, respectively. The physical meaning of $\phi_{ilq}(\mathbf{r})$ is that it presents the ilq-th free vibration with respect to the system described by equations (2.9)–(2.17), and its vibration frequencies are λ_{x_i}, λ_{y_l} and λ_{z_q} in x-, y- and z-directions, respectively. The physical meaning of λ_{ilq}^2 is that it presents the spectral magnitude of $\phi_{ilq}(\mathbf{r})$.

2.2.2 System Transformation for Time-Varying Coefficients

Since the generated bases $\{\phi_{ilq}(\mathbf{r})\}$ are completely ortho-normal in the spatial domain of die, $T(\mathbf{r}, t)$ can be approximated as the following finite integral transform pair [65–67].

$$T(\mathbf{r}, t) \approx \widehat{T}(\mathbf{r}, t) = \sum_{q=0}^{N_z-1}\sum_{l=0}^{N_y-1}\sum_{i=0}^{N_x-1}\psi_{ilq}(t)\phi_{ilq}(\mathbf{r}), \tag{2.18}$$

$$\psi_{ilq}(t) = \int_{-L_z}^{0}\int_0^{L_y}\int_0^{L_x} T(\mathbf{r}, t)\phi_{ilq}(\mathbf{r})dxdydz, \tag{2.19}$$

where each $\psi_{ilq}(t)$ is an unknown transformed time-varying coefficient, and N_x, N_y and N_z are truncation points in x-, y- and z- directions, respectively.

After utilizing the energy conservation law and Divergence theorem [66], and executing a series of derivations[3], the following un-coupled system is established to find each time-varying coefficient function $\psi_{ilq}(t)$.

$$\begin{cases} \sigma\psi'_{ilq}(t) = -\kappa\lambda^2_{ilq}\psi_{ilq}(t) + \widehat{p}_{ilq}(t), \\ \psi_{ilq}(0) = 0, \end{cases} \tag{2.20}$$

for $0 \leq i \leq N_x - 1, 0 \leq l \leq N_y -, 0 \leq q \leq N_z - 1$. In equation (2.20), the calculating formula of $\widehat{p}_{ilq}(t)$ is

$$\widehat{p}_{ilq}(t) = \int_{-j_d}^{0}\int_{0}^{L_y}\int_{0}^{L_x} p(\mathbf{r}, t)\phi_{ilq}(\mathbf{r})dxdydz. \tag{2.21}$$

Since equation (2.20) is un-coupled for different 'ilq', each $\psi_{ilq}(t)$ can be individually solved as

$$\psi_{ilq}(t) = \frac{1}{\sigma}\int_{0}^{t}\widehat{p}_{ilq}(\tau)e^{-\frac{k}{\sigma}\lambda^2_{ilq}(t-\tau)}d\tau. \tag{2.22}$$

For the steady state simulation, $p_{mn}(t)$ is a step function with its magnitude being equal to the average power density of grid (m, n) for a long time period. Thus, t is set to be infinity to find the steady state value of $\psi_{ilq}(\infty)$ which is $\widehat{p}_{ilq}(\infty)/(k\lambda^2_{ilq})$. Therefore, the evaluation of steady state temperature can be done without any time step approaching.

2.2.3 Average Rising Temperature Evaluation of Grid Cells

In general, hot spots occur in regions which are close to power sources. Hence, we focus on evaluating the average temperature of each grid cell on the top surface (z=0) of die[4]. First, we present the formulation to calculate the average rising temperature of steady state and discuss its decaying rate of truncation error. Then, the fast evaluating algorithms are developed for realizing the formulation. Finally, the transient (dynamic) thermal simulation is given.

[3]The detail description is shown in APPENDIX A.1.

[4]Our method can be used to find the average temperature of each grid cell at arbitrary lateral plane of the die by substituting suitable z into the bases.

Steady State Formulation

Plugging $\phi_{ilq}(\mathbf{r})$'s and $\psi_{ilq}(\infty)$'s into equation (2.18), the average steady state rising temperature $\overline{T}_{mn}$ for each grid cell (m, n) on the top surface of die is

$$\begin{aligned} \overline{T}_{mn} &= \frac{1}{\Delta x \Delta y} \int_{n\Delta y}^{(n+1)\Delta y} \int_{m\Delta x}^{(m+1)\Delta x} \widehat{T}(x, y, 0, \infty) dxdy \\ &= \sum_{l=0}^{N_y-1} \sum_{i=0}^{N_x-1} K_{il} \cos\left(\frac{i\pi(2m+1)}{2M}\right) \cos\left(\frac{l\pi(2n+1)}{2N}\right), \end{aligned} \tag{2.23}$$

where

$$K_{il} = \frac{\widehat{P}_{il}}{\kappa} \sum_{q=0}^{N_z-1} \frac{\Gamma_q C_{ilq}}{N_{ilq}} \phi_q(0), \tag{2.24}$$

$$C_{ilq} = \begin{cases} \frac{\Delta x \Delta y}{\lambda_{ilq}^2} & ; i = 0, l = 0 \\ \frac{4NL_y \Delta x \sin^2\left(\frac{l\pi}{2N}\right)}{l^2 \pi^2 \lambda_{ilq}^2} & ; i = 0, l \neq 0 \\ \frac{4ML_x \Delta y \sin^2\left(\frac{i\pi}{2M}\right)}{i^2 \pi^2 \lambda_{ilq}^2} & ; i \neq 0, l = 0 \\ \frac{16MNL_x L_y \sin^2\left(\frac{i\pi}{2M}\right) \sin^2\left(\frac{l\pi}{2N}\right)}{i^2 l^2 \pi^4 \lambda_{ilq}^2} & ; i \neq 0, l \neq 0 \end{cases} \tag{2.25}$$

and

$$\widehat{P}_{il} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} p_{mn} \cos\left(\frac{i\pi(2m+1)}{2M}\right) \cos\left(\frac{l\pi(2n+1)}{2N}\right), \tag{2.26}$$

$$\Gamma_q = \frac{2\kappa}{\lambda_{z_q}} \cos\left(\lambda_{z_q}(L_z - j_d/2)\right) \sin(\lambda_{z_q} j_d/2) - \frac{2h_p}{\lambda_{z_q}^2} \sin\left(\lambda_{z_q}(L_z - j_d/2)\right) \sin(\lambda_{z_q} j_d/2), \tag{2.27}$$

where p_{mn} is the average power density of grid (m, n) for a long time period, and M and N are numbers of divisions in the x- and y-directions, respectively.

An error bound of employing $\overline{T}_{mn}$ to approximate the temperature in a gird cell is given by **Theorem 1** stated in APPENDIX A.2. As shown in **Theorem 1**, the error decaying rate of employing $\overline{T}_{mn}$ to approximate the temperature in a gird cell is dominated by $i^2 l^2 \lambda_{z_q}((i\pi/L_x)^2 + (l\pi/L_y)^2 + \lambda_{z_q}^2)$. To compare the above error decaying rate with the Green's function based method's [5], the boundary conditions and power source location are set to be the same and substituted in to their formula. As shown in APPENDIX A.2, the error decaying rate of our GIT based formulation is in the order of $i^2 l^2((i\pi/L_x)^2 + (l\pi/L_y)^2 + \lambda_{z_q}^2))$, and the error decaying rate of [5] is in the order of $i^2 l^2 \sqrt{(i\pi/L_x)^2 + (l\pi/L_y)^2}$. Therefore, the error decaying rate of the proposed GIT based method is faster than that of [5]. The reason is that the bases in z-direction

of the GIT based method are different with [5], and our constructed bases can fully fill the eigenspace of heat diffusion equation. This fact leads to different coefficients in the approximating form even if the bases in x- and y- directions of our GIT based method are the same with [5]. Furthermore, the error decaying rate of the proposed GIT based method is not only faster than that of [5], the experimental results also show that it can maintain the same accuracy level of [5] even if its truncation points, N_x and N_y are far less than the numbers of divisions, M and N.

Although the truncation points N_xN_y can be far less than the number of grid cells MN, there is no actual efficiency improvement over [5] if we directly apply the standard FFT to evaluate each $\overline{T}_{mn}$. The reason is that the standard IFFT (Inverse Fast Fourier Transform) algorithm needs to pad zeros to the input data when the dimension of input data is less than the dimension of output data, such as equation (2.23). Moreover, the dimension of output data in standard FFT algorithm is restricted to be equal to the dimension of input data. However, the dimension of output data in equation (2.26) is only N_xN_y which is far less than its dimension of input data, MN. To overcome this limitation, we develop FFT like fast evaluating algorithms for our GIT formulation in the next subsection.

Fast Evaluating Algorithms for GIT Formulation

To efficiently realize our formulation for the steady state temperature distribution, we first derive a one-dimensional radix-two based FFT like algorithm for the length of output data being larger than the length of input data, *1D-STL-FFT*. Then, based on *1D-STL-FFT*, we develop a one-dimensional FFT like algorithm for the length of output data being smaller than the length of input data, *1D-LTS-FFT*. Finally, we extend these one-dimensional algorithms to two two-dimensional algorithms by the row-column procedure, and we call them as *2D-STL-FFT* and *2D-LTS-FFT*. Finally, these two algorithms are integrated to calculate equations (2.23) and (2.26). The computational complexity of our GIT based thermal simulator can be analyzed to be only $\underline{O(MN\log_2 N_xN_y)}$. The overview of the above evaluating algorithms are shown in Figure 2.5. Given the power density profile of chip, *2D-STL-FFT* computes the transformed coefficients of power density profile, and *2D-LTS-FFT* transforms these transformed coefficients to obtain the average rising temperature of grid cells.

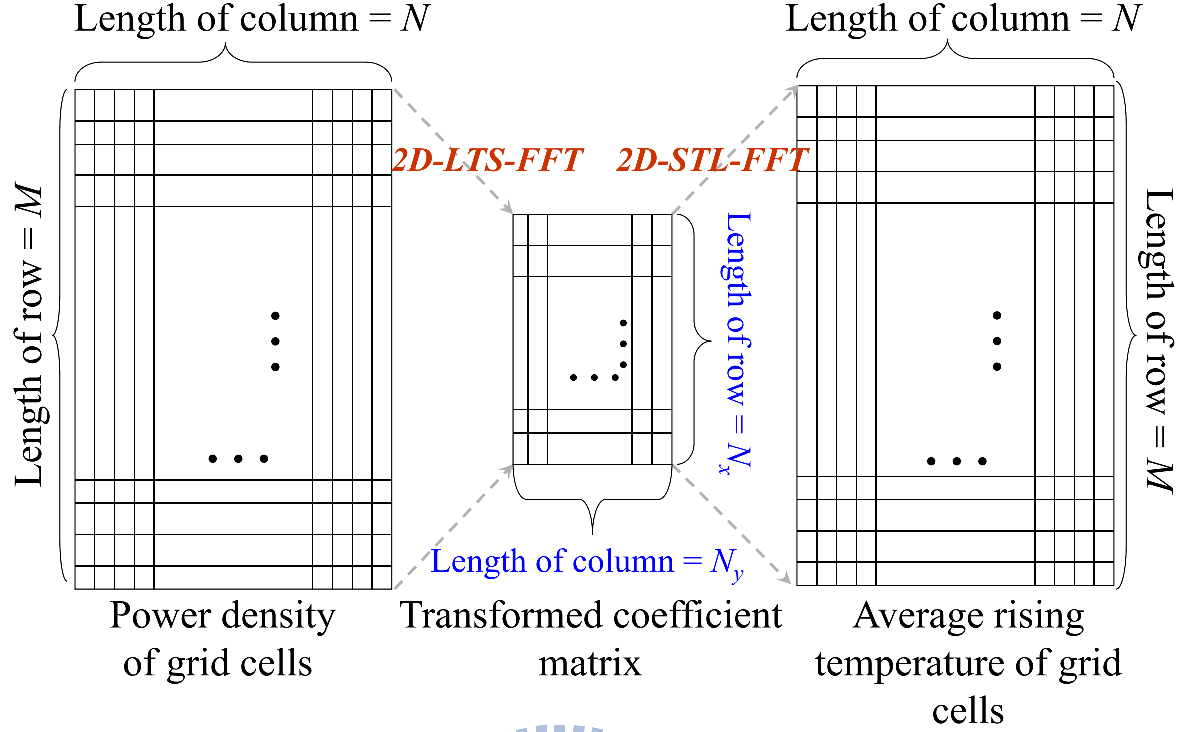


Figure 2.5: The overview of using *2D-SLT-FFT* and *2D-LTS-FFT* to evaluate the average rising temperature of grid cells.

1D-STL-FFT The prototype of *1D-STL-FFT* is

$$\overline{F}_k = \sum_{i=0}^{\widetilde{M}-1} f_i e^{j2\pi ik/2M}; \quad k = 0, \cdots, 2M-1, \tag{2.28}$$

where $\widetilde{M} < M$ and both are power of 2, $j = \sqrt{-1}$, and f_i's and $\overline{F}_k$'s are complex input and output data with lengths being equal to $\widetilde{M}$ and M, respectively.

Because the length of $\overline{F}_k$'s is larger than the length of f_i's, the zeros-padding step of f_i's like in the standard FFT algorithm needs to be avoided for saving the runtime. Therefore, the *1D-STL-FFT* algorithm shown in Figure 2.6 is developed to calculate equation (2.28) without the zeros-padding. In Figure 2.6, the "**Reverse-bit**" means the reverse-bit algorithm [69].

In the beginning, the "**Reverse-bit**(f)" reorders the input data for those sub DFTs (Discrete Fourier Transforms) which will be generated by recursively performing the Danielson-Lanczos Lemma (DL-Lemma) [69] to the prototype of *1D-STL-FFT* in equation (2.28). The DL-Lemma is used to rewrite the original DFT as the sum of two sub DFTs with half output length. One of the two is formed from the even-numbered points of the input data, and the other is formed from the odd-numbered points. In this step, the DL-Lemma is used recursively for these two

Algorithm Radix-two *1D-STL-FFT*
Input: *Complex vector f with length $\widetilde{M}$*
Output: *Complex vector $\overline{F}$ with length $2M$*

1 **Begin**
2 $f_{\mathbf{R}}$ = **Reverse-bit**(f) ;
3 $L = 4M/\widetilde{M}$;
4 $N_{SubDFTs} = \widetilde{M}/2$;
5 **For** $SubIndex = 0$ to $N_{SubDFTs} - 1$
6 $k = L \times SubIndex$;
7 $i = 2 \times SubIndex$;
8 **For** $SubK = 0$ to $L - 1$
9 $\overline{F}[k] = f_{\mathbf{R}}[i] + f_{\mathbf{R}}[i+1] \times e^{j2\pi \times SubK/L}$;
10 $k = k + 1$;
11 **EndFor**
12 **EndFor**
13 Apply the bottom up procedure of standard FFT to execute the Danielson-Lanczos Lemma $\log_2 \widetilde{M} - 1$ times for evaluating $\overline{F}$
14 **End**

Figure 2.6: Procedure of *1D-STL-FFT*.

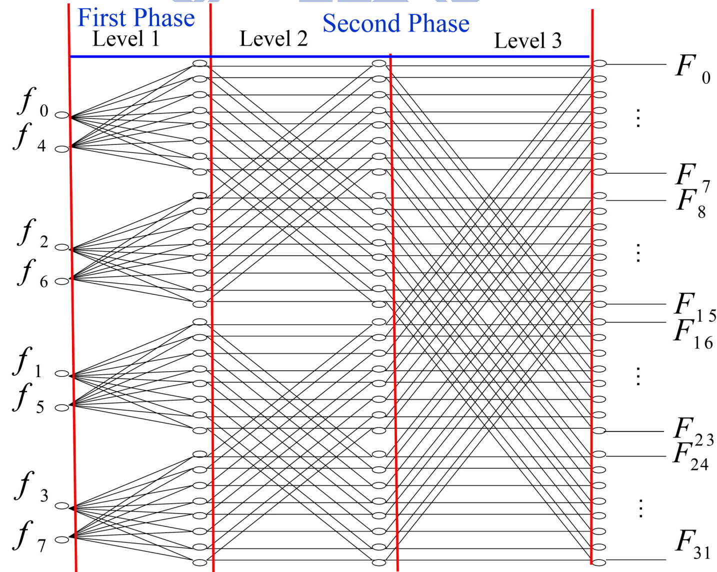


Figure 2.7: The sketch of the computational flow for *1D-STL-FFT* with $\widetilde{M} = 8$ and $M = 16$.

sub DFTs. Because $\widetilde{M}$ is less than M, this bisecting procedure is executed only $\log_2 \widetilde{M}$ times, and we have $\log_2 \widetilde{M}$ bisecting levels. After *Line* 2 in Figure 2.6 is performed, the *1D-STL-FFT* algorithm evaluates the output of those L sub DFTs in the bottom level by using *Lines* 3~12, and performs *Line* 13 to get the output of remaining levels. An example with $M = 16$ and $\widetilde{M} = 8$ is given in Figure 2.7. There are 3 bisecting levels, and 4 sub DFTs in the bottom level.

Algorithm Radix-two *1D-LTS-FFT*
Input: *Real vector* $\widehat{f}$ *with length* M
Output: *Complex vector* $\widehat{F}$ *with length* $\widetilde{M}$

```
1  Begin
2    f̂_R = Reverse-bit(f̂) ;
3    N_SubDFTs = 2M/M̃ ;
4    For Sub_i = 0 to N_SubDFTs − 1
5      Start = Sub_i × M̃ ;
6      End = Start + M̃;
7      F_t(Start : End − 1) = 1D-LTS-FFT(f̂_R(Start/2 : End/2 − 1));
8    EndFor
9    L = M̃ ;
10   For level = 0 to log_2(M/M̃)
11     n = 0 ;
12     Sub_i = 0 ;
13     N_SubDFTs = N_SubDFTs/2 ;
14     While Sub_i < N_SubDFTs
15       For i = 0 to M̃ − 1 ;
16         i* = i + Sub_i × M̃ ;
17         F_t[i + n] = F_t[i*] + F_t[i* + M̃] × e^{j2πi/L} ;
18       EndFor
19       Sub_i = Sub_i + 2 ;
20       n = n + M̃ ;
21     EndWhile
22     L = 2 × L ;
23   EndFor
24   F̂ = F_t(0 : M̃ − 1) ;
25 End
```

Figure 2.8: Procedure of *1D-LTS-FFT*.

After performing the reverse-bit algorithm to the input data, two phases are executed. The first phase is done by using *Lines* 3~12 of Figure 2.6. The second phase is to get the output of the remaining levels by executing the bottom up procedure of standard FFT as stated in *Line* 13 of Figure 2.6.

The complexity of *1D-STL-FFT* is $O(M \log_2 \widetilde{M})$ since there are $\log_2 \widetilde{M}$ bisecting levels and each complexity is $O(M)$.

1D-LTS-FFT The prototype of *1D-LTS-FFT* is

$$\widehat{F}_i = \sum_{m=0}^{M-1} \widehat{f}_m e^{j2\pi i m/2M}; \quad i = 0, \cdots, \widetilde{M} - 1, \tag{2.29}$$

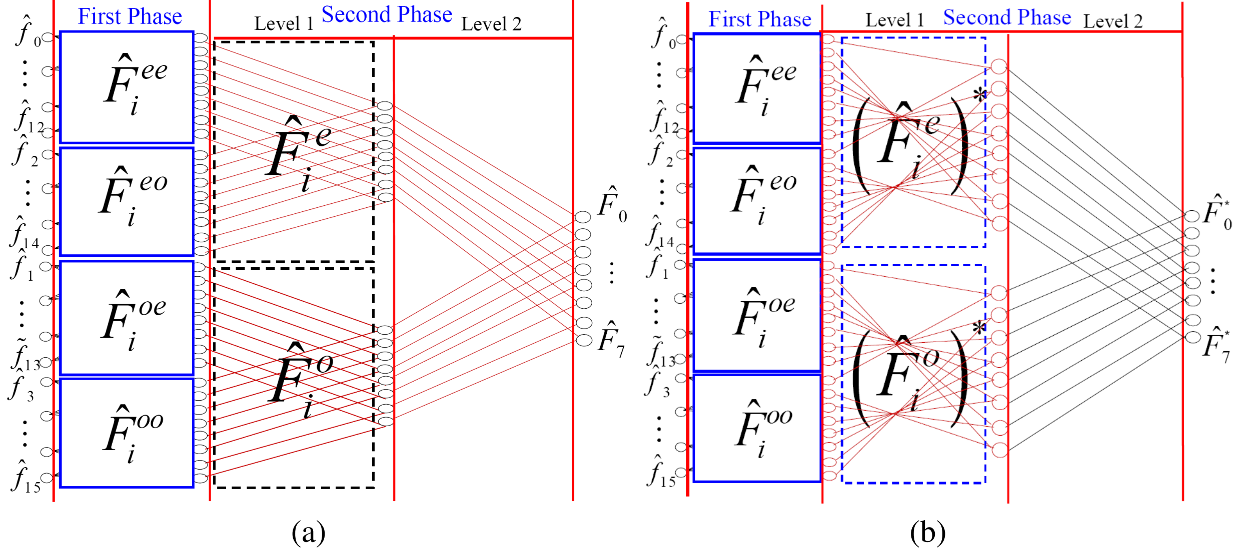


Figure 2.9: The sketch of the computational flow for *1D-LTS-FFT* with $\widetilde{M} = 8$ and $M = 16$. (a) The *1D-LTS-FFT*. (b) The *1D-LTS-FFT* for negative frequencies.

where $\widetilde{M} < M$, and $\widehat{f_m}$ and $\widehat{F_i}$ are real input and complex output data with lengths being equal to M and $\widetilde{M}$, respectively.

Applying the DL-Lemma to the prototype of *1D-LTS-FFT* for generating $\log_2(M/\widetilde{M}) + 1$ bisecting levels, $\widehat{F_i}$ can be written as the sum of $2M/\widetilde{M}$ sub DFTs. Each sub DFT has the same form as the *1D-STL-FFT* with the lengths of input and output being equal to $\widetilde{M}/2$ and $\widetilde{M}$, respectively. Two phases are utilized to evaluate $\widehat{F_i}$, and the *1D-LTS-FFT* algorithm is shown in Figure 2.8. First, *Line* 2 performs the reverse-bit algorithm to the input data, and *Lines* 4~8 use the *1D-STL-FFT* algorithm to obtain each bisected sub DFT. After each sub DFT has been done, a bottom up procedure is applied to the remaining $\log_2(M/\widetilde{M}) + 1$ bisecting levels for finding $\widehat{F_i}$, and the executing steps are from *Line* 9 to *Line* 24.

An example with $M = 16$ and $\widetilde{M} = 8$ is shown in Figure 2.9.(a). In the first phase, the input data are reordered by using the reverse-bit algorithm, and the reordered data are fed into the corresponding *1D-STL-FFT* blocks. This can be done by using *Lines* 3~8 in Figure 2.8. Then, the output of top block in the level 1 of the second phase is calculated by

$$\widehat{F_i^e} = \widehat{F_i^{ee}} + e^{j2\pi i/16}\widehat{F_i^{eo}}, \tag{2.30}$$

and $\widehat{F_i^o}$ can be done by a similar way. Finally, $\widehat{F_i}$ is equal to

$$\widehat{F_i} = \widehat{F_i^e} + e^{j2\pi i/32}\widehat{F_i^o}. \tag{2.31}$$

Algorithm Radix-two *2D-STL-FFT*
Input: *Complex matrix* $\overline{K}$ *with length* $N_x \times N_y$
Output: *Complex matrix* $\overline{F}$ *with length* $2M{\times}2N$
1 **Begin**
2 **For** i = 0 to $N_x - 1$
3 $T_{Row}(i, 0 : 2N - 1) = \mathit{1D\text{-}STL\text{-}FFT}\big(\overline{K}(i, 0 : N_y - 1)\big)$;
4 **EndFor**
5 **For** j = 0 to $2N - 1$
6 $\overline{F}(0 : 2M - 1, j) = \mathit{1D\text{-}STL\text{-}FFT}(T_{Row}(0 : N_x - 1, j))$;
7 **EndFor**
8 **End**

Figure 2.10: Procedure of *2D-STL-FFT*.

The second phase is summarized in *Lines* 9~24 of Figure 2.8.

For the general case, the sub DFTs in each level of the second phase can be obtained by combining those sub DFTs of their previous level with the similar formula of equation (2.30) by replacing 16 to be $2^1\widetilde{M}, 2^2\widetilde{M}, \cdots, 2M$ in each level. The computational complexity of the first phase is $O(M \log_2 \widetilde{M})$ because the *1D-STL-FFT* needs to be executed $2M/\widetilde{M}$ times, and each complexity is $O(\widetilde{M} \log_2 \widetilde{M})$. The complexity is $O(M)$ for the second phase. Hence, the computational complexity of *1D-LTS-FFT* is $O(M \log_2 \widetilde{M})$.

Temperature Evaluation The average rising temperature of steady state shown in equation (2.23) can be got as

$$\overline{T}_{mn} = \frac{1}{2} R_e \left\{ \overline{F}_{m,n} + \overline{F}_{2M-(m+1),n} \right\}, \tag{2.32}$$

where $R_e\{\cdot\}$ is the real part operator, and

$$\overline{F}_{k_1,k_2} = \sum_{i=0}^{N_x-1} \sum_{l=0}^{N_y-1} \overline{K}_{il} e^{\frac{j2\pi i k_1}{2M}} e^{\frac{j2\pi l k_2}{2N}}. \tag{2.33}$$

Here, $0 \leq k_1 \leq 2M - 1$, $0 \leq k_2 \leq 2N - 1$, $\overline{K}_{il} = K_{il} e^{j2\pi i/4M} e^{j2\pi l/4N}$, and each K_{il} is equal to equation (2.24).

To obtain $\overline{T}_{mn}$'s, the values of $\overline{F}_{k_1,k_2}$'s and K_{il}'s need to be firstly obtained. Therefore, as shown in Figure 2.10, a row-column based *2D-STL-FFT* method is developed to calculate $\overline{F}_{k_1,k_2}$'s by utilizing the *1D-STL-FFT* algorithm. In Figure 2.10, *Lines* 2~4 perform the *1D-STL-FFT* for each row of the input matrix $\overline{K}$ which each (i, l) entry is $\overline{K}_{il}$, and *Lines* 5~7 apply

the *1D-STL-FFT* to each column of the output matrix got from the row procedure for obtaining the desired matrix $\overline{F}$ which each (k_1, k_2) entry is $\overline{F}_{k_1,k_2}$. The complexity for obtaining $\overline{F}_{k_1,k_2}$'s is $\underline{O(MN \log_2 N_x N_y)}$ because the complexities of row and column procedures are $O(N_x N \log_2 N_y)$ and $O(NM \log_2 N_x)$, respectively.

To calculate each K_{il} from equation (2.24), $\widehat{P}_{il}$'s need to be known from equation (2.26). Therefore, the two dimensional prototype with the similar form as equation (2.29) is needed to get related $\widehat{F}_{i,l}$'s for the input data being p_{mn}'s. A row-column based *2D-LTS-FFT* algorithm can be constructed by using the similar procedure shown in Figure 2.10 with the *1D-STL-FFT* replaced by the *1D-LTS-FFT*. The *2D-LTS-FFT* method is then used to get those related $\widehat{F}_{i,l}$'s.

However, equation (2.32) can not be utilized to calculate $\widehat{P}_{il}$'s because the lengths of those related $\widehat{F}_{i,l}$'s in the row and column directions are less than $2M$ and $2N$, respectively. Therefore, the complex conjugates of $\widehat{F}_{i,l}$'s are required to complete the calculation of $\widehat{P}_{il}$'s. Fortunately, the complex conjugate of the output from each sub *1D-STL-FFT* in calculating $\widehat{F}_{i,l}$'s can be directly obtained by reversing these sub DFTs. Therefore, the complex conjugates of $\widehat{F}_{i,l}$'s can be got by reversing the data of F_t in *Line* 7 of Figure 2.8, and performing *Lines* 9~24 in Figure 2.8 during the row-column procedure of $\widehat{F}_{i,l}$'s.

The complexity of row procedure for obtaining those related $\widehat{F}_{i,l}$'s is $O(MN \log_2 N_y)$ because the *1D-LTS-FFT* needs to be executed $2M$ times. The complexity of column procedure is $O(N_y M \log_2 N_x)$ because the *1D-LTS-FFT* needs to be executed N_y times. Hence, the complexity for obtaining $\widehat{F}_{i,l}$'s is $O(MN \log_2 N_x N_y)$. The complexity for calculating the complex conjugates of $\widehat{F}_{i,l}$'s is $O(MN) + O(N_y M)$ since only the second phase needs to be recomputed. Therefore, the complexity for computing equation (2.26) is $\underline{O(MN \log_2 N_x N_y)}$.

From the above discussion, we conclude that the complexity of our GIT based thermal simulator is $\underline{O(MN \log_2 N_x N_y)}$. Finally, the completely proposed simulating algorithm is illustrated in Figure 2.11.

Transient (Dynamic) Thermal Simulation

While performing the dynamic thermal simulation, each $p_{mn}(t)$ can be modeled as a user-specified time interval function with the magnitude in each interval being equal to the average power in each time interval. By using equation (2.22), each time-varying coefficient

Input: Geometries of die and package, and related thermal parameters.
The steady power density of grid cells.
Output: The average steady state rising temperature $\overline{T}_{mn}$ for each grid cell (m, n).

Pre-calculating stage

1. Set thermal parameters of die by using the roughly average temperature obtained by the simplified 1-D model described in section 2.1.

2. Obtain the eigenfunctions and eigenvalues described in section 2.2.1.

3. Obtain C_{ilq} and the summation term for each q in equations (2.25) and (2.24), respectively.

Post-calculating stage

1. Obtain $\widehat{P}_{il}$ by *2D-LTS-FFT* discribed in section 2.2.3, and K_{il} in equation (2.24).

2. Obtain $\overline{K}_{il}$ in equation (2.33) and feed it into *2D-STL-FFT* described in section 2.2.3. Then, apply equation (2.32) to obtain $\overline{T}_{mn}$.

Figure 2.11: Simulating algorithm of the proposed steady state thermal simulator.

$\psi^t_{ilq} \equiv \psi_{ilq}(t)$ is

$$\psi^t_{ilq} = \psi^{t-\Delta t}_{ilq} + \frac{\widehat{p}^t_{ilq}}{\kappa\lambda^2_{ilq}}(1 - e^{-\frac{k}{\sigma}\lambda^2_{ilq}\Delta t}), \tag{2.34}$$

where Δt is the time step and is equal to the time interval of power density waveforms, $\widehat{p}^t_{ilq}$ is equal to equation (2.21) with $p(\mathbf{r}, t)$ being equal to the average power density profile in the time interval $(t - \Delta t, t)$, and $\psi^{t-\Delta t}_{ilq} = \psi_{ilq}(t - \Delta t)$.

After ψ^t_{ilq}'s are calculated, the average temperature of each grid cell at the sampling time t can be obtained by equation (2.18) with the same evaluating method presented in section 2.2.3. In addition, applying equation (2.34) to compute each ψ^t_{ilq} wouldn't induce any un-stable issue with a large Δt because equation (2.34) is the exact solution of the system equation (2.20), *i.e.* without the error caused from finite difference approximations such as the backward-Euler method, the trapezoidal method and the Runge-Kutta method. Furthermore, since the thermal time constant of heat conduction is much larger than the clock period of circuit [51, 56], the time step Δt can be far larger than the clock period of the circuit to save runtime with acceptable errors.

2.3 Thermal Simulation for Stacked-Layer 3-D ICs

As mentioned in section 1.3, the thermal issue will be one major concern for 3-D ICs. Recently, the tradeoff between the circuit performance and the thermal issue of early-stage 3-D ICs design has been studied by estimating the uniform average temperature of each layer [32, 33]. However, the uniform average temperature loses information of the spatial temperature gradient. To efficiently obtain the non-uniform temperature distribution, we develop a fast 3-D IC thermal simulator by combining the GIT and numerical schemes. This thermal simulator is developed for the 3-D ICs with the wire bonded, microbump-3D package, face-to-face, contactless interconnect structures shown in the Figure 1.7 (a)–(f) in section 1.3.

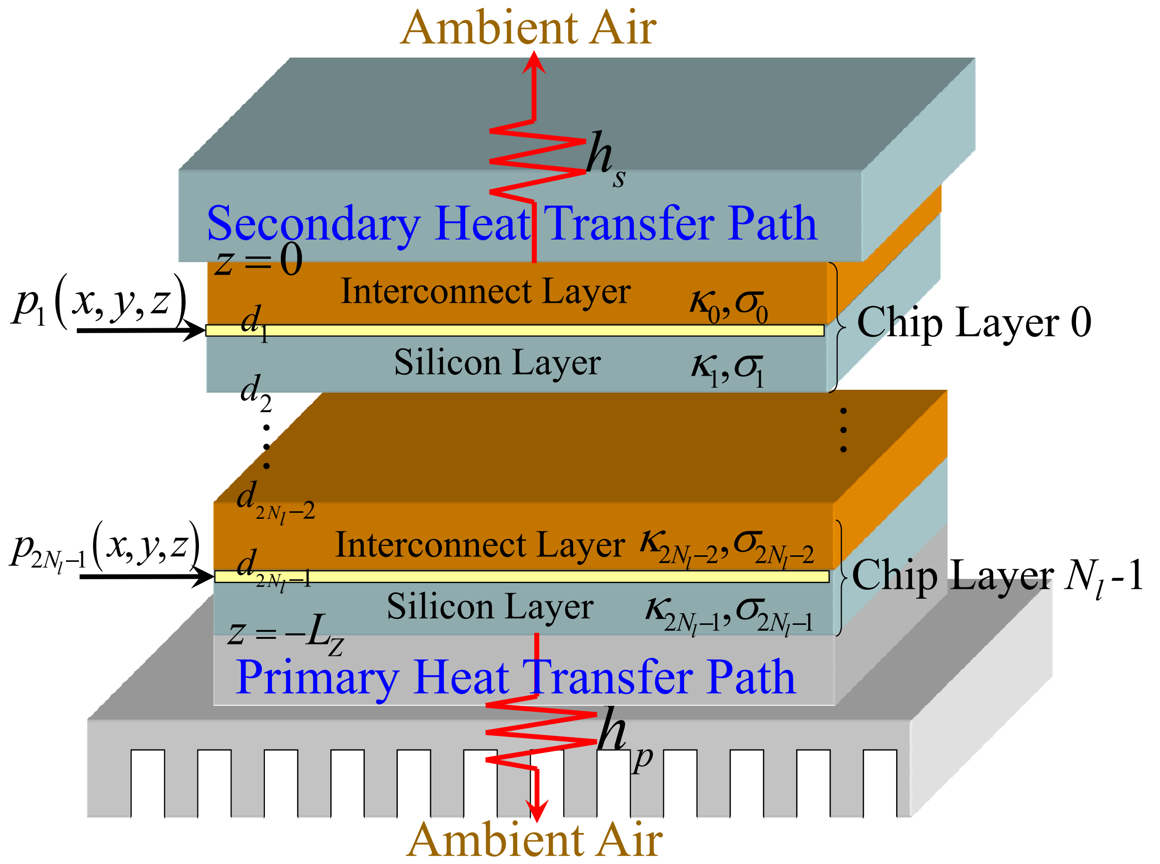


Figure 2.12: The schematic diagram of a 3-D IC with N_l chip layers.

As shown in Figure 2.12, the structure of 3-D ICs is a multilayer structure with stacking silicon and insulator layers one by one [30, 32, 33]. The power sources are distributed in a thin layer close to the top surface of each active silicon layer in the z-direction, and each insulator layer consists of Cu, ILD and glue materials. The heat transfer equations of 3-D ICs can be built

by combining the governing equations of each layer with suitable boundary conditions. The heat diffusion equation inside each layer is similar to equation (2.4) with their corresponding thermal parameters κ_ζ and σ_ζ. Here, ζ is the layer index. The boundary conditions on the lateral surfaces are flux isolated, and the boundary conditions at $z = -L_z$ and $z = 0$ are convection types with equivalent heat transfer coefficients h_p and h_s for the primary and secondary heat flow paths[5], respectively.

With the thermal model show in Figure 2.12, the corresponding heat transfer equations can also be governed to simulate the temperature profile of 3-D ICs with stacked-layer structures. By using a similar deviation stated in Appendix A.1, the heat transfer equations of 3-D ICs can be transformed into a one-dimensional subproblem by utilizing the following ortho-normal spatial bases in the x- and y-directions.

$$\phi_{il}(x,y) = \frac{1}{\sqrt{N_{il}}}\cos(\frac{i\pi x}{L_x})\cos(\frac{l\pi y}{L_y}), \tag{2.35}$$

where $N_{il} = \rho_{il}L_xL_y, \rho_{00} = 1, \rho_{i0} = \rho_{0l} = 1/2$ and $\rho_{il} = 1/4$ with $i \neq 0, l \neq 0$. These ortho-normal spatial bases satisfy the following two dimensional Sturm-Liouville problem.

$$\lambda_{il}^2\phi_{il}(x,y) = -\nabla^2\phi_{il}(x,y); \quad (x,y) \in D_{xy}, \tag{2.36}$$

where $D_{xy} = (0, L_x) \times (0, L_y)$ and $\lambda_{il}^2 = \lambda_{x_i}^2 + \lambda_{y_l}^2$. The boundary conditions of equation (2.36) are flux isolated and equal to equation (2.10) with replacing $\phi_{ilq}(\mathbf{r})$ by $\phi_{il}(x,y)$.

Since $\phi_{il}(x,y)$'s are ortho-normal spatial bases, the approximated rising temperature $\widehat{T}(\mathbf{r},t)$ can be expressed as

$$\widehat{T}(\mathbf{r},t) = \sum_{i=0}^{N_x-1}\sum_{l=0}^{N_y-1}\psi_{il}(z,t)\phi_{il}(x,y), \tag{2.37}$$

where each $\psi_{il}(z,t)$ is an unknown function, and needs to be found.

Combining the interface conditions, the temperature continuity and the heat flux conservation law on the interface of two different layers, performing Galerkin's scheme along the x- and y-directions, and using equation (2.36), each $\psi_{il}(z,t)$ can be got by solving the following

[5]Although different materials in the primary and secondary heat flow paths of Figure 2.1 are described by effective heat transfer coefficients for the fast temperature estimation, those materials should be modeled as an inhomogeneous structure for the further accuracy consideration. Since the structures of the components in the primary and secondary heat flow paths are also layer stacked, its non-homogeneity can also be handled by the proposed simulation method

one-dimensional sub-problem.

$$\sigma_\zeta \frac{\partial}{\partial t}\psi_{il}(z,t) = \kappa_\zeta\left(\frac{\partial^2}{\partial^2 z}\psi_{il}(z,t) - \lambda_{il}^2\psi_{il}(z,t)\right) + \widehat{p}_{il,\zeta}(z,t), \quad (2.38)$$

$$\psi_{il}(z,t)|_{z=d_\zeta^+} = \psi_{il}(z,t)|_{z=d_\zeta^-}, \quad (2.39)$$

$$\kappa_{\zeta-1}\left.\frac{\partial\psi_{il}(z,t)}{\partial z}\right|_{z=d_\zeta^+} = \kappa_\zeta \left.\frac{\partial\psi_{il}(z,t)}{\partial z}\right|_{z=d_\zeta^-}, \quad (2.40)$$

$$\left.\frac{\partial\psi_{il}(z,t)}{\partial z}\right|_{z=0} = h_s\psi_{il}(0,t), \quad (2.41)$$

$$\kappa_{2N_l-1}\left.\frac{\partial\psi_{il}(z,t)}{\partial z}\right|_{z=-L_z} = h_p\psi_{il}(-L_z,t). \quad (2.42)$$

where

$$\widehat{p}_{il,\zeta}(z,t) = \int_0^{L_y}\int_0^{L_x} p_\zeta(x,y,z,t)\phi_{il}(x,y)dxdy, \quad (2.43)$$

and ζ is the layer index satisfying $1 \le \zeta \le 2N_l - 1$, d_ζ is the position of the ζ-th interface in the z-direction, $p_\zeta(x,y,z,t)$ is the power density in the thin layer of the ζ-th active silicon substrate and is equal to zero as ζ is even (insulator layer), and each $\psi_{il}(z,0) = 0$.

Though the ortho-normal spatial bases in the z-direction of the above one-dimensional sub-problem can be analytically solved by the *sign-count* method [65] or the method proposed in [70], their computational efforts[6] are relatively high for the practical purpose. Hence, we adopt the numerical scheme to obtain $\psi_{il}(z,t)$ because its runtime is linear in the number of grid points along the z-direction.

By discretizing this one-dimensional sub-problem along the z-direction, the value of $\psi_{il}(z,t)$ at each grid point in the z-direction can be obtained by the following matrix equation.

$$\mathbf{G}_{il}\boldsymbol{\psi}_{il}(t) + \mathbf{C}\boldsymbol{\psi}_{il}^{'}(t) = \mathbf{p}_{il}(t), \quad (2.44)$$

where $\boldsymbol{\psi}_{il}(t) = [\psi_{il}(z_0,t),\cdots\psi_{il}(z_r,t),\cdots\psi_{il}(z_{\Lambda-1},t)]^T$, z_r's are positions of grid points in the z-direction, $z_0 = 0$, $z_{\Lambda-1} = -L_z$, and Λ is the number of grid points. The $\mathbf{G}_{il}$'s and $\mathbf{C}$ are tri-diagonal and diagonal matrices, respectively, and $\mathbf{p}_{il}(t)$ is

$$\mathbf{p}_{il}(t) = [0,...0,\widehat{p}_{il}(d_1,t),0,...0,\widehat{p}_{il}(d_3,t),0...0,\widehat{p}_{il}(d_{2N_l-1},t),0...0]^T. \quad (2.45)$$

[6]The complexity of *sign-count* method [65] for obtaining the ortho-normal spatial bases in the z-direction for each 'il' is proportional to "$\#Layers \times \sum_{q=0}^{q=N_z-1} K_{ilq}$". Here, N_z is the truncation number in the z-direction, and K_{ilq} is the sign-count iterations for obtaining the eigenvalue λ_{ilq} of each ortho-normal spatial basis. The complexity of using [70] to obtain the spatial bases in the z-direction for each 'il' is extremely high because it needs symbolic expression for the determinant of a $\#Layers \times \#Layers$ matrix and needs to perform the inverse Laplace transform.

When performing steady state thermal simulation, $\mathbf{p}_{il}(t)$ is a constant vector, and $\psi'_{il}(t)$ is a zero vector. Hence, $\psi_{il}(\infty)$ can be obtained without the time step evaluation. Moreover, because each $\mathbf{G}_{il}$ is tri-diagonal, each $\psi_{il}(\infty)$ can be solved in linear time. After solving $\psi_{il}(\infty)$, the steady state temperature of equation (2.37) at any z position of grid point can be cast into the similar form developed for 2-D ICs, and the proposed fast evaluating method can be used to calculate the temperature.

The transient analysis can be done by performing the time step evaluation to equation (2.44) for getting the value of $\psi_{il}(t)$ at each time step. Then, the proposed evaluating method is used to calculate the temperature at each time step. Note that, each $\mathbf{G}_{il}$ will not change after functional blocks are replaced. Hence, once the LU decompositions of $\mathbf{G}_{il}$'s are done, they can be reused during the temperature-aware design flow.

2.4 Approach to Handle the Temperature Dependent Issue of Leakage Powers

Our algorithm can be extended to deal with the temperature dependence issue of leakage power by combining the widely used temperature-power iterative framework [33,55–59] with our proposed thermal simulation method. The executing flow is shown in Figure 2.13. In the beginning,

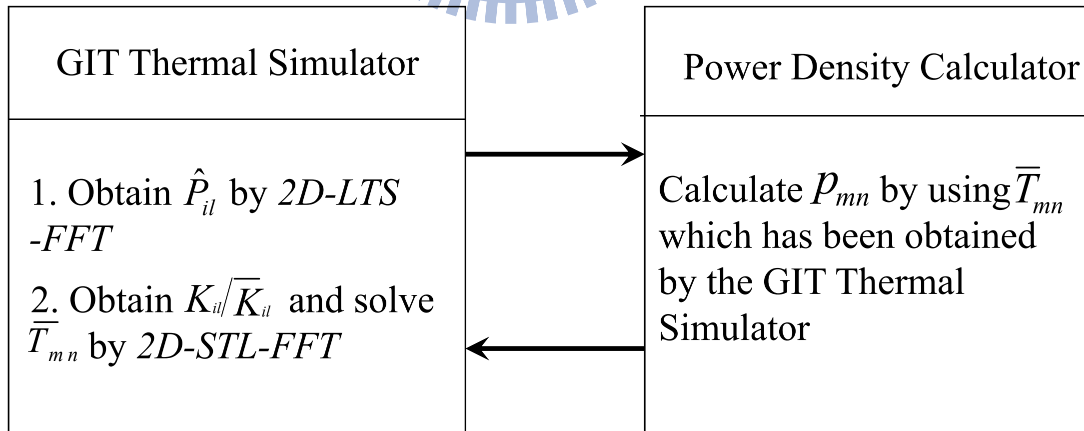


Figure 2.13: Temperature-power iterative framework for dealing with the temperature dependence issue of leakage power.

the power density profile can be obtained by setting the chip temperature to be room temperature. Then, the power density profile can be immediately updated by applying temperature-power iterative framework to the 1-D thermal model before performing the detail thermal sim-

ulation. After that, the temperature-power iterative framework are performed by using the GIT thermal simulator and power density calculator until they converge. One should note that the pre-calculating stage only needs to be executed once during this iterative framework since it is independent of the power density profile.

Remarks: The average temperature in the power source layer of each grid cell can be easily obtained by integrating equation (2.18) from $-j_d$ to 0 and converting it to the form suitable for performing *2D-STL-FFT* to get more accuracy result. However, the difference between top surface temperature and the average temperature in the power source layer of each grid cell is very small because the thickness of power source layer is very thin.

2.5 Experimental Results

We implement the proposed GIT based thermal simulator and the Algorithm II of a highly efficient Green's function based method [5] in C++ language. The state-of-the-art FFT package, FFTW [71], is used to realize the DCT and IDCT for [5]. All methods are tested on a HP xw9300 workstation with 16 GB memory. For demonstrating the accuracy, the results of proposed method and [5] are compared with that of the commercial computational fluid dynamic software, ANSYS.

2.5.1 Accuracy and Fast Convergence of the GIT Based Thermal Simulator

A chip, DEC Alpha 21264 [72], is employed to demonstrate the accuracy of our method, and its size is scaled down to 3.3 mm $\times$ 3.3 mm $\times$ 0.5 mm for the 65 nm technology. Its floorplan is shown in Figure 2.14(a), and its die and package geometries are shown in Figure 2.14(b). The equivalent thermal resistance of the package is set to be 45.5 °C/W [68]. The interconnect layer consists of 25% copper and 75% oxide with the thickness being equal to 0.06 *mm*, and its effective thermal conductivity is 101W/(m·°C). The thickness of the power source layer is set to be 20 *nm* which is the nominal value of the device junction depth for the 65 *nm* technology [73]. The equivalent heat transfer coefficient of the primary heat flow path, h_p, is 8700 W/(m^2·°C) [5], and the equivalent heat transfer coefficient of the secondary heat flow path, h_s, is 2017 W/(m^2·°C)

To appropriately set the thermal conductivity of die, we apply the 1-D thermal model shown in Figure 2.3 to compute the average temperature of die. To calculate the thermal resistance R_p, we apply the formula stated in [49, 51] to obtain $R_p = 1/(h_p A_{dz}) = 10.55\,^{\circ}\mathrm{C/W}$. Here, A_{dz} is the cross area of die among the z-direction. The R_s is equivalent thermal resistance of the successively connected package and interconnect layers which is equal to 45.52 °C/W. The room temperature T_a is set to be 27 °C. Based on the iteration process stated in section 2.1, the average temperature of die is calculated as 90.9°C, the thermal conductivity of die is 113.5 W/(m·°C), and $R_{die} = 0.4\,^{\circ}\mathrm{C/W}$.

The top surface of die is divided into 128×128 grid cells and the average power density profile is shown in Figure 2.14(c). The average steady state rising temperature distribution on the top surface of the die computed by the proposed method with the truncation points being 32 in each x-, y- and z-direction is shown in Figure 2.14(d). The maximum relative error compared with the result of ANSYS is 0.24%, and its relative error distribution is shown in Figure 2.14(e). The relative error of each grid cell (m, n) is measured by

$$e_{mn} = \left| \frac{T_{mn}^{ANSYS} - \overline{T}_{mn}}{T_{mn}^{ANSYS}} \right|, \tag{2.46}$$

where T_{mn}^{ANSYS} is the average rising temperature of grid cell (m, n) obtained by ANSYS. Note that, the $T_{avg}(0) = 65.14\,^{\circ}\mathrm{C}$ got by the 1-D thermal model is consistent with the $T_{avg}(0) = 65.15\,^{\circ}\mathrm{C}$ got by the proposed GIT based method. This verifies the ability of 1-D thermal model for predicting the average temperature of the entire die.

To further demonstrate our fast error decaying rate, we plot the maximum relative errors with different truncation points in Fig 2.14(f). The result shows that the proposed GIT based analyzer can achieve an extremely accurate solution even when the truncation points are very small.

2.5.2 Thermal Simulation for the Full-Chip Containing Lots of Functional Blocks

To demonstrate the capability of the proposed GIT based method for the thermal simulation of full-chip with containing lots of functional blocks and the efficiency improvement over the Algorithm II of [5], a test chip with dimension of $1\,\mathrm{cm} \times 1\,\mathrm{cm} \times 0.5\mathrm{mm}$ and one million functional

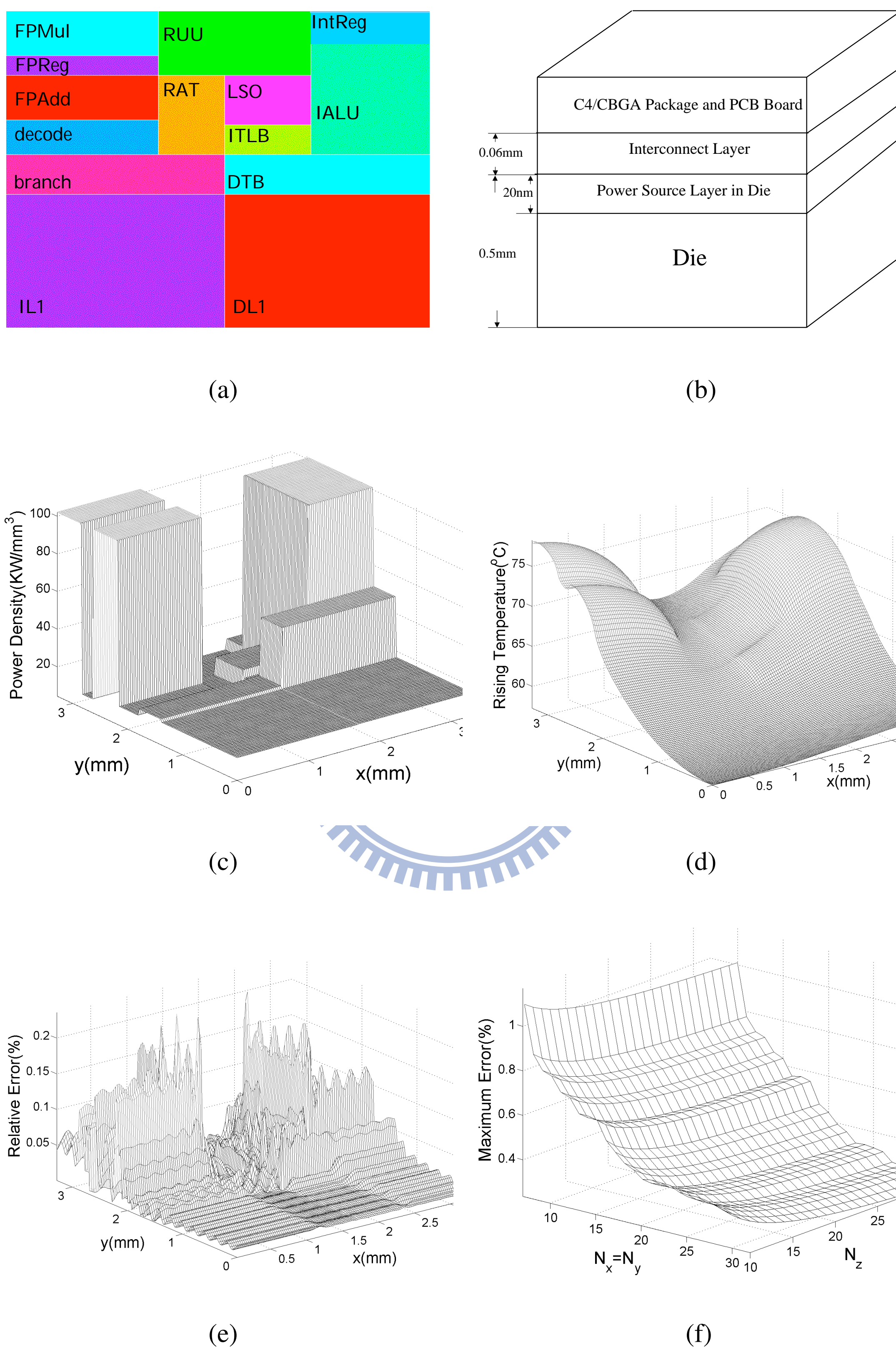


Figure 2.14: Accuracy and the maximum error trend of a test chip. (a) Floorplan, (b) geometries of the test chip, (c) power distribution, (d) the rising temperature distribution of the top surface of the die, (e) the relative error distribution, and (f) the maximum relative error versus truncation point.

blocks is considered. The top surface of the chip is set to be adiabatic, and the power sources are assumed to be attached on the top surface of the die[7]. The setting is consistent with the setting in [5]. Figure 2.15(a) shows the power density distribution of the functional blocks in W/cm^2. The top surface of the chip is divided into 1024×1024 grid cells. The truncation points of our GIT based method are $16 \times 16 \times 8$ and the truncation points of [5] are 2048×2048 to achieve the same maximum error level. The average rising temperature distribution of the top surface got by our GIT based method is shown in Figure 2.15(b), and the maximum error is 0.3576% presented in Table 3.5.

		Algorithm II of [5]	Our method
number of functional blocks		1 million	
number of grid cells		2^{20}	
number of bases		2^{22}	2^{11}
maximum error (%)		0.4143	0.3576
runtime (sec)	pre-calculating	2.47850	0.00005
	post-calculating	2.7642	0.1312
speedup (post-calculating)		21.0686	

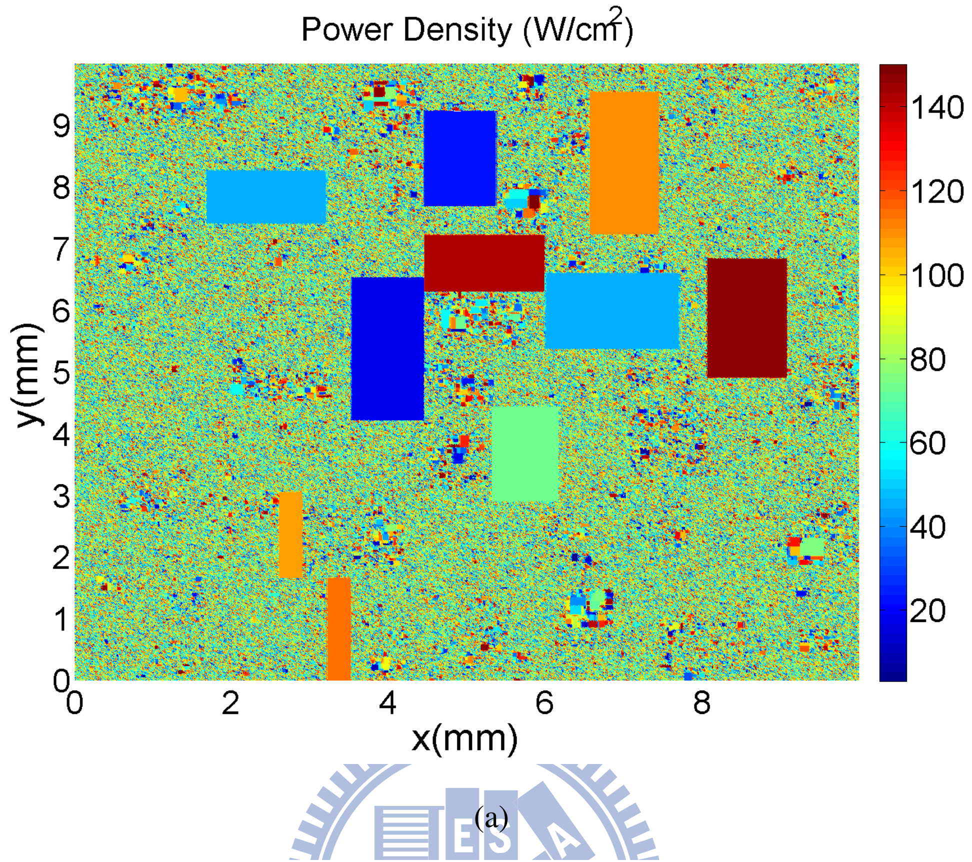
Table 2.1: Accuracy and Runtime Comparison of the proposed GIT based method and the Algorithm II of [5].

The runtime comparison is shown in Table 3.5. The runtime of the post-calculating stage in our method is 0.1312 seconds while the runtime of the post-calculating stage in [5] is 2.7642 seconds. The speedup of our method over [5] is 21.07 at the post-calculating stage. This result demonstrates the substantial efficiency improvement of our thermal analyzer over [5].

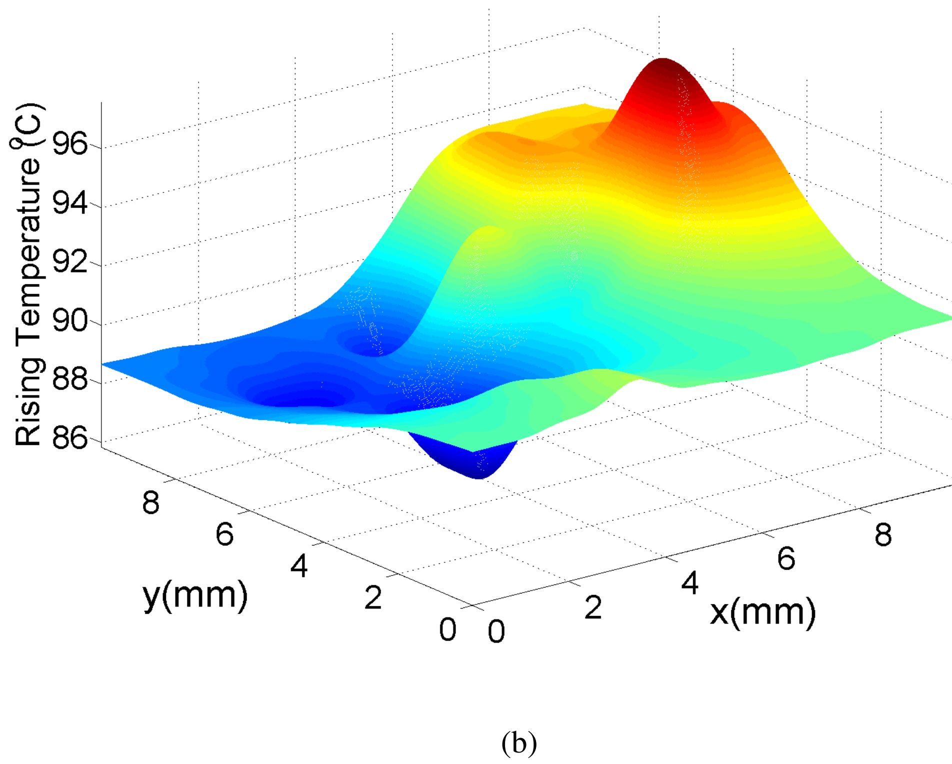
2.5.3 Accuracy and Efficiency of the GIT Based Thermal Simulator for the 3-D IC Thermal Analysis

To demonstrate the accuracy of our GIT based thermal simulator for 3-D ICs, three chip layers are stacked and the power sources are distributed in three thin layers with the thickness being equal to the device junction depth. The lateral dimension of each chip layer is 3.3 mm $\times$ 3.3 mm. The thicknesses of insulator and silicon layers on both top and middle chips are scaled down to 15 μm and 10 μm, respectively. The thicknesses of insulator and silicon layers (including the substrate) for the bottom chip are 15 μm and 500 μm, respectively. The thermal parameters of

[7]The power sources which are attached on the top surface of the die can be easily handled by deriving the integral transform pair with the assumption of the plane-power density on the top surface of die. The general solution can be found in [65–67].



(a)

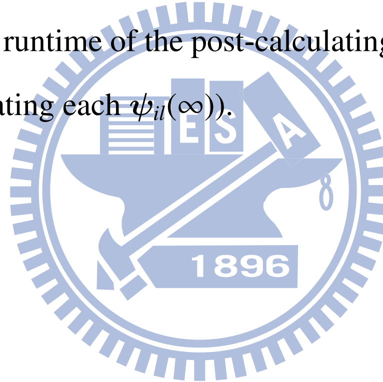


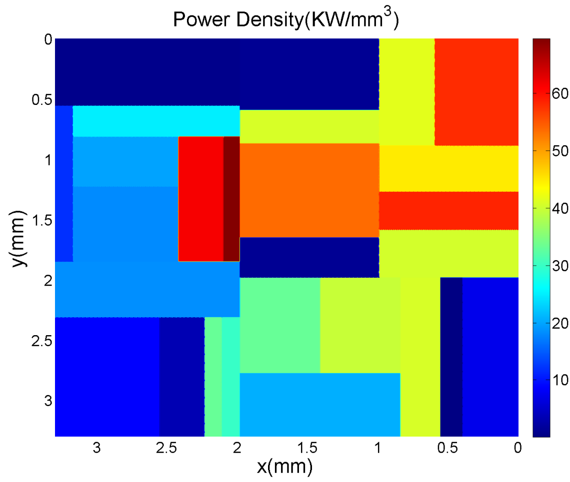
(b)

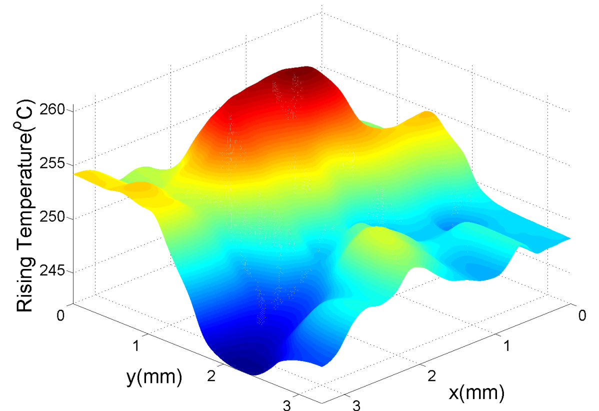
Figure 2.15: The power density and temperature distribution of a 1 cm × 1 cm chip with one million functional blocks. (a) The power density distribution, and (b) the rising temperature distribution.

each layer are referred to [32]. The top surface of each silicon layer is divided into 128×128 grid cells. The truncation point is 32 in each x- and y-direction, and the number of sampling points in the z-direction is 10 for each layer. Comparing with the result of ANSYS, our maximum error is 0.24% which demonstrates the accuracy of our method for 3-D ICs.

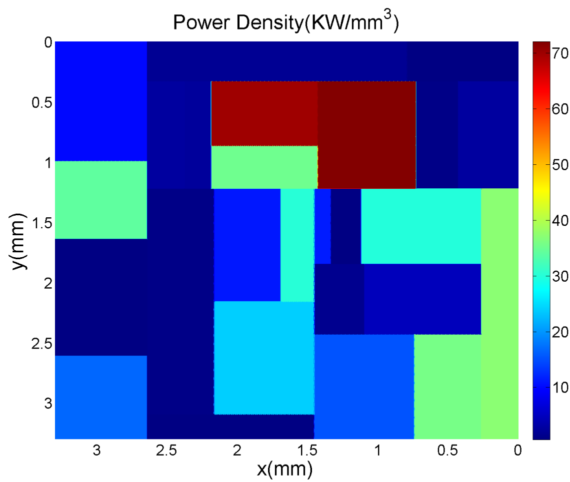
To show the efficiency of our GIT based method for the cell-level thermal analysis in 3-D ICs, the top surface of each silicon layer is divided into 1024×1024 grid cells to mimic 1.05 million power sources. The truncation point and the number of sampling points in the z-direction are the same as the case of 128×128 grid cells. The average power density profile of each silicon layer is shown in Figure 2.16(a), (c) and (e). The estimated average steady state rising temperature distribution on the top surface of each silicon layer is shown in Figure 2.16(b), (d), and (f) from the top layer to the bottom layer. The runtime of our GIT based method is 0.031 seconds for the pre-calculating stage (including the LU decomposition of each tri-diagonal matrix $\mathbf{G}_{il}$). The runtime of the post-calculating stage is only 0.48 seconds (including 0.016 seconds for calculating each $\psi_{il}(\infty)$).

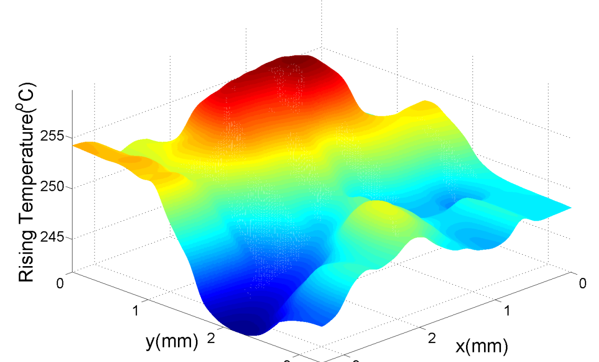




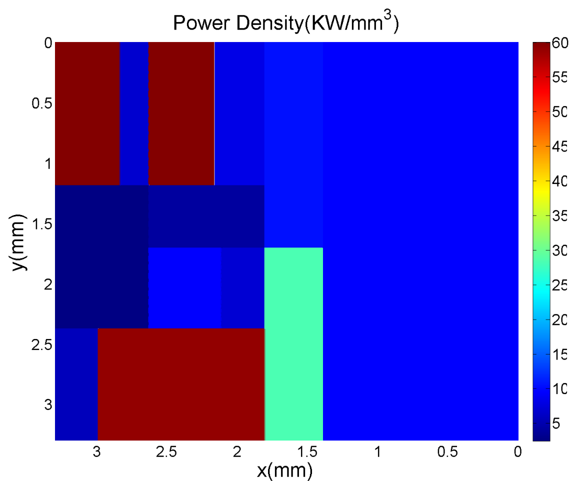


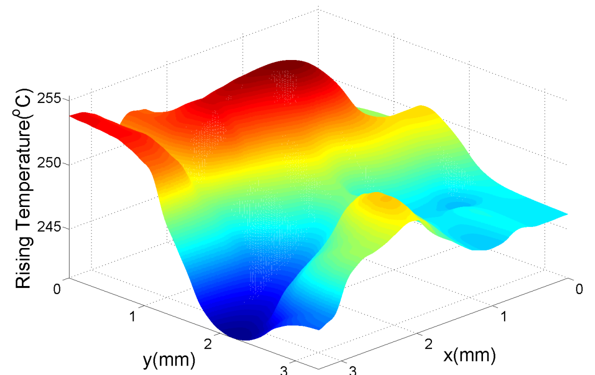
(a) (b)





(c) (d)





(e) (f)

Figure 2.16: Power density and temperature distribution of a test 3-D chip. Figures (a), (c) and (e) are the power density profiles on the top surface of the top, middle and bottom silicon layers, respectively. Figures (b), (d) and (f) are the temperature distribution on the top surface of the top, middle and bottom silicon layers, respectively.

Chapter 3

Simulation Method II – *An Efficient Method for Analyzing the Process Variations Considered On-Chip Thermal Reliability*

The context of this chapter is organized as follows. Firstly, section 3.1 describes, the illustration of the importance of statistical electro-thermal simulation, the concept and the essentiality of the on-chip *thermal yield* profile, and the accuracy comparison between the proposed leakage current models and several existing leakage current models. Then, section 3.2 introduces the problem formulation, the modeling technique of the device parameters, and th concept of the Hermite polynomial chaos (H-PC). After that, the developed statistical electro-thermal analyzer is detailed in section 3.3. Finally, experimental results are given in section 3.4.

3.1 Motivation Illustrations

3.1.1 Electro-Thermal Coupling Issue under Process Variations

On-chip power consumption consists of dynamic and leakage powers. Basically, dynamic power consumption weakly and negatively depends on the operating temperature. On the contrary, leakage power consumption is sensitive to the operating temperature and the variations of device parameters. Under the *nominal* value of device parameters, as pointed out by [26], the thermal analyzer should take into account the electro-thermal coupling mechanism to ensure the thermal reliability. The electro-thermal coupling mechanism is proceeded as follows. With an initial temperature of the chip, the initial power consumption of the chip is obtained. Based on

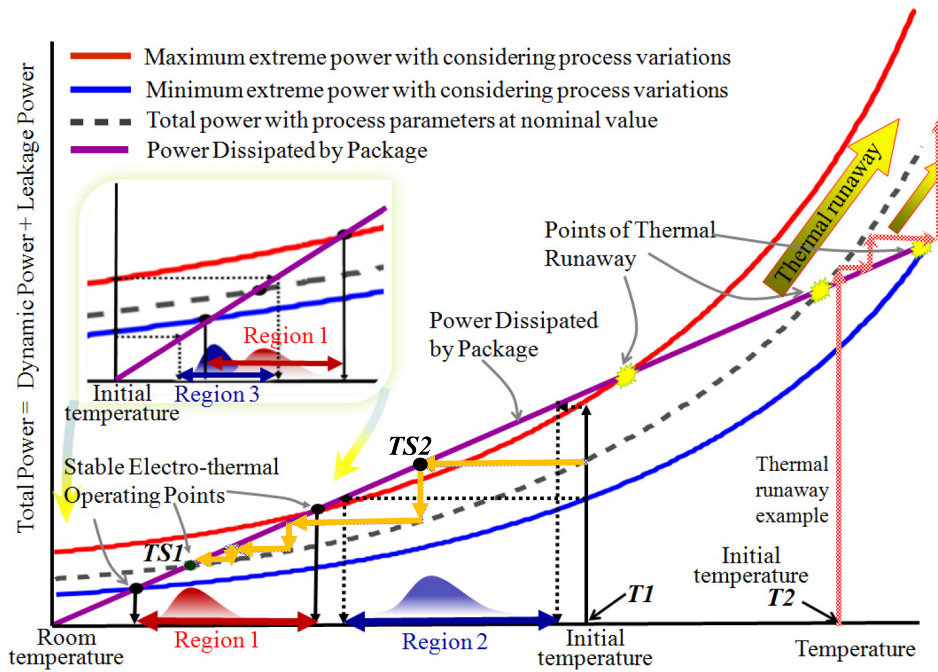


Figure 3.1: An example for the electro-thermal coupling mechanism under process variations.

the zeroth law of thermodynamics [27], the on-chip temperature increases because the surplus power consumption that cannot be dissipated will transform into heat for achieving the equilibrium between the power consumption of chip and the power dissipated by the package and cooling system. On the other hand, as the power dissipation capacity of package is larger than the generating power of system, the on-chip temperature decreases. Since the leakage power is temperature dependent, the on-chip power consumption is updated. The above mechanism repeats until a stable operating temperature is achieved. Otherwise, the chip thermally runs away if the chip is operated at an inappropriately initial temperature [26].

Under process variations, the equilibrium temperature is no longer a deterministic value and can no longer be predicted by a deterministic thermal analyzer. As shown in Figure 3.1, The red and blue curves are the maximum and minimum on-chip total power consumptions that a chip operates at different temperatures under process variations, respectively. With an initial temperature *T1*, the distribution of equilibrium temperatures falls into Region 1 if the electro-thermal coupling effect is considered while performing the statistical thermal analysis. However, the distribution of the temperature falls into Region 2 if the electro-thermal coupling effect is not considered. On the other hand, with a different initial temperature such as the room temperature shown in the sub-plot of Figure 3.1, the temperature distribution falls into

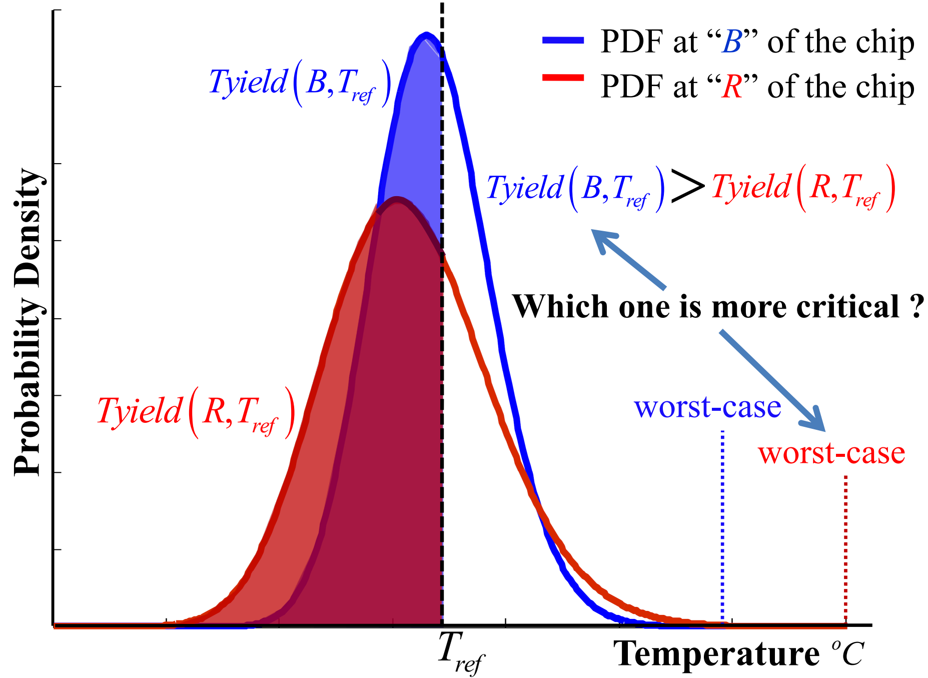


Figure 3.2: PDFs of on-chip temperature values at two different positions (B and R) of a die for indicating which one is the *statistically hot-spot* location.

a different Region 3 if the electro-thermal coupling effect is not considered while performing the statistical thermal analysis. However, the equilibrium temperature distribution still falls into Region 1 if the electro-thermal coupling effect is considered.

Therefore, the uncertainty of the confidential region of equilibrium temperature and the drastic errors of Region 2 or Region 3 show that it is necessary to consider the electro-thermal coupling effect while performing the statistical thermal analysis.

3.1.2 Concept of On-Chip Thermal Yield Profile

Because of process variations, the on-chip temperature at an arbitrary position **r** is a random variable. To identify possible hot-spot regions of a chip, its *thermal yield profile*, $Tyield(\mathbf{r}, T_{ref})$, can be defined as *the probability profile of the on-chip temperature at arbitrary position* **r** *being at or less than a reference temperature* T_{ref}.

To illustrate the concept of the thermal yield, two probability density functions (PDFs) of the on-chip temperature values at two different positions ('R' and 'B') are shown in Figure 3.2. 'R' is indicated to be more critical than 'B' by the conventional worst-case thermal analysis. However, the probability of temperature at 'B' being at or less than a specific temperature (T_{ref})

is larger than that of 'R'. Generally, the result of the worst-case thermal analysis without further considering the statistical behavior of on-chip temperature values might lead to an immoderately conservative related thermal cost for thermal-aware optimization engines. To design effectively, the location that more likely exceeds the tolerable temperature needs to be well-concerned. Therefore, the design around location 'B' should be concerned more seriously than that of 'R' since it has a smaller thermal yield, $Tyield(B, T_{ref})$. According to the above discussion, an efficient on-chip thermal yield profile analyzer is essential to provide useful related thermal cost for thermal-aware optimization engines under process variations.

Instead of applying the thermal yield profile, one can realize that the figure of merit for identifying statistical hot-spot locations is ambiguous if only the mean and variance profiles are provided [9]. For example, if only the mean profile of on-chip temperature distribution is used as a figure of merit, it is very likely (about 50%) to incorrectly indicate hot-spot locations. Furthermore, if only the mean profile ($\mu_T(\mathbf{r})$) and standard deviation profile ($\sigma_T(\mathbf{r})$) of on-chip temperature distribution are provided, by utilizing the Chebyshev inequality, a large temperature value is estimated to ensure the 90% lower bound of thermal reliability, i.e. T_{ref} needs to be $\mu_T(\mathbf{r}) + 3\sigma_T(\mathbf{r})$ to ensure $\mathbf{Prob}(T(\mathbf{r}) \leq T_{ref}) \geq 0.9$. Here, $T(\mathbf{r})$ is the statistical on-chip temperature profile. Since the Chebyshev inequality does not always get a tight lower bound for any type of random variable[1], T_{ref} might be an immoderately conservative constrain for the thermal reliability. This undesirable phenomenon can result in the immoderate guard-banding for the circuit design.

3.2 Preliminaries

3.2.1 Leakage Power Modeling

The leakage currents of a gate not only depend on physical device parameters and operating temperature but also on its input patterns [16, 20, 74, 75]. To build the leakage power models, different input patterns, physical parameters and operating temperatures are set for each gate in the cell library, and HSPICE simulation is performed with the industry design kit to generate

[1]For example, suppose that x is a standard normal random variable, $\mathbf{Prob}(x \leq 1.28\sigma_x) = 0.9$. However, the Chebyshev inequality requires a larger reference value to obtain the same probability as the lower bound, i.e. $\mathbf{Prob}(x \leq 3\sigma_x) \geq 0.9$.

Table 3.1: Accuracy comparison of leakage power models for an NAND gate under 65*nm* technology node. The results of HSPICE simulation with TSMC model card are employed to be the reference solution. The second column represents the fitting components of $f_g(L, t_{ox}, T)$ and $f_s(L, t_{ox}, T)$ adopted by the models proposed by [6–8] and our proposed models.

	$f_g(L, t_{ox}, T)$	max. error	avg. error	error > 3%
Without temperature	t_{ox}, L, t_{ox}^2, L^2 [6,76]	6.48%	2.70%	4.37%
With temperature	L, t_{ox}, T	3.20%	0.97%	0.35%
	†L, t_{ox}, T, t_{ox}^2	1.55%	0.29%	0.00%

	$f_s(L, t_{ox}, T)$	max. error	avg. error	error > 3%
Without temperature	$L, t_{ox}, t_{ox}^2, t_{ox}^{-1}$ [6]	347.32%	70.65%	98.27%
	$L, t_{ox}, Lt_{ox}, L^2, t_{ox}^2, t_{ox}^{-1}, Lt_{ox}^{-1}, L^{-1}t_{ox},$ [7,76]	314.13%	70.52%	100.00%
With temperature	L, T, t_{ox} [8]	32.23%	8.73%	76.62%
	(L, t_{ox}, T) are fully expanded to 2nd order $\Longrightarrow$ $L, t_{ox}, T, Lt_{ox}, t_{ox}T, TL, L^2, t_{ox}^2, T^2$	10.31%	1.53%	8.47%
	† (L, t_{ox}, T) are fully expanded to 3rd order $\Longrightarrow$ $L, t_{ox}, T, Lt_{ox}, t_{ox}T, TL, L^2, t_{ox}^2, T^2, Lt_{ox}T,$ $L^2t_{ox}, t_{ox}^2T, T^2L, L^3, t_{ox}^3, T^3$	1.31%	0.19%	0.00%

† The adoptive forms of f_g and f_s in this work.

the data of leakage currents. After that, the average leakage currents of input patterns are fitted by the least square fitting method. With the fact that leakage currents exponentially relate to physical parameters and operating temperatures, using the least square fitting method, the average gate tunneling leakage I_g and subthreshold leakage I_s currents for each type of gate can be fitted as [6,7,76,77]

$$I_g = a_0 \exp\left(f_g(L, t_{ox}, T)\right), \tag{3.1}$$

$$I_s = b_0 \exp\left(f_s(L, t_{ox}, T)\right). \tag{3.2}$$

Here, a_0 and b_0 are fitting constants, L is the channel length, t_{ox} is the oxide thickness, and T is the operating temperature. The f_g and f_s are specific fitting forms[2].

Basically, I_g occurs in both on and off states, and I_s is the off-state leakage mechanism [6]. Therefore, the leakage power of a gate can be represented as

$$P_{leak} = V_{dd} \times \left(I_g + (1 - Sw)\, I_s\right), \tag{3.3}$$

where V_{dd} is the supply voltage, and Sw is the switching activity.

Many compact leakage current models have been developed in [6–9, 76]. To examine their accuracies, we have implemented their proposed models and compared their results with that

[2] The variations of device channel length and oxide thickness are considered in this work since leakage power is more sensitive to these parameters [6,7]. It should be noted that although only these two parameters are considered, the developed framework can be easily extended to include any other process variation types such as the channel dopant variation.

of HSPICE simulation under TSMC 65nm model card. To model the leakage currents under process variation, researchers have developed several cell-based compact models [6, 7, 76]. However, they ignored the temperature effect in their models. Therefore, the test results show that the ignorance of temperature effect induces considerable errors. As shown in the first row of Table 3.1, the model of [6, 76] can provide an acceptable accuracy for the gate tunneling leakage current because of its insensitivity to the temperature. However, since the subthreshold leakage current is sensitive to temperature, as shown in the 6-th and 7-th rows of Table 3.1, the models of [6, 7] are not adequate for accurately capturing the subthreshold leakage currents.

To simultaneously take into account the temperature and process variations, Yu et. al. [8] proposed a first-order exponential model, $b_0 \exp(b_1 L + b_2 t_{ox} + b_3 T)$, for the subthreshold leakage current. As reported in [8], their model can provide accurate results for 90nm technology node. However, since the variability of the subthreshold leakage current to the temperature and physical device parameters will increase for more advanced technology(about $5\times \sim 10\times$ increase per technology generation [78]), as shown in the 8-th row of Table 3.1, considerable errors occur for the test results under 65nm technology node.

To improve the accuracy for modeling leakage currents, we increase the order of the fitting components for $f_g(L, t_{ox}, T)$ and $f_s(L, t_{ox}, T)$ shown in equations (3.1) and (3.2). With fitting components shown in 4-th and 10-th rows of Table 3.1, the explicit forms of $f_g(L, t_{ox}, T)$ and $f_s(L, t_{ox}, T)$ of our models are

$$f_g(L, t_{ox}, T) = (a_1 L + a_2 t_{ox} + a_3 T + a_4 t_{ox}^2), \quad (3.4)$$

$$\begin{aligned} f_s(L, t_{ox}, T) = (&b_1 L + b_2 t_{ox} + b_3 T + b_4 L t_{ox} + b_5 T t_{ox} + b_6 LT + \\ &b_7 L^2 + b_8 t_{ox}^2 + b_9 T^2 + b_{10} L t_{ox}^2 + b_{11} LT^2 + \\ &b_{12} T t_{ox}^2 + b_{13} T L^2 + b_{14} t_{ox} L^2 + b_{15} t_{ox} T^2 + \\ &b_{16} t_{ox} T L + b_{17} L^3 + b_{18} t_{ox}^3 + b_{19} T^3), \end{aligned} \quad (3.5)$$

where a_i's and b_i's are fitting constants. As shown in the 4-th and 10-th rows of Table 3.1, our proposed model can present accurate results for both gate tunneling and subthreshold leakage currents. As shown in the 4-th and 10-th rows of Table 3.1, comparing with [6–8, 76], our proposed model can present accurate results for both gate tunneling and subthreshold leakage currents. Although Table 3.1 only shows the results of an NAND gate, the ranges of the errors

Table 3.2: Accuracy comparison of leakage current models in [9] for an NAND gate under 65*nm* technology node.

Leakage Current	Fitting Model	maximum error	average error	error > 3%
Subthreshold	$a_0(1 + a_1T + a_2T^2)e^{a_3L+a_4t_{ox}}$	35.53%	9.82%	79.34%
Gate Tunneling	$b_0(1 + b_1T + b_2T^2)e^{b_3L+b_4t_{ox}}$	4.51%	1.07%	6.32%

of our model for all gates in the cell library are 0.78%–9.66% and 1.2%–3.49% for subthreshold and gate tunneling leakage currents,respectively.

Besides the cell-based leakage current models [6–8, 76], Jaffari et. al. [9] proposed a bin (grid) based model for leakage powers that are also simultaneously take into account the temperature and process variations effects. However, instead of the cell-based leakage power model, leakage powers of bins will be changed after each optimization iteration of thermal-aware optimization engines, such as floorplanner or placer, has been done. Therefore, the time-consumed HSPICE simulation and least-squire fitting process need to be re-performed for re-building their leakage power models of bins (grids) for each optimization iteration. This will degrade their efficiency to provide thermal reliability or thermal related cost for thermal-aware optimization engines. Nevertheless, we implement their leakage power models as cell-based framework for examining the accuracy. Although, as reported in [9], their leakage power model can present accurate result for 90*nm* technology node, considerable errors occur in the test results under 65*nm* technology node. As shown in Table 3.2, their leakage power models result in 35.53% maximum and 9.82% average errors for subthreshold leakage current of an NAND gate under 65*nm* technology node. For all gates in the cell library, the ranges of errors induced by their model are 27.25%–111.27% and 2.93%–5.07% for subthreshold and gate tunneling leakage currents, respectively.

As demonstrating by the above test results, exquisite approaches are still required for modern statistical power analyzers [6, 7, 76] to refine their estimated result while the temperature dependence of the model for leakage powers is included. Besides, more accurate leakage power models should be adopted in Jaffari's electro-thermal analysis framework [9] to refine their estimating results because the temperature is transformed from power. However, their baseline framework requires exquisite extending strategies because their recursive log-normal approximation algorithm is restricted to their leakage power models. Comparing with Jaffari's framework [9], our proposed thermal reliability estimator can handle accurate but more complicated

leakage power models and present accurate estimating results.

3.2.2 Modeling of Variations for Physical Device Parameters

Generally, variations of physical parameters can be classified into two categories, the die-to-die (D2D) variations and the within-die (WID) variations. Due to the different stages of fabrication process, D2D and WID variations can be treated as two independent variation sources. Since D2D variations are smooth on a die, it is reasonable to model all devices having the same D2D variations. On the other hand, the WID variations present considerable gradients within die, and they are spatially correlated because the spatial imperfection of chemical-mechanical polishing and lithography processes. There, WID variations are generally be treated as a correlated random process. As shown in the measured results reported by Cheng et. al. [79], the distributions of physical parameters are similar to Gaussian random variable, the WID variations are generally assumed to be a correlated Gaussian random process and the D2D variations are generally treated as a Gaussian random variable [6, 7, 76, 80].

Combining the models of the D2D and WID variations, the physical parameter $Par(\mathbf{r}_{xy})$ with its nominal value $\mu_{Par}(\mathbf{r}_{xy})$ at position $\mathbf{r}_{xy} = (x, y) \in (0, L_x) \times (0, L_y)$, can be represented as

$$Par(\mathbf{r}_{xy}) = \mu_{Par}(\mathbf{r}_{xy}) + \delta_{WID}(\mathbf{r}_{xy}) + \delta_{D2D}, \tag{3.6}$$

where $\delta_{WID}(\mathbf{r}_{xy})$ is the Gaussian random process of the WID variations, and δ_{D2D} is the Gaussian random variable of the D2D variations.

Since the spatial correlations of $\delta_{WID}(\mathbf{r}_{xy})$ have different decreasing rates in x- and y-directions [81], the following spatial covariance function proposed by [80] is adopted for modeling the spatial correlation of $\delta_{WID}(\mathbf{r}_{xy})$[3].

$$C(\mathbf{r}_{x_1y_1}, \mathbf{r}_{x_2y_2}) = \sigma^2 \exp\left(-\frac{|x_1 - x_2|}{\lambda_x}\right) \exp\left(-\frac{|y_1 - y_2|}{\lambda_y}\right), \tag{3.7}$$

where λ_x and λ_y are correlation lengths of δ_{WID} in the x- and y-directions, respectively. σ is the standard deviation of $\delta_{WID}(\mathbf{r}_{xy})$, $\mathbf{r}_{x_1y_1} = (x_1, y_1)$ and $\mathbf{r}_{x_1y_2} = (x_2, y_2)$.

[3] Although this specific spatial covariance function is adopted, the Karhunen-Loève expansion of a Gaussian random process with any arbitrary spatial covariance function can be efficiently obtained by a finite-element method [82]. Hence, more advanced spatial covariance functions [79, 83–85] can also be incorporated into our analysis framework.

In this dissertation, the Karhunen-Loève (KL) expansion is utilized to simplify $\delta_{WID}(\mathbf{r}_{xy})$, since its number of transformed random variables is much smaller than that of principal component analysis [80]. By applying the KL expansion, $\delta_{WID}(\mathbf{r}_{xy})$ with the spatial covariance function shown in equation (3.7) can be approximated as

$$\delta_{WID}(\mathbf{r}_{xy}) \approx \sum_{l=1}^{N_{Par}} \sqrt{\chi_l}\vartheta_l(\mathbf{r}_{xy})\zeta_l. \tag{3.8}$$

Here, N_{Par} is the truncation number, each $(\chi_l, \vartheta_l(\mathbf{r}_{xy}))$ is an eigen-pair of $C(\mathbf{r}_{x_1y_1}, \mathbf{r}_{x_2y_2})$, and ζ_l's are independent standard normal random variables because the target random process is Gaussian [86].

The closed-form expressions of an eigen-pair $(\chi_l, \vartheta_l(\mathbf{r}_{xy}))$ for $C(\mathbf{r}_{x_1y_1}, \mathbf{r}_{x_2y_2})$ shown in equation (3.7) can be derived as follows [87].

$$\chi_l = \frac{4\sigma^2\lambda_x\lambda_y}{(\lambda_x^2\nu_{x,i}^2+1)(\lambda_y^2\nu_{y,j}^2+1)}, \tag{3.9}$$

$$\vartheta_l(\mathbf{r}_{xy}) = \vartheta_{x,i}(x)\vartheta_{y,j}(y), \tag{3.10}$$

where l, i and j are indices, and there is a one-to-one mapping between (i, j) and l.

The closed forms of $\vartheta_{x,i}(x)$ and $\vartheta_{y,j}(y)$ are

$$\vartheta_{x,i}(x) = \frac{\lambda_x\nu_{x,i}\cos(\nu_{x,i}x)+\sin(\nu_{x,i}x)}{\sqrt{(\lambda_x^2\nu_{x,i}^2+1)L_x/2+\lambda_x}}, \tag{3.11}$$

$$\vartheta_{y,j}(y) = \frac{\lambda_y\nu_{y,j}\cos(\nu_{y,j}y)+\sin(\nu_{y,j}y)}{\sqrt{(\lambda_y^2\nu_{y,j}^2+1)L_y/2+\lambda_y}}. \tag{3.12}$$

Here, $\nu_{x,i}$ and $\nu_{y,j}$ are positive values which satisfy

$$(\lambda^2\nu^2-1)\sin(\nu\gamma) = 2\lambda\nu\cos(\nu\gamma), \tag{3.13}$$

with $(\nu = \nu_{x,i}, \gamma = L_x, \lambda = \lambda_x)$ and $(\nu = \nu_{y,j}, \gamma = L_y, \lambda = \lambda_y)$, respectively.

To get reasonable truncation numbers of the KL expansions for the target physical parameters L and t_{ox}, in this dissertation, N_{Par} for $Par \in \{L, t_{ox}\}$ is decided by the following criterion,

$$\frac{\chi_{N_{Par}+1}}{\sum_{i=1}^{N_P+1}\chi_i} \leq \varepsilon \tag{3.14}$$

with $\varepsilon = 1\%$.

Since the variation of a physical parameter is generally in a controllable range [6, 7, 76, 79, 80], they are the second order random processes [86]. Practically, the spatial covariance

functions shown in equation (3.7) and in [79, 83–85] are continuous. With the above properties of practical covariance functions, the KL expansion of WID variations is valid for the practical implementation.

Generally, the devices located adjacently have similar physical characteristics [6,76]. Therefore, top surface of the die is partitioned into rectangular grids for modeling physical parameters. After that, with the KL expansion of $Par \in \{L, t_{ox}\}$, the device channel length $L_{\underline{m}}$ and oxide thickness $t_{ox\underline{m}}$ in the m-th parameter modeling grid can be approximated as

$$L_{\underline{m}} = \mu_{L_{\underline{m}}} + \boldsymbol{g}_{L_{\underline{m}}}^T \boldsymbol{\eta}_L, \quad (3.15)$$

$$t_{ox\underline{m}} = \mu_{t_{ox\underline{m}}} + \boldsymbol{g}_{t_{ox\underline{m}}}^T \boldsymbol{\eta}_{t_{ox}}. \quad (3.16)$$

Here, $\mu_{L_{\underline{m}}}$ and $\mu_{t_{ox\underline{m}}}$ are nominal values of $L_{\underline{m}}$ and $t_{ox\underline{m}}$, respectively. The $\boldsymbol{g}_{L_{\underline{m}}}$ and $\boldsymbol{g}_{t_{ox\underline{m}}}$ are coefficient vectors for $\boldsymbol{\eta}_L$ and $\boldsymbol{\eta}_{t_{ox}}$, respectively. The $\boldsymbol{\eta}_L = [\eta_{L_1}, \cdots, \eta_{L_{N_L}}]^T$ and $\boldsymbol{\eta}_{t_{ox}} = [\eta_{t_{ox1}}, \cdots, \eta_{t_{oxN_{tox}}}]^T$ are standard normal random vectors including KL expanded WID and D2D random variables for representing the device channel length and the oxide thickness in all parameter modeling grids, respectively.

In the rest of this chapter, $\boldsymbol{\xi}^T$ is employed to represent $[\boldsymbol{\eta}_L^T, \boldsymbol{\eta}_{t_{ox}}^T]$ for the sake of notation simplicity.

3.2.3 Problem Formulation

As addressed by [40–42, 56–59], temperature-aware design should be brought to early design stages such as thermal-aware floor-planning and placement. Therefore, the scope of this dissertation is on providing the thermal reliability analysis under process variations for early design stages. With similar modeling techniques mentioned in section 2.1 of Chapter 2, the structure of the compact thermal model for physical design stages is shown in Figure 3.3. The difference between the thermal models shown in Figure 3.3 and the thermal models mentioned in section 2.1 of Chapter 2 is that powers of the functional blocks are treated statistically because the leakage powers will be random under process variations. Therefore, the profile of power generating sources, $p(\mathbf{r}, L, t_{ox}, T)$, shown in Figure 3.3 is modeled as a function of device channel length L, oxide thickness t_{ox} and the on-chip temperature distribution T.

Combining the compact thermal model and statistical powers of functional blocks, the sta-

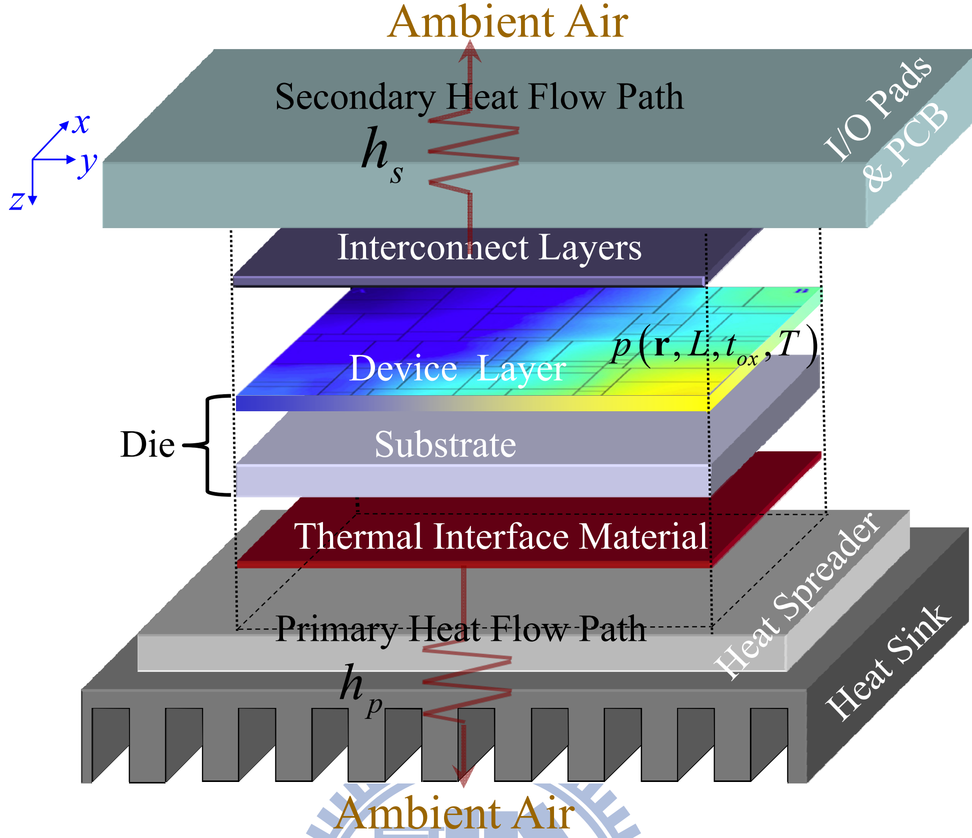


Figure 3.3: Compact thermal model of physical design stages under process variations.

tistical on-chip temperature distribution $T(\mathbf{r}, L, t_{ox})$ can be governed by the statistical steady state heat transfer equation[4].

$$\nabla \cdot (\kappa(\mathbf{r}, T)\nabla T(\mathbf{r}, L, t_{ox})) = -p(\mathbf{r}, L, t_{ox}, T), \tag{3.17}$$

subject to the boundary condition

$$\kappa(\mathbf{r}_{b_s}, T)\frac{\partial T(\mathbf{r}_{b_s}, L, t_{ox})}{\partial b_s} + h_{b_s}T(\mathbf{r}_{b_s}, L, t_{ox}) = f_{b_s}(\mathbf{r}_{b_s}). \tag{3.18}$$

Here, $\mathbf{r} = (x, y, z) \in D$, $D = (0, L_x) \times (0, L_y) \times (-L_z, 0)$ is the domain of die, L_x and L_y are lateral sizes of die, L_z is the thickness of die, $\kappa(\mathbf{r}, T)$ is the thermal conductivity (W/m·°C) of die, and ∇ is the diverge operator. The b_s is any specific boundary surfaces of the die, $\mathbf{r}_{b_s}$ is the position on b_s, h_{b_s} is the heat transfer coefficient on b_s, $f_{b_s}(\mathbf{r}_{b_s})$ is the heat flux function on b_s, and $\partial/\partial n_{b_s}$ is the differentiation along the outward direction which is normalized to b_s.

[4]Because the time constant of heat conduction is much larger than the clock period of circuit [51, 56], the steady state characteristics of the on-chip temperature distribution are more concerned in thermal-aware physical design engines [40–42, 44]. The scope of this dissertation is to provide a simulation framework for thermal-aware physical design engines although the temporary characteristics of on-chip temperature are also important for the post floorplanning or placement real-time task scheduling or workload assignment [56, 88, 89].

```
1   Set μ_T and μ_{T_old} to be the room temperature values;
2   Obtain thermal conductivity by using μ_T;
3   Obtain μ_P by μ_T and set μ_{P_old} to be μ_P;
4   error_P ← 1.0;
5   While error_P > ε_P
6     error_T ← 1.0;
7     While error_T > ε_T
8       Obtain μ_T by the 1-D thermal model shown in
        Figure 2.3 of section 2.1 in Chapter 2 with μ_{P_old};
9       Update thermal conductivity by using μ_T;
10      error_T ← |μ_T − μ_{T_old}| / μ_T;
11      μ_{T_old} ← μ_T;
12    EndWhile
13    Obtain μ_P by using μ_T;
14    error_P ← |μ_P − μ_{P_old}| / μ_P;
15    μ_{P_old} ← μ_P;
16  EndWhile
```

Figure 3.4: An iterative scheme for computing the appropriate thermal conductivity of die. μ_T is a roughly average mean temperature of die, and μ_P is the mean of total on-chip power consumption after executing an iteration. μ_P can be obtained by the zeroth order of H-PC projected power of gates proposed in Figure 3.7 and Figure 3.9 of section 3.3.1.

The $p(\mathbf{r}, L, t_{ox}, T)$ is the power density profile that consists of the deterministic dynamic power density profile $p_d(\mathbf{r})$, the statistical gate tunneling leakage power density profile $p_g(\mathbf{r}, L, t_{ox}, T)$, and the statistical subthreshold leakage power density profile $p_s(\mathbf{r}, L, t_{ox}, T)$. Since the major part of device current flows through the channel, the power density distribution has its value only when $\mathbf{r} \in (0, L_x) \times (0, L_y) \times (-j_d, 0)$. Here, j_d is the junction depth of device [73].

Generally, the values of $\kappa(\mathbf{r}, T)$ are temperature dependent. However, since several iteration loops of the entire thermal analysis procedure need to be executed to correct the error induced by the temperature dependent issue of thermal parameters, the effort of dealing with this issue can be relatively high. Practically, they can be set as an appropriate value, which is suggested to be the thermal conductivity operating at the average temperature of die [41, 42, 44], while performing temperature-aware physical design procedures. Therefore, the iterative computation scheme shown in Figure 3.4 is employed to set the value of $\kappa(\mathbf{r}, T)$ at the steady state mean temperature of die. As mentioned in section 2.1 of Chapter 2, with using the 1-D thermal model, it is fairly efficient to set the thermal conductivity of die at the roughly steady state mean

temperature of die.

With the appropriate thermal conductivity, the statistical steady state heat transfer equation can be re-written as

$$\kappa \nabla^2 T(\mathbf{r}, L, t_{ox}) = -p(\mathbf{r}, L, t_{ox}, T), \tag{3.19}$$

subject to the boundary condition

$$\kappa \frac{\partial T(\mathbf{r}_{b_s}, L, t_{ox})}{\partial \vec{n}_{b_s}} + h_{b_s} T(\mathbf{r}_{b_s}, L, t_{ox}) = f_{b_s}(\mathbf{r}_{b_s}), \tag{3.20}$$

where κ is the thermal conductivity of die that is obtained by utilizing the procedure presented in Figure 3.4.

With the statistical steady state heat transfer equations (3.19) and (3.20), the goals of this work are to evaluate the mean, variance and thermal yield profiles of on-chip temperature distribution.

3.3 Statistical Electro-Thermal Analyzer

The executing flow of the proposed statistical electro-thermal analyzer is summarized in Figure 3.5. Given the information of physical parameters, the KL expansion is performed to transform the spatial correlated physical parameters into a set of un-correlated random variables. Then, the Hermite polynomials (HPs) of these uncorrelated random variables are generated to serve as bases for approximating the statistical on-chip temperature distribution. With the design information and the generated HPs, the statistical expression of the on-chip temperature distribution can be generated using one of our developed statistical expression generators, the stochastic projection based statistical expression generator and the stochastic collocation based statistical expression generator. After that, the on-chip thermal yield profile is estimated using the generated statistical expression of the on-chip temperature distribution. The statistical expression generators and the on-chip thermal yield profile estimation are summarized as follows.

Stochastic Projection Based Statistical Expression Generator

First, with the average temperature obtained by using the 1-D thermal model under the nominal physical parameters, the projected leakage power profiles of the HPs are obtained using

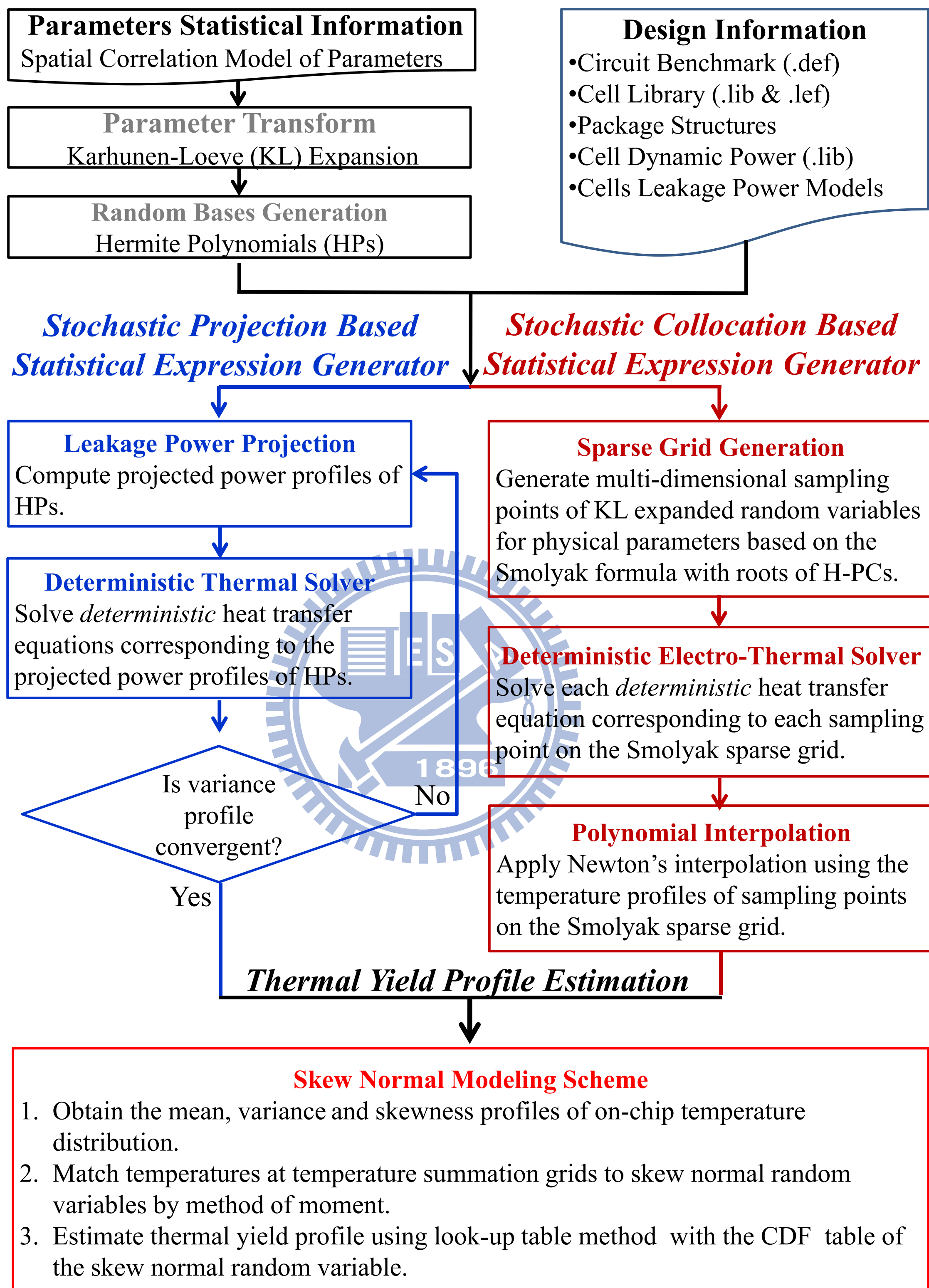


Figure 3.5: Overview of the developed statistical electro-thermal analyzer.

the algorithms presented in section 3.3.1. Then, as detailed in section 3.3.1, the coefficient functions of the Hermite polynomial (HP) representation for the statistical on-chip temperature distribution can be acquired by solving the deterministic heat transfer equations corresponding to the projected power profiles of HPs. After that, the variance profiles of the statistical on-chip temperature distribution are computed. Finally, whether the computed variance profile is convergent or not is checked. If it is convergent, the obtained statistical thermal expression is delivered to the estimating engine of on-chip thermal yield profile. Otherwise, the projected leakage power profiles of HPs are re-calculated by using the computed HP representation of the statistical on-chip temperature distribution, and the above statistical expression generating procedure is repeated.

Stochastic Collocation Based Statistical Expression Generator

First, as described in section 3.3.2, the multi-dimensional sampling points of the KL expanded random variables are generated by using the Smolyak sparse grid formula with the roots of HPs. Then, as stated in section 3.3.2, the deterministic temperature profile corresponding to each sampling point is obtained by solving the deterministic heat transfer equation with the power profile obtained from each sampling point. After that, the statistical expression of on-chip temperature distribution is calculated by applying the Newton's polynomial interpolating formula presented in section 3.3.2. Finally, the calculated the statistical expression is delivered to the estimating engine of the thermal yield profile.

Thermal Yield Profile Estimation

Using the expression generated by one of the developed statistical expression generator, the skew normal modeling scheme first calculates the mean, variance and skewness profiles of the on-chip temperature distribution. After that, each temperature at a specific position is matched to a skew normal random variable by method of moment. Finally, the on-chip thermal yield profile is estimated by look-up table method with the CDF table of the skew normal random variable. The detail of the above computation scheme of the thermal yield profile will be detail in section 3.3.4.

3.3.1 Stochastic Projection Based Statistical Expression Generator

In the recent years, analysis frameworks of the statistical performances, such as the statistical static timing analysis [90], the statistical leakage analysis [6, 7, 76], the statistical waveform/delay analysis of interconnects [60, 91–93] and the statistical power grid analysis [94], have been proposed. The general analysis frameworks of statistical performances analyzers expressed the target performance as the parametric form of the physical parameters up to second order polynomials. The statistical static timing analyzer [90] and interconnect waveform/delay analyzer [60, 91–93] applied the second order Taylor expansion to simulate the target performance. Due to the variations of delays for gate or interconnect corresponding to the parameters are usually in a controllable range, the Taylor expansion can present reasonable estimations for the delays of gates and interconnects [60, 90–93]. However, as shown in section 3.2.1, leakage currents/powers are sensitive to the variations of the physical parameters, and their variation ranges will be large due to the exponential dependency to the physical parameters. In this situation, since the Taylor expansion presents accurate results under the assumption that the variation of the estimating waveforms is small, it might be not suitable for estimating the performance correlated to the leakage currents/powers. For example, Mi et. al [94] have addressed that the second order Taylor expansion did not present accurate voltage waveform analysis for the power grid considering the leakage currents.

Comparing with the Taylor expansion, the polynomial chaos (PC)) [86] is another framework to estimate an output quantity of a system with the sources having statistical fluctuations. Similar with the Taylor expansion framework, the PC expresses the target quantity as a polynomial of the variation sources fluctuating the input sources of the system. The difference between PC and Taylor expansion is that PC generates orthonormal bases corresponding to the probability distribution of the variation sources to represent the output of the system. With the orthonormal property for the representing bases, the PC achieves the minimal mean square error estimation for the target quantity with a specific approximation order of the polynomials. Therefore, under a specific approximation order, PC can be more accurate than Taylor expansion. In other words, under a specific accuracy, Taylor expansion requires higher approximation orders that PC. Besides the advantage of the accuracy, the efficiency of PC is equal to that of

Taylor expansion if the transformed systems for calculating the coefficient corresponding to each orthonormal bases can be solved individually.

Based on the framework of PC, the statistical leakage power analyzers [76,90], the statistical power grid analysis [94] and the author's previous works on the statistical on-chip thermal analysis [8, 95] have been developed. However, the leakage power models adopted by [76, 90, 95] do not take the temperature effects into account. As addressed in section 3.2.1, this leads to considerable errors on the leakage power prediction. Although the author's previous work [8] has took into account the temperature effect in the leakage power model, as addressed in section 3.2.1, the leakage power models is still not adequate for the accurate leakage power prediction. Since the on-chip temperature is transformed from the on-chip power, our previous works can not provide sufficient accuracy for the statistical thermal estimation. Therefore, although the framework of PC is adopted in this statistical expression generator, we propose an adaptive leakage power modeling technique to deal with the issue of complex leakage power models for advanced technologies.

Polynomial Bases

With the random vector $\boldsymbol{\xi} = [\xi_1, \xi_2, \cdots, \xi_{|\boldsymbol{\xi}|}]^T$ constructed by the KL expansion stated in section 3.2.2, a set of $|\boldsymbol{\xi}|$-dimensional Hermite polynomial (HPs) [86] can be constructed to serve as the bases to approximate the on-chip temperature distribution. The HPs of $\boldsymbol{\xi}$ with order r is

$$\Gamma_r(\xi_{i_1}, \cdots, \xi_{i_r}) = (-1)^r \frac{\partial^r}{\partial \xi_{i_1} \cdots \xi_{i_r}} \exp\left(-\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\xi}\right), \tag{3.21}$$

where each i_n with $1 \le n \le r$ is the index of the selected random variable in $\boldsymbol{\xi}$. With equation 3.21, the zeroth, first, and second orders of HPs are

$$\Gamma_0 = 1, \tag{3.22}$$

$$\Gamma_1(\xi_{i_1}) = \xi_{i_1}, \tag{3.23}$$

$$\Gamma_2(\xi_{i_1}, \xi_{i_2}) = \xi_{i_1}\xi_{i_2} - \delta_{i_1 i_2}, \tag{3.24}$$

respectively. Here, $\delta_{i_1 i_2}$ is the Kronecker delta. The HPs satisfy the following orthogonal property [86].

$$\mathrm{E}\left\{\Phi_i(\boldsymbol{\xi})\Phi_j(\boldsymbol{\xi})\right\} = \mathrm{E}\left\{\Phi_i^2(\boldsymbol{\xi})\right\}\delta_{ij}, \tag{3.25}$$

where $\Phi_i(\boldsymbol{\xi})$ is the concise expression of $\Gamma_r(\xi_{i_1}, \cdots, \xi_{i_r})$. There is a one-to-one mapping between $\Phi[\cdot]$ and $\Gamma[\cdot]$, and between i and $i_1 \cdots i_r$.

Stochastic Projection Based Electro-Thermal Updating Scheme

With the KL expanded random vector $\boldsymbol{\xi}$ of the channel length L and oxide thickness t_{ox}, the on-chip temperature $T(\mathbf{r}, L, t_{ox})$ can be approximated as $T(\mathbf{r}, \boldsymbol{\xi})$. Since L and t_{ox} are the variation sources of the input source, the leakage powers, of the heat transfer equations, based on the framework of PC [86], the on-chip temperature $T(\mathbf{r}, L, t_{ox})$ can be approximated by the following polynomial expression.

$$T(\mathbf{r}, L, t_{ox}) \approx \widehat{T}(\mathbf{r}, \boldsymbol{\xi}) = \sum_{k=0}^{N_{PC}} T_k(\mathbf{r})\Phi_k(\boldsymbol{\xi}), \tag{3.26}$$

where each $T_k(\mathbf{r}) = \mathrm{E}\{T(\mathbf{r}, \boldsymbol{\xi})\Phi_k(\boldsymbol{\xi})\}$ is the projection temperature profile at any arbitrary position $\mathbf{r}$ of die corresponding to $\Phi_k(\boldsymbol{\xi})$, and N_{PC} is the truncation number that is equal to $1 + \sum_{n=1}^{p} \frac{1}{n!} \prod_{r=0}^{n-1}(N_{KL} + r)$. Here, p is the order of HPs, and $N_{KL} = N_{t_{ox}} + N_L$.

Substituting equation (3.26) into equation (3.19) and approximating $p(\mathbf{r}, L, T_{ox}, \widehat{T})$ to be $p(\mathbf{r}, \boldsymbol{\xi}, \widehat{T})$, the residual of equation (3.19) is

$$R(\mathbf{r}, \boldsymbol{\xi}) \equiv \kappa \sum_{k=0}^{N_{PC}} \nabla^2 T_k(\mathbf{r})\Phi_k(\boldsymbol{\xi}) - p(\mathbf{r}, \boldsymbol{\xi}, \widehat{T}). \tag{3.27}$$

With a similar procedure, the residual of equation (3.20) can also be obtained. Based on the principle of stochastic Galerkin projection [86], the residuals of the statistical heat transfer equations (3.19)–(3.20) are enforced to be orthogonal to each H-PC, i.e. $\mathrm{E}\{R(\mathbf{r}, \boldsymbol{\xi})\Phi_k(\boldsymbol{\xi})\} = 0$ for each k. Therefore, we have the following un-coupled deterministic heat transfer equation for solving each $T_k(\mathbf{r})$.

$$\kappa \nabla^2 T_k(\mathbf{r}) = -\frac{\mathrm{E}\{p(\mathbf{r}, \boldsymbol{\xi}, \widehat{T})\Phi_k(\boldsymbol{\xi})\}}{\mathrm{E}\{\Phi_k^2(\boldsymbol{\xi})\}}, \tag{3.28}$$

subject to the boundary condition

$$\kappa \frac{\partial T_k(\mathbf{r}_{b_s})}{\partial \vec{n}_{b_s}} + h_{b_s} T_k(\mathbf{r}_{b_s}) = f_{b_s}(\mathbf{r}_{b_s})\delta_{0k} \tag{3.29}$$

for each b_s.

$\mathrm{E}\left\{p(\mathbf{r}, \boldsymbol{\xi}, \widehat{T})\Phi_k(\boldsymbol{\xi})\right\}$ in equation (3.28) is equal to

$$\mathrm{E}\left\{p(\mathbf{r}, \boldsymbol{\xi}, \widehat{T})\Phi_k(\boldsymbol{\xi})\right\} = p_d(\mathbf{r})\delta_{0k} + \mathrm{E}\left\{p_g(\mathbf{r}, \boldsymbol{\xi}, \widehat{T})\Phi_k(\boldsymbol{\xi})\right\} + \mathrm{E}\left\{p_s(\mathbf{r}, \boldsymbol{\xi}, \widehat{T})\Phi_k(\boldsymbol{\xi})\right\}. \tag{3.30}$$

Algorithm Stochastic Projection Based Electro-Thermal Updating Scheme
Input: Initial average die temperature μ_T^{ini} obtained by 1-D thermal model
Output: The H-PC expression of on-chip temperature $\widehat{T}(\mathbf{r},\boldsymbol{\xi})$

1 **Begin**
2 $T_k(\mathbf{r}) \leftarrow 0$ for $0 \le k \le N_{PC}$;
3 $\widehat{T}(\mathbf{r},\boldsymbol{\xi}) \leftarrow \mu_T^{ini}$;
4 $MaxStdError \leftarrow \infty$;
5 **While** ($MaxStdError > \epsilon$)
6 $\widehat{T}_{pre}(\mathbf{r},\boldsymbol{\xi}) \leftarrow \widehat{T}(\mathbf{r},\boldsymbol{\xi})$;
7 **For** $k \leftarrow 0$ to N_{PC}
8 $p_{s,k}(\mathbf{r}) \leftarrow \mathrm{E}\left\{p_s(\mathbf{r},\boldsymbol{\xi},\widehat{T})\Phi_k(\boldsymbol{\xi})\right\}$;
9 $p_{g,k}(\mathbf{r}) \leftarrow \mathrm{E}\left\{p_g(\mathbf{r},\boldsymbol{\xi},\widehat{T})\Phi_k(\boldsymbol{\xi})\right\}$;
10 Obtain the projected power density profile onto the
k-th HP, $p_k(\mathbf{r}) \leftarrow p_d(\mathbf{r})\delta_{0k} + p_{s,k}(\mathbf{r}) + p_{g,k}(\mathbf{r})$;
11 † Solve equations (3.28) and (3.29) with $p_k(\mathbf{r})$ to update $T_k(\mathbf{r})$;
12 **EndFor**
13 $\widehat{T}(\mathbf{r},\boldsymbol{\xi}) \leftarrow \sum_{k=0}^{N_{PC}} T_k(\mathbf{r})\Phi_k(\boldsymbol{\xi})$;
14 $MaxStdError \leftarrow \max_{\mathbf{r}} \left|\mathrm{stdev}\left(\widehat{T}(\mathbf{r},\boldsymbol{\xi})\right) - \mathrm{stdev}\left(\widehat{T}_{pre}(\mathbf{r},\boldsymbol{\xi})\right)\right|$;
15 **EndWhile**
16 **End**

† The deterministic thermal simulator mentioned in Chpater 2 is employed to solve equations (3.28) and (3.29). Note that any deterministic thermal simulators can be used here. "stdev" means the standard deviation.

Figure 3.6: The electro-thermal updating scheme of the stochastic projection based statistical expression generator.

Here, $p_d(\mathbf{r})$ is the dynamic power density profile. $\mathrm{E}\left\{p_g(\mathbf{r},\boldsymbol{\xi},\widehat{T})\Phi_k(\boldsymbol{\xi})\right\}$ and $\mathrm{E}\left\{p_s(\mathbf{r},\boldsymbol{\xi},\widehat{T})\Phi_k(\boldsymbol{\xi})\right\}$ are the statistical projected power density profiles onto the k-th H-PC basis for the gate-leakage and subthreshold-leakage power density profiles $p_g(\mathbf{r},\boldsymbol{\xi},\widehat{T})$ and $p_s(\mathbf{r},\boldsymbol{\xi},\widehat{T})$, respectively. The term δ_{0k} in both equations (3.29) and (3.30) is from $\mathrm{E}\{\Phi_k(\boldsymbol{\xi})\} = \delta_{0k}$ [86].

Any existing deterministic thermal simulators, such as [5, 51–59] and the GIT thermal simulator mentioned in Chapter 2, can be utilized to obtain each $T_k(\mathbf{r})$ after $\mathrm{E}\left\{p(\mathbf{r},\boldsymbol{\xi},\widehat{T})\Phi_k(\boldsymbol{\xi})\right\}$ has been calculated. Since the subthreshold leakage and gate tunneling leakage power density profiles are temperature dependent, as shown in Figure 3.6, a developed electro-thermal updating scheme is performed to obtain each $T_k(\mathbf{r})$ in equation (3.26) for expressing $\widehat{T}(\mathbf{r},\boldsymbol{\xi})$.

First, the initial $\widehat{T}(\mathbf{r},\boldsymbol{\xi})$ is estimated by utilizing the 1-D equivalent thermal circuit with the nominal total on-chip power, and all of the projected coefficient functions $\widehat{T}_k(\mathbf{r})$'s are set to be zeros. Then, by executing the projection algorithms presented in the next subsection 3.3.1 with the leakage power models shown in section 3.2.1, the projection power profiles of corresponding

to each $\Phi_k(\boldsymbol{\xi})$ of the circuit, which are $p_{s,k}(\mathbf{r})$ and $p_{s,k}(\mathbf{r})$ shown in *Lines* 8~10, can be obtained. After that, each $T_k(\mathbf{r})$ is solved by using the deterministic thermal simulator mentioned in 2, and the HP expression of $\widehat{T}(\mathbf{r},\boldsymbol{\xi})$ is updated by using *Line* 11. Finally, the *MaxStdError* is calculated as the maximum absolute error between the standard deviation of $\widehat{T}(\mathbf{r},\boldsymbol{\xi})$ and the standard deviation of $\widehat{T}_{pre}(\mathbf{r},\boldsymbol{\xi})$. The above computation process is repeated until *MaxStdError* is less than a given threshold value.

The developed stochastic projection based electro-thermal updating scheme has the following advantage. Because the deterministic heat transfer equations for solving different $T_k(\mathbf{r})$'s are un-coupled, each $T_k(\mathbf{r})$ can be solved individually. Moreover, since equations (3.28)-(3.29) corresponding to each $T_k(\mathbf{r})$ have the same thermal conductivity κ, the system handling process of an employed deterministic thermal simulator, such as the LU decomposition of the tridiagonal matrix [51], the establishment of the multi-grid cycle [54, 55], and the basis construction of the deterministic thermal simulator stated in Chapter 2, can be performed only once for solving all $T_k(\mathbf{r})$'s. In this work, we employ the deterministic thermal simulator stated in Chapter 2 to solve $T_k(\mathbf{r})$'s because of its high efficiency for the thermal estimation in early design stages[5]

With the statistical expression shown in equation (3.26), the mean and variance profiles of the statistical on-chip temperature distribution can be approximated as

$$\mathrm{E}\left\{\widehat{T}(\mathbf{r},\boldsymbol{\xi})\right\} = T_0(\mathbf{r}), \tag{3.31}$$

$$\mathrm{Var}\left\{\widehat{T}(\mathbf{r},\boldsymbol{\xi})\right\} = \sum_{k=1}^{N_{PC}} T_k^2(\mathbf{r})\mathrm{E}\left\{\Phi_k^2(\boldsymbol{\xi})\right\}. \tag{3.32}$$

Projection Coefficient Calculation of Leakage Power Consumption

In this subsection, for a specific type of gate located at an arbitrary parameter modeling grid, two algorithms are proposed to calculate the projection coefficient of leakage powers corresponding to each HP. As the locations of gates are given, for completing the electro-thermal updating scheme shown in Figure 3.6, the projected power density profiles $p_{s,k}(\mathbf{r})$ and $p_{g,k}(\mathbf{r})$ shown in *Lines* 7~8 of Figure 3.6 can be obtained.

According to the deterministic thermal simulator stated in Chapter 2, the die is divided into a mesh for obtaining $T_k(\mathbf{r})$'s. Similarly, as mentioned in section 3.2.2, the die is divided into

[5]For the post-routing thermal verification, one of the detailed thermal simulators [51, 54, 55] can be employed to solve $T_k(\mathbf{r})$'s with the complicated thermal model for the interconnect layer.

a mesh for obtaining the explicit forms of the device channel length and oxide thickness at the parameter modeling grids. Under an acceptable accuracy, the required mesh sizes for modeling physical parameters and obtaining $T_k(\mathbf{r})$'s are different. Therefore, the mesh sizes of parameter modeling and temperature simulation grids is set to be different. Based on the above setting, if the grids of the temperature simulation mesh overlap those of the parameter modeling mesh, the values of $T_k(\mathbf{r})$ are averaged for calculating the projection coefficients of leakage powers in an arbitrary parameter modeling grid.

As mentioned in section 3.2.1, the complex fitting form are required to be adopted for accurately modeling the leakage powers. Therefore, we adopt the accurate but complex leakage power models presented in section 3.2.1 and propose two approximating strategies to trace the temperature dependency of the leakage powers. For both the subthreshold and the gate tunneling leakage powers, in each parameter modeling grid, the HP expression of the temperature distribution by each iteration, $\widehat{T}_{pre}(\mathbf{r}$ shown in Figure 3.6, is substituted in to the leakage power models. Although the temperature distribution is approximated by the second order HPs in the output of this statistical expression generator, for reducing the complexity, the first order HP expression of the temperature is employed to obtain the explicit forms of leakage powers . Besides, with the mean temperature profile obtained by each iteration shown in Figure 3.6 being the expansion point, an adaptive Taylor expansion is proposed for simplifying the explicit form of the subthreshold leakage power model.

Projection Coefficient of Gate Tunneling Leakage Power For a specific type of gate in the m-th parameter modeling grid, the proposed gate tunneling leakage power model mentioned in section 3.2.1 can be written as

$$P_{g_{\underline{m}}}(L_{\underline{m}}, t_{ox\underline{m}}, T_{\underline{m}}) = V_{dd} \times a_0 e^{a_1 L_{\underline{m}} + a_2 T_{\underline{m}} + a_3 t_{ox\underline{m}} + a_4 t_{ox\underline{m}}^2}, \tag{3.33}$$

where $L_{\underline{m}}$, $t_{ox\underline{m}}$ and $T_{\underline{m}}$ are the device channel length, the device oxide thickness and the average temperature in the m-th parameter modeling grid, respectively. And a_i's are the fitting constants of the specific type of gate.

Approximating $L_{\underline{m}}$ and $t_{ox\underline{m}}$ as the KL expansions shown in equations (3.15)–(3.16) and utilizing the first order HP expression obtained by each iteration shown in Figure 3.6, the average

Algorithm Gate Tunneling Leakage Power Projection

Input: *V_{dd} and a_i's of each gate leakage power model;*
$\mu_{L_{\underline{m}}}$, $\mu_{tox_{\underline{m}}}$ and $\mu_{T_{\underline{m}}}$; *vectors* $\boldsymbol{g}_{L_{\underline{m}}}$, $\boldsymbol{g}_{tox_{\underline{m}}}$, $\boldsymbol{h}_{L_{\underline{m}}}$ and $\boldsymbol{h}_{tox_{\underline{m}}}$;
$\boldsymbol{V}_{tox_{\underline{m}}}$ *and* $\boldsymbol{\Lambda}_{tox_{\underline{m}}}$ *which are the eigen-vector matrix and the diagonal eigen-value matrix of* $\boldsymbol{G}_{tox_{\underline{m}}}$*, respectively.*

Output: $\boldsymbol{q}_{\underline{m}}[k] = \mathrm{E}\left\{P_{g_{\underline{m}}}\left(L_{\underline{m}}, tox_{\underline{m}}, \widehat{T}_{\underline{m}}\right)\Phi_k(\boldsymbol{\xi})\right\}$ for $k = 1 \sim N_{PC}$

1 **Begin**
2 $\boldsymbol{c}_{L_{\underline{m}}} \leftarrow a_1\boldsymbol{g}_{L_{\underline{m}}} + a_2\boldsymbol{h}_{L_{\underline{m}}}$,
$\boldsymbol{c}_{tox_{\underline{m}}} \leftarrow (a_3 + 2a_4\mu_{t_{ox}})\boldsymbol{g}_{tox_{\underline{m}}} + a_2\boldsymbol{h}_{tox_{\underline{m}}}$,
$N_{t_{ox}} \leftarrow \left|\boldsymbol{\eta}_{t_{ox}}\right|$, $N_L \leftarrow |\boldsymbol{\eta}_L|$,
$\mu_{P_{g_{\underline{m}}}} \leftarrow V_{dd} \times a_0 e^{a_1\mu_L + a_2\mu_{T_{\underline{m}}} + a_3\mu_{tox} + a_4\mu_{tox}^2}$,
$\varpi_{L_{\underline{m}}} \leftarrow \boldsymbol{c}_{L_{\underline{m}}}^T\boldsymbol{c}_{L_{\underline{m}}}/2$;
3 PROVECs($N_{t_{ox}}, \boldsymbol{c}_{tox_{\underline{m}}}, \boldsymbol{V}_{tox_{\underline{m}}}, a_4\boldsymbol{\Lambda}_{tox_{\underline{m}}}, \varphi_{tox_{\underline{m}}}, \varpi_{tox_{\underline{m}}}, \boldsymbol{\rho}, \boldsymbol{\Theta}$);
4 $E_{P_g} \leftarrow \mu_{P_{g_{\underline{m}}}}\varphi_{tox_{\underline{m}}} e^{\varpi_{L_{\underline{m}}} + \varpi_{tox_{\underline{m}}}}$;
5 **For** $k \leftarrow 0$ **to** N_{PC}
6 **if** $\Phi_k(\boldsymbol{\xi}) = 1$,
7 $\boldsymbol{q}_{\underline{m}}[k] \leftarrow E_{P_g}$;
8 **else if** $\Phi_k(\boldsymbol{\xi}) = \eta_{L_i}$, $i \in \mathcal{N}_L$,
9 $\boldsymbol{q}_{\underline{m}}[k] \leftarrow \boldsymbol{c}_{L_{\underline{m}}}[i]E_{P_g}$;
10 **else if** $\Phi_k(\boldsymbol{\xi}) = \eta_{tox_i}$, $i \in \mathcal{N}_{t_{ox}}$,
11 $\boldsymbol{q}_{\underline{m}}[k] \leftarrow \boldsymbol{\rho}[i]E_{P_g}$;
12 **else if** $\Phi_k(\boldsymbol{\xi}) = \eta_{L_i}\eta_{L_j} - \delta_{ij}$, $i \in \mathcal{N}_L$, $j \in \mathcal{N}_L$,
13 $\boldsymbol{q}_{\underline{m}}[k] \leftarrow \boldsymbol{c}_{L_{\underline{m}}}[i]\boldsymbol{c}_{L_{\underline{m}}}[j]E_{P_g}$;
14 **else if** $\Phi_k(\boldsymbol{\xi}) = \eta_{tox_i}\eta_{tox_j} - \delta_{ij}$, $i \in \mathcal{N}_{t_{ox}}$, $j \in \mathcal{N}_{t_{ox}}$,
15 $\boldsymbol{q}_{\underline{m}}[k] \leftarrow \boldsymbol{\Theta}[i][j]E_{Pg}$;
16 **else if** $\Phi_k(\boldsymbol{\xi}) = \eta_{tox_i}\eta_{L_j}$, $i \in \mathcal{N}_{t_{ox}}$, $j \in \mathcal{N}_L$,
17 $\boldsymbol{q}_{\underline{m}}[k] \leftarrow \boldsymbol{c}_{L_{\underline{m}}}[j]\boldsymbol{\rho}[i]E_{P_g}$;
18 **EndFor**
19 **End**

$* \mathcal{N}_{t_{ox}} = \{1, 2, 3, \cdots, N_{t_{ox}}\}$; $\mathcal{N}_L = \{1, 2, 3, \cdots, N_L\}$

Figure 3.7: The evaluating algorithm of projection coefficients of gate tunneling leakage power up to second order of HPs.

Function PROVECs$\left(N_{Par}, \boldsymbol{c}_{Par_{\underline{m}}}, \boldsymbol{V}_{Par_{\underline{m}}}, \boldsymbol{\Lambda}_{Par_{\underline{m}}}, \varphi, \varpi, \boldsymbol{\rho}, \boldsymbol{\Theta}\right)$
Input: *The dimension of related vectors and matrices* N_{Par}*;*
Vector $\boldsymbol{c}_{Par_{\underline{m}}}$; Matrices $\boldsymbol{V}_{Par_{\underline{m}}}$, and $\boldsymbol{\Lambda}_{Par_{\underline{m}}}$.
Output: *scalars* φ *and* ϖ*; vector* $\boldsymbol{\rho}$*; matrix* $\boldsymbol{\Theta}$

1 **Begin**
2 $\tilde{\boldsymbol{c}} \leftarrow \boldsymbol{V}^T \boldsymbol{c}_{Par_{\underline{m}}},$

$\boldsymbol{s} \leftarrow \left[\sqrt{1-2\boldsymbol{\Lambda}_{Par_{\underline{m}}}[1][1]}, \cdots, \sqrt{1-2\boldsymbol{\Lambda}_{Par_{\underline{m}}}[N_{Par}][N_{Par}]}\right]^T,$

$\boldsymbol{u} \leftarrow \left[\frac{\tilde{\boldsymbol{c}}[1]}{\boldsymbol{s}[1]}, \cdots, \frac{\tilde{\boldsymbol{c}}[N]}{\boldsymbol{s}[N]}\right]^T,$

$\boldsymbol{w} \leftarrow \left[\frac{1}{\boldsymbol{s}[1]^2}, \cdots, \frac{1}{\boldsymbol{s}[N]^2}\right]^T,$

$\boldsymbol{m} \leftarrow \left[\frac{\tilde{\boldsymbol{c}}[1]}{\boldsymbol{s}[1]^2}, \cdots, \frac{\tilde{\boldsymbol{c}}[N]}{\boldsymbol{s}[N]^2}\right]^T,$

$\varphi \leftarrow 1\Big/\prod_{i=1}^{i=N} \boldsymbol{s}[i],\ \varpi \leftarrow \boldsymbol{u}^T\boldsymbol{u}/2,\ \boldsymbol{\rho} \leftarrow \boldsymbol{V}_{Par_{\underline{m}}}^T \boldsymbol{m};$

3 **For** $i \leftarrow 1$ **to** N
4 **For** $j \leftarrow 1$ **to** N
5 $\boldsymbol{\Theta}[i][j] \leftarrow \boldsymbol{\rho}[i]\boldsymbol{\rho}[j] + \sum_{l=1}^{N} \boldsymbol{w}[l]\boldsymbol{V}_{Par_{\underline{m}}}^T[i][l]\boldsymbol{V}_{Par_{\underline{m}}}[l][j] - \delta_{ij};$
6 **EndFor**
7 **EndFor**
8 **End**

Figure 3.8: The function that evaluates the related vectors for calculating the leakage powers.

temperature in m-th parameter modeling grid $T_{\underline{m}} = T_{\underline{m}}(\boldsymbol{\eta}_L, \boldsymbol{\eta}_{t_{ox}})$ can be written as

$$\widehat{T}_{\underline{m}}(\boldsymbol{\eta}_L, \boldsymbol{\eta}_{t_{ox}}) = \mu_{T_{\underline{m}}} + \boldsymbol{h}_{L_{\underline{m}}}^T \boldsymbol{\eta}_L + \boldsymbol{h}_{t_{ox\underline{m}}}^T \boldsymbol{\eta}_{t_{ox}}, \tag{3.34}$$

where $\mu_{T_{\underline{m}}}$ is the mean of average temperature in the m-th parameter modeling grid. $\boldsymbol{h}_{L_{\underline{m}}}$ and $\boldsymbol{h}_{t_{ox\underline{m}}}$ are the vectors of projection coefficients of $T_{\underline{m}}(\boldsymbol{\eta}_L, \boldsymbol{\eta}_{t_{ox}})$ corresponding to HPs. Substituting $\widehat{T}_{\underline{m}}(\boldsymbol{\eta}_L, \boldsymbol{\eta}_{t_{ox}})$ into equation (3.33), for each electro-thermal iteration, the projection coefficient corresponding to the k-th HP of $P_{g_{\underline{m}}}(L_{\underline{m}}, t_{ox\underline{m}}, T_{\underline{m}})$ can be approximated by

$$\mathrm{E}\left\{P_{g_{\underline{m}}}(L_{\underline{m}}, t_{ox\underline{m}}, \widehat{T}_{\underline{m}})\Phi_k(\boldsymbol{\xi})\right\} = \mu_{P_{g_{\underline{m}}}} \mathrm{E}\left\{e^{\boldsymbol{c}_{L_{\underline{m}}}^T \boldsymbol{\eta}_L} \cdot e^{\boldsymbol{c}_{t_{ox\underline{m}}}^T \boldsymbol{\eta}_{t_{ox}} + a_4 \boldsymbol{\eta}_{t_{ox}}^T \boldsymbol{G}_{t_{ox\underline{m}}} \boldsymbol{\eta}_{t_{ox}}} \Phi_k(\boldsymbol{\xi})\right\}, \tag{3.35}$$

where $\mu_{P_{g_{\underline{m}}}} = V_{dd} a_0 e^{a_1\mu_{L_{\underline{m}}} + a_2\mu_{T_{\underline{m}}} + a_3\mu_{t_{ox\underline{m}}} + a_4\mu_{t_{ox\underline{m}}}^2}$, $\boldsymbol{c}_{t_{ox\underline{m}}} = \left(a_3 + 2a_4\mu_{t_{ox\underline{m}}}\right)\boldsymbol{g}_{t_{ox\underline{m}}} + a_2\boldsymbol{h}_{t_{ox\underline{m}}}$, $\boldsymbol{c}_{L_{\underline{m}}} = a_1\boldsymbol{g}_{L_{\underline{m}}} + a_2\boldsymbol{h}_{L_{\underline{m}}}$ and $\boldsymbol{G}_{t_{ox\underline{m}}} = \boldsymbol{g}_{t_{ox\underline{m}}}\boldsymbol{g}_{t_{ox\underline{m}}}^T$.

According to the derivation given in APPENDIX B and replacing the subscript Par, which is shown in Figure 3.8, with $t_{ox\underline{m}}$, the evaluating algorithm of equation (3.35) is summarized in Figure 3.7. The function PROVECs shown in Fig 3.8 evaluates related vectors for calculating $\mathrm{E}\left\{\Phi_{k_{Par}}(\boldsymbol{\xi})\exp(\boldsymbol{c}_{Par_{\underline{m}}}^T \boldsymbol{\eta}_{Par} + a\boldsymbol{\eta}_{Par}^T \boldsymbol{G}_{Par_{\underline{m}}} \boldsymbol{\eta}_{Par})\right\}$. Here, the subscript Par means the physical

Algorithm Subthreshold Leakage Power Projection

Input: $\mu_{P_{s\underline{m}}}$ and β_i's in equation (3.38); vectors $\boldsymbol{g}_{L_{\underline{m}}}$, $\boldsymbol{g}_{t_{ox\underline{m}}}$, $\boldsymbol{h}_{L_{\underline{m}}}$ and $\boldsymbol{h}_{t_{ox\underline{m}}}$; $\boldsymbol{V}_{L_{\underline{m}}}$ and $\boldsymbol{\Lambda}_{L_{\underline{m}}}$ which are eigen-vector matrix and diagonal eigen-value matrix of $\boldsymbol{G}_{L_{\underline{m}}}$, respectively. $\boldsymbol{V}_{t_{ox\underline{m}}}$ and $\boldsymbol{\Lambda}_{t_{ox\underline{m}}}$ which are eigen-vector matrix and diagonal eigen-value matrix of $\boldsymbol{G}_{t_{ox\underline{m}}}$, respectively

Output: $\boldsymbol{q}_{\underline{m}}[k] = \mathrm{E}\left\{P_{s\underline{m}}\left(L_{\underline{m}}, t_{ox\underline{m}}, \widehat{T}_{\underline{m}}\right)\Phi_k(\boldsymbol{\xi})\right\}$ for $k = 1 \sim N_{PC}$

1 **Begin**
2 $\boldsymbol{c}_{L_{\underline{m}}} \leftarrow \beta_1 \boldsymbol{g}_{L_{\underline{m}}} + \beta_2 \boldsymbol{h}_{L_{\underline{m}}}$,
  $\boldsymbol{c}_{t_{ox\underline{m}}} \leftarrow \beta_3 \boldsymbol{g}_{t_{ox\underline{m}}} + \beta_5 \boldsymbol{h}_{t_{ox\underline{m}}}$,
  $N_L \leftarrow |\boldsymbol{\eta}_L|$, $N_{t_{ox}} \leftarrow \left|\boldsymbol{\eta}_{t_{ox}}\right|$;
3 PROVECs$(N_L, \boldsymbol{c}_{L_{\underline{m}}}, \boldsymbol{V}_{L_{\underline{m}}}, \beta_2 \boldsymbol{\Lambda}_{L_{\underline{m}}}, \varphi_{L_{\underline{m}}}, \varpi_{L_{\underline{m}}}, \boldsymbol{\rho}_L, \boldsymbol{\Theta}_L)$;
4 PROVECs$(N_{t_{ox}}, \boldsymbol{c}_{t_{ox\underline{m}}}, \boldsymbol{V}_{t_{ox\underline{m}}}, \beta_4 \boldsymbol{\Lambda}_{t_{ox\underline{m}}}, \varphi_{t_{ox\underline{m}}}, \varpi_{t_{ox\underline{m}}}, \boldsymbol{\rho}_{t_{ox}}, \boldsymbol{\Theta}_{t_{ox}})$;
5 $E_{P_s} \leftarrow \mu_{P_{s\underline{m}}} \varphi_{L_{\underline{m}}} \varphi_{t_{ox\underline{m}}} e^{\varpi_{L_{\underline{m}}} + \varpi_{t_{ox\underline{m}}}}$;
6 **For** $k \leftarrow 0$ **to** N_{PC}
7 **if** $\Phi_k(\boldsymbol{\xi}) = 1$,
8 $\boldsymbol{q}_{\underline{m}}[k] \leftarrow E_{P_s}$;
9 **else if** $\Phi_k(\boldsymbol{\xi}) = \eta_{L_i}$, $i \in \mathcal{N}_L$,
10 $\boldsymbol{q}_{\underline{m}}[k] \leftarrow \boldsymbol{\rho}_L[i] E_{P_s}$;
11 **else if** $\Phi_k(\boldsymbol{\xi}) = \eta_{t_{ox i}}$, $i \in \mathcal{N}_{t_{ox}}$,
12 $\boldsymbol{q}_{\underline{m}}[k] \leftarrow \boldsymbol{\rho}_{t_{ox}}[i] E_{P_s}$;
13 **else if** $\Phi_k(\boldsymbol{\xi}) = \eta_{L_i}\eta_{L_j} - \delta_{ij}$, $i \in \mathcal{N}_L$, $j \in \mathcal{N}_L$,
14 $\boldsymbol{q}_{\underline{m}}[k] \leftarrow \boldsymbol{\Theta}_L[i][j] E_{P_s}$;
15 **else if** $\Phi_k(\boldsymbol{\xi}) = \eta_{t_{ox i}}\eta_{t_{ox j}} - \delta_{ij}$, $i \in \mathcal{N}_{t_{ox}}$, $j \in \mathcal{N}_{t_{ox}}$,
16 $\boldsymbol{q}_{\underline{m}}[k] \leftarrow \boldsymbol{\Theta}_{t_{ox}}[i][j] E_{P_s}$;
17 **else if** $\Phi_k(\xi) = \eta_{t_{ox i}}\eta_{L_j}$, $i \in \mathcal{N}_{t_{ox}}$, $j \in \mathcal{N}_L$
18 $\boldsymbol{q}_{\underline{m}}[k] \leftarrow \boldsymbol{\rho}_{t_{ox}}[i] \boldsymbol{\rho}_L[j] E_{P_s}$;
19 **EndFor**
20 **End**

$* \ \mathcal{N}_{t_{ox}} = \{1, 2, 3, \cdots, N_{t_{ox}}\}$; $\mathcal{N}_L = \{1, 2, 3, \cdots, N_L\}$

Figure 3.9: Subthreshold leakage power projection algorithm.

parameter L or t_{ox}, and $\underline{m}$ means the m-th grid. $\boldsymbol{\eta}_{Par}$ is a standard normal random vector representing the KL expanding random vector of the physical parameter, and a is a constant. $\boldsymbol{V}_{Par_{\underline{m}}}$ is the eigen-vector matrix of $\boldsymbol{G}_{Par_{\underline{m}}}$, and $\boldsymbol{\Lambda}_{Par_{\underline{m}}}$ is the eigen-value matrix of $\boldsymbol{G}_{Par_{\underline{m}}}$ multiplied by a. $\Phi_{k_{Par}}(\boldsymbol{\xi})$ is the HPs of $\boldsymbol{\xi}$ up to the second order. k_{Par} is the index of HPs.

Although the algorithm in Figure 3.7 only calculates the projection coefficient of the gate tunneling leakage power up to the second order of HPs, it can be easily extended to the higher order of HPs.

Subthreshold Leakage Power Projection As mentioned in section 3.2.1, for a specific type of gate in the m-th parameter modeling grid, the adopted subthreshold leakage power model can be written as

$$P_{s\underline{m}}(L_{\underline{m}}, t_{ox\underline{m}}, T_{\underline{m}}) = V_{dd} \times b_0 e^{\tilde{f}_s(L_{\underline{m}}, t_{ox\underline{m}}, T_{\underline{m}})} \times e^{b_1 L_{\underline{m}} + b_2 L_{\underline{m}}^2 + b_3 t_{ox\underline{m}} + b_4 t_{ox\underline{m}}^2 + b_5 T_{\underline{m}}}, \tag{3.36}$$

where b_i's are the fitting constants, and $\tilde{f}_s$ equals to the remaining terms of f_s shown in equation (3.2) with excluding the polynomial terms $\{L_{\underline{m}}, L_{\underline{m}}^2, t_{ox\underline{m}}, t_{ox\underline{m}}^2, T_{\underline{m}}\}$. By utilizing the Taylor expansion with the expansion point at $(\mu_{L_{\underline{m}}}, \mu_{t_{ox\underline{m}}}, \mu_{T_{\underline{m}}})$, $\tilde{f}_s(L_{\underline{m}}, t_{ox\underline{m}}, T_{\underline{m}})$ can be approximated as $d_0 + d_1 \Delta L_{\underline{m}} + d_2 \Delta L_{\underline{m}}^2 + d_3 \Delta t_{ox\underline{m}} + d_4 \Delta t_{ox\underline{m}}^2 + d_5 \Delta T_{\underline{m}}$. Here, d_i's are $\mu_{T_{\underline{m}}}$ dependent Taylor expansion coefficients. With the approximated $\tilde{f}_s(L_{\underline{m}}, t_{ox\underline{m}}, T_{\underline{m}})$, $P_{s\underline{m}}$ can be approximated as

$$P_{s\underline{m}}(L_{\underline{m}}, t_{ox\underline{m}}, T_{\underline{m}}) \approx \mu_{P_{s\underline{m}}} e^{\beta_1 \Delta L_{\underline{m}} + \beta_2 \Delta L_{\underline{m}}^2 + \beta_3 \Delta t_{ox\underline{m}} + \beta_4 \Delta t_{ox\underline{m}}^2 + \beta_5 \Delta T_{\underline{m}}}, \tag{3.37}$$

where $\beta_1 = b_1 + d_1, \beta_2 = b_1 + 2b_2 + d_2, \beta_3 = b_3 + d_3, \beta_4 = b_3 + 2b_4 + d_4, \beta_5 = b_5 + d_5$, and $\mu_{P_{s\underline{m}}}$ is equal to $V_{dd} \times b_0 e^{(b_1+d_0)\mu_{L_{\underline{m}}} + b_2 \mu_{L_{\underline{m}}}^2 + b_3 \mu_{t_{ox\underline{m}}} + b_4 \mu_{t_{ox\underline{m}}}^2 + b_5 \mu_{T_{\underline{m}}}}$.

Utilizing the KL expansions of $L_{\underline{m}}$ and $t_{ox\underline{m}}$, and the first order HP expression of $T_{\underline{m}}$, we have $\Delta L_{\underline{m}} = \boldsymbol{g}_{L_{\underline{m}}}^T \boldsymbol{\eta}_L$, $\Delta t_{ox\underline{m}} = \boldsymbol{g}_{t_{ox\underline{m}}}^T \boldsymbol{\eta}_{t_{ox}}$, and $\Delta T_{\underline{m}} = \boldsymbol{h}_{L_{\underline{m}}}^T \boldsymbol{\eta}_L + \boldsymbol{h}_{t_{ox\underline{m}}}^T \boldsymbol{\eta}_{t_{ox}}$. Then, for a specific type of gate located in the m-th parameter modeling grid, the projection coefficients of $P_{s\underline{m}}(L_{\underline{m}}, t_{ox\underline{m}}, T_{\underline{m}})$ corresponding to k-th HP basis can be approximated by equation (3.38) for each electro-thermal iteration.

$$\mathrm{E}\left\{P_{s\underline{m}}(L_{\underline{m}}, t_{ox\underline{m}}, \widehat{T}_{\underline{m}})\Phi_k(\boldsymbol{\xi})\right\} = \mu_{P_{s\underline{m}}} \mathrm{E}\left\{e^{\boldsymbol{c}_{L_{\underline{m}}}^T \boldsymbol{\eta}_L + \beta_2 \boldsymbol{\eta}_L^T \boldsymbol{G}_{L_{\underline{m}}} \boldsymbol{\eta}_L} \cdot e^{\boldsymbol{c}_{t_{ox\underline{m}}}^T \boldsymbol{\eta}_{t_{ox}} + \beta_4 \boldsymbol{\eta}_{t_{ox}}^T \boldsymbol{G}_{t_{ox\underline{m}}} \boldsymbol{\eta}_{t_{ox}}} \Phi_k(\boldsymbol{\xi})\right\}, \tag{3.38}$$

where $\mu_{P_{s\underline{m}}} = V_{dd} b_0 e^{(b_1+d_0)\mu_L + b_2 \mu_L^2 + b_3 \mu_{t_{ox}} + b_4 \mu_{t_{ox}}^2 + b_5 \mu_{T_{\underline{m}}}}$, $\boldsymbol{c}_{t_{ox\underline{m}}} = \beta_3 \boldsymbol{g}_{t_{ox\underline{m}}} + \beta_5 \boldsymbol{h}_{t_{ox\underline{m}}}$, $\boldsymbol{c}_{L_{\underline{m}}} = \beta_1 \boldsymbol{g}_{L_{\underline{m}}} + \beta_5 \boldsymbol{h}_{L_{\underline{m}}}$, $\boldsymbol{G}_{t_{ox\underline{m}}} = \boldsymbol{g}_{t_{ox\underline{m}}} \boldsymbol{g}_{t_{ox\underline{m}}}^T$ and and $\boldsymbol{G}_{L_{\underline{m}}} = \boldsymbol{g}_{L_{\underline{m}}} \boldsymbol{g}_{L_{\underline{m}}}^T$.

According to the derivation given in APPENDIX B, the calculating algorithm of equation (3.38) up to the second order of HPs is summarized in Figure 3.9. Since the expressions of both exponents in equation (3.38) are similar with the expression of the second exponent in equation (3.35), the calculating steps (*Lines* 6 ∼ 19) in Figure 3.9 are similar with the calculating steps (*Lines* 5 ∼ 18) in Figure 3.7.

As indicated by [6, 76], the number of parameter modeling grids can be much less than the number of gates while maintaining the acceptable accuracy. Thus, the simulated die is divided

```
Algorithm  Stochastic Projection Based Electro-Thermal Analysis
Input: Geometries of the die, spatial correlation models of channel
       length and oxide thickness, design information such as .def file,
       .lef file, .lib file, package structure and leakage power models
Output: The H-PC expression of T̂(r, ξ).
        The mean profile and the variance profile of T̂(r, ξ).
```

1. **Begin**
2. Set the thermal parameters, and get the initial average die temperature μ_T^{ini} by 1-D thermal model described in section 3.2.3;
3. **For** $m \leftarrow 1$ to N_g
4. Obtain $\boldsymbol{g}_{L_{\underline{m}}}$ of $L_{\underline{m}}$ and $\boldsymbol{g}_{t_{ox\underline{m}}}$ of $t_{ox\underline{m}}$ by the KL expansion;
5. Eigen-decompose $\boldsymbol{G}_{L_{\underline{m}}}$ to obtain $\boldsymbol{V}_{L_{\underline{m}}}$ and $\boldsymbol{\Lambda}_{L_{\underline{m}}}$;
6. Eigen-decompose $\boldsymbol{G}_{t_{ox\underline{m}}}$ to obtain $\boldsymbol{V}_{t_{ox\underline{m}}}$ and $\boldsymbol{\Lambda}_{t_{ox\underline{m}}}$;
7. **EndFor**
8. $T_k(\mathbf{r}) \leftarrow 0$ for $0 \le k \le N_{PC}$;
9. $\widehat{T}(\mathbf{r}, \boldsymbol{\xi}) \leftarrow \mu_T^{ini}$;
10. **While** ($MaxStdError > \epsilon$)
11. $\widehat{T}_{prev}(\mathbf{r}, \boldsymbol{\xi}) \leftarrow \widehat{T}(\mathbf{r}, \boldsymbol{\xi})$;
12. **For** $m \leftarrow 1$ to N_g
13. **For** $n \leftarrow 1$ to *NumGateType*
14. Obtain the projected gate-tunneling leakage powers onto the H-PC bases for the n-th gate type in the m-th parameter modeling grid by the algorithm shown in Figure 3.7;
15. Obtain the projected sub-threshold leakage powers onto the H-PC bases for the n-th gate type in the m-th parameter modeling grid by the algorithm shown in Figure 3.9;
16. **EndFor**
17. **EndFor**
18. **For** $k \leftarrow 0$ to N_{PC}
19. Obtain the projected power density profile onto the k-th H-PC basis, $p_k(\mathbf{r})$, by using the projected powers calculated from *Lines* 14 and 15;
20. Solve equations (3.28) and (3.29) with $p_k(\mathbf{r})$ to update $T_k(\mathbf{r})$;
21. **EndFor**
22. $\widehat{T}(\mathbf{r}, \boldsymbol{\xi}) \leftarrow \sum_{k=0}^{N_{PC}} T_k(\mathbf{r})\Phi_k(\boldsymbol{\xi})$;
23. Update mean and variance profiles by equations (3.31) and (3.32);
24. $MaxStdError \leftarrow \max_{\mathbf{r}} \left| \text{stdev}\left(\widehat{T}(\mathbf{r}, \boldsymbol{\xi})\right) - \text{stdev}\left(\widehat{T}_{prev}(\mathbf{r}, \boldsymbol{\xi})\right) \right|$;
25. **EndWhile**
26. **End**

Figure 3.10: Stochastic projection based electro-thermal analysis algorithm. *NumGateType* in *Line* 13 is the number of gate types given from the industrial library file.

into N_g grids for modeling parameters that is much less than the number of simulated temperature grids. Gates locate in the same parameter modeling grid share the same KL expansions of the channel length and oxide thickness; hence, they share the same $\boldsymbol{G}_{L_{\underline{m}}}$ and $\boldsymbol{G}_{t_{ox\underline{m}}}$ in the m-th parameter modeling grid. Therefore, the number of eigen-decompositions is N_g rather than the number of gates. In addition, the eigen-functions and eigen-values only depend on the spatial covariance functions of the channel length and oxide thickness; thus, each $\boldsymbol{G}_{L_{\underline{m}}}$ and $\boldsymbol{G}_{t_{ox\underline{m}}}$ are known after the spatial covariance functions are given. Generally, the empirical spatial covariance functions of the channel length and the oxide thickness can be extracted before the circuit design [79, 80, 83, 85]. Therefore, the eigen-decomposition of each $\boldsymbol{G}_{L_{\underline{m}}}$ and $\boldsymbol{G}_{t_{ox\underline{m}}}$ can be calculated before the thermal simulation. The complete analysis algorithm is presented in Figure 3.10.

3.3.2 Stochastic Collocation Based Statistical Expression Generator

Smolyak Sparse Grid Formulation

The primary advantage of Smolyak sparse grid formulation is to construct an interpolating polynomial of the multivariate function $u \in C^r$ by using much less samples of the desired function than those of the full tensor product interpolation formula and the Monte Carlo method but still maintains an acceptable error bound [96, 97]. Here, C^r is the set of all functions which have continuous derivatives of all orders up to r. With the stochastic collocation technique, the statistical expression of the on-chip temperature distribution can be efficiently constructed.

The difference between Monte Carlo method and Smolyak sparse grid formulation is that the Monte Carlo method randomly generates the samples of random variables and, hence, requires a large number of samples for achieving an accurate estimate. On contrary to the Monte Carlo Method, the Smolyak sparse grid technique uses the roots of H-PCs or the extrema of Chebyshev polynomial [97] to generate the samples of random variables and employs these fewer samples to effectively interpolate the desired solution. For a two-dimensional random variable, its possible sample sets of the Monte Carlo method and the Smolyak sparse grid formulation are illustrated in Figure 3.11.

According to the Smolyak sparse grid formulation, the on-chip temperature distribution can

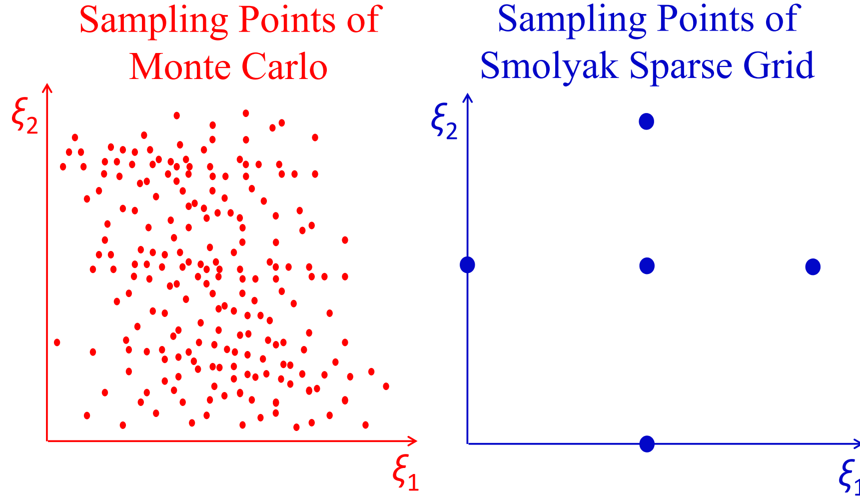


Figure 3.11: The number of sampling random variables comparison between the Monte Carlo method and the Smolyak sparse grid formulation. Here, the samples of Smolyak sparse grid are adopted for achieving a level two approximation.

be explicitly approximated as follows [96, 97].

$$\widehat{T}_q^{N_{KL}}(\mathbf{r},\boldsymbol{\xi}) = \sum_{q-N_{KL}+1\leq|\mathbf{i}|\leq q} (-1)^{q-|\mathbf{i}|}\binom{N_{KL}-1}{q-|\mathbf{i}|}\left(Q^{i_1}(T)\otimes\cdots\otimes Q^{i_{N_{KL}}}(T)\right). \tag{3.39}$$

Here, $N_{KL} = N_{t_{ox}} + N_L$ is the number of random variables in $\boldsymbol{\xi}$, $q = N_{KL} + l$, $l \geq 1$ is the formulation level, and $|\mathbf{i}| = i_1 + \cdots + i_n + \cdots + i_{N_{KL}}$. With level $i_n \geq 1$, Q^{i_n} is an interpolating polynomial of $T(\mathbf{r},\boldsymbol{\xi})$ by only utilizing the random variable ξ_n, and $\otimes$ is the functional cross product. The level i_n is the index to decide the number of samples (m_{i_n}) for the interpolating polynomial Q^{i_n}. As suggested by [97], the relation between m_{i_n} and i_n is that $m_1 = 1$ and $m_{i_n} = 2^{i_n-1} + 1$ for $i_n > 1$.

From (3.39), only the corresponding temperature values of a small set of samples for $\boldsymbol{\xi}$ [97] need to be known. This set is called the sparse grid and is equal to [97]

$$\mathcal{H}(q, N_{KL}) = \bigcup_{q-N_{KL}+1\leq|\mathbf{i}|\leq q} \left(\hbar^{i_1}\times\cdots\times\hbar^{i_n}\times\cdots\times\hbar^{i_{N_{KL}}}\right), \tag{3.40}$$

where $\hbar^{i_n} = \left\{\xi_{i_n}^1, \cdots, \xi_{i_n}^{m_{i_n}}\right\}$ is the set of sample points used by $Q^{i_n}(T)$, and the operator '$\times$' is the cross product of sets. The number of sample points from the Smolyak sparse grid formulation increases as $O\left(N_{KL}^l/l!\right)$, and the runtime complexity for obtaining $\widehat{T}_q^{N_{KL}}(\mathbf{r},\boldsymbol{\xi})$ is in the order of

$C_{det} \cdot O\left(N_{KL}^{l}/l!\right)$. Here, C_{det} is the runtime complexity for performing the deterministic electro-thermal simulation once.

For a function having bounded derivatives up to order r, the Smolyak sparse grid formulation ensures a error bound, $|E_l| = c_{N_{KL},r} \cdot N_{\mathcal{H}}^{-r} \cdot (\log N_{\mathcal{H}})^{(r+1)(N_{KL}-1)}$ [97]. Here, $N_{\mathcal{H}}$ is the number of sample points in $\mathcal{H}(q, N_{KL})$, and $c_{N_{KL},r}$ is a constant that only depends on N_{KL} and r. In our experience, the accurate estimation of thermal yield profile can be obtained by setting the level l to be 1. Therefore, the number of sample points in the Smolyak sparse grid formulation can be much less than that of the Monte Carlo method.

An example with $N_{KL} = 2$ and $q = N_{KL} + 1 = 3$ is given to illustrate the Smolyak sparse grid formulation. Since $q - N_{KL} + 1 \leq |\mathbf{i}| \leq q$, we have $i_1 = 1, i_2 = 1$ for $|\mathbf{i}| = 2$, and $i_1 = 1, i_2 = 2$ or $i_1 = 2, i_2 = 1$ for $|\mathbf{i}| = 3$. Therefore, the numbers of sample values for random variables ξ_1 and ξ_2 are $m_{i_1=1} = 1, m_{i_2=1} = 1$ for $|\mathbf{i}| = 2$, and $m_{i_1=1} = 1, m_{i_2=2} = 3$ or $m_{i_1=2} = 3, m_{i_2=1} = 1$ for $|\mathbf{i}| = 3$, respectively. According to various values of i_1 and i_2, the interpolating polynomial forms by individually utilizing each random variable at different levels can be determined. After that, the interpolating polynomial forms corresponding to $\boldsymbol{\xi}^T = [\xi_1, \xi_2]$ at different combined levels (i_1, i_2) can be constructed by the functional cross product operation. For example, $Q^{i_1=1}(T) = 1$ and $Q^{i_1=2}(T) = a_0 + a_1\xi_2 + a_2\xi_2^2$ are the first order and second order interpolating polynomial forms by utilizing ξ_1, respectively; $Q^{i_1=1}(T) \otimes Q^{i_2=2}(T) = 1 \otimes (b_0 + b_1\xi_2 + b_2\xi_2^2) = b_0 + b_1\xi_2 + b_2\xi_2^2$ and $Q^{i_1=2}(T) \otimes Q^{i_2=1}(T) = (a_0 + a_1\xi_1 + a_2\xi_1^2) \otimes 1 = a_0 + a_1\xi_1 + a_2\xi_1^2$. Here, a_j's and b_j's are coefficients that can be determined by using the sample values of $T(\mathbf{r}, \boldsymbol{\xi})$. To obtain $Q^{i_1}(T) \otimes Q^{i_2}(T)$ for each pair (i_1, i_2), only the chip temperature distribution excited by the point that belongs to the following sample set of $\boldsymbol{\xi}$ needs to be known. Given $\hbar^1 = \{p_0^1\}$ and $\hbar^2 = \{p_0^2, p_1^2, p_2^2\}$, we have

$$
\begin{aligned}
\mathcal{H}(3,2) &= \left(\hbar^{i_1=1} \times \hbar^{i_2=1}\right) \cup \left(\hbar^{i_1=1} \times \hbar^{i_2=2}\right) \cup \left(\hbar^{i_2=2} \times \hbar^{i_2=1}\right) \\
&= \left\{\left[p_0^1, p_0^1\right]^T\right\} \cup \left\{\left[p_0^1, p_0^2\right]^T, \left[p_0^1, p_1^2\right]^T, \left[p_0^1, p_2^2\right]^T\right\} \cup \left\{\left[p_0^2, p_0^1\right]^T, \left[p_1^2, p_0^1\right]^T, \left[p_2^2, p_0^1\right]^T\right\} \\
&= \left\{\left[p_0^1, p_0^1\right]^T, \left[p_0^1, p_0^2\right]^T, \left[p_0^1, p_1^2\right]^T, \left[p_0^1, p_2^2\right]^T, \left[p_0^2, p_0^1\right]^T, \left[p_1^2, p_0^1\right]^T, \left[p_2^2, p_0^1\right]^T\right\}.
\end{aligned}
\tag{3.41}
$$

The sampling values of $\hbar^i$ for each level i must be properly decided. Adopting the roots of H-PCs with its order being corresponding to the level i can achieve the most accurate result as

Algorithm Temperature Profile Calculation for a Sample Point
Input: A sampling point $\boldsymbol{\xi}^j$, initial temperature $T^{ini}_{\boldsymbol{\xi}^j}$ and $p_d(\mathbf{r})$
Output: Temperature profile $T(\mathbf{r}, \boldsymbol{\xi}^j)$

1 **Begin**
2 $T(\mathbf{r}, \boldsymbol{\xi}^j) \leftarrow T^{ini}_{\boldsymbol{\xi}^j}$;
3 $MaxError \leftarrow \infty$;
4 Obtain $t_{ox\underline{m}}(\boldsymbol{\xi}^j)$ and $L_{\underline{m}}(\boldsymbol{\xi}^j)$ for each m-th parameter modeling grid according to $\boldsymbol{\xi}^j$;
5 **While** ($MaxError > \epsilon$)
6 $T_{pre}(\mathbf{r}, \boldsymbol{\xi}^j) \leftarrow T(\mathbf{r}, \boldsymbol{\xi}^j)$;
7 Update $p_{leak}(\mathbf{r}, \boldsymbol{\xi}^j, T_{pre})$ by $T_{pre}(\mathbf{r}, \boldsymbol{\xi}^j)$;
8 $p(\mathbf{r}, \boldsymbol{\xi}^j, T_{pre}) \leftarrow p_{\text{leak}}(\mathbf{r}, \boldsymbol{\xi}^j, T_{pre}) + p_d(\mathbf{r})$;
9 †Solve equations (3.42) and (3.43) with $p(\mathbf{r}, \boldsymbol{\xi}^j, T_{pre})$ to obtain a new $T(\mathbf{r}, \boldsymbol{\xi}^j)$;
10 **if** ($T(\mathbf{r}, \boldsymbol{\xi}^j) = \infty$) **then** Thermal runaway;
11 $MaxError \leftarrow \max_{\mathbf{r}} \left|T(\mathbf{r}, \boldsymbol{\xi}^j) - T_{pre}(\mathbf{r}, \boldsymbol{\xi}^j)\right|$;
12 **EndWhile**
13 **End**

†Any deterministic thermal simulators can be used to execute *Line* 9. Here, the simulator stated in Chapter 2 is adopted.

Figure 3.12: Deterministic electro-thermal analysis for each sampling point, $\boldsymbol{\xi}^j$, in sparse grid. p_{leak}, p_d and p are the leakage, dynamic and total power density profiles for each sampling point, respectively.

$\boldsymbol{\xi}$ is a normal random vector [98]. Choosing the extrema of the Chebyshev polynomial with its order being corresponding to the level i can achieve the nested sparse grid structure, i.e. $\hbar^i \subset \hbar^k$ for $i < k$, for any levels and the acceptable accuracy [97]. In this work, we select the roots of H-PCs as the sampling values since the result is shown to be very accurate by using the low level approximation, and the nested sparse grid structure is still preserved for $q = N_{KL} + 1$[6].

Temperature Profile Calculation for a Given Sample Point

After the sparse grid $\mathcal{H}(q, N_{KL})$ of $\boldsymbol{\xi}$ is obtained, the samples of channel length and oxide thickness in the m-th parameter modeling grid corresponding to the j-th sample point, $\boldsymbol{\xi}^j$, of $\mathcal{H}(q, N_{KL})$ can be obtained by equations (3.15) and (3.16). Hence, the deterministic power density profile corresponding to $\boldsymbol{\xi}^j$ can be obtained. With the deterministic power density profile,

[6]If the high order approximation is needed for the accuracy, we suggest to use the extrema of the Chebyshev polynomial because the nested sparse grid structure is preserved for any levels; hence, the number of sample points can be much less.

we have the following deterministic steady-state heat transfer equation

$$\kappa\nabla^2 T(\mathbf{r},\boldsymbol{\xi}^j) = -p(\mathbf{r},\boldsymbol{\xi}^j,T), \tag{3.42}$$

subject to the following boundary condition

$$\kappa\frac{\partial T(\mathbf{r}_{b_s},\boldsymbol{\xi}^j)}{\partial \vec{n}_{b_s}} + h_{b_s}T(\mathbf{r}_{b_s},\boldsymbol{\xi}^j) = f_{b_s}(\mathbf{r}_{b_s}). \tag{3.43}$$

Here, $p(\mathbf{r},\boldsymbol{\xi}^j,T)$ and $T(\mathbf{r},\boldsymbol{\xi}^j)$ are the deterministic power density and temperature profiles with respect to $\boldsymbol{\xi}^j$, respectively. Since the power density profile in equation (3.42) is temperature dependent, a deterministic electro-thermal analysis procedure summarized in Figure 3.12 is built to obtain each $T(\mathbf{r},\boldsymbol{\xi}^j)$.

Temperature Profile Construction by Using Polynomial Interpolation

Instead of directly using equation (3.39) to obtain $Q^{i_1}(T)\otimes\cdots\otimes Q^{i_{N_{KL}}}(T)$ for each different $|\mathbf{i}| = i_1+\cdots+i_{N_{KL}}$, we take the advantage of nested sparse grid structure and then perform the Newton interpolating method [98] to globally interpolate $T(\mathbf{r},\boldsymbol{\xi})$.[7] Based on the Newton interpolating formula, the approximated on-chip temperature at a specified position of the die, $T(\mathbf{r}^*,\boldsymbol{\xi})$, can be expressed as

$$\widehat{T}(\mathbf{r}^*,\boldsymbol{\xi}) = \sum_{j=0}^{j=N_{\mathcal{H}}-1} \hat{u}_j(\mathbf{r}^*)\phi_j(\boldsymbol{\xi}). \tag{3.44}$$

Here, each $\phi_j(\boldsymbol{\xi})$ is an interpolating polynomial with respect to the j-th sampling vector $\boldsymbol{\xi}^j$, and the form of each $\phi_j(\boldsymbol{\xi})$ can be found in [98]. $N_{\mathcal{H}} = |\mathcal{H}(q,N_{KL})|$ and $|\mathcal{H}(q,N_{KL})|$ is the number of the sampling vectors in sparse grid. Each $\hat{u}_j(\mathbf{r}^*)$ is an unknown coefficient which needs to be determined.

Based on the basic idea of interpolation that the approximation function must match each known data, the interpolated polynomial in (3.44) satisfies the following equation for each $\boldsymbol{\xi}^n$.

$$\sum_{j=0}^{j=N_{\mathcal{H}}-1} \hat{u}_j(\mathbf{r}^*)\phi_j(\boldsymbol{\xi}^n) = T(\mathbf{r}^*,\boldsymbol{\xi}^n). \tag{3.45}$$

[7] For the sparse grid that does not preserve the nested structure, the Newton interpolating method can also be applied to obtain each $Q^{i_1}(T)\otimes\cdots\otimes Q^{i_{N_{KL}}}(T)$.

Algorithm Stochastic Collocation Based Electro-thermal Analysis

Input: Geometries of the die; spatial correlation models of device channel length and oxide thickness; design informations such as .def, .lef, and .lib files; package structure and leakage power models

Output: Mean profile, variance profile, and the Smolyak sparse grid interpolation formula, $\widehat{T}(\mathbf{r}, \boldsymbol{\xi})$, of on-chip temperature distribution

1 **Begin**
2 Set thermal parameters and the initial average mean temperature, μ_T^{ini}, of the die by 1-D thermal model;
3 **For** $m \leftarrow 1$ to N_g
4 Obtain $\boldsymbol{g}_{L_{\underline{m}}}$ and $\boldsymbol{g}_{t_{ox\underline{m}}}$ of $L_{\underline{m}}$ and $t_{ox\underline{m}}$ by the KL expansion, respectively;
5 **EndFor**
6 Generate the Smolyak sparse grid, $\mathcal{H}(q, N_{KL})$, for the KL expanded random variables.
7 **For** $n \leftarrow 0$ to $|\mathcal{H}(q, N_{KL})| - 1$
8 Obtain $T(\mathbf{r}, \boldsymbol{\xi}^n)$ by using the algorithm shown in Figure 3.12.
9 **EndFor**
10 Solve equation (3.46) to obtain the Newton interpolation formula in equation (3.44), and calculate the mean and variance profiles.
11 **End**

Figure 3.13: Stochastic Collocation Based Statistical Expression Generating Algorithm.

With the property of $\phi_j(\boldsymbol{\xi})$ described in [98], equation (3.45) can be rewritten as the following matrix form for finding each $\hat{u}_j(\mathbf{r}^*)$ at the chip position $\mathbf{r}^*$.

$$\begin{bmatrix} \phi_0(\boldsymbol{\xi}^0) & 0 & \cdots & 0 \\ \phi_0(\boldsymbol{\xi}^1) & \phi_1(\boldsymbol{\xi}^1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\boldsymbol{\xi}^{N_\mathcal{H}-1}) & \phi_1(\boldsymbol{\xi}^{N_\mathcal{H}-1}) & \cdots & \phi_{N_\mathcal{H}-1}(\boldsymbol{\xi}^{N_\mathcal{H}-1}) \end{bmatrix} \begin{bmatrix} \hat{u}_0(\mathbf{r}^*) \\ \hat{u}_1(\mathbf{r}^*) \\ \vdots \\ \hat{u}_{N_\mathcal{H}-1}(\mathbf{r}^*) \end{bmatrix} = \begin{bmatrix} T(\mathbf{r}^*, \boldsymbol{\xi}^0) \\ T(\mathbf{r}^*, \boldsymbol{\xi}^1) \\ \vdots \\ T(\mathbf{r}^*, \boldsymbol{\xi}^{N_\mathcal{H}-1}) \end{bmatrix} \tag{3.46}$$

Each $\hat{u}_j(\mathbf{r}^*)$ can be calculated by using the forward substitution. After each $\hat{u}_j(\mathbf{r}^*)$ is calculated, the mean and variance profiles of the temperature distribution can be estimated as

$$\mathrm{E}\left\{\widehat{T}(\mathbf{r}^*, \boldsymbol{\xi})\right\} = \mathrm{E}\left\{\sum_{j=0}^{j=N_\mathcal{H}-1} \hat{u}_j(\mathbf{r}^*)\phi_j(\boldsymbol{\xi})\right\}, \tag{3.47}$$

$$\mathrm{Var}\left\{\widehat{T}(\mathbf{r}^*, \boldsymbol{\xi})\right\} = \mathrm{Var}\left\{\sum_{j=0}^{j=N_\mathcal{H}-1} \hat{u}_j(\mathbf{r}^*)\phi_j(\boldsymbol{\xi})\right\}. \tag{3.48}$$

The algorithm of the developed stochastic collocation based statistical expression generator is shown in Figure 3.13.

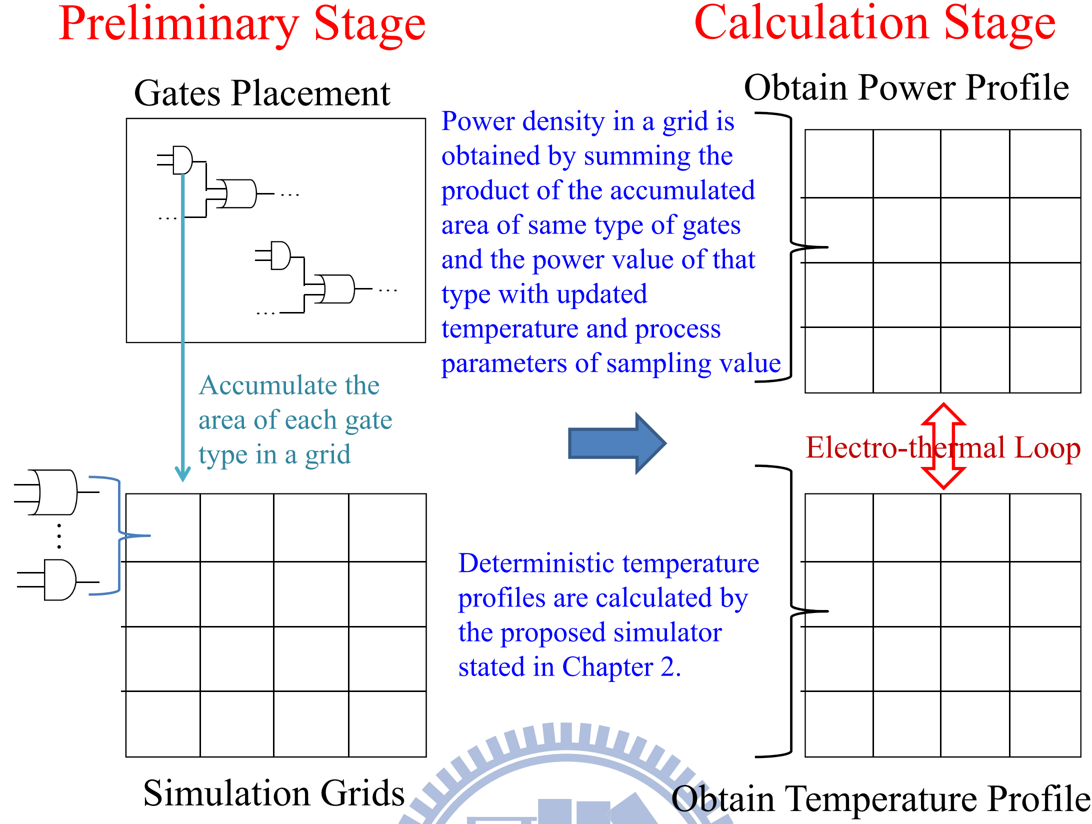


Figure 3.14: Implementation of solving the deterministic heat transfer equations.

3.3.3 Implementation of the Deterministic Electro-Thermal Simulation

The implementation of solving the deterministic heat transfer equations for the stochastic projection and collocation based methods is shown in Figure 3.14. In the preliminary stage, the accumulated area of each gate type in each simulating temperature grid is pre-calculated and stored. With the accumulated area of each gate type in each simulating temperature grid, the deterministic power density profile for each sampling point in $\mathcal{H}(q, N_{KL})$ or each projected power density profile corresponding to HP can be obtained in the order of $O(N_x N_y N_{type})$. Here, N_x and N_y are the division numbers of simulation grid along the $x-$ and $y-$directions, respectively, and N_{type} is the number of total gate types for the given design. Generally, N_{type} is determined by the specified cell library, and it is far less than the number of simulation grid, $N_x N_y$.

The deterministic thermal simulator stated in Chapter 2 is adopted for solving the deterministic heat transfer equations. In our experimental setting, the number of simulation grid is 128×128, and it takes 0.018 seconds to execute one time of deterministic thermal simulation.

3.3.4 On-Chip Thermal Yield Computation

As mention in section 3.1.2, the on-chip thermal yield profile can be defined as

$$Tyield(\mathbf{r}, T_{ref}) \stackrel{\text{def}}{=} Prob(T(\mathbf{r}, \boldsymbol{\xi}) \leq T_{ref}). \tag{3.49}$$

With the definition of (3.49), the target is to approximate the CDF of the on-chip temperature distribution. Because the first order H-PC approximation obtains a linear combination of KL expanded normal random variables, the first order expression of $\widehat{T}(\mathbf{r}, \boldsymbol{\xi})$ in equation (3.26) is a normal random variable. Thus, the on-chip thermal yield estimation based on the first order H-PC expression can be easily obtained by looking up the table of the tail probability of the standard normal random variables with using the estimated mean and variance profiles shown in equations (3.31) and (3.32), respectively.

For designs with large scale variations, the second order approximation has been suggested to be taken into account for the performance analysis [6, 7, 76, 90–92, 94]. In this work, based on the second order H-PC expression for the stochastic projection method and the Level-1 Smolyak sparse grid formula for the stochastic collocation method, an on-chip thermal yield profile estimator is proposed. Two different approaches of estimating the thermal yield profile for the stochastic projection method up to the second order of H-PCs, without including the cross product terms and with including the cross product terms, will be presented.

The approach for the second order of H-PCs without including the cross product terms will be introduced first because it has the same form as the Level-1 Smolyak sparse grid formula of the stochastic collocation method. The statistical expression of the on-chip temperature distribution at a specific location $\mathbf{r}^*$ of the die without including the cross product terms can be written as

$$\widehat{T}(\mathbf{r}^*, \boldsymbol{\xi}) = \sum_{k=1}^{k=N_{KL}} \left(\hat{a}_k(\mathbf{r}^*)\xi_k^2 + \hat{b}_k(\mathbf{r}^*)\xi_k \right) + \hat{c}(\mathbf{r}^*). \tag{3.50}$$

Here, $\hat{a}_k(\mathbf{r}^*)$, $\hat{b}_k(\mathbf{r}^*)$ and $\hat{c}(\mathbf{r}^*)$ are the coefficients calculated by the stochastic projection or the stochastic collocation generators. Equation (3.50) can be re-written as

$$\widehat{T}(\mathbf{r}^*, \boldsymbol{\xi}) = \sum_{k=1}^{k=N_{KL}} \hat{a}_k(\mathbf{r}^*)\chi_k(\mathbf{r}^*, \xi_k) + \tilde{c}(\mathbf{r}^*), \tag{3.51}$$

where each $\chi_k(\mathbf{r}^*, \xi_k) = \left(\xi_k + \hat{b}_k(\mathbf{r}^*)/2\hat{a}_k(\mathbf{r}^*)\right)^2$ is a non-central chi-square random variable since ξ_k is a normal random variable, and $\tilde{c}(\mathbf{r}^*) = \hat{c}(\mathbf{r}^*) - \sum_{k=1}^{N_{KL}} \hat{b}_k^2(\mathbf{r}^*)/4\hat{a}_k(\mathbf{r}^*)$ is a constant.

Since $\boldsymbol{\xi}$ is an independent normal random vector, $\widehat{T}(\mathbf{r}^*, \boldsymbol{\xi})$ is a weighted sum of independent non-central chi-square random variables. To estimate the on-chip thermal yield profile defined by equation (3.49) can be done by approximating the CDF of $\widehat{T}(\mathbf{r}^*, \boldsymbol{\xi})$ at each specified location of the die. Although, theoretically, the CDF of $\widehat{T}(\mathbf{r}^*, \boldsymbol{\xi})$ can be obtained by convolving the PDFs of $\chi_k(\mathbf{r}^*, \xi_k)$'s, it is not suitable for the practical propose because of numerous numerical convolutions. Thus, an innovated statistical moment matching based method is performed to efficiently approximate the CDF of $\widehat{T}(\mathbf{r}^*, \boldsymbol{\xi})$.

APEX [99], a state-of-the-art method for estimating the CDF, approximates the CDF of random variable with the similar form of equation (3.50) by a set of linear-combined exponential waveforms and can achieve an arbitrarily required matching order of statistical moments. However, the Padé approximation, which does not guarantee to be stable for obtaining poles/zeros even in the low order approximation, is essential for APEX. To remedy the unstable issue, the technique proposed by [100] can be adopted to obtain the first two dominated pole/zero pairs for APEX. However, the first two dominated pole/zero pairs can only construct an approximated CDF of $\widehat{T}(\mathbf{r}^*, \boldsymbol{\xi})$ that matches up to the first two statistical moments. Please refer [99] and [100] for the details of APEX and the stable two-pole technique, respectively.

Here, we are going to present a skew-normal based statistical moment matching technique that can stably approximate the CDF of $\widehat{T}(\mathbf{r}^*, \boldsymbol{\xi})$ and matches the statistical moments up to the third order. The basic idea is to approximate a Gaussian-like but skewed random variable by matching its mean, variance and skewness to be a skew-normal random variable. The validation for approximating the PDF/CDF of $\widehat{T}(\mathbf{r}^*, \boldsymbol{\xi})$ by the skew-normal random variable is explained by using Figure 3.15. The sketches of PDFs for the weighted sum of two independent non-central chi-square random variables in two different cases are shown in Figure 3.15. In *Case1*, the skewness of PDF decreases because a left-skewed distribution and a right-skewed distribution are moving integrated. In *Case2*, the skewness of PDF increases because two right-skewed distributions (or two left-skewed distributions, only the right-skewed case shown in Figure 3.15) are moving integrated. Both integrated PDFs in *Case1* and *Case2* are Gaussian-like. Since

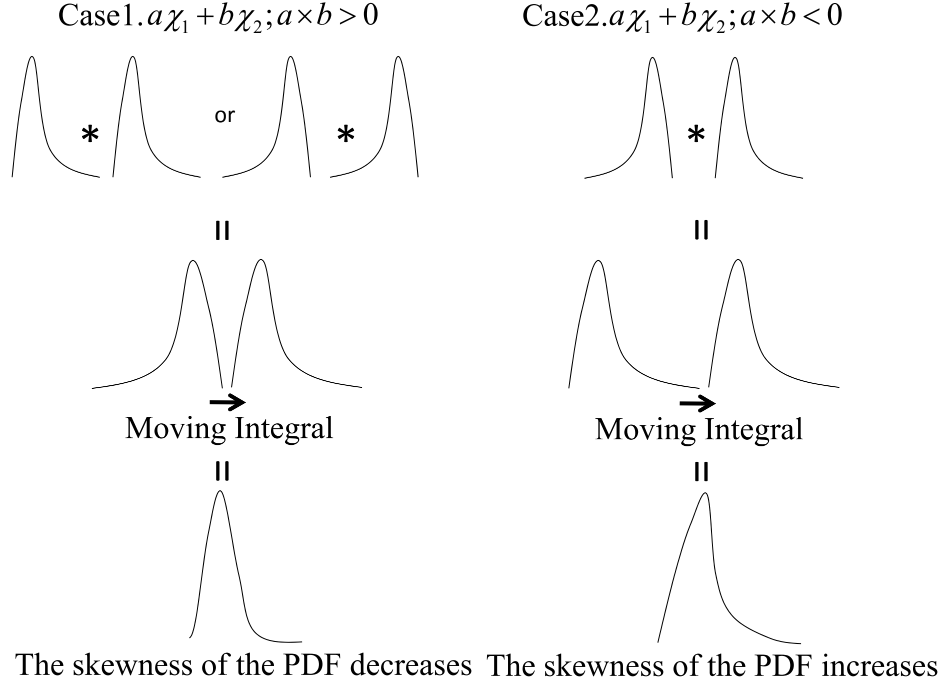


Figure 3.15: Weighted sum of two independent non-central chi-square random variables. *Case1*: the skewness of the PDF decreases because a left-skewed distribution and a right-skewed distribution are moving integrated. *Case2*: the skewness of the PDF increases because two right-skewed distributions are moving integrated.

$\widehat{T}(\mathbf{r}^*, \boldsymbol{\xi})$ is the weighted sum of independent non-central chi-square random variables, $\widehat{T}(\mathbf{r}^*, \boldsymbol{\xi})$ is a Gaussian-like and skewed random variable.

The moment generating function and CDF of a skew-normal random variable, $Z \sim SN(\upsilon, \omega, \alpha)$, are characterized by the given parameters υ, ω and α [101]. With these parameters, the first three statistical moments of Z are

$$\mathrm{E}(Z) = \upsilon + \omega\delta\sqrt{2/\pi}, \tag{3.52}$$

$$\mathrm{Var}(Z) = \omega^2(1 - 2\delta^2/\pi), \tag{3.53}$$

$$\mathrm{Skew}(Z) = \frac{(4-\pi)\left(\delta\sqrt{2/\pi}\right)^3}{2(1-2\delta^2/\pi)^{3/2}}, \tag{3.54}$$

where $\delta = \alpha/\sqrt{1+\alpha^2}$.

With the approximated expression of on-chip temperature distribution $\widehat{T}(\mathbf{r}^*, \boldsymbol{\xi})$, the thermal yield at a specified location $\mathbf{r}^*$ can be approximated as

$$\begin{aligned} Tyield(\mathbf{r}^*, T_{ref}) &\approx Prob\left(\widehat{T}(\mathbf{r}^*, \boldsymbol{\xi}) \leq T_{ref}\right) \\ &= Prob\left(\Delta\widehat{T}(\mathbf{r}^*, \boldsymbol{\xi}) \leq \rho(\mathbf{r}^*)\right), \end{aligned} \tag{3.55}$$

where $\rho(\mathbf{r}^*) = (T_{ref} - \mu_{\widehat{T}}(\mathbf{r}^*))/\sigma_{\widehat{T}}(\mathbf{r}^*)$ and $\Delta\widehat{T}(\mathbf{r}^*,\boldsymbol{\xi}) = (\widehat{T}(\mathbf{r}^*,\boldsymbol{\xi}) - \mu_{\widehat{T}}(\mathbf{r}^*))/\sigma_{\widehat{T}}(\mathbf{r}^*)$. $\mu_{\widehat{T}}(\mathbf{r}^*)$ and $\sigma_{\widehat{T}}(\mathbf{r}^*)$ are the mean and the standard deviation of $\widehat{T}(\mathbf{r}^*,\boldsymbol{\xi})$, and they can be calculated by using equations (3.31) and (3.32) or equations (3.47) and (3.48), respectively.

By matching the first three statistical moments of $\Delta\widehat{T}(\mathbf{r}^*,\boldsymbol{\xi})$ to the skew-normal random variable Z, we have

$$\upsilon + \omega\delta\sqrt{2/\pi} = \mu_{\Delta\widehat{T}}(\mathbf{r}^*), \tag{3.56}$$

$$\omega^2(1 - 2\delta^2/\pi) = \sigma^2_{\Delta\widehat{T}}(\mathbf{r}^*), \tag{3.57}$$

$$\frac{(4-\pi)\left(\delta\sqrt{2/\pi}\right)^3}{2(1-2\delta^2/\pi)^{3/2}} = \gamma_{\Delta\widehat{T}}(\mathbf{r}^*). \tag{3.58}$$

Here, $\mu_{\Delta\widehat{T}}(\mathbf{r}^*) = 0$ and $\sigma^2_{\Delta\widehat{T}}(\mathbf{r}^*) = 1$ are the mean and variance of $\Delta\widehat{T}(\mathbf{r}^*,\boldsymbol{\xi})$, respectively. $\gamma_{\Delta\widehat{T}}(\mathbf{r}^*)$ is the skewness of $\Delta\widehat{T}(\mathbf{r}^*,\boldsymbol{\xi})$, and its value can be calculated by the binomial moment evaluation algorithm [99]. After solving equations (3.56)–(3.58), we have

$$\delta(\mathbf{r}^*) = \sqrt{\frac{\gamma_{\Delta\hat{T}}^{\frac{2}{3}}(\mathbf{r}^*)}{\left(\frac{2(4-\pi)}{2\pi}\sqrt{\frac{2}{\pi}}\right)^{\frac{2}{3}} + \frac{2}{\pi}\gamma_{\Delta\hat{T}}^{\frac{2}{3}}(\mathbf{r}^*)}}, \tag{3.59}$$

$$\omega(\mathbf{r}^*) = \sqrt{\frac{1}{1 - \frac{2}{\pi}\delta^2(\mathbf{r}^*)}}, \tag{3.60}$$

$$\upsilon(\mathbf{r}^*) = -\omega(\mathbf{r}^*)\delta(\mathbf{r}^*)\sqrt{\frac{2}{\pi}}, \tag{3.61}$$

and $\alpha(\mathbf{r}^*) = \delta(\mathbf{r}^*)\Big/\sqrt{1-\delta^2(\mathbf{r}^*)}$.

With $\upsilon(\mathbf{r}^*)$, $\omega(\mathbf{r}^*)$ and $\alpha(\mathbf{r}^*)$, the thermal yield $Tyield(\mathbf{r}^*, T_{ref})$ is approximated by the CDF of the skew normal random variable as follows.

$$Tyield(\mathbf{r}^*, T_{ref}) \approx \Phi(\beta(\mathbf{r}^*)) - 2T_{Owen}(\beta(\mathbf{r}^*), \alpha(\mathbf{r}^*)). \tag{3.62}$$

Here, Φ is the CDF of the standard normal random variable, T_{Owen} is Owen's T function, and $\beta(\mathbf{r}^*) = (\rho(\mathbf{r}^*) - \upsilon(\mathbf{r}^*))/\omega(\mathbf{r}^*)$. Based on equation (3.62), the approximated thermal yield at a specific location on the die can be evaluated by using a look-up table method.

For the stochastic projection method up to the second order H-PCs with cross product terms, the H-PC expression is a quadratic polynomial with the form of $\mathbf{c}^T(\mathbf{r}^*)\boldsymbol{\xi} + \boldsymbol{\xi}^T\mathbf{A}(\mathbf{r}^*)\boldsymbol{\xi}$. In this category, the quadratic model diagonalization [99] needs to be performed for calculating the

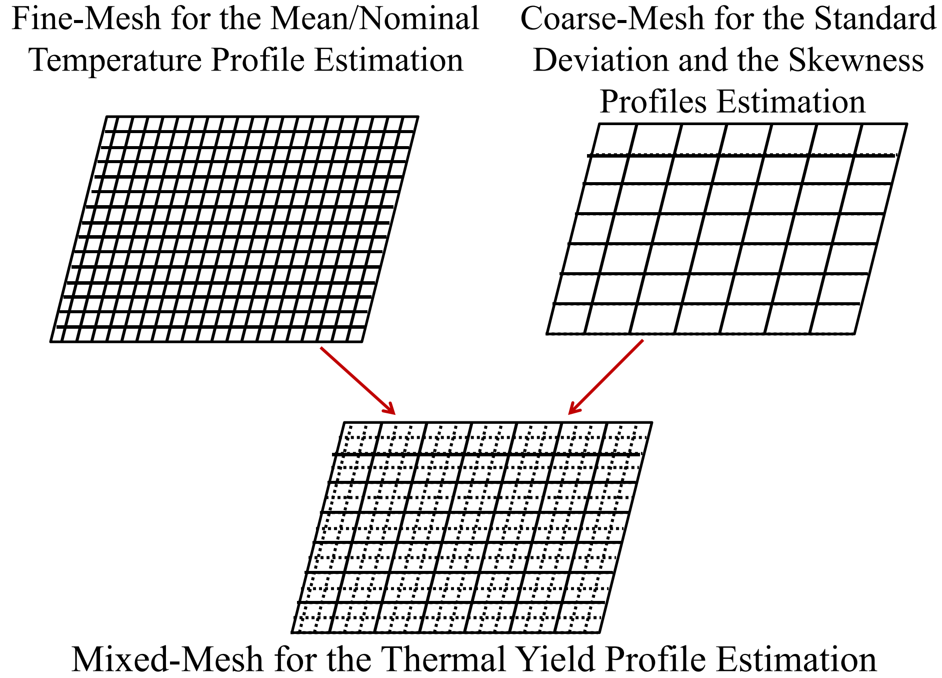


Figure 3.16: The executing sketch of the mixed-mesh thermal yield estimation.

statistical moments. Then, the skew normal modeling technique mentioned previously can be applied to the thermal yield profile approximation for this category. In this paper, the second order H-PC expression without cross product terms of the stochastic projection method is implemented, and the experimental result shows that this category can provide an accurate on-chip thermal yield profile estimation.

3.3.5 Mixed-Mesh Thermal Yield Estimation

As mentioned in Sections 3.3.1 and 3.3.2, the developed statistical expression generators need to solve several deterministic heat transfer equations to obtain the statistical expressions of the on-chip temperature distribution. Although our results show that the Level-1 Smolyak sparse grid formula of the stochastic collocation based method and the second order H-PC expression without the cross product terms of the stochastic projection based method can obtain the accurate statistical expressions of on-chip temperature distribution, we still need to solve $2\times(N_L+N_{t_{ox}})+1$ deterministic heat transfer equations. Therefore, the runtime of the thermal yield estimation is dominated by the statistical expression generators. To save the runtime for feeding the thermal yield to be the thermal cost of thermal-aware optimization engines, such as thermal-aware

floorplanners or placers, a mixed-mesh strategy is proposed to estimate on-chip thermal yield profile under an allowable temperature resolution, T_{res}.

The mixed-mesh strategy is inspired by the following observations. The developed statistical polynomial expression generator stated in Section 3.3.1/3.3.2 consists of a deterministic thermal simulation for calculating the mean/nominal temperature profile and $2 \times (N_L + N_{t_{ox}})$ deterministic thermal simulations for calculating the variations of the temperature distribution. Practically, since the process variations of parameters are usually within a controllable range, the mean/nominal value of the circuit performance is larger than the values of variance and skewness of the circuit performance [6, 7, 76, 90, 92, 94]. Since the mean is the PDF/CDF location parameter of the temperature at a specific position of die, it contributes the major portion to the value of thermal yield.

Based on the above observations, the mixed-mesh strategy for generating the statistical polynomial expression of temperature distribution is exhibited in Figure 3.16. For preserving the estimation accuracy, a fine-mesh deterministic thermal simulation is performed to obtain the mean/nominal temperature profile. Then, the difference $\Delta\overline{T}_{\text{max}}$ between the maximum and minimum temperatures for the mean/nominal temperature profile is extracted, and a temperature resolution T_{res} is chosen. Then, a N_{CM} by N_{CM} coarse-mesh is utilized for executing the remaining $N_{PC} - 1$ or $N_{\mathcal{H}} - 1$ deterministic thermal simulations. Here, N_{CM} can be calculated using the criterion $\left\lceil \Delta\overline{T}_{\text{max}}/T_{\text{res}} \right\rceil$. After that, using the statistical polynomial expression generated by these $N_{PC} - 1$ or $N_{\mathcal{H}} - 1$ coarse-mesh temperature simulations, the coarse-mesh variance and skewness profiles of temperature distribution are obtained. Finally, the thermal yield profile is calculated by using the mixed-mesh mean/nominal, variance and skewness profiles of temperature distribution.

With the above mixed-mesh strategy, the complexity of statistical polynomial expression can be significant reduced. For example, in our implementation, the deterministic thermal simulator stated in Chapter 2 is employed to calculate the deterministic temperature profile. The complexity of the baseline algorithm stated in Sections 3.3.1 is $N_{PC}N_{\text{FM}}N_{\text{FM}}\text{O}(\log N_{\text{Base}})$, and the complexity of the mixed-mesh strategy is $(N_{\text{FM}}N_{\text{FM}} + (N_{PC} - 1)N_{\text{CM}}N_{\text{CM}})\,\text{O}(\log N_{\text{Base}})$. Here, N_{FM} is the number of girds in x- and y-directions for the fine-mesh, and N_{Base} is the number of

Table 3.3: Parameters and Truncation Points for the Channel Length and the Oxide Thickness.

Nominal L	Nominal t_{ox}	$3\sigma_L$	$3\sigma_{t_{ox}}$	N_L	$N_{t_{ox}}$	N_{KL_g}
$65nm$	$1.5nm$	12%	5%	13	13	49

bases for expressing the deterministic temperature profile.

The complexity ratio of the mixed-mesh strategy to the deterministic thermal simulation is $(1+(N_{PC}-1)(N_{\mathrm{CM}}/N_{\mathrm{FM}})^2)$. In our experimental results, an accurate thermal yield profile can be estimated with the setting $N_{PC} = 53$, $N_{\mathrm{FM}} = 128$, $N_{\mathrm{CM}} = 16$ and $T_{\mathrm{res}} = 0.65^o$C. The complexity ratio is 1.8125. Therefore, the mixed-mesh strategy enhances the efficiency of the thermal yield profile estimator for catching up with those of deterministic thermal simulators.

3.4 Experimental Results

The developed stochastic projection based thermal analyzer and the stochastic collocation based thermal analyzer are implemented in C++ language and tested on a Linux system with Intel Xeon 3.0-GHz CPU and 32GB memory. The die size is $2.5mm \times 2.5mm \times 0.5mm$. The junction depth is set to be $20nm$ that is the nominal value for the $65nm$ technology [73] and the Debye length is set to be $2nm$ [102]. The floorplanning of test chip having 1.2 million functional gates is shown as Figure 3.17(a), and the geometries of chip and package are shown in Figure 3.17(b).

The device parameters, the truncation points of KL expansions for the channel length (N_L) and the oxide thickness ($N_{t_{ox}}$), and the number of device modeling grid (N_{KL_g}) are summarized in Table 3.3. Both N_L and $N_{t_{ox}}$ are decided by satisfying $\gamma_{N_L+1}/\sum_{i=1}^{N_L+1}\gamma_i \le 1\%$ and $\gamma_{N_{t_{ox}}+1}/\sum_{i=1}^{N_{t_{ox}}+1}\gamma_i \le 1\%$, respectively. To model the spatial correlation, both η_x/L_x and η_y/L_y are set to 0.98 for the correlation function shown in equation (3.7) [81].

By applying the modeling skill of thermal parameter mention in Figure 3.4 of Section 3.2.3 and the modeling skill for both of the heat transfer paths mentioned in [57], the thermal conductivity and the equivalent heat transfer coefficients of the primary and secondary heat flow paths for executing the deterministic simulator stated in Chapter 2 are summarized in Table 3.4. The boundary condition of each vertical surface is set to be isothermal. The top surface of the test circuit is divided into 128×128 grids for executing the deterministic thermal simulator.

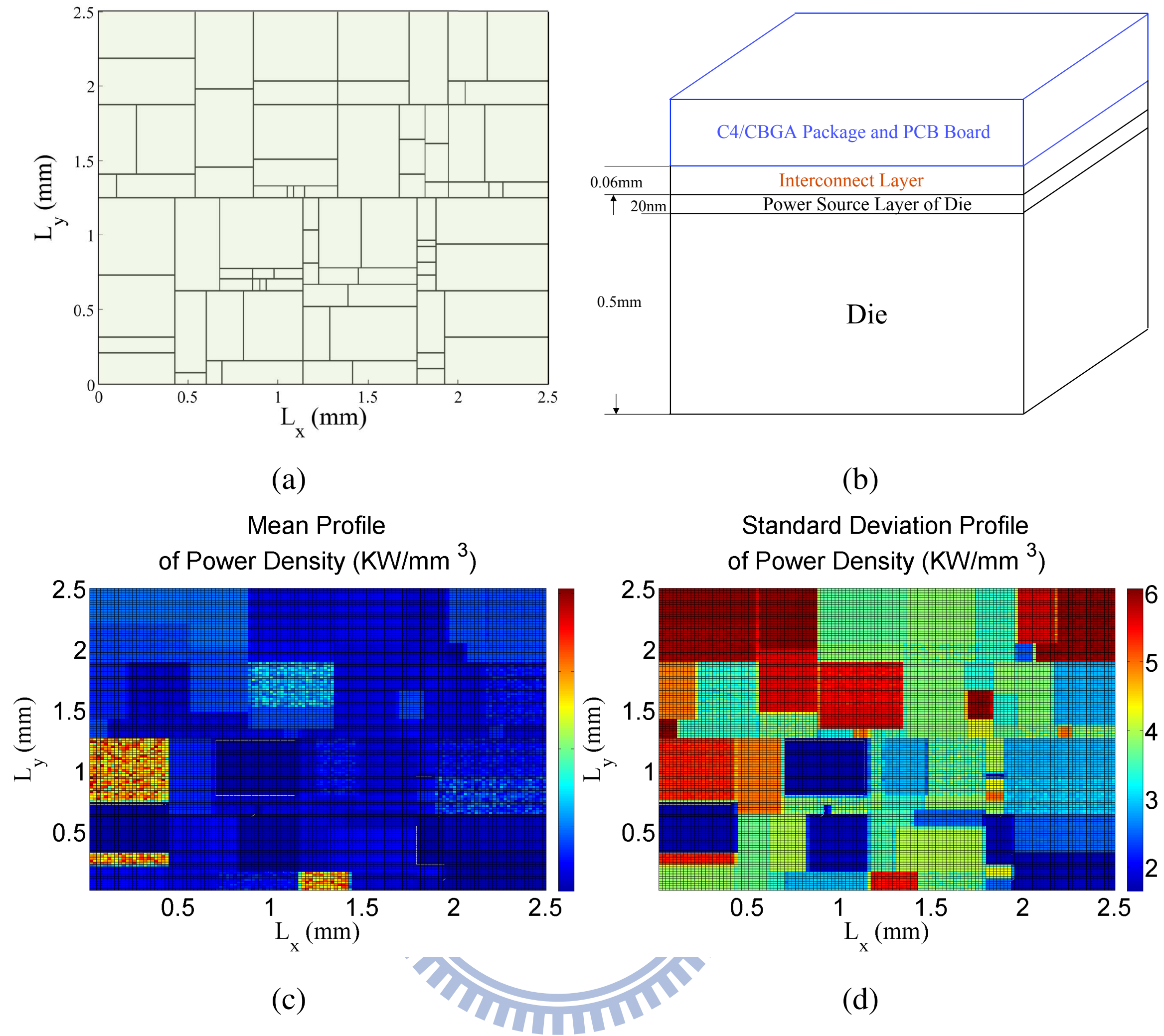


Figure 3.17: Floorplan of the test die, geometries of the test chip and package, and mean and standard deviation profiles of the power density on the test chip. (a) Floorplan of the test die. (b) Geometries of the test chip and package. (c) The mean profile of power density. (d) The standard deviation profile of power density. Here, L_x and L_y are the width and length of the test chip, respectively.

Table 3.4: Equivalent Thermal Parameters.

Parameter	Value
κ	104.6 W/(m·°C)
h_p	12000 W/(m^2·°C)
h_s	2017 W/(m^2·°C)

κ: the thermal conductivity of the die.
h_p: the equivalent primary heat transfer coefficient.
h_s: the equivalent secondary heat transfer coefficient.

Table 3.5: Accuracy and Efficiency of the Developed Statistical Expression Generators.

Device Variation		Monte Carlo‡ Method		Statistical Expression Generator†						Speedup	
				Projection Based			Collocation Based			Projection Based	Collocation Based
$\frac{\text{WID}}{\text{WID+D2D}}$	$\frac{\text{D2D}}{\text{WID+D2D}}$	#Samples	Runtime ①	Maximum Error		Runtime ②	Maximum Error		Runtime ③	①/②	①/③
				Mean	STDEV		Mean	STDEV			
40%	60%	6921	442.94	0.92%	2.69%	2.47s	0.91%	2.70%	2.68s	179.3×	165.2×
50%	50%	7011	448.70	0.93%	2.43%	2.42s	0.91%	2.68%	2.72s	185.4×	164.9×
60%	40%	7031	449.98	0.90%	2.53%	2.47s	0.90%	2.72%	2.74s	182.1×	164.2×

† The maximum error is obtained by comparing with the golden solution constructed by the Monte Carlo method using 2×10^5 samples.
‡ To demonstrate the efficiency, here, the Monte Carlo method is simulated till achieving the same accuracy of standard deviation as the developed methods. The runtime does not include the time of parsing input that is performed only once for all above methods. In this table, "STDEV" represents the standard deviation.

The estimated mean and standard deviation profiles of the power density under the settings of 60% of WID and 40% D2D variations to the total variation are shown in Figure 3.17(c)–(d), respectively.

3.4.1 Statistical Thermal Simulations With/Without Considering Electro-Thermal Effects

The Monte Carlo method with 2×10^5 samples, 100 grids for modeling device parameters, 60% of WID and 40% D2D variations to the total variations is performed to demonstrate the essentialness of the statistical electro-thermal simulation loop. Figure 3.18 presents the mean and standard deviation profiles of the on-chip temperature distribution with and without considering the temperature dependent effect of leakage powers. According to the mean profile results, the difference between Figure 3.18(a) (with considering the electro-thermal effect) and Figure 3.18(b) (without considering the electro-thermal effect) is over 16%. Furthermore, according to the standard deviation profile results, the difference between Figure 3.18(c) (with considering the electro-thermal effect) and Figure 3.18(d) (without considering the electro-thermal effect) can be over 31%. These substantial differences indicate that the statistical electro-thermal analysis is essential.

3.4.2 Accuracy and Efficiency

Given various ratios of WID variation and D2D variation to the total variation and 100 grids for modeling device parameters, the results of Monte Carlo method with 2×10^5 samples are used as the reference (golden) solution of statistical electro-thermal simulation.

The Level-1 Smolyak sparse grid formula that chooses the roots of H-PCs to be the sampling points is executed for the stochastic collocation based method. To compare the accuracy

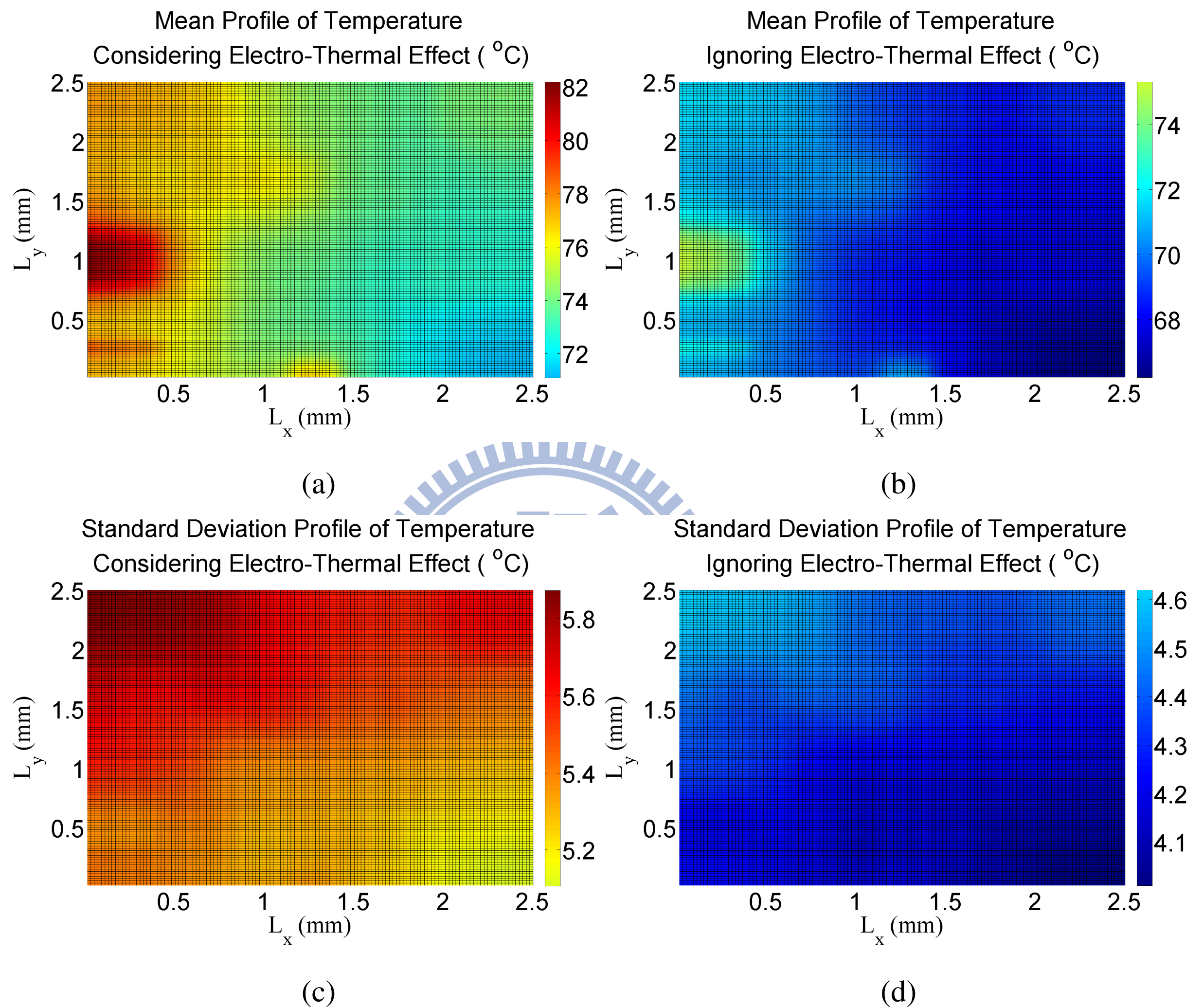


Figure 3.18: Results of the Monte Carlo method with or without considering electro-thermal effects. (a) The mean temperature profile with considering the electro-thermal effect. (b) The mean temperature profile without considering the electro-thermal effect. (c) The standard deviation profile of temperature distribution with considering the electro-thermal effect. (d) The standard deviation profile of temperature distribution without considering the electro-thermal effect.

between the stochastic projection based method and the stochastic collocation based method, the temperature distribution is expanded by the second order H-PCs without the cross product terms for the stochastic projection based method since it generate the similar form as that of the Level-1 Smolyak sparse grid formula. The number of executing the deterministic electro-thermal simulation is 53 for both of the stochastic projection based and stochastic collocation based methods because the stochastic projection based method generates $2 \times (N_L + N_{t_{ox}}) + 1$ H-PCs to approximate the temperature distribution, and the Level-1 Smolyak sparse grid formula uses $2 \times (N_L + N_{t_{ox}}) + 1$ sampling points. Both N_L and $N_{t_{ox}}$ are 13 as shown in Table 3.3.

In Table 3.5, the first two columns are the ratios of WID variation and D2D variation to the total variation, respectively. As shown in Table 3.5, the maximum errors of two proposed statistical expression generators are less than 3% for both estimated mean and standard deviation profiles among three different ratios of WID variation and D2D variation.

As shown in Table 3.5, the runtime of the stochastic collocation based method is larger than that of the stochastic projection based method because the required number of iterations for solving the deterministic electro-thermal problem of the stochastic collocation based method is larger; especially for those samples hitting the $(\mu - 3\sigma)$ value of each KL expanded random variable. The "Speedup" indicates the speedup of each developed method over the Monte Carlo method. The speedup of the stochastic projection method and the stochastic collocation method are over 179× and 164×, respectively. The results show that the proposed methods can be orders of magnitude faster than the Monte Carlo method.

The mean and standard deviation profiles of the temperature distribution on the test chip with 60% of WID variation and 40% of D2D variation to the total variation are shown in Figure 3.19(a)–(d). Figure 3.19(a) and (b) are the mean and standard deviation profiles estimated by the stochastic projection based method, respectively. Figure 3.19(c) and (d) are the mean and standard deviation profiles estimated by the stochastic collocation based method, respectively. The error distributions of the mean and standard deviation of the temperature distribution estimated by the stochastic projection based method are shown in Figure 3.19(e) and (f), respectively.

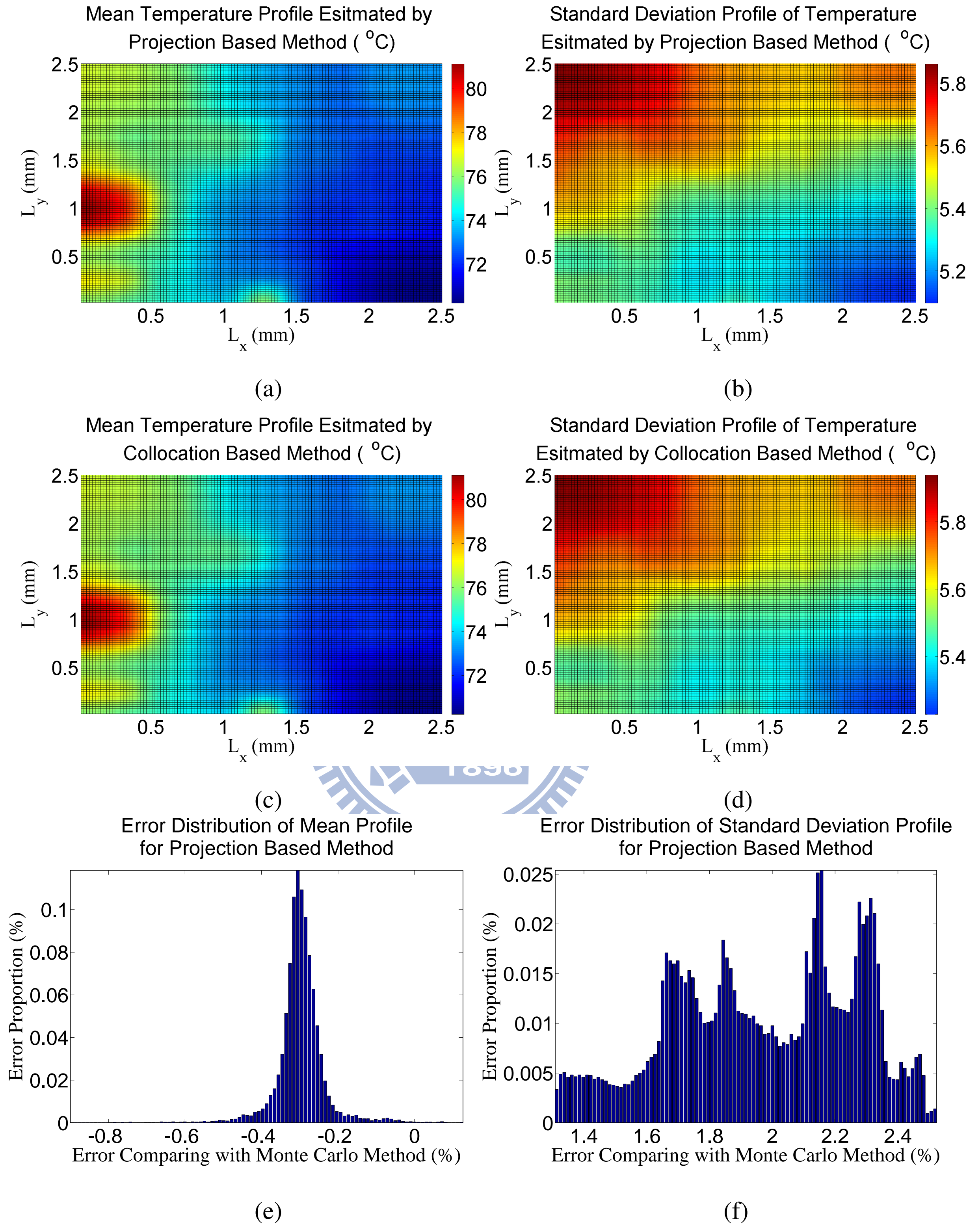


Figure 3.19: Simulation results of the developed methods. (a) and (b) the mean and standard deviation profiles of the estimated temperature distribution got by the stochastic projection method, respectively. (c) and (d) the mean and standard deviation profiles of the estimated temperature distribution obtained by the stochastic collocation method, respectively. (e) and (f) the error distributions of the mean and standard deviation of the estimated temperature distribution got by the stochastic projection method, respectively.

Table 3.6: Accuracy and Efficiency Comparison of the Skew Normal Model and APEX for Estimating Thermal Yield Profiles. The results are compared with the Monte Carlo method with 2×10^5 samples.

Variation		T_{ref}	Skew Normal				APEX			
			Projection		Collocation		Projection		Collocation	
$\frac{WID}{WID+D2D}$	$\frac{D2D}{WID+D2D}$		Runtime	MaxError	Runtime	MaxError	Runtime	MaxError	Runtime	MaxError
40%	60%	88.40°C	0.013s	1.51%	0.013s	1.63%	2.80s	1.91%	2.80s	1.97%
50%	50%	88.48°C	0.013s	1.46%	0.013s	1.52%	2.80s	1.87%	2.80s	1.90%
60%	40%	88.54°C	0.013s	1.37%	0.013s	1.41%	2.80s	2.27%	2.80s	2.32%

Thermal Yield Estimation

Based on the second order H-PCs without the cross product terms for the stochastic projection based method and the Level-1 Symolyak sparse grid formula for the stochastic collocation based method, the skew-normal based statistical moment matching method and APEX [99] are implemented for estimating thermal yield profiles. To avoid the instability of the Padè approximation for APEX, the stable two pole model [100] is implemented for finding the poles/zeros. Based on the average mean ($\bar{\mu}_T$) and the average standard deviation ($\bar{\sigma}_T$) of temperature obtained by the Monte Carlo method, the reference temperature (T_{ref}) specified by the designer is set to be $\bar{\mu}_T + 2.5\bar{\sigma}_T$. With various ratios of WID variation and D2D variation to the total variation, the results of the skew-normal based method and APEX for estimating thermal yield profiles are summarized in Table 3.6.

The "Projection" and "Collocation" indicate that the statistical expressions of the temperature distribution are generated by the stochastic projection method and stochastic collocation method, respectively. The "Runtime" is the execution time to obtain the thermal yield profile, and "MaxError" is the maximum error of the estimated thermal yield profile compared with the golden solution obtained by the Monte Carlo method. As shown in Table 3.6, both our statistical expression generators can provide accurate statistical on-chip temperature expressions for the thermal yield estimation. The maximum error of the skew-normal based method is less than 1.63% for all test situations, and the maximum error of APEX is less than 2.32%. It can be observed that the accuracy of the skew-normal based methods outperforms that of APEX.

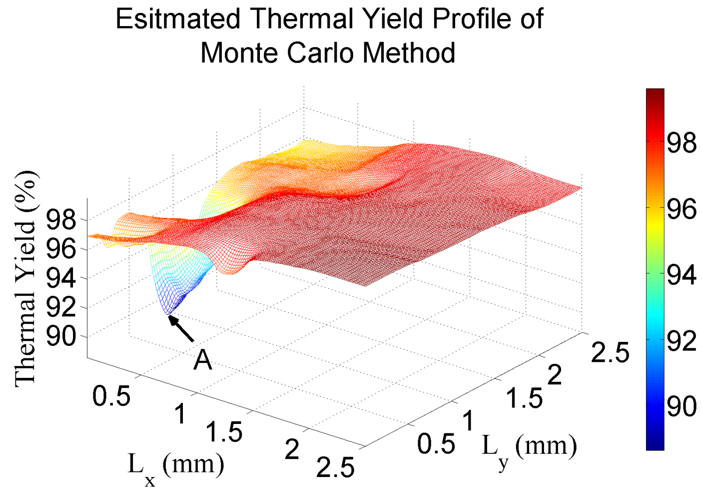
Furthermore, as shown in Table 3.6, the proposed skew-normal based method can achieve 215× speedup over APEX. It is because of two reasons. First, APEX needs a high order of statistical moments to get a tight bound of their generalized Chebshev inequality for the PDF/CDF shifting process. In our experimental results, it requires the first nine statistical moments to

achieve an accurate thermal yield profile even though [100] only needs the first four statistical moments to get the first two dominated poles. Rather than APEX, the skew-normal based method only needs to match the first three statistical moments to construct the model and can accurately estimate the thermal yield profile. Second, after the first two dominated poles are computed, APEX needs to solve equations to obtain the zeros of the first two dominated poles for constructing its exponential model. Rather than APEX, the skew-normal based method only needs to perform a constant-time lookup-table method to estimate the thermal yield profile after the first three statistical moments have been computed.

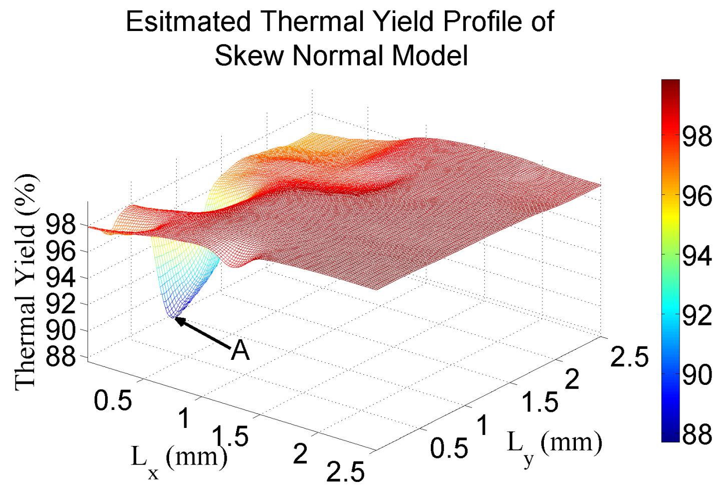
With $\left(\frac{WID}{WID+D2D}, \frac{D2D}{WID+D2D}\right) = (60\%, 40\%)$, results of the thermal yield profile estimation are shown in Figure 3.20. The thermal yield profile got by the Monte Carlo method is drawn in Figure 3.20(a). Figure 3.20(b) and (c) show the estimated thermal yield profiles of the proposed skew-normal based method and APEX, respectively. Comparing with the Monte Carlo method, the error distributions of the estimated thermal yield profiles of the proposed skew-normal based method and APEX are shown in Figure 3.21. Figure 3.21(a) and (b) are the error distributions of the proposed skew-normal based method and APEX, respectively. From Figure 3.20(a)–(b) and Figure 3.21(a), it can be observed that the developed skew-normal based method can accurately deliver the on-chip thermal yield profile. However, Figure 3.20(c) reveals that the estimated thermal yield profile got by APEX exceeds 100% in some region since APEX doesn't guarantee to generate a statistical model for preserving the property of CDF.

To further demonstrate that the skew-normal model based method can accurately estimate the temperature CDF at a position on the chip. Figure 3.22 plots the CDF curve of temperature at position **A** in Fig 3.20(a) got by the Monte Carlo method and its estimated CDF curves got by the skew-normal model bases method, and APEX with the 9-th order and the 4-th order for the PDF/CDF shifting process.

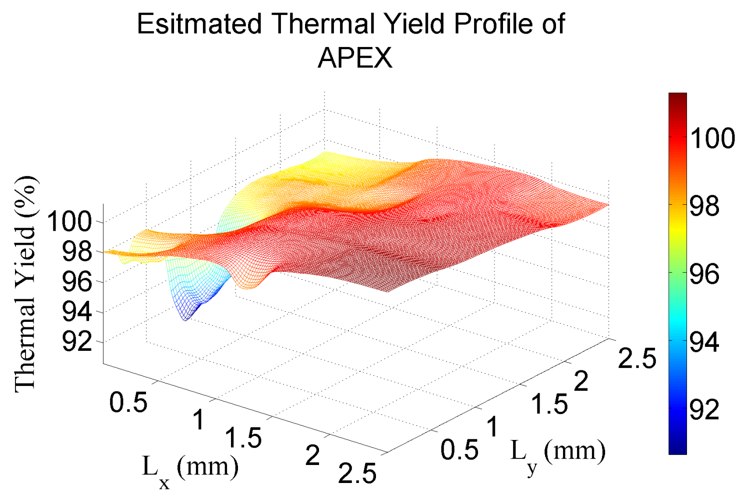
As shown in Figure 3.22, the estimated CDF curve got by the skew-normal model based method can tightly fit the CDF curve obtained by the Monte Carlo method. However, APEX with the 4-th order can not meet the result got by the Monte Carlo method. Although the accuracy of estimated CDF curve got by APEX can be improved by increasing the order to 9, it still cannot accurately estimate the thermal yield for a smaller reference temperature value as



(a)



(b)



(c)

Figure 3.20: Thermal yield profiles of the test chip with $\left(\frac{WID}{WID+D2D}, \frac{D2D}{WID+D2D}\right) = (60\%, 40\%)$. (a) Profile obtained by the Monte Carlo method. (b) Profile obtained by the proposed skew-normal based method. (c) Profile obtained by APEX.

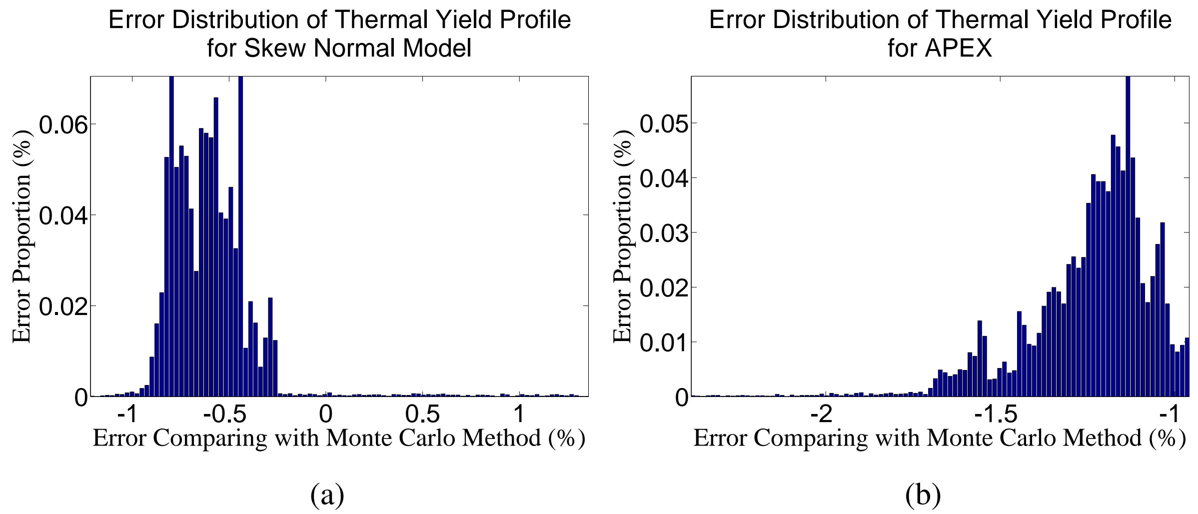


Figure 3.21: The error distributions of the skew-normal based method and APEX. (a) Distribution of the skew-normal based method comparing with the Monte Carlo method. (b) Distribution of APEX comparing with the Monte Carlo method.

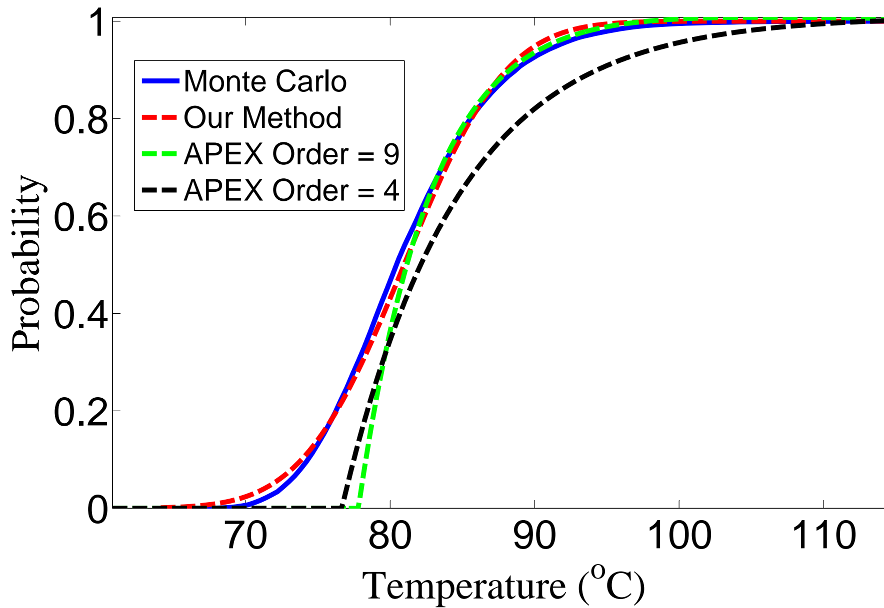
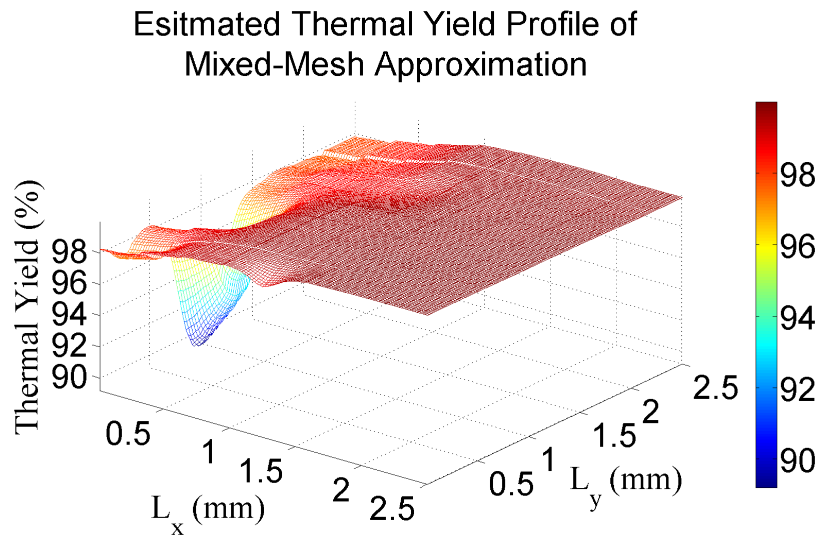


Figure 3.22: The temperature CDF curve at position **A** in Figure 3.20(a) got by the Monte Carlo method, and its estimated CDF curves obtained by the skew-normal model based method, APEX with the 4-th oredr and the 9-th order for the PDF/CDF shifting process.

illustrated in Figure 3.22.

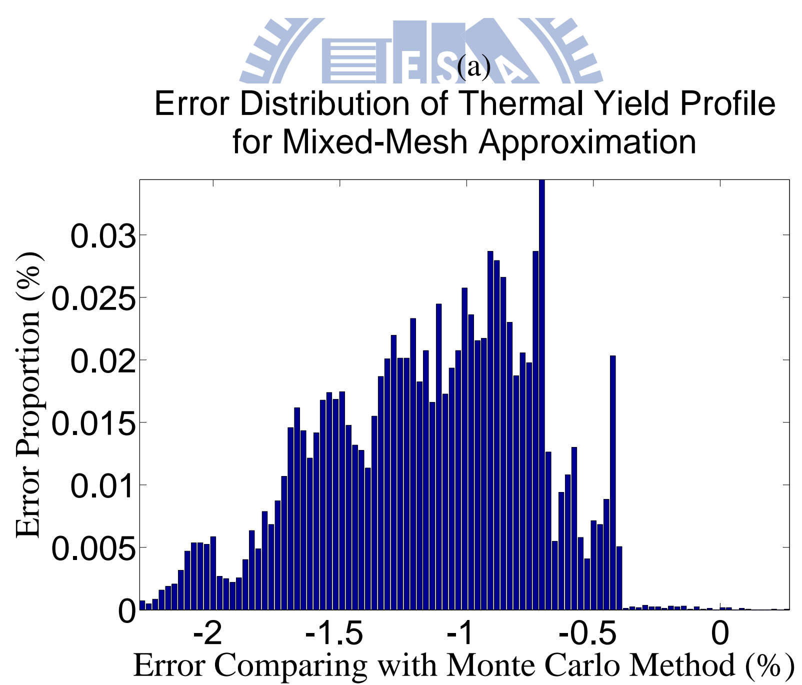
Mixed-Mesh Thermal Yield Estimation

The mixed-grid thermal yield estimation strategy presented in section 3.3.5 has been implemented into the statistical thermal expression generators to demonstrate its effectiveness. The estimated thermal yield profile of the test chip with the stochastic projection based statistical expression generator is shown in Figure 3.23. In this test case, the difference between the maximum and minimum mean temperatures, $\Delta\overline{T}_{\max}$, can be calculated as 11.1°Cwith the number of fine grid being 128×128, and the temperature resolution, T_{res}, is set to be 0.65°C. Hence, the number of coarse gird for the remaining $N_{PC} - 1$ deterministic thermal simulations can be calculated as 16×16. Comparing with the result from the Monte Carlo method, the maximum error of the estimated thermal yield profile obtained by the mixed-grid strategy is only 2.24% which is slightly larger than the result shown in Table 3.6. However, the runtime of building the statistical expression of on-chip temperature distribution can be reduced to 0.019 seconds (The runtime without using the mixed-grid strategy is 2.47 seconds as shown in Table 3.5.). Thus, the mixed-mesh strategy achieves 130× speedup over the baseline statistical polynomial expression generator. The runtime for estimating the thermal yield profile is still 0.013 seconds. Totally, the runtime for executing the entire flow of the mixed-mesh thermal yield estimation is 0.032s.



(a)

ES A



(b)

Figure 3.23: The estimated thermal yield profile and the error distribution of the mixed-grid thermal estimation strategy.

Chapter 4

Simulation Method III – *LUTSim: A Look-Up Table Based Thermal Simulator for 3-D ICs*

In this chapter, a look-up table based thermal simulator, LUTSim, is presented to efficiently estimate the temperature profile of three-dimensional integrated circuits. With utilizing the pre-built tables of the temperature response induced by a unit power source, the superposition, interpolation, and a recursive table look-up techniques are applied to estimate the temperature profile of the three-dimensional integrated circuits.

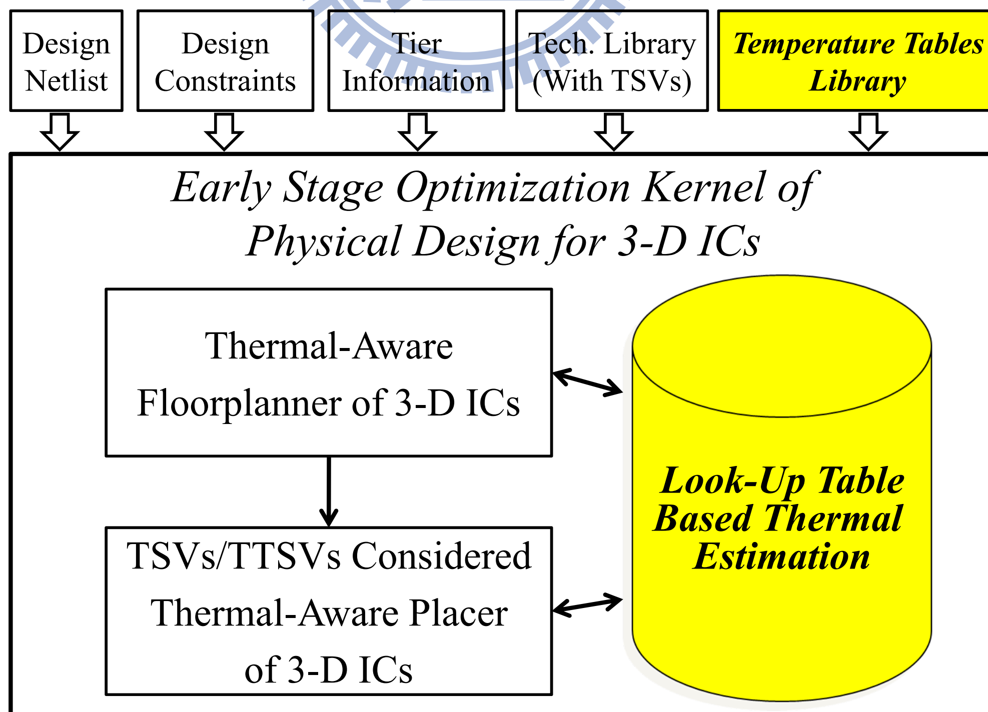


Figure 4.1: Key points of LUTSim for the early physical design stages in 3-D ICs.

As shown in Figure 4.1, LUTSim is conceptually similar with the circuit performance analysis, such timing and power analysis, using the standard cell library. For the thermal simulation (the circuit performance analysis), the thermal (electrical) characteristics of modeling grids (gates) are first pre-characterized by the detailed thermal simulation (the SPICE simulation), and the temperature profiles (delays/powers of gates) are tabled in the library files. With the pre-characterized tables of the temperature (electrical characteristics such as delays or powers), the thermal analysis (the circuit performance analysis such as the static timing analysis) can be efficiently performed via table look-up instead of executing the time-consuming detail thermal simulation (SPICE simulation).

With the framework of look-up table (LUT), LUTSim can efficiently calculate full-chip temperature profile *without* solving the large scale system of the modified nodal analysis (MNA), which is the major computation effort of the prior arts [50–59], of the equivalent thermal circuit. More important, besides the advantage of the full-chip thermal simulation, if TSVs are moved by the optimization engines, this simulation method can update the on-chip temperature by table looking-up without re-performing the dealing process of the large scale thermal conductance matrix.

The organization of this chapter is summarized as follows. The compact thermal models for early design stages of 3-D ICs is stated in section 4.1. Then, LUTSim is described in section 4.2. Finally, the experimental results are given in section 4.3.

4.1 Thermal Model for Early Design Stages of TSVs based 3-D IC Structures

As mentioned in section 1.3, the TSVs based 3-D IC structures can provide much interconnect density, and is the most popular implementation categories. Therefore, this work focuses on thermal analysis of these structures of 3-D ICs. As exhibited in Figure 4.2, the thermal model for the early physical design stages of TSVs based 3-D ICs consists of following portions[1].

1. The primary heat flow path consists of the heat spreader, heat sink and package. The secondary heat flow path consists of the input/output pads, the package substrate and the

[1]Although LUTSim adopts the thermal model of TSV based 3-D ICs, its framework can be extended to other structures of 3-D ICs, e.g. face-to-face, contactless interconnection and wire-bound structures [3].

print circuit board. Using the techniques stated in section 2.1, the heat transfer coefficients of the primary and secondary heat flow paths can be equalized to two different effective heat transfer coefficients h_p and h_s, respectively.

2. Interconnect layers consists of the interconnects and the dielectric. Because the routing information is unknown in the early physical design stages, each interconnect layer can be modeled as a homogeneous layer with an effective thermal resistance or conductivity using the modeling techniques in [56] with the empirical density and the regularity structure assumption of wires.

3. Functional blocks of tiers are modeled as power sources attached to the thin layers that are close to the top surfaces of the silicon bulk and the stacked silicon substrates, and there are TSVs in each silicon substrate of each stacked tier, e.g. tiers 2 and 3 in Figure 4.2.

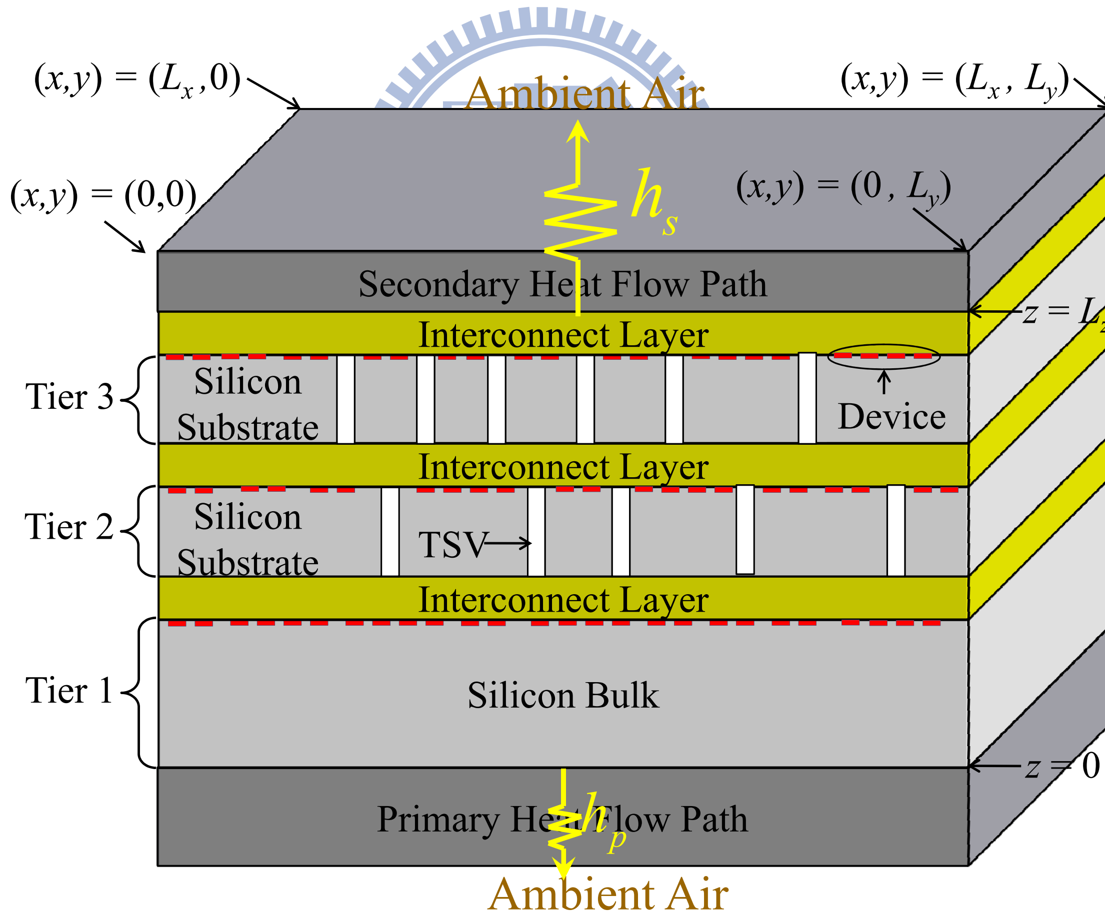


Figure 4.2: Thermal model for the early design stage of a 3-D IC with three tiers.

As stated previously, this work focuses on the steady-state thermal analysis because the steady state on-chip temperature is more concerned in the physical design stages. With the

above thermal model, the temperature profile of a TSVs based 3-D IC, $T(\mathbf{r})$, can be governed by the following steady-state heat transfer equation.

$$\nabla \cdot (\kappa(\mathbf{r}) \nabla T(\mathbf{r})) = -p(\mathbf{r}), \tag{4.1}$$

subject to the boundary condition as

$$\kappa(\mathbf{r}_{b_s}) \frac{\partial T(\mathbf{r}_{b_s})}{\partial \vec{n}_{b_s}} + h_{b_s} T(\mathbf{r}_{b_s}) = f_{b_s}(\mathbf{r}_{b_s}). \tag{4.2}$$

Here, $\mathbf{r} = (x, y, z) \in D$, $D = (0, L_x) \times (0, L_y) \times (0, L_z)$ is the domain of the chip, L_x and L_y are the lateral sizes of the chip, L_z is the thickness of the chip, $\kappa(\mathbf{r})$ is the thermal conductivity (W/m·°C) of the chip, and ∇ is the diverge operator. The b_s is any specific boundary surfaces of the chip, $\mathbf{r}_{b_s}$ is the position on b_s, h_{b_s} is the heat-transfer coefficient on b_s, $f_{b_s}(\mathbf{r}_{b_s})$ is the heat flux function on b_s, $\vec{n}_{b_s}$ is the outward normal to b_s, $\partial/\partial\vec{n}_{b_s}$ denotes the differentiation along the outward normal to b_s, and $p(\mathbf{r})$ is the power density profile of the chip. Since the major portion of device current flows through the channel, $p(\mathbf{r})$ has its value only when $\mathbf{r}$ is in the thin layers close to the top surfaces of tiers. The thicknesses of these thin layers are equal to the junction depth of devices.

Although the steady state heat transfer equations shown in equations (4.1) and (4.2) are similar with the steady state heat transfer equations for 2-D ICs stated in section 2.1, the difference is that the thermal conductivity profile $\kappa(\mathbf{r})$ in the silicon substrates and bulk can not be treated as a constant value because there are TSVs these layers. Therefore, the analytical simulation framework stated in Chapter 2 should be modified for the TSV based 3-D ICs. Nevertheless, the author refers the above modification to be an open research topic. Instead of the analytical simulation approaches, using the finite difference method (FDM), the steady-state heat transfer equations (4.1) and (4.2) can be transformed into a SPICE-compatible equivalent thermal circuit [50], and the steady-state temperature profile of a 3-D IC can be obtained by solving the following modified nodal analysis (MNA) system.

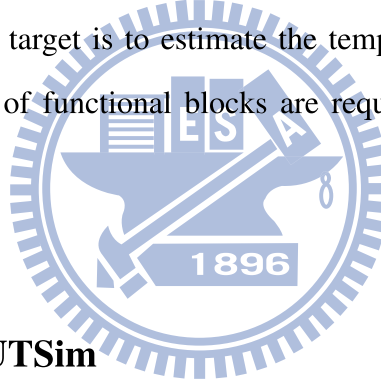
$$\mathbf{G}\mathbf{T} = \mathbf{p}. \tag{4.3}$$

Here, $\mathbf{G}$ is the thermal conductance matrix, and $\mathbf{T}$ is the temperature profile vector of simulation grids. $\mathbf{p}$ is the power vector of modeling grids, its entries have non-zero values only for grids in

S_g, and S_g is the set of grids close to the top surfaces of tiers. With equation (4.3), our target is to estimate the temperature profile of the grids in S_g because temperatures of functional blocks are required to be well concerned in the early physical design stages.

Once the thermal conductance matrix of the TSV based 3-D IC is constructed, advanced numerical simulation framework such as [51, 54, 55] can be adopted to solve the temperature profile in simulation grids, $\mathbf{T}$. However, since the positions of TSVs will be moved by early stage design engines such as floorplanners or placers, the thermal conductance matrix, $\mathbf{G}$, will be different after an optimization step is executed. Therefore, the handling process[2] of the thermal conductance matrix needs to be re-performed for each optimization loop, and this decreases the efficiency of [51,54,55]. Therefore, to avoid the re-handling process of the thermal conductance while thermal-aware design engines are executing, LUTSim employs the look-up table framework to simulate the temperature profile of 3-D ICs.

With equation (4.3), our target is to estimate the temperature profile of grids in S_g, $\mathbf{T}_{S_g}$, because temperature values of functional blocks are required to be well concerned in early physical design stages.



4.2 LUTSim

4.2.1 Overview of LUTSim

The flowchart of LUTSim is summarized in Figure 4.3. Before the design information, such as the floorplan/placement and the powers of macros/gates, is given, the table establishment is executed to pre-building the tables for the temperature response of unit powers. To execute the table establishment, the following chip information is required. 1) The thicknesses of silicon substrates and silicon bulk, and the material of TSV/TTSV; 2) The number of tiers and the effective heat transfer coefficients of primary and secondary heat flow paths; 3) The thicknesses and effective thermal conductivities of interconnect layers; 4) The outline of the chip.

As shown in Figure 4.1, since one of major objectives of the proposed thermal analyzer is the temperature estimation for floorplanners, the table establishment is required being performed

[2]The handling processes of the thermal conductance matrix in advanced numerical simulation frameworks [51, 54, 55] are the LU decomposition of a tri-diagonal matrix in each propagating direction [51] and the multilevel restriction-interpolation construction [54, 55].

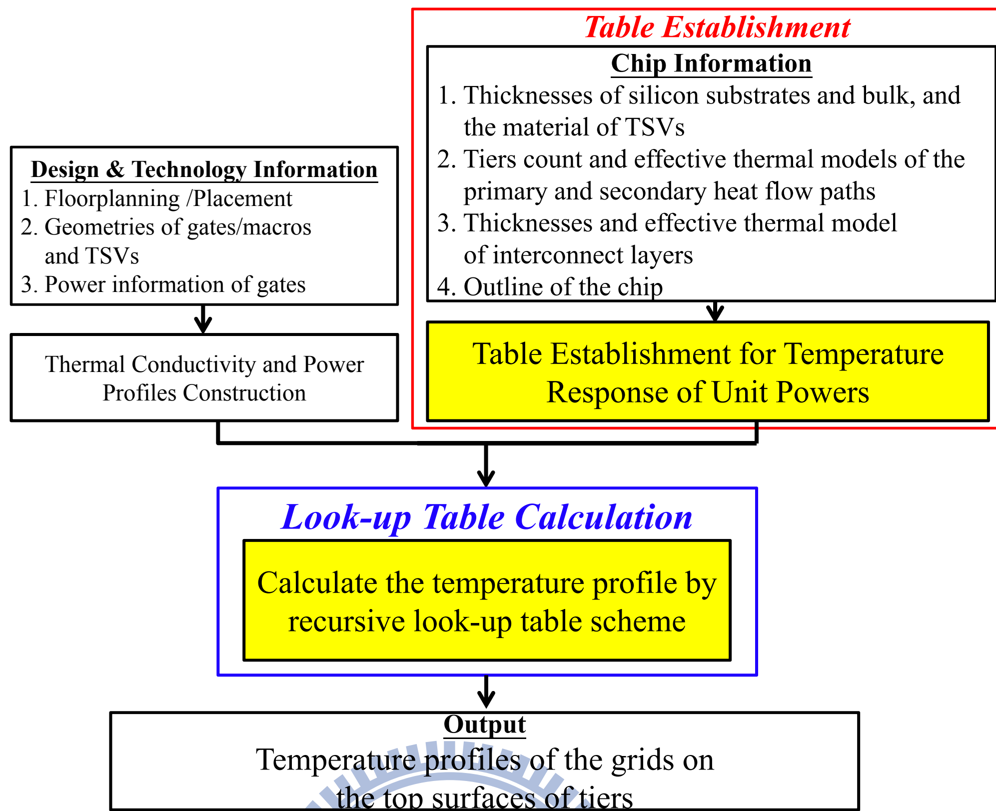


Figure 4.3: The flowchart of LUTSim.

before the floorplanning stage. Due to the following reasons, the chip information for the table establishment is practically available before the floorplanning stage.

1. The thicknesses of the silicon substrates and silicon bulk, and the material of TSV/TTSV are manufacturing parameters.

2. The number of tiers is practically determined by the partition [3,103] or is given during the heterogeneous integration before the floorplanning stage. The heat transfer coefficients of primary and secondary heat flow paths are generally determined in the high level system design [68].

3. The thicknesses and effective thermal conductivities of interconnect layers can be estimated by the modeling technique [57] before the floorplanning is processed.

4. Recently, the fixed-outline floorplanning [38,104–106] brings more and more attention in modern ASIC designs of 2-D and 3-D ICs. It enables the hierarchical framework, which is the prevalent framework for dealing with the rapid increasing design complexity and is

not supported by the classical outline-free floorplanning [104], for modern ASIC designs. The outline estimation technique of 2-D ICs has been proposed in [105], which is based on the total area of blocks and whitespace threshold. Xiao et. al. [38] has further extended the outline estimation technique to 3-D ICs, which can be written as

$$W = ((1+\epsilon)A\gamma/L)^{1/2}, H = ((1+\epsilon)A/\gamma L)^{1/2}. \tag{4.4}$$

Here, W/H is the chip width/height, A is the total area of macros, L is the number of tiers, γ is the aspect ratio of the chip, and ϵ is the maximum allowable fraction of the white space. Therefore, outlines of 3-D ICs are practically available before the floorplanning stage of modern VLSI designs.

Therefore, the table establishment can be executed before the floorplanning stage, and the pre-built tables can be reduced while the floorplaners and placers are executed.

As the design information and the pre-built tables of the temperature responses are inputted, a recursive look-up table technique is executed to calculate the temperature profile of the 3-D IC. Finally, the temperature profile of grids on top surfaces of stacked tiers are reported.

The following sections will detail the technical contents of LUTSim. For the sake of simplicity, "temperature response induced by a unit power source at a specific position with ignoring the effect of TSVs/TTSVs" is abbreviated as "TR-UPS".

4.2.2 Recursive Look-Up Table based Full-Chip Thermal Simulation Framework

Since the junction depth is much thiner than those of silicon substrates and silicon bulk, while a floorplanner or a placer [36, 38, 42, 107] is executing, the thermal conductivity variation of a tier due to the shifting of gates can be reasonably ignored. Therefore, with ignoring TSVs/TTSVs, the thermal model of each tier can be characterized as a homogeneous-material layer. Replacing the thermal model of this 3-D IC with a thermal model without considering TSVs/TTSVs, the MNA equation for calculating its steady-state temperature profile becomes

$$\mathbf{G}_h \mathbf{T}_h = \mathbf{p}. \tag{4.5}$$

Here, $\mathbf{G}_h$ is the thermal conductance matrix of TSVs/TTSVs ignored 3-D IC, $\mathbf{T}_h$ is the temperature profile vector for the simulation grids of TSVs/TTSVs ignored 3-D IC, and $\mathbf{T}_h$ can be

computed as

$$\mathbf{T}_h = \mathbf{G}_h^{-1}\mathbf{p} = \sum_{i\in S_g} \mathbf{G}_h^{-1}\mathbf{p}_i = \sum_{i\in S_g} p_i\mathbf{G}_h^{-1}\mathbf{e}_i = \sum_{i\in S_g} p_i\mathbf{T}_i^{\mathrm{I}}. \tag{4.6}$$

Here, $\mathbf{p}_i = p_i\mathbf{e}_i$, each p_i is the power of grid i, $\mathbf{e}_i$ is the vector with i-th component 1 and everywhere else 0, and $\mathbf{T}_i^{\mathrm{I}} \stackrel{\mathrm{def}}{=} \mathbf{G}_h^{-1}\mathbf{e}_i$ is the temperature response induced by the unit power source vector $\mathbf{e}_i$. $\sum_{i\in S_g} \mathbf{p}_i = \mathbf{p}$ because $p_j = 0$ for each grid $j \notin S_g$. Since the temperature values of grids in S_g are required to be well concerned, they are extracted as

$$\mathbf{T}_{h,S_g} = \sum_{i\in S_g} p_i\mathbf{h}_i, \tag{4.7}$$

where $\mathbf{T}_{h,S_g}$ is an $N_{S_g} \times 1$ vector for the temperature profile of grids in S_g, each $\mathbf{h}_i$ is an $N_{S_g} \times 1$ vector that extracts the values of $\mathbf{T}_i^{\mathrm{I}}$ for grids in S_g, and N_{S_g} is the number of grids in S_g.

Each $\mathbf{h}_i$ can be pre-calculated and stored as the technology and chip information is given.[3] With the pre-built $\mathbf{h}_i$'s, $\mathbf{T}_{h,S_g}$ can be computed by using the superposition of $p_i\mathbf{h}_i$'s. However, the material of tiers, which have TSVs/TTSVs passing through, actually is non-homogeneous. Hence, LUTSim is developed to calculate the temperature profile of TSVs/TTSVs considered 3-D ICs.

Considering the effect of TSVs/TTSVs, the MNA equation for calculating steady-state temperature profile of the 3-D IC can be written as

$$(\mathbf{G}_h + \Delta\mathbf{G})\mathbf{T} = \mathbf{p}, \tag{4.8}$$

where $\Delta\mathbf{G}$ is the variation of thermal conductance matrix induced by TSVs/TTSVs corresponding to $\mathbf{G}_h$.

Before proceeding the rest of contents, the convergence of LUTSim is guaranteed by *Proposition 1*, and its proof is presented in APPENDIX C.

Proposition 1. *Given an FDM based SPICE-compatible equivalent thermal circuit for a 3-D IC with the thermal conductance matrix* $\mathbf{G} = \mathbf{G}_h + \Delta\mathbf{G}$, *the temperature profile of this 3-D IC is*

$$\mathbf{T} = \sum_{i=0}^{\infty}(-1)^i \left(\mathbf{G}_h^{-1}\Delta\mathbf{G}\right)^i \mathbf{G}_h^{-1}\mathbf{p}. \tag{4.9}$$

■

[3] In this work, although the sparse LU decomposition based fast MNA solver [10] is employed to calculate each $\mathbf{h}_i$, advanced thermal simulation methods such as [51, 54] and the GIT based 3-D IC thermal simulation method stated in section 2.3 can be adapted to speed up the pre-simulation runtime.

Truncating the high order terms in equation (4.9), the q-th order approximation of $\mathbf{T}$ is

$$\mathbf{T} \approx \mathbf{T}^q \stackrel{\text{def}}{=} \mathbf{m}_0 + \sum_{i=1}^{q} \mathbf{m}_i, \tag{4.10}$$

where $\mathbf{m}_0 = \mathbf{T}_h = \mathbf{G}_h^{-1}\mathbf{p}$, and $\mathbf{m}_i = \mathbf{G}_h^{-1}\mathbf{b}_{i-1}$ with $\mathbf{b}_{i-1} = -\Delta\mathbf{G}\mathbf{m}_{i-1}$ for $i \geq 1$.

Since $\mathbf{m}_i = \mathbf{G}_h^{-1}\mathbf{b}_{i-1}$ has the same form as $\mathbf{T}_h$, each $\mathbf{m}_i$ can be recursively calculated by using the look-up table technique. Because we mainly concern the temperature profile $\mathbf{T}_{S_g}$ for those grids in S_g, with equation (4.10), $\mathbf{T}_{S_g}$ can be written as

$$\mathbf{T}_{S_g} \approx \mathbf{T}_{S_g}^q \stackrel{\text{def}}{=} \mathbf{T}_{h,S_g} + \sum_{i=1}^{q} (\mathbf{m}_i)_{S_g}, \tag{4.11}$$

where each symbol with the sub-index S_g means that only the related values in set S_g are calculated and concerned.

The complexity of directly solving $\mathbf{T}_{h,S_g}$ from equation (4.7) is $O(N_{S_g}^2)$, and the memory usage is $O(N_{S_g}^2)$ for storing $\mathbf{h}_i$'s. Moreover, the complexity for solving equation (4.11) is much higher due to the recursive look-up table procedure. Hence, a double-mesh look-up table temperature calculation technique is developed to improve the efficiency and save the memory usage.

4.2.3 Fine-Mesh Table Establishment

Because the on-chip heat flow mainly passes through the vertical direction (the directions along the primary and second heat flow paths shown in Figure 4.2) of the chip, TR-UPS has the *lateral locality* property, which means that TR-UPS has significant values in the local lateral region close to the grid with the inserted unit power source. Moreover, since the effective lateral thermal conductances of simulation grids in a local region are similar, TR-UPS also has the *local similarity* property, which means that the temperature responses of unit power sources inserted at different grids in a local lateral region have similar waveforms. These two properties are illustrated in Fig 4.4.

Based on lateral locality, as shown in Figure 4.5, the table establishing process for $\mathbf{h}_i$ of the grid $i \in S_g$ is proceed as follows. After inserting a unit power source to the grid $i \in S_g$ (Figure 4.5.(a)), the thermal simulation is performed for calculating $\mathbf{h}_i$. Then, as shown in Figure 4.5.(b), a truncation window W is chosen to extract the significant portion of $\mathbf{h}_i$. Then,

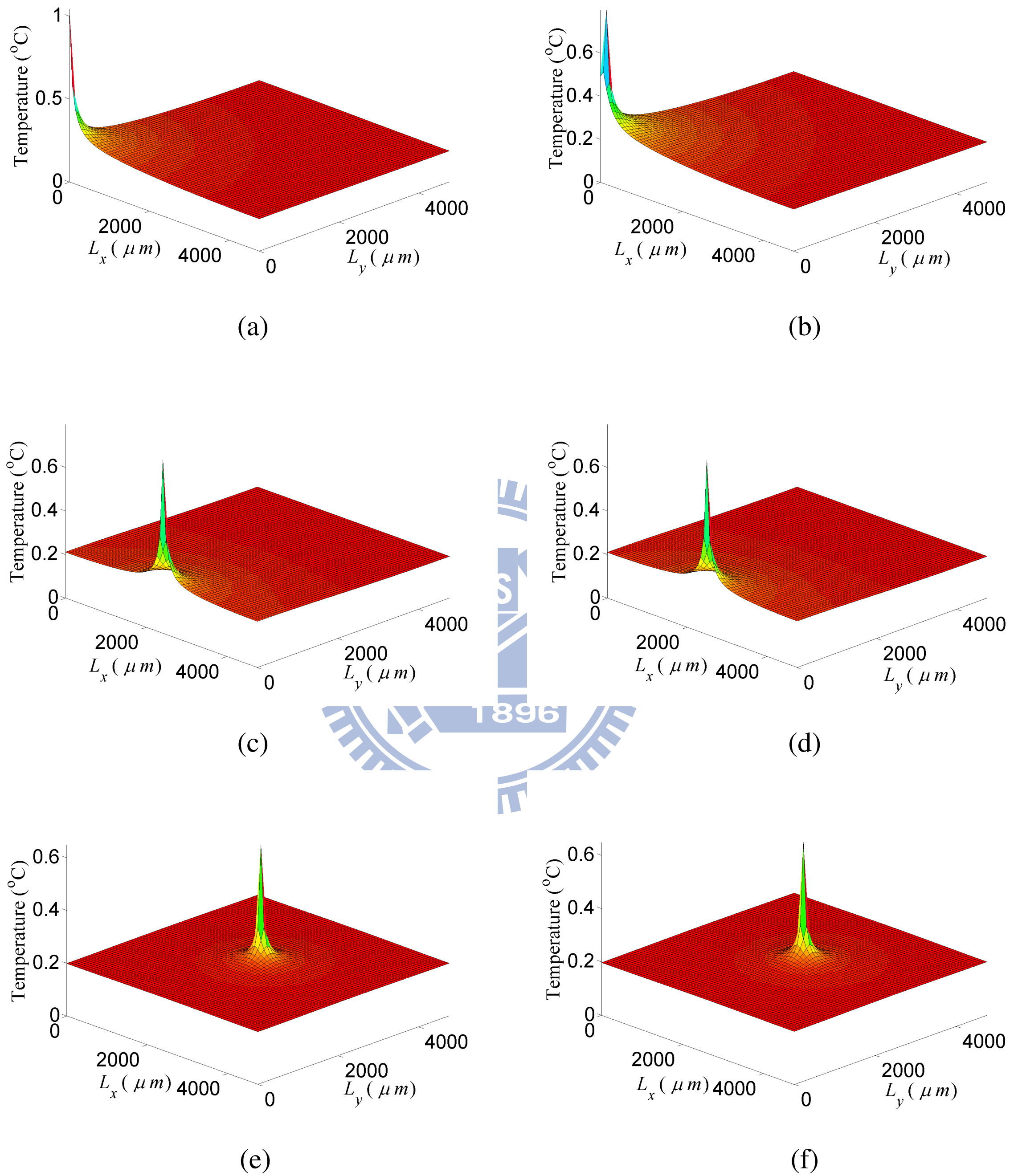


Figure 4.4: Examples for the *lateral locality* and *local similarity* of the temperature response induced by a unit power source. Each unit power source is inserted to a grid on the top-surface of the tier adjacent to the secondary heat flow path. (a)–(f) are the temperature responses with inserting a unit power source on grids (0, 0), (1, 1), (32, 0), (33, 0), (32, 32) and (33, 33), respectively.

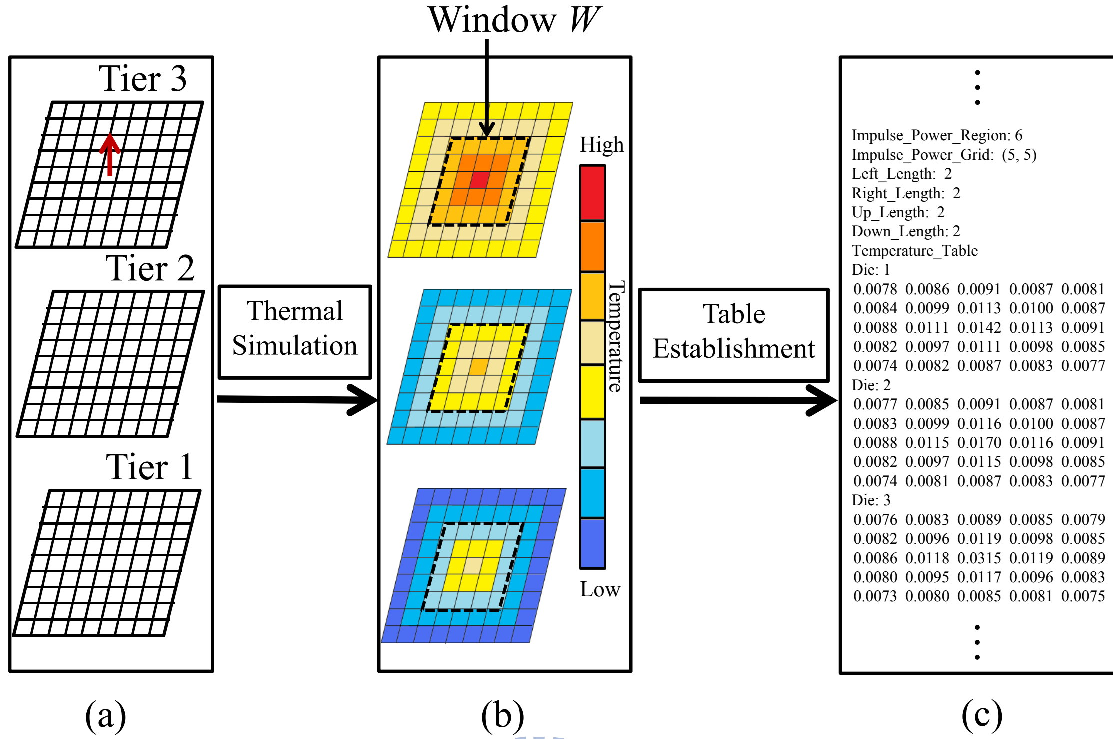


Figure 4.5: The table establishing process of TR-UPS of a specific grid in S_g.

the values of $\mathbf{h}_i$ for grids within the region of W are stored as a table, TRTAB$_i$. The following strategy is applied to obtain the size of W. Based on the local similarity, we separately insert a unit power source to a representative grid in S_g among different regions of the chip, and simulate their inducing temperature responses. As shown in Figure 4.4, although the minimal value of each temperature response is are not equal to zero, the waveform of the temperature response outside the region with significant values are smooth. Therefore, for $\mathbf{h}_i$ of a representative grid i, whether a grid k belongs to a window W_i is decided by the following criterion,

$$\frac{(h_i^k - h_i^{i_{min}})}{(h_i^{i_{max}} - h_i^{i_{min}})} \geq \eta. \tag{4.12}$$

Here, h_i^k is the entry of $\mathbf{h}_i$ corresponding to the grid $k \in S_g$, $h_i^{i_{min}}$ is the minimum entry of $\mathbf{h}_i$, i_{min} is the grid having $h_i^{i_{min}}$, $h_i^{i_{max}}$ is the maximum entry of $\mathbf{h}_i$, i_{max} is the grid having $h_i^{i_{max}}$, and η is a user specified threshold value. In the experimental results, an accurate result can be achieved with setting η as 5%. Then, the maximum distance between each grid $k \in W_i$ and the grid i_{max} is set to be the window size of W_i. After the size of each W_i is obtained, the size of W is chosen to be the maximum value among the sizes of W_i's.

With local similarity, several grids in a local region can share the temperature response of

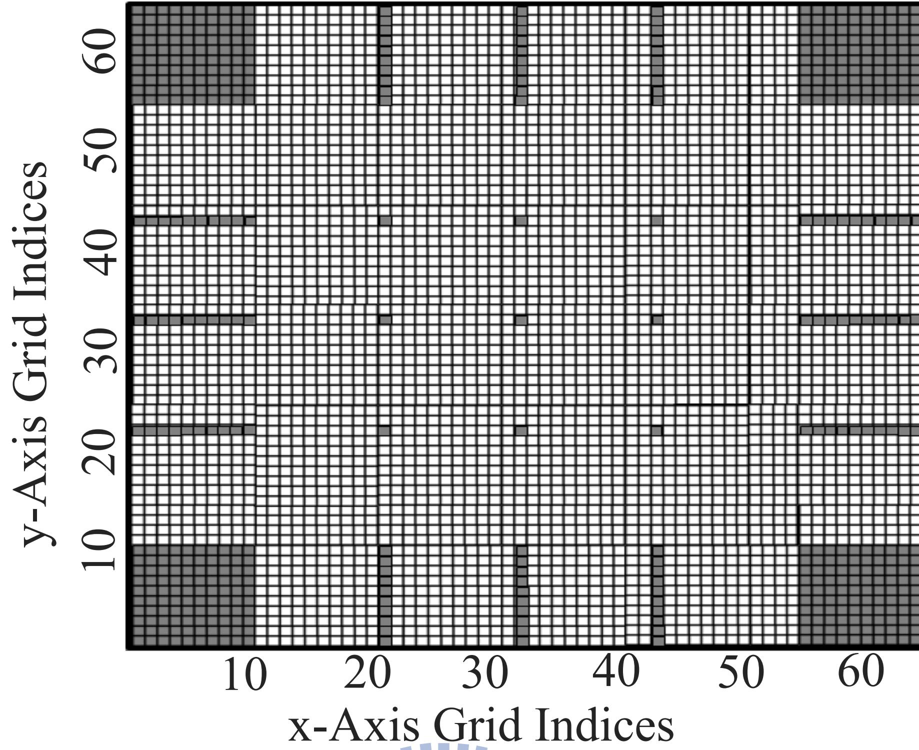


Figure 4.6: An example of the selected representative grids in S_g of a specific tier. Gray color grids are the representative grids.

a unit power source located at a representative grid and still achieve the accurate temperature profile. Therefore, to save the memory usage, only a small amount of representative grids (much less than the total grids of simulation mesh) distributed on the simulation mesh are selected to construct TR-UPS tables. As shown in Fig 4.4, the waveform of $\mathbf{h}_m$ corresponding to the gird m at the region around the corner of the chip is different from that of $\mathbf{h}_n$ corresponding to the gird n locates at the region around the center of the chip.

Hence, the selection of representative grids can be proceeded as follows. First, the grids at the region around the corner of the chip are successively selected. After a grid i is selected, with performing the table establishing process shown in Figure 4.5, the table of its $\mathbf{h}_i$ is obtained and saved. The above process is repeated until the waveform of $\mathbf{h}_i$ in the x- or y-direction are similar with that of the $\mathbf{h}_j$ corresponding to the grid j at the region around the center of the chip. Then, the rest area of the chip is uniformly divided into several regions, and the grid at the center of each region is chosen as a representative grid for building the TR-UPS table. After performing the above procedure, an example of the selected representative grids is shown in Figure 4.6.

To calculate $\mathbf{T}_{h,S_g}$, we also need $\mathbf{h}_l$ for the gird $l \in S_g$ but not a representative grid, i.e. lack of

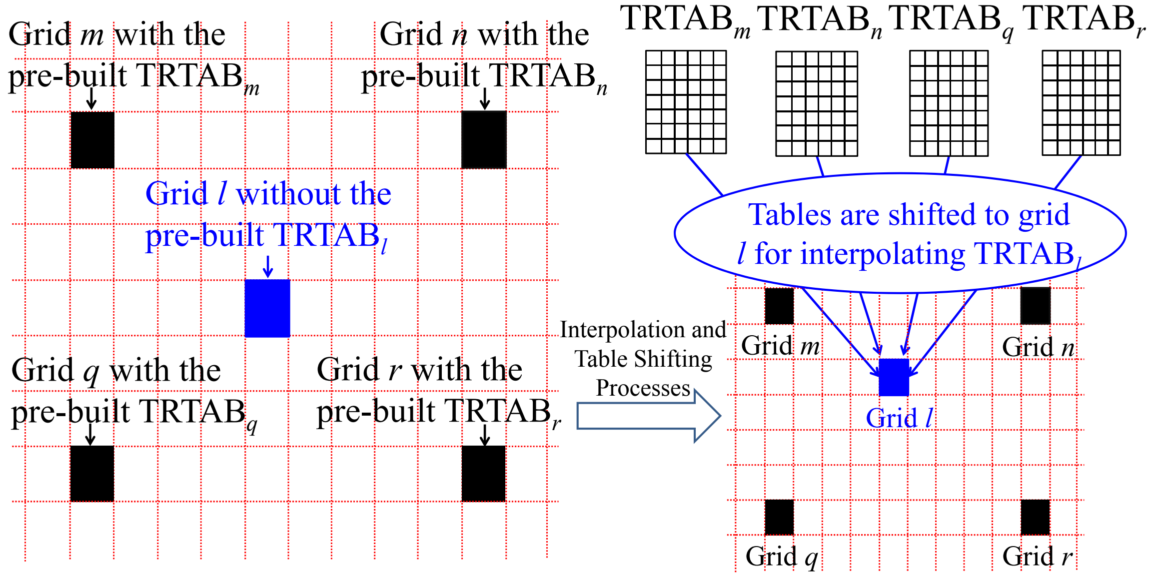


Figure 4.7: The table shifting and interpolation processes for the grid having no pre-built unit power temperature response table.

the pre-built TR-UPS table (TRTAB). Thus, a proposed table shifting and interpolation process is proceeded to obtain the approximated value of each entry of $\mathbf{h}_l$. As shown in Figure 4.7, grids m, n, q and r are four representative grids that are the closest grids to grid l. TRTAB$_m$, TRTAB$_n$, TRTAB$_q$ and TRTAB$_r$ are the tables of $\mathbf{h}_m$, $\mathbf{h}_n$, $\mathbf{h}_q$ and $\mathbf{h}_r$, respectively. By shifting and interpolating these four tables, the TR-UPS table of $\mathbf{h}_l$ can be approximated and represented as TRTAB$_l$. Here, TRTAB$_l$ can be interpolated as [108]

$$\text{TRTAB}_l \approx c_1\text{TRTAB}_m + c_2\text{TRTAB}_n + c_3\text{TRTAB}_q + c_4\text{TRTAB}_r, \tag{4.13}$$

where each $c_i = d_i^{-1} / \sum_{j=1}^{4} d_j^{-1}$ with $d_i = \sqrt{(x^* - x_i)^2 + (y^* - y_i)^2}$ for $1 \leq i \leq 4$. Here, (x^*, y^*), (x_1, y_1), (x_2, y_2), (x_3, y_3) and (x_4, y_4) are the lateral position at the center of grids l, m , n, q and r, respectively.

With the above table shifting and interpolation processes, an approximation of $\mathbf{T}_{h,S_g}$ can be obtained by table lookup. However, the error caused by ignoring the temperature responses outside the truncation window W will be accumulated.

With the above table shifting and interpolation process, $\mathbf{T}_{h,S_g}$ can be approximated by table lookup. However, the error caused by ignoring the temperature response outside the truncation window W will be accumulated. This phenomenon is illustrated in Figure 4.8. Because TRTAB$_i$ for grid i and TRTAB$_j$ for gird j do not cover grids m and n, the errors induced by ignoring the

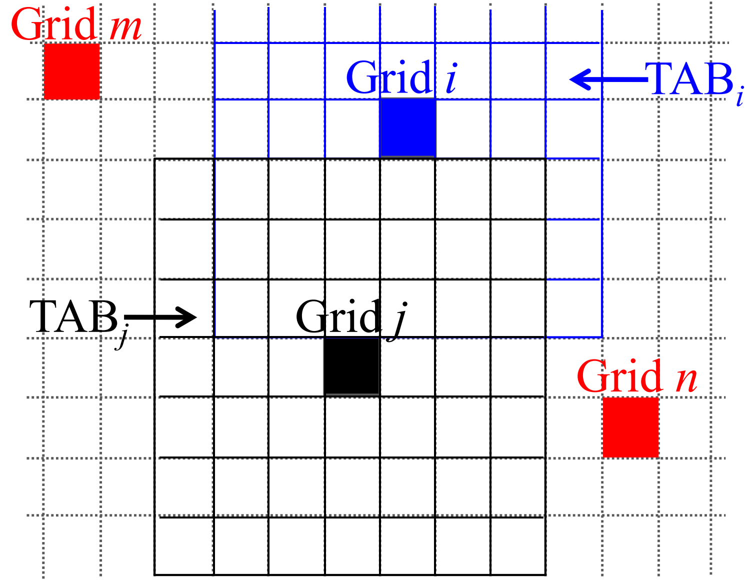


Figure 4.8: The error accumulation phenomenon of the fine-mesh look-up table strategy.

values of $p_i\mathbf{h}_i$ and $p_j\mathbf{h}_j$ outside the window W are accumulated in grids m and n. To alleviate the error, the size of window needs to be increased; nevertheless, the efficiency is degraded. Therefore, in the next subsection, a double-mesh look-up table technique is developed to release the trade-off between the efficiency and the accuracy of the fine-mesh look-up table technique.

4.2.4 Double-Mesh Table Establishment

As shown in Fig. 4.4, the portion of TR-UPS outside the truncation window is very smooth. Therefore, we can construct the temperature response of a coarser mesh (fewer grids) to represent that portion by averaging the temperature values of fine grids in each coarse grid. The establishing process of coarse-mesh table is shown in Fig. 4.9. First, a unit power is inserted to a representative grid in S_g (PT1 of Fig. 4.9), and the detail thermal simulation is performed to generate its TR-UPS (PT2 of Fig. 4.9). After that, the chip is divided into a coarse mesh (PT3 of Fig. 4.9), and those fine grids with the significant portion of temperature response are shown in PT4 of Fig. 4.9. Finally, the temperature response in fine grids shown in PT4 is subtracted from the temperature response in fine grids shown in PT3, and the results of fine grids in each coarse grid are averaged (PT5 of Fig. 4.9) and stored into the coarse-mesh table (PT6 of Fig. 4.9). Meanwhile, the temperature values of coarse mesh (PT5 of Fig. 4.9) in the significant region are subtracted from those of fine grids (PT4 of Fig. 4.9) if they are overlapped. After that, the

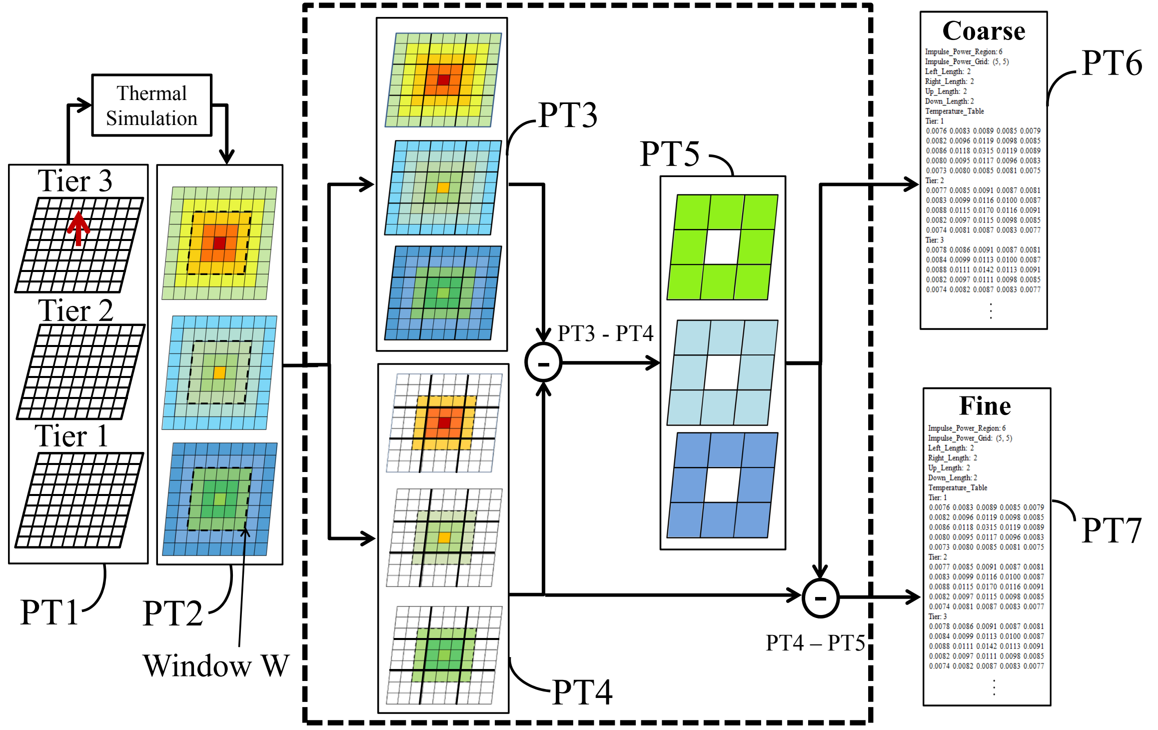


Figure 4.9: The double-mesh table establishment of the unit power temperature response.

results of fine girds in the significant region are stored into the fine-mesh table (PT7 of Fig. 4.9).

With the above fine-mesh and coarse-mesh tables, the double-mesh look-up table calculating process is exhibited in Fig. 4.10. First, the temperature profile of fine mesh, $\mathbf{T}_\mathrm{F}$, is computed by $\mathbf{T}_\mathrm{F} = \sum_{i \in S_g} p_i \times \mathrm{TAB}_{\mathrm{F}_i}$. Here, $\mathrm{TAB}_{\mathrm{F}_i}$ is the fine-mesh temperature response table with a unit power source at the fine grid i. Meanwhile, the temperature profile of coarse mesh, $\mathbf{T}_\mathrm{C}$, is computed by $\mathbf{T}_\mathrm{C} = \sum_{i \in S_g} p_i \times \mathrm{TAB}_{\mathrm{C}_i}$. Here, $\mathrm{TAB}_{\mathrm{C}_i}$ is the coarse-mesh temperature response table induced by a unit power source at the fine grid i. Mapping each entry in $\mathbf{T}_\mathrm{C}$ into its corresponding entries in $\mathbf{T}_\mathrm{F}$, $\mathbf{T}_{h,S_g}$ can be approximated as $\mathbf{T}_\mathrm{F} + \mathbf{T}_\mathrm{C}$.

With the double-mesh look-up table technique, the complexity for solving $\mathbf{T}_{h,S_g}$ is $\mathrm{O}\left((N_W + N_C + 1)N_{S_g}\right)$. Here, N_W and N_C are the sizes of the fine-mesh and the coarse-mesh tables, respectively. N_W is much less than N_{S_g} due to the lateral locality, and N_C can be much less than N_W. The memory usage for storing double-mesh tables is $\mathrm{O}\left((N_W + N_C)N_{R_g}\right)$. Here, N_{R_g} is the number of representative grids and is much less than N_{S_g} due to the local similarity. Moreover, with the double mesh look-up table technique, the complexity for calculating $\mathbf{T}^q_{S_g}$ is $\mathrm{O}\left((N_W + N_C + 1)\,(N_{S_g} + qN_{TSV_g})\right)$. Here, N_{TSV_g} is the number of grids in S_g that have TSVs/TTSVs passing through.

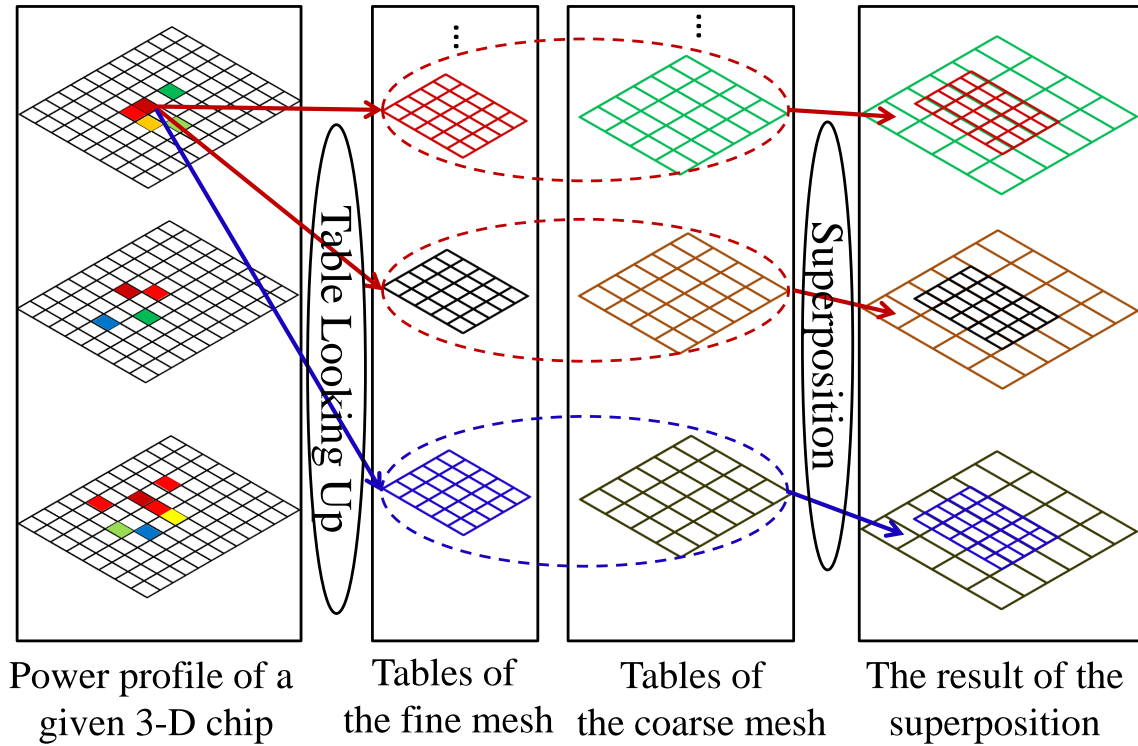


Figure 4.10: The calculating process of double-mesh look-up table technique.

4.3 Experimental Results

4.3.1 Experimental Settings

The developed thermal simulator, LUTSim, is implemented in C++ language and tested on Intel Core 2 Quad 2.83-GHz CPU with 8GB memory. In our experiments, the material and width of TSVs/TTSVs are copper and 45μm [109], and the thermal conductivities (W/(m×°C)) of silicon, copper, and oxide are set to 148, 406 and 0.83 under the room temperature 27°C, respectively.

In the table establishing process, the mesh of equivalnet thermal circuit is set to $64 \times 64 \times 15$, and a fast MNA solver employed in [10] is adopted to build the pre-simulated tables. For building TR-UPS tables, the size of truncation window W is 11×11. The sizes of W_d and W_p are 10×10 and 7×7 for building TDR-TSV tables, respectively. The size of coarse-mesh tables is set to 3×3 for both TR-UPS and TDR-TSV. The number of specific thermal conductivities for building TDR-TSV tables is 3.

4.3.2 Validation

In this section, LUTSim is compared with the commercial tool ANSYS to validate its accuracy. An industrial 3-D IC design with two tiers is tested. It has about 254K and 258K macros/cells in the top and bottom tiers, respectively, and the number of TSVs is 222. The thickness of interconnect layer is $12\mu m$, and its effective thermal conductivity is 243.9W/(m×°C), which is composed of 60% copper and 40% oxide. Under the cooling system and package provided by the industrial manufacturer, the heat transfer coefficients in the primary and secondary heat flow path are 1903.55 ($W/(m^2 \times^o C)$) and 103.204 ($W/(m^2 \times^o C)$), respectively. The heat transfer coefficients in lateral surfaces around the chip are 1112.65 ($W/(m^2 \times^o C)$). Its placement of macros/gates is shown in Fig. 4.11(a), and the geometry of each tier is $4832\mu m \times 4832\mu m \times 50\mu m$. The power profile of each tier is shown in Fig. 4.11(b), and the power consumption of top and bottom tier is $0.49W$ and $0.58W$, respectively. The temperature profiles of top surfaces of tiers estimated by ANSYS and LUTSim are shown in Fig. 4.11(c) and (d), respectively. Comparing with these two figures, the result of LUTSim consists with that of ANSYS. Compared with ANSYS, the error distribution of LUTSim is plotted in Fig. 4.12, and the errors are within the range of $[-0.76\%, 0.56\%]$. Here, the error between the results of ANSYS, T_i^{ANSYS}, and LUTSim, T_i^{LUTSim}, in gird $i \in S_g$ is measured as $e_i = (T_i^{\text{ANSYS}} - T_i^{\text{LUTSim}})/T_i^{\text{ANSYS}}$.

4.3.3 Robustness Verification

Since the two-tier industrial design contains a small amount of TSVs, extra test cases with large amounts of TSVs are generated to further demonstrate the accuracy and efficiency of LUTSim. To generate the power profile of each test chip, millions of gates are randomly picked from the TSMC $90nm$ standard cell library and randomly inserted into each test chip. Moreover, several TSVs are randomly inserted to each test chip. Since the area occupied by TSVs is usually under a threshold decided by designers, it is set to about 10% of chip area [36]. All experimental settings of extra test cases are the same with those of two-tier industrial design. Since ANSYS requires huge execution time for the model building and meshing process, the fast MNA solver employed in [10] is adopted to generate the reference solution.[4] Comparing with ANSYS, its

[4]The fast MNA solver [10] is employed because the TSV modeling of 3-D ICs is not yet explicitly supported by the latest version of HotSpot [110].

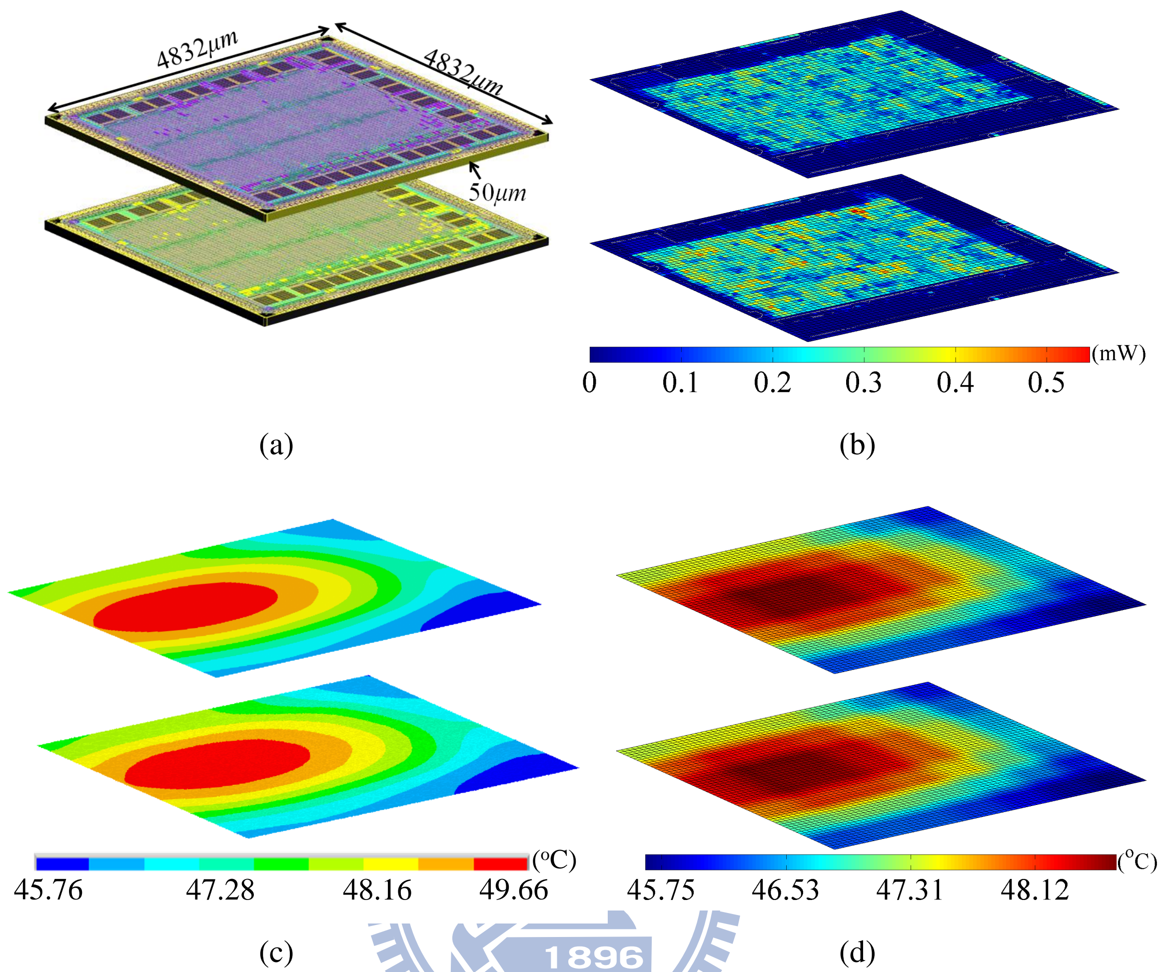


Figure 4.11: Placement, power profiles, estimated temperature profiles of a two-tier industrial chip by ANSYS and R-LUTSim. (a) Placement. (b) Power profiles. (c) Estimated temperature profiles by ANSYS. (d) Estimated temperature profiles by LUTSim.

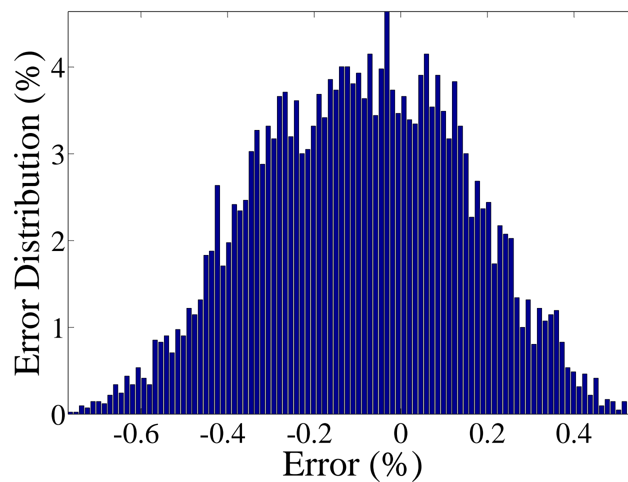


Figure 4.12: Error distribution of LUTSim compared with ANSYS.

Table 4.1: Comparison between LUTSim and the fast MNA solver [10].

Test Chip	Tier Count	Cell Count	TSV Count	Total Power (W)	Truncation Order	Maximum Error (%)	†Runtime (second)		Speedup Ratio
							MNA [10]	LUTSim	
industrial	2	0.5M	222	1.1	2	0.23	61.05	0.06	1000.8
g-chip1	3	8.3M	3320	5.8	2	0.27	256.92	0.41	626.6
g-chip2	3	8.4M	3320	6.3	2	0.26	257.92	0.42	614.8
g-chip3	3	8.2M	4000	6.0	2	0.25	257.65	0.42	613.5

† The runtime does not include the execution time for parsing files.

maximum error is only 1.6% for the two-tier industrial design but execution runtime is fairly less. The insertion numbers of TSVs for a three tiers generated test chip is shown in Figure 4.13.

60	30	260	30
70	110	200	150
40	200	70	150
110	120	100	220

(a) Insertion numbers of TSVs in lateral regions between the tier 3 and tier 2 shown in Figure 4.2.

60	30	260	200
40	200	180	150
180	220	30	150
110	60	100	110

(b) Insertion numbers of TSVs in lateral regions between the tier 2 and tier 1 shown in Figure 4.2.

Figure 4.13: The distribution of insertion numbers of TSVs for the test chip, “g-chip3”, stated in Table 4.1.

The comparison between LUTSim and the fast MNA solver are shown in TABLE 4.1. In TABLE 4.1, “industrial” is the industrial design stated in section 4.3.2, and “g-Chip1”–“g-Chip3” are the generated test chips with different power and TSV profiles. Columns 2–5 are the tier count, cell count, TSV count and the total power consumption of test chips, respectively. “Truncation Order” is the truncation order of R-LUTSim, and “Maximum Error” is the maximum absolute error of R-LUTSim comparing with the fast MNA solver. As shown in columns 7–8, with the maximum error only 0.27% for all test cases, R-LUTSim only requires the approximating order with two for accurately estimating the temperature profile. In summary, the runtime of R-LUTSim is less than 0.42 seconds, and its speedup ratio to the fast MNA solver is over 613.5 for all test chips. The power profile, the estimation temperature profiles of the fast

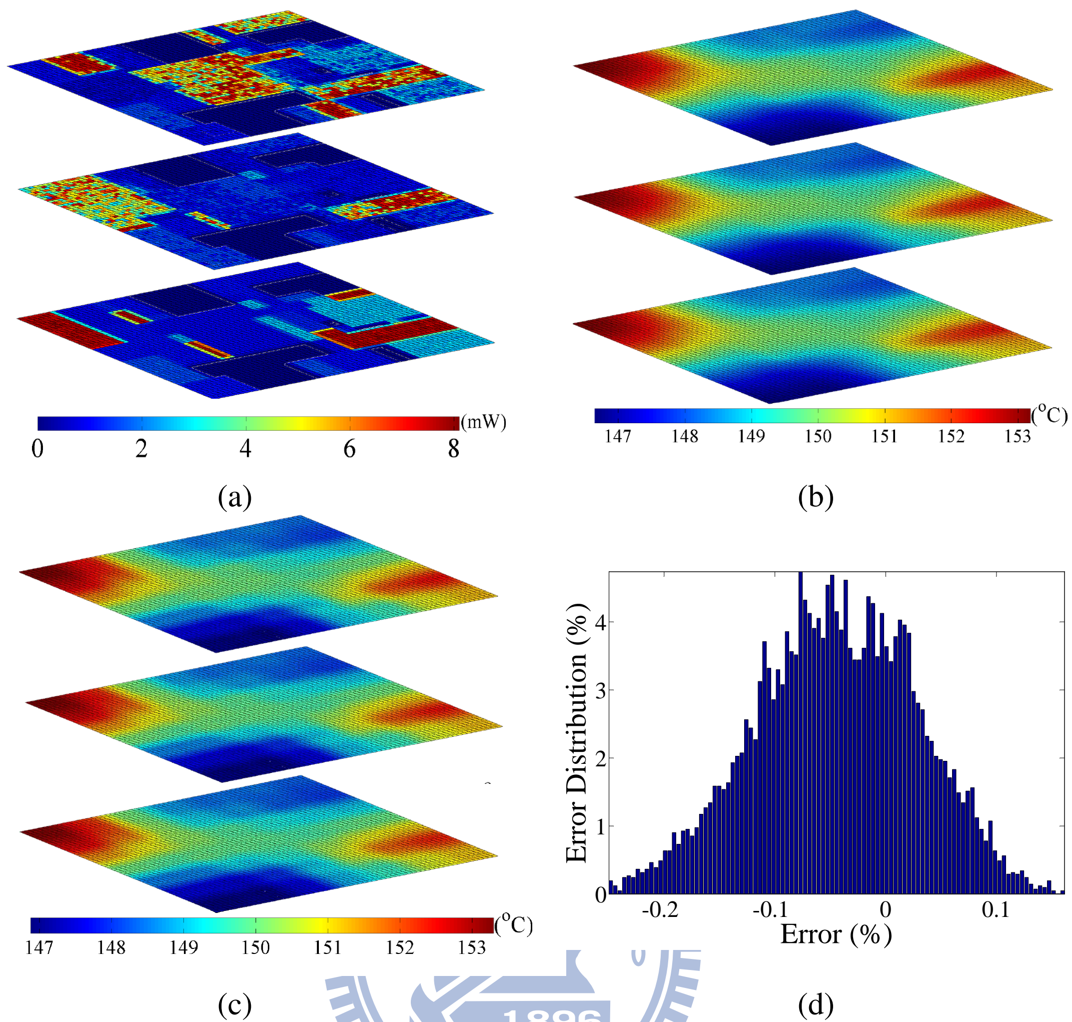


Figure 4.14: Power profiles, estimated temperature profile of fast MNA solver, estimated temperature profile of LUTSim and the error distribution between fast MNA solver and LUTSim of the test chip “g-Chip3”. a two-tier industrial chip by ANSYS and R-LUTSim. (a) Power profiles. (b) Estimated temperature profile of fast MNA solver. (c) Estimated temperature profile of LUTSim (d) Error distribution between fast MNA solver and LUTSim.

MNA solver and LUTSim, and the error distribution between fast MNA solver and LUTSim of the test case “g-Chip3” are shown in Figure 4.14. As shown in Figure 4.14, the result of LUTSim matches that of the fast MNA solver and the errors are in the range $[-0.25\%, 0.17\%]$. The results demonstrate that LUTSim can efficiently provide accurate temperature estimation while the number of TSVs is large.

Chapter 5

Conclusion

5.1 Summary of Current Research Results

In this dissertation, issues of the thermal, power, and process variations in thermal-aware physical design flows of modern VLSI have been investigated. To predict the temperature induced performance and reliability degradations, thermal-aware design engines require accurate and efficient thermal simulators to calculate the their corresponding thermal costs. There, this dissertation contributes three accurate and efficient thermal simulators for early design stages.

Under nominal values of physical parameters, after the positions, geometries and powers of the marcos/cells is given by early stage design engines, the first proposed thermal simulator can efficiently provide accurate temperature profile estimation. In the test results, the first simulator presents accurate estimation comparing with the commercial thermal analysis tool ANSYS. Moreover, comparing with the existing state of the art, Green function based thermal simulator, the first simulator presents an order of magnitude speedup. Under the 1024×1024 thermal simulation mesh, the first proposed thermal simulator can obtain the temperature distribution of a chip with millions gates in 0.13 seconds. Besides thermal simulation of 2-D ICs, the first proposed thermal simulator can accurately and efficiently provide the thermal simulation of the stacked layer or the contactless interconnection 3-D ICs. Comparing with ANSYS, the maximum error of the first proposed thermal simulator is 0.24% for a three tiers stacked layer 3-D IC. The simulation time of the first proposed thermal simulator for the test 3-D IC is only 0.48 seconds.

Under the physical device parameter process variations, the second proposed simulator has took into account process variation inducing fluctuations and the temperature dependence of the

leakage power models for the thermal reliability estimation. Two statistical polynomial expression generators, which have the ability to deal with complex leakage power models for more advance technologies, are developed in the second proposed simulator to efficiently generate the approximating expression of the statistical on-chip temperature distribution. Comparing with the Monte Carlo method, both proposed statistical expression generators can approximate the temperature distribution under 0.93% maximum error for mean estimation and 0.72% maximum error for standard deviation estimation. As the results demonstrating, both proposed statistical expression generators present the 164× speedup over the Monte Carlo method with the simulation time being less than 2.74 seconds. Besides the statistical expression generators, the second proposed simulator provides the thermal yield profile estimation to obtain the probability profile that the temperature distribution being less than or equal to a user specified reference temperature. Comparing with the Monte Carlo method, the developed thermal yield estimator can obtain the thermal yield profile under the maximum error being 1.63%, and the execution is less than 0.013 seconds. Comparing with an existing state-of-the-art, APEX, the developed thermal yield estimator presents the 215× runtime improvement. To overcome the efficiency bottleneck of the baseline algorithm of both proposed statistical expression generators, the second proposed simulator provides a mixed-mesh thermal yield estimation strategy. Under an acceptable accuracy, results have demonstrated that the mixed-mesh strategy can make the efficiency of the thermal yield estimation to be catching up with that of the deterministic thermal simulation. In the test results, the entire flow of mixed-mesh thermal yield profile estimation can be completed in 0.032 seconds with 2.24% maximum error.

To provide the thermal estimation for early stage thermal-aware design engines of the TSV based 3-D ICs, the third proposed simulator provides a look-up table based simulation framework to avoid the time consumed dealing process of the thermal conductance matrix of the SPICE-compatible thermal circuit. Comparing with the commercial thermal analysis tool ANSYS, the third proposed simulator provide accurately thermal simulation for an industrial test chip with the errors are within the range of $[-0.76\%, 0.56\%]$. To further examine the robustness, several test chips are generated and tested. Comparing with a fast MNA solver, under 0.26% maximum absolute error, the third proposed simulator achieves 613.5× runtime improve-

ment.

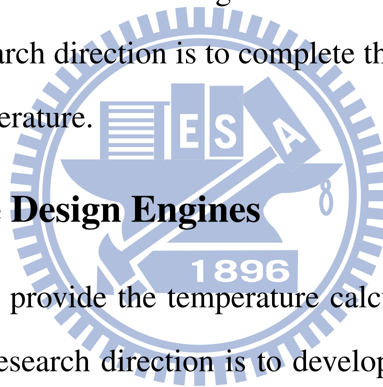
5.2 Future Research Directions

5.2.1 Statistical Thermal Simulation of 3-D ICs

In this dissertation, a statistical thermal simulation framework of 2-D ICs has been proposed. Therefore, a future research direction is to study the variation phenomenon of the physical device parameters, model the variation of the physical device parameters, and develop an extension strategy of the proposed statistical thermal simulation framework for 3-D ICs.

5.2.2 Thermal-aware Timing Analysis

Since delays of gates and wires are temperature-dependent, the on-chip temperature should be fed into the timing analysis tools for calibrating the error terms induced by the temperature variations. Thus, the future research direction is to complete the calibration of the timing variation induced by the on-chip temperature.



5.2.3 Thermal-aware Design Engines

In this dissertation, we have provide the temperature calculation techniques for 2-D and 3-D ICs. Therefore, the future research direction is to develop strategies that can incorporate the proposed thermal simulators into the thermal-aware design engines, such as floorplaner, placer and voltage island generator, of 2-D and 3-D ICs.

Appendix A

Derivation of the time-varying coefficients for GIT based thermal simulation method and error bound analysis of GIT based steady state temperature formulae

A.1 Derivation of the Analytical Expression of Time-Varying Coefficients for the Approximated Temperature

The derivation of the un-coupled first order differential equation (2.20) for each time-varying coefficient $\psi_{ilq}(t)$ is now proceeded. Both sides of equation (2.4) are multiplied by $\phi_{ilq}(\mathbf{r})$ and integrated over the region of die D. After that, we have

$$\int_D \nabla^2 T(\mathbf{r},t)\phi_{ilq}(\mathbf{r})dv = \frac{\sigma}{\kappa}\int_D \frac{\partial T(\mathbf{r},t)}{\partial t}\phi_{ilq}(\mathbf{r})dv - \frac{1}{\kappa}\int_D p(\mathbf{r},t)\phi_{ilq}(\mathbf{r})dv, \tag{A.1}$$

where $\int_D(\cdot)dv = \int_{-L_z}^{0}\int_0^{L_y}\int_0^{L_x}(\cdot)dxdydz$.

Here, the inward and outward flows of equation (A.1) must be balanced to satisfy the law of energy conservation. Therefore, the Divergence theorem [66] is then applied to the left hand side of equation (A.1), and we have

$$\int_D \nabla^2 T(\mathbf{r},t)\phi_{ilq}(\mathbf{r})dv = \int_S \phi_{ilq}(\mathbf{r})\frac{\partial T(\mathbf{r},t)}{\partial n}ds - \int_D \nabla T(\mathbf{r},t)\cdot\nabla\phi_{ilq}(\mathbf{r})dv, \tag{A.2}$$

where $\int_S(\cdot)ds$ is the surface integral of all the boundary surfaces S of die, and $\partial/\partial n$ is the normal derivative on the boundary surfaces in the outward direction.

Then, the Divergence theorem is applied to the second term in the right side of (A.2), and we have

$$\int_D \nabla T(\mathbf{r},t)\cdot\nabla\phi_{ilq}(\mathbf{r})dv = \int_S T(\mathbf{r},t)\frac{\partial\phi_{ilq}(\mathbf{r})}{\partial n}ds - \int_D T(\mathbf{r},t)\nabla^2\phi_{ilq}(\mathbf{r})dv. \tag{A.3}$$

By plugging equation (A.3) into equation (A.2) and the result is plugged into equation (A.2), we have

$$\begin{aligned} &\sigma \int_D \frac{\partial T(\mathbf{r},t)}{\partial t} \phi_{ilq}(\mathbf{r}) dv - \kappa \int_D T(\mathbf{r},t) \nabla^2 \phi_{ilq}(\mathbf{r}) dv \\ &\quad = \kappa \int_S \left[\phi_{ilq}(\mathbf{r}) \frac{\partial T(\mathbf{r},t)}{\partial n} - T(\mathbf{r},t) \frac{\partial \phi_{ilq}(\mathbf{r})}{\partial n} \right] ds + \int_D p(\mathbf{r},t) \phi_{ilq}(\mathbf{r}) dv. \end{aligned} \tag{A.4}$$

By plugging equation (2.9), the expression of $\psi_{ilq}(t)$ (equation (2.19)), and boundary conditions (equations (2.5)–(2.7) and (2.10)–(2.12)), into equation (A.4), we have the un-coupled first order differential equation (2.20) for each time-varying coefficient $\psi_{ilq}(t)$.

A.2 Error Bound Analysis of GIT Based Steady State Temperature Formulation

To proceed the error bound analysis of GIT based steady state temperature formulation stated in section 2.2.3, the following lemma is introduced.

Lemma 1. *The magnitude of each time-varying coefficient $|\psi_{ilq}(\infty)|$ at the steady state is bounded by*

$$|\psi_{ilq}(\infty)| \le \begin{cases} \dfrac{2 j_d P_T}{\lambda_{ilq}^2 \kappa \sqrt{N_{ilq}}} \left(\dfrac{\kappa}{\lambda_{zq}} + \dfrac{h_p}{\lambda_{zq}^2} \right); & i = 0,\ l = 0 \\ \dfrac{4N j_d P_T}{l \lambda_{ilq}^2 \kappa \pi \sqrt{N_{ilq}}} \left(\dfrac{\kappa}{\lambda_{zq}} + \dfrac{h_p}{\lambda_{zq}^2} \right); & i = 0,\ l \neq 0 \\ \dfrac{4M j_d P_T}{i \lambda_{ilq}^2 \kappa \pi \sqrt{N_{ilq}}} \left(\dfrac{\kappa}{\lambda_{zq}} + \dfrac{h_p}{\lambda_{zq}^2} \right); & i \neq 0,\ l = 0 \\ \dfrac{8MN j_d P_T}{i l \lambda_{ilq}^2 \kappa \pi^2 \sqrt{N_{ilq}}} \left(\dfrac{\kappa}{\lambda_{zq}} + \dfrac{h_p}{\lambda_{zq}^2} \right); & i \neq 0,\ l \neq 0, \end{cases} \tag{A.5}$$

where P_T is the total steady power consumption of die and j_d is the junction depth of device.

Proof. Since the time domain waveform of steady power profile can be treated as a step function, the bound shown in **Lemma 1** can be easily proved by plugging equations (2.13) and (2.21) into equation (2.22), setting t to be infinity, and with several manipulations. □

With the above lemma, an error bound of GIT based formulation is given by the following theorem.

Theorem 1. *The absolute error of average steady state temperature for each grid cell (m, n) by using the GIT based formulation with truncation points N_x, N_y, and N_z in x-, y-, and z-directions*

is bounded by

$$\sum_{(i,l,q)\in S_1} \frac{\alpha_1\gamma_q}{i^2l^2\lambda_{ilq}^2} + \sum_{(i,q)\in S_2} \frac{\alpha_2\gamma_q}{i^2\lambda_{i0q}^2} + \sum_{(l,q)\in S_3} \frac{\alpha_3\gamma_q}{l^2\lambda_{0lq}^2} + \sum_{q\in S_4} \frac{\alpha_4\gamma_q}{\lambda_{00q}^2}, \tag{A.6}$$

where

$$\gamma_q = \frac{1}{\lambda_{z_q}^2}\left(\kappa + \frac{h_p}{\lambda_{z_q}}\right)\frac{\kappa\lambda_{z_q}^2 + h_p^2}{\kappa\lambda_{z_q}^2 - h_p h_s},$$

and $S_1 = [1, N_x] \times (N_y, \infty) \times [0, \infty) \cup (N_x, \infty) \times [1, N_y] \times [0, \infty) \cup [1, N_x] \times [1, N_y] \times (N_z, \infty)$, $S_2 = [1, N_y] \times (N_z, \infty) \cup (N_x, \infty) \times [0, \infty)$, $S_3 = [1, N_x] \times (N_z, \infty) \cup (N_y, \infty) \times [0, \infty)$, $S_4 = (N_z, \infty)$, $\alpha_1 = 256M^2N^2 j_d P_T/(\kappa L_x L_y L_z \pi^4)$, $\alpha_2 = 32M^2 j_d P_T/(\kappa L_x L_y L_z \pi^2)$, $\alpha_3 = 32N^2 j_d P_T/(\kappa L_x L_y L_z \pi^2)$, $\alpha_4 = 4 j_d P_T/(\kappa L_x L_y L_z)$.

Proof. As pointed out in [66, 67], equation (2.18) is convergent in mean when truncation points are infinities. Hence, the absolute truncation error is bounded as

$$|\epsilon_{mn}(z, \infty)| \le \sum_{(i,l,q)\notin S} \left|\frac{\psi_{ilq}(\infty)}{\Delta x \Delta y}\int_{n\Delta y}^{(n+1)\Delta y}\int_{m\Delta x}^{(m+1)\Delta x} \phi_{ilq}(\mathbf{r})dxdy\right|, \tag{A.7}$$

where $S = [0, N_x] \times [0, N_y] \times [0, N_z]$. Plugging equation (2.13) into (A.7), utilizing **Lemma 1** and with several manipulations, we can get the error bound (A.6). □

Since the decaying rate of γ_q is dominated by $1/\lambda_{z_q}^2$, the error decaying rate of GIT based steady state temperature formulation stated in section 2.2.3 is dominated by $i^2l^2\lambda_{z_q}((i\pi/L_x)^2 + (l\pi/L_y)^2 + \lambda_{z_q}^2)$. To compare the error bond of GIT based formulation with [5] under same boundary conditions and power source location, the error bound (A.6) can be simplified to

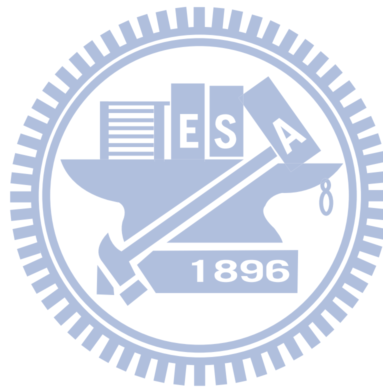
$$\sum_{(i,l,q)\in S_1} \frac{\alpha_1}{i^2l^2\lambda_{ilq}^2} + \sum_{(i,q)\in S_2} \frac{\alpha_2}{i^2\lambda_{i0q}^2} + \sum_{(l,q)\in S_3} \frac{\alpha_3}{l^2\lambda_{0lq}^2} + \sum_{q\in S_4} \frac{\alpha_4}{\lambda_{00q}^2}, \tag{A.8}$$

where $\alpha_1 = 128M^2N^2P_T/(V\kappa\pi^4)$, $\alpha_2 = 16M^2P_T/(\kappa V\pi^2)$, $\alpha_3 = 16N^2P_T/(\kappa V\pi^2)$, $\alpha_4 = 2P_T/(\kappa V)$, $V = L_xL_yL_z$, and the definitions of S_1, S_2, S_3, and S_4 are the same with these in **Theorem 1**. The above result shows that the error decaying rate of our GIT based method can be in the order of $i^2l^2((i\pi/L_x)^2 + (l\pi/L_y)^2 + \lambda_{z_q}^2)$.

On the other hand, the error bound of the Green's function based method shown in [5] can be similarly derived as

$$\sum_{(i,l)\in B_1} \frac{\beta_1}{i^2l^2\gamma_{il}} + \sum_{i\in B_2, l=0} \frac{\beta_2}{i^2\gamma_{il}} + \sum_{i=0, l\in B_3} \frac{\beta_3}{l^2\gamma_{il}}, \tag{A.9}$$

where $\gamma_{il} = \sqrt{(i\pi/L_x)^2 + (l\pi/L_y)^2}$, $B_1 = (N_x, \infty) \times (N_y, \infty)$, $B_2 = (N_x, \infty)$, $B_3 = (N_y, \infty)$, $\beta_1 = 64M^2N^2P_T/(L_xL_y\kappa\pi^4)$, $\beta_2 = 8M^2P_T/(L_xL_y\kappa\pi^2)$, and $\beta_3 = 8N^2P_T/(L_xL_y\kappa\pi^2)$. This bound shows that the error decaying rate of the Green's function based method [5] is in the order of $i^2l^2\sqrt{(i\pi/L_x)^2 + (l\pi/L_y)^2}$.



Appendix B

Derivation of the Projection Coefficients of Subthreshold and Gate Tunneling Leakage Powers

In this section, the calculation algorithms stated in Figure 3.7 and Figure 3.9 are derived. Before going through the derivation, the following preliminary lemma is given.

Lemma 2. *Given two constants, γ and $\lambda < 1/2$, then for a standard normal random variable, x, we have*

$$\mathrm{E}\left\{x^n e^{\gamma x+\lambda x^2}\right\} = \sigma_y e^{\mu_y^2/(2\sigma_y^2)} \mathrm{E}\left\{y^n\right\}, \tag{B.1}$$

where $y \sim N(\mu_y, \sigma_y)$ is a normal random variable with $\mu_y = \gamma/(1-2\lambda)$ and $\sigma_y = 1/\sqrt{(1-2\lambda)}$.

Proof. The result of **Lemma 2** is concluded by re-writing $\gamma x+\lambda x^2$ as $(x-\mu_y)^2/(2\sigma_y^2)+\mu_y^2/(2\sigma_y^2)$, and then substituting it into $\mathrm{E}\left\{x^n e^{\gamma x+\lambda x^2}\right\}$. □

B.1 Derivation of the Evaluating Algorithm for the Projection Coefficient of Gate Tunneling Leakage Power

For the sake of notation simplicity, $\phi(\boldsymbol{\eta}_L) = \boldsymbol{c}_{L_{\underline{m}}}^T \boldsymbol{\eta}_L$, $\psi(\boldsymbol{\eta}_{t_{ox}}) = \boldsymbol{c}_{t_{ox\underline{m}}}^T \boldsymbol{\eta}_{t_{ox}} + a_4 \boldsymbol{\eta}_{t_{ox}}^T \boldsymbol{G}_{t_{ox\underline{m}}} \boldsymbol{\eta}_{t_{ox}}$, $\mathcal{N}_L = \{1, 2, \cdots, N_L\}$, and $\mathcal{N}_{t_{ox}} = \{1, 2, \cdots, N_{t_{ox}}\}$ are employed for the derivation. The objective is to derive the expression for $\mathrm{E}\{\Phi_k(\boldsymbol{\xi}) e^{\phi(\boldsymbol{\eta}_L)} e^{\psi(\boldsymbol{\eta}_{t_{ox}})}\}$ up to the second order HPs.

With $\phi(\boldsymbol{\eta}_L)$, and $\psi(\boldsymbol{\eta}_{t_{ox}})$, the projection equation of gate tunneling leakage power onto H-PCs as shown in equation (3.35) can be rewritten as

$$\mathrm{E}\left\{\Phi_k(\boldsymbol{\xi}) P_{g_{\underline{m}}}(L_{\underline{m}}, t_{ox\underline{m}}, \widehat{T}_{\underline{m}})\right\} = \mu_{P_{g_{\underline{m}}}} \mathrm{E}\left\{\Phi_k(\boldsymbol{\xi}) e^{\phi(\boldsymbol{\eta}_L)} e^{\psi(\boldsymbol{\eta}_{t_{ox}})}\right\}. \tag{B.2}$$

Since $\boldsymbol{\eta}_L$ and $\boldsymbol{\eta}_{t_{ox}}$ are mutually independent standard normal random vectors, we have

$$\mathrm{E}\left\{\Phi_k(\boldsymbol{\xi})e^{\phi(\boldsymbol{\eta}_L)}e^{\psi(\boldsymbol{\eta}_{tox})}\right\} = \mathrm{E}\left\{\Phi_{k_L}(\boldsymbol{\eta}_L)e^{\phi(\boldsymbol{\eta}_L)}\right\}\mathrm{E}\left\{\Phi_{k_{tox}}(\boldsymbol{\eta}_{t_{ox}})e^{\psi(\boldsymbol{\eta}_{tox})}\right\}. \tag{B.3}$$

Here, $\Phi_{k_L}(\boldsymbol{\eta}_L)$ and $\Phi_{k_{tox}}(\boldsymbol{\eta}_{t_{ox}})$ are the H-PCs of $\boldsymbol{\eta}_L$ and $\boldsymbol{\eta}_{t_{ox}}$ up to the second order, respectively. k_L and $k_{t_{ox}}$ are indices for $\Phi_{k_L}(\boldsymbol{\eta}_L)$ and $\Phi_{k_{tox}}(\boldsymbol{\eta}_{t_{ox}})$, respectively; their values are $\{1,\cdots,N_L(N_L-1)/2\}$ and $\{1,\cdots,N_{t_{ox}}(N_{t_{ox}}-1)/2\}$ for $\Phi_{k_L}(\boldsymbol{\eta}_L)$ and $\Phi_{k_{tox}}(\boldsymbol{\eta}_{t_{ox}})$, respectively. $N_L = |\boldsymbol{\eta}_L|$ and $N_{t_{ox}} = |\boldsymbol{\eta}_{t_{ox}}|$.

The computation formulae of $\mathrm{E}\left\{\Phi_{k_L}(\boldsymbol{\eta}_L)e^{\phi(\boldsymbol{\eta}_L)}\right\}$ corresponding to the zeroth, the first and the second orders of $\Phi_{k_L}(\boldsymbol{\eta}_L)$ are derived as follows.

The zeroth order of HPs : $\Phi_{k_L}(\boldsymbol{\eta}_L) = 1$.

Since entries of $\boldsymbol{\eta}_L$ are independent standard normal random variables, according to *Lemma* 2, we have

$$\mathrm{E}\left\{e^{\phi(\boldsymbol{\eta}_L)}\right\} = \prod_{i=1}^{N_L}\mathrm{E}\left\{e^{\boldsymbol{c}_{L_{\underline{m}}}[i]\times\eta_{L_i}}\right\} = e^{\boldsymbol{c}_{L_{\underline{m}}}^T\boldsymbol{c}_{L_{\underline{m}}}/2}. \tag{B.4}$$

Here, each $\boldsymbol{c}_{L_{\underline{m}}}[i]$ is the i-th entry of $\boldsymbol{c}_{L_{\underline{m}}}$.

The first order of HPs : $\Phi_{k_L}(\boldsymbol{\eta}_L) = \eta_{L_i}, i \in \mathcal{N}_L$**.**

Applying *Lemma* 2, we have

$$\mathrm{E}\left\{\eta_{L_i}e^{\phi(\boldsymbol{\eta}_L)}\right\} = \boldsymbol{c}_{L_{\underline{m}}}[i]\mathrm{E}\left\{e^{\phi(\boldsymbol{\eta}_L)}\right\}. \tag{B.5}$$

The second order of HPs : $\Phi_{k_L}(\boldsymbol{\eta}_L) = \eta_{L_i}\eta_{L_j} - \delta_{ij}$**,** $i \in \mathcal{N}_L$ and $j \in \mathcal{N}_L$**.**

Applying *Lemma* 2, we have

$$\mathrm{E}\left\{\left(\eta_{L_i}\eta_{L_j} - \delta_{ij}\right)e^{\phi(\boldsymbol{\eta}_L)}\right\} = \boldsymbol{c}_{L_{\underline{m}}}[i]\boldsymbol{c}_{L_{\underline{m}}}[j]\mathrm{E}\left\{e^{\phi(\boldsymbol{\eta}_L)}\right\}. \tag{B.6}$$

The zeroth order of HPs : $\Phi_{k_{tox}}(\boldsymbol{\eta}_{t_{ox}}) = 1$**.**

Employing the eigen decomposition, $\boldsymbol{G}_{t_{ox\underline{m}}}$ can be written as $\boldsymbol{G}_{t_{ox\underline{m}}} = \boldsymbol{V}_{t_{ox\underline{m}}}^T\boldsymbol{\Lambda}_{t_{ox\underline{m}}}\boldsymbol{V}_{t_{ox\underline{m}}}$. Here, $\boldsymbol{V}_{t_{ox\underline{m}}}$ is the matrix composed of the eigenvectors of $\boldsymbol{G}_{t_{ox\underline{m}}}$, and $\boldsymbol{\Lambda}_{t_{ox\underline{m}}}$ is the diagonal matrix with its i-th diagonal entry, $\boldsymbol{\Lambda}_{t_{ox\underline{m}}}[i][i]$, being the eigenvalue corresponding to the i-th eigenvector of $\boldsymbol{G}_{t_{ox\underline{m}}}$. Plugging the eigen decomposition of $\boldsymbol{G}_{t_{ox\underline{m}}}$ into $\psi(\boldsymbol{\eta}_{t_{ox}})$, setting $\tilde{\boldsymbol{c}} = \boldsymbol{V}_{t_{ox\underline{m}}}\boldsymbol{c}_{t_{ox\underline{m}}}$ and $\boldsymbol{z} = \boldsymbol{V}_{t_{ox\underline{m}}}\boldsymbol{\eta}_{t_{ox}}$,

using the property of independent standard normal random variables for the entries in $\boldsymbol{z}$, and applying **Lemma 2**, we have

$$\begin{aligned} \mathrm{E}\left\{e^{\psi(\boldsymbol{\eta}_{tox})}\right\} &= \mathrm{E}\left\{e^{\tilde{\boldsymbol{c}}\boldsymbol{z}+a_4\boldsymbol{z}^T\boldsymbol{\Lambda}_{tox\underline{m}}\boldsymbol{z}}\right\} \\ &= \prod_{i=1}^{N_{tox}} \mathrm{E}\left\{e^{\tilde{\boldsymbol{c}}[i]\boldsymbol{z}[i]+\boldsymbol{d}[i]\boldsymbol{z}[i]^2}\right\} \\ &= \frac{e^{\boldsymbol{u}^T\boldsymbol{u}/2}}{\prod_{i=1}^{N_{tox}} \boldsymbol{s}[i]}. \end{aligned} \tag{B.7}$$

Here, $\tilde{\boldsymbol{c}}[i]$ is the i-th entry of $\tilde{\boldsymbol{c}}$, $\tilde{\boldsymbol{z}}[i]$ is the i-th entry of $\boldsymbol{z}$, $\boldsymbol{d}$ is a vector with its i-th entry $\boldsymbol{d}[i] = a_4\boldsymbol{\Lambda}_{tox\underline{m}}[i][i]$, $\boldsymbol{s}$ is a vector with its i-th entry $\boldsymbol{s}[i] = \sqrt{1-2\boldsymbol{d}[i]}$, and $\boldsymbol{u}$ is a vector with its i-th entry $\boldsymbol{u}[i] = \tilde{\boldsymbol{c}}[i]/\boldsymbol{s}[i]$.

The first order of HPs : $\Phi_{k_{tox}}(\boldsymbol{\eta}_{tox}) = \eta_{tox_i}, i \in \mathcal{N}_{tox}$.

$$\begin{aligned} \mathrm{E}\left\{\eta_{tox_j} e^{\psi(\boldsymbol{\eta}_{tox})}\right\} &= \mathrm{E}\left\{\left(\boldsymbol{v}_i^T\boldsymbol{z}\right) e^{\tilde{\boldsymbol{c}}\boldsymbol{z}+a_4\boldsymbol{z}^T\boldsymbol{\Lambda}_{tox\underline{m}}\boldsymbol{z}}\right\} \\ &= \boldsymbol{\rho}[i]\mathrm{E}\left\{e^{\tilde{\boldsymbol{c}}\boldsymbol{z}+a_4\boldsymbol{z}^T\boldsymbol{\Lambda}_{tox\underline{m}}\boldsymbol{z}}\right\} \\ &= \boldsymbol{\rho}[i]\mathrm{E}\left\{e^{\psi(\boldsymbol{\eta}_{tox})}\right\}. \end{aligned} \tag{B.8}$$

Here, $\boldsymbol{\rho}$ is a vector with its i-the $\boldsymbol{\rho}[i] = \left(\boldsymbol{v}_i^T\boldsymbol{m}\right)$, $\boldsymbol{v}_i^T$ is the i-th row vector of $\boldsymbol{V}_{tox\underline{m}}^T$, and $\boldsymbol{m}$ is a vector with its j-th entry $\boldsymbol{m}[j] = \tilde{\boldsymbol{c}}[j]/(1-2\boldsymbol{d}[j])$.

The second order of HPs : $\Phi_{k_{tox}}(\boldsymbol{\eta}_{tox}) = \eta_{tox_i}\eta_{tox_j} - \delta_{ij}, i \in \mathcal{N}_{tox}$ and $j \in \mathcal{N}_{tox}$.

Plugging the eigen decomposition of $\boldsymbol{G}_{tox\underline{m}}$ into $\psi(\boldsymbol{\eta}_{tox})$, setting $\boldsymbol{z} = \boldsymbol{V}_{tox\underline{m}}\boldsymbol{\eta}_{tox}$, and applying *Lemma* 2 with several manipulations, we have

$$\begin{aligned} \mathrm{E}\left\{\left(\eta_{tox_i}\eta_{tox_j} - \delta_{ij}\right) e^{\psi(\boldsymbol{\eta}_{tox})}\right\} &= \mathrm{E}\left\{\left(\boldsymbol{z}^T\boldsymbol{v}_i\boldsymbol{v}_j^T\boldsymbol{z} - \delta_{ij}\right) e^{\tilde{\boldsymbol{c}}\boldsymbol{z}+a_4\boldsymbol{z}^T\boldsymbol{\Lambda}_{tox\underline{m}}\boldsymbol{z}}\right\} \\ &= \boldsymbol{\Theta}[i][j]\mathrm{E}\left\{e^{\tilde{\boldsymbol{c}}\boldsymbol{z}+a_4\boldsymbol{z}^T\boldsymbol{\Lambda}_{tox\underline{m}}\boldsymbol{z}}\right\} \\ &= \boldsymbol{\Theta}[i][j]\mathrm{E}\left\{e^{\psi(\boldsymbol{\eta}_{tox})}\right\}. \end{aligned} \tag{B.9}$$

Here, $\boldsymbol{\Theta}$ is a matrix with its (i, j) entry $\boldsymbol{\Theta}[i][j] = \boldsymbol{\rho}[i]\boldsymbol{\rho}[j] + \sum_{l=1}^{N_{tox}} \boldsymbol{w}[l]\boldsymbol{v}_i[l]\boldsymbol{v}_j[l] - \delta_{ij}$. And $\boldsymbol{w}$ is a vector with its l-th entry $\boldsymbol{w}[l] = 1/(1-2\boldsymbol{d}[l])$. $\boldsymbol{v}_i[l]$ and $\boldsymbol{v}_j[l]$ are the l-th entries of $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$, respectively.

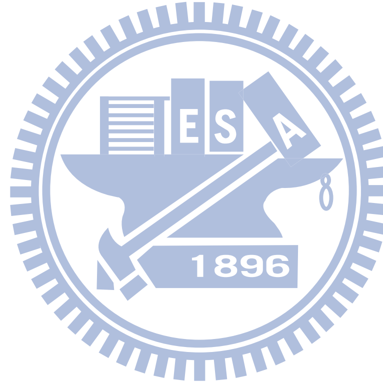
With the equations (B.3)–(B.9), the evaluating algorithm shown in Figure 3.7 is concluded.

B.1.1 Derivation of Evaluating Algorithm for the Projection Coefficient of Subthreshold Leakage Power

Expressing $\psi_L(\boldsymbol{\eta}_L) = \boldsymbol{c}_{L_{\underline{m}}}^T \boldsymbol{\eta}_L + \beta_2 \boldsymbol{\eta}_L^T \boldsymbol{G}_{L_{\underline{m}}} \boldsymbol{\eta}_L$ and $\psi_{t_{ox}}(\boldsymbol{\eta}_{t_{ox}}) = \boldsymbol{c}_{t_{ox\underline{m}}}^T \boldsymbol{\eta}_{t_{ox}} + \beta_4 \boldsymbol{\eta}_{t_{ox}}^T \boldsymbol{G}_{t_{ox\underline{m}}} \boldsymbol{\eta}_{t_{ox}}$ and employing the independent property of $\boldsymbol{\eta}_L$ and $\boldsymbol{\eta}_{t_{ox}}$, we have

$$\mathrm{E}\left\{\Phi_k(\boldsymbol{\xi}) e^{\psi_L(\boldsymbol{\eta}_L)} e^{\psi_{t_{ox}}(\boldsymbol{\eta}_{t_{ox}})}\right\} = \mathrm{E}\left\{\Phi_{k_L}(\boldsymbol{\eta}_L) e^{\psi_L(\boldsymbol{\eta}_L)}\right\} \mathrm{E}\left\{\Phi_{k_{t_{ox}}}(\boldsymbol{\eta}_{t_{ox}}) e^{\psi_{t_{ox}}(\boldsymbol{\eta}_{t_{ox}})}\right\}. \tag{B.10}$$

The computation formulae of $\mathrm{E}\left\{\Phi_{k_L}(\boldsymbol{\eta}_L) e^{\psi_L(\boldsymbol{\eta}_L)}\right\}$ and $\mathrm{E}\left\{\Phi_{k_{t_{ox}}}(\boldsymbol{\eta}_{t_{ox}}) e^{\psi_{t_{ox}}(\boldsymbol{\eta}_{t_{ox}})}\right\}$ can be derived by using the deviation similar with equations (B.7)–(B.9). Consequently, the evaluating algorithm shown in Figure 3.9 is concluded.



Appendix C

Proof of Proposition 1

First, we introduce a lemma for thermal conductance matrices $\mathbf{G} = \mathbf{G}_h + \Delta\mathbf{G}$ and $\mathbf{G}_h$.

Lemma 3. *Both conductance matrices* $\mathbf{G}$ *and* $\mathbf{G}_h$ *are positive definite.*

Proof. Since $\mathbf{G}$ and $\mathbf{G}_h$ are conductance matrices, they are symmetric matrices that satisfy the irreducible diagonal dominant property [111] and have positive real diagonal entries. Hence, $\mathbf{G}$ and $\mathbf{G}_h$ are positive definite [112]. □

Based on **Lemma 3**, we have the following lemma.

Lemma 4. *The absolute value of each eigenvalue of* $\mathbf{G}_h^{-1}\Delta\mathbf{G}$ *is less than* 1.

Proof. Let λ be an eigenvalue of $\mathbf{G}_h^{-1}\Delta\mathbf{G}$, and $\mathbf{y}$ be its corresponding eigenvector. Considering $\mathbf{y}^T(\mathbf{G}_h + \Delta\mathbf{G})\mathbf{y}$, we have

$$\begin{aligned} \mathbf{y}^T(\mathbf{G}_h + \Delta\mathbf{G})\mathbf{y} &= \mathbf{y}^T\mathbf{G}_h(\mathbf{I} + \mathbf{G}_h^{-1}\Delta\mathbf{G})\mathbf{y} \\ &= (1+\lambda)\mathbf{y}^T\mathbf{G}_h\mathbf{y}. \end{aligned} \quad \text{(C.1)}$$

From **Lemma 3**, we have $\mathbf{y}^T(\mathbf{G}_h + \Delta\mathbf{G})\mathbf{y} > 0$ and $\mathbf{y}^T\mathbf{G}_h\mathbf{y} > 0$. Therefore, $\lambda > -1$.

Since a 3-D IC has a nonzero thermal conductivity value at any position, $(\mathbf{G}_h - \Delta\mathbf{G})$ is also a thermal conductance matrix. Applying a similar derivation to $\mathbf{y}^T(\mathbf{G}_h - \Delta\mathbf{G})\mathbf{y} > 0$, we have $\lambda < 1$. Therefore, $|\lambda| < 1$, and **Lemma 4** is concluded. □

To complete the proof of **Proposition 1**, we require the following proposition stated in [113].

Proposition 2. *Let* $\mathbf{A} \in \mathbf{F}^{n \times n}$*, and assume that* $\operatorname{spard}(\mathbf{A}) < 1$*. Then, the series* $\sum_{i=0}^{\infty} \mathbf{A}^i$ *converges absolutely, and*

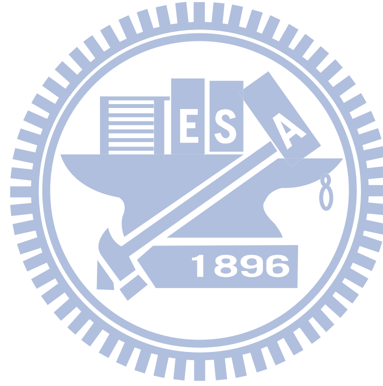
$$(\mathbf{I} - \mathbf{A})^{-1} = \sum_{i=0}^{\infty} \mathbf{A}^i, \tag{C.2}$$

where $\mathbf{F}^{n \times n}$ *is the set of* $n \times n$ *real or complex matrices, and* $\operatorname{spard}(\mathbf{A})$ *is the maximum absolute eigenvalue of* $\mathbf{A}$.

With setting $\mathbf{A} = -\mathbf{G}_h^{-1}\Delta\mathbf{G}$ in equation (C.2) and using **Lemma 4**, we have

$$\mathbf{T} = (\mathbf{G}_h + \Delta\mathbf{G})^{-1}\mathbf{p} = \mathbf{G}_h(\mathbf{I} + \mathbf{G}_h^{-1}\Delta\mathbf{G})\mathbf{p} = \sum_{i=0}^{\infty}(-1)^i(\mathbf{G}_h^{-1}\Delta\mathbf{G})^i\mathbf{G}_h^{-1}\mathbf{p}. \tag{C.3}$$

Consequently, **Proposition 1** is concluded.



Bibliography

[1] K. Banerjee, M. Pedram, and A.H. Ajami. Analysis and optimization of thermal issues in high-performance vlsi. In *Proceedings of the 2001 international symposium on Physical design*, pages 230–237. ACM, 2001.

[2] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De. Parameter variations and impact on circuits and microarchitecture. *Proc. Des. Autom. Conf.*, pages 338–342, June 2003.

[3] W.R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A.M. Sule, M. Steer, and P.D. Franzon. Demystifying 3d ics: the pros and cons of going vertical. *Design & Test of Computers, IEEE*, 22(6):498–510, 2005.

[4] J.A. Burns, B.F. Aull, C.K. Chen, C.L. Chen, C.L. Keast, J.M. Knecht, V. Suntharalingam, K. Warner, P.W. Wyatt, and D.R.W. Yost. A wafer-scale 3-d circuit integration technology. *Electron Devices, IEEE Transactions on*, 53(10):2507–2516, 2006.

[5] Y. Zhan and S.S. Sapatnekar. High-efficiency green function-based thermal simulation algorithms. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 26(9):1661–1675, 2007.

[6] H. Chang and S.S. Sapatnekar. Prediction of leakage power under process uncertainties. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 12(2):12–es, 2007.

[7] R. Shen, S.X.D. Tan, N. Mi, and Y. Cai. Statistical modeling and analysis of chip-level leakage power by spectral stochastic method. *Integration, the VLSI Journal*, 43(1):156–165, 2010.

[8] S.A. Yu, P.Y. Huang, and Y.M. Lee. A multiple supply voltage based power reduction method in 3-d ics considering process variations and thermal effects. In *Proceedings of the 2009 Asia and South Pacific Design Automation Conference*, pages 55–60. IEEE Press, 2009.

[9] J. Jaffari and M. Anis. Statistical thermal profile considering process variations: Analysis and applications. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 27(6):1027–1040, 2008.

[10] T.H. Chen, C. Luk, and C.C.P. Chen. Inductwise: Inductance-wise interconnect simulator and extractor. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 22(7):884–894, 2003.

[11] B.C. Paul, K. Kang, H. Kufluoglu, M.A. Alam, and K. Roy. Impact of nbti on the temporal performance degradation of digital circuits. *Electron Device Letters, IEEE*, 26(8):560–562, 2005.

[12] S. Khan and S. Hamdioui. Temperature dependence of nbti induced delay. In *2010 IEEE 16th International On-Line Testing Symposium*, pages 15–20. IEEE, 2010.

[13] R. Kumar and V. Kursun. Reversed temperature-dependent propagation delay characteristics in nanometer CMOS circuits. *IEEE Trans. Circuits Syst. II, Exp. Briefs*, 53(10):1078–82, Oct. 2006.

[14] S. Bota, M. Rosales, J. Rosello, A. Keshavarzi, and J. Segura. Within die thermal gradient impact on clock-skew: a new type of delay-fault mechanism. In *Proc. IEEE Int. Test Conf.*, pages 1276–83, Oct. 2004.

[15] J.R. Black. Electromigration failure modes in aluminum metallization for semiconductor devices. *Proceedings of the IEEE*, 57(9):1587–1594, 1969.

[16] S. Yang, W. Wolf, N. Vijaykrishnan, Y. Xie, and W. Wang. Accurate stacking effect macro-modeling of leakage power in sub-100nm circuits. 2005.

[17] K. Roy, S. Mukhopadhyay, and h. Mahmoodi-Meima. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer cmos circuits. *Proceedings of the IEEE*, pages 305–327, Feb. 2003.

[18] A. Asenov, S. Kaya, and J.H. Davies. Intrinsic threshold voltage fluctuations in decanano mosfets due to local oxide thickness variations. *Electron Devices, IEEE Transactions on*, 49(1):112–119, 2002.

[19] KM Cao, W.C. Lee, W. Liu, X. Jin, P. Su, SKH Fung, JX An, B. Yu, and C. Hu. Bsim4 gate leakage model including source-drain partition. In *Electron Devices Meeting, 2000. IEDM Technical Digest. International*, pages 815–818. IEEE, 2000.

[20] K.K. Kim, Y.B. Kim, M. Choi, and N. Park. Accurate macro-modeling for leakage current for iddq test. In *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*, pages 1–4. IEEE.

[21] K. Bernstein, C.T. Chuang, R. Joshi, and R. Puri. Design and cad challenges in sub-90nm cmos technologies. In *Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design*, page 129. IEEE Computer Society, 2003.

[22] A. Ono, K. Fukasaku, T. Hirai, S. Koyama, M. Makabe, T. Matsuda, M. Takimoto, Y. Kunimune, N. Ikezawa, Y. Yamada, et al. A 100 nm node cmos technology for practical soc application requirement. In *Electron Devices Meeting, 2001. IEDM Technical Digest. International*, pages 22–5. IEEE, 2001.

[23] Y. Zhang, D. Parikh, K. Sankaranaraynan, K. Skadron, and M. Stan. Hotleakage: A temperature-aware model of subthreshold and gate leakage for architects. Technical Report CS-2003-05, Univ. of Virginia, May 2003.

[24] N. Sung, T. Austin, J.S. Hu, and M. Jane. Leakage current: Moores law meets static power. *IEEE Computer Magazine*, 2003.

[25] X. Zhang. High performance low leakage design using power compiler and multi-Vt libraries. http://www.synopsys.com, Synopsys, SNUG, Europe, Oct. 2003.

[26] A. Vassighi and M. Sachdev. Thermal runaway in integrated circuits. *IEEE Trans. Device Mater. Rel.*, 6(2):300–5, Jun. 2006.

[27] M. J. Moran and H. N. Shapiro. *Fundamentals of Engineering Thermodynamics*. Wiley, 6 edition, May 2007.

[28] N. Magen, A. Kolodny, U. Weiser, and N. Shamir. Interconnect-power dissipation in a microprocessor. In *Proceedings of the 2004 international workshop on System level interconnect prediction*, pages 7–13. ACM, 2004.

[29] A. Rahman, A. Fan, and R. Reif. Comparison of key performance metrics in two-and three-dimensional integrated circuits. In *Interconnect Technology Conference, 2000. Proceedings of the IEEE 2000 International*, pages 18–20. IEEE, 2000.

[30] K. Banerjee, S.J. Souri, P. Kapur, and K.C. Saraswat. 3-d ics: A novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration. *Proceedings of the IEEE*, 89(5):602–633, 2001.

[31] C.C. Liu, I. Ganusov, M. Burtscher, and S. Tiwari. Bridging the processor-memory performance gap with 3d ic technology. *Design & Test of Computers, IEEE*, 22(6):556–564, 2005.

[32] G.L. Loi, B. Agrawal, N. Srivastava, S.C. Lin, T. Sherwood, and K. Banerjee. A thermally-aware performance analysis of vertically integrated (3-d) processor-memory hierarchy. In *Proceedings of the 43rd annual Design Automation Conference*, pages 991–996. ACM, 2006.

[33] H. Hua, C. Mineo, K. Schoenfliess, A. Sule, S. Melamed, R. Jenkal, and W.R. Davis. Exploring compromises among timing, power and temperature in three-dimensional integrated circuits. In *Proceedings of the 43rd annual Design Automation Conference*, pages 997–1002. ACM, 2006.

[34] J. Cong, A. Jagannathan, Y. Ma, G. Reinman, J. Wei, and Y. Zhang. An automated design flow for 3d microarchitecture evaluation. In *Proceedings of the 2006 Asia and South Pacific Design Automation Conference*, pages 384–389. IEEE Press, 2006.

[35] J. Cong, G. Luo, J. Wei, and Y. Zhang. Thermal-aware 3d ic placement via transformation. In *Proceedings of the 2007 Asia and South Pacific Design Automation Conference*, pages 780–785. IEEE Computer Society, 2007.

[36] B. Goplen and S.S. Sapatnekar. Placement of thermal vias in 3-d ics using various thermal objectives. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 25(4):692–709, 2006.

[37] B. Goplen and S. Spatnekar. Placement of 3d ics with thermal and interlayer via considerations. In *Proceedings of the 44th annual Design Automation Conference*, pages 626–631. ACM, 2007.

[38] L. Xiao, S. Sinha, J. Xu, and E.F.Y. Young. Fixed-outline thermal-aware 3d floorplanning. In *Proceedings of the 2010 Asia and South Pacific Design Automation Conference*, pages 561–567. IEEE Press, 2010.

[39] R. Viswanath, V. Wakharkar, A. Watwe, V. Lebonheur, et al. Thermal performance challenges from silicon to systems. 2000.

[40] K. Sankaranarayanan, S. Velusamy, M. Stan, and K. Skadron. A case for thermal-aware floorplanning at the microarchitectural level. *Journal of Instruction-Level Parallelism*, 8(1-16), 2005.

[41] Y. Han and I. Koren. Simulated annealing based temperature aware floorplanning. *Journal of Low Power Electronics*, 3(2):141–155, 2007.

[42] J. L. Tsai, C. C. P. Chen, G. Chen, B. Goplen, H. Qian, Y. Zhan, S. M. Kang, M. D. F. Wong, and S. S. Sapatnekar. Temperature-aware placement for SOCs. *Proc. IEEE*, 94(8):1502–18, Aug. 2006.

[43] M. Cho, S. Ahmedtt, and D.Z. Pan. Taco: temperature aware clock-tree optimization. In *Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, pages 582–587. IEEE Computer Society, 2005.

[44] A. Chakraborty, K. Duraisami, A. Sathanur, P. Sithambaram, L. Benini, A. Macii, E. Macii, and M. Poncino. Dynamic thermal clock skew compensation using tunable delay buffers. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 16(6):639–649, 2008.

[45] W.L. Hung, GM Link, Y. Xie, N. Vijaykrishnan, N. Dhanwad, and J. Conner. Temperature-aware voltage islands architecting in system-on-chip design. 2005.

[46] Y. Cai, B. Liu, Q. Zhou, and X. Hong. A thermal aware floorplanning algorithm supporting voltage islands for low power soc design. *Integrated Circuit and System Design*, pages 257–266, 2005.

[47] A. Gupta, N.D. Dutt, F.J. Kurdahi, K.S. Khouri, and M.S. Abadir. Thermal aware global routing of vlsi chips for enhanced reliability. In *9th International Symposium on Quality Electronic Design*, pages 470–475. IEEE, 2008.

[48] Y.T. Lee, Y.J. Chang, and T.C. Wang. A temperature-aware global router. In *VLSI Design Automation and Test (VLSI-DAT), 2010 International Symposium on*, pages 279–282. IEEE.

[49] Y.K. Cheng, P. Raha, C.C. Teng, E. Rosenbaum, and S.M. Kang. Illiads-t: An electrothermal timing simulator for temperature-sensitive reliability diagnosis of cmos vlsi chips. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 17(8):668–681, 1998.

[50] P. Wilkerson, A. Raman, and M. Turowski. Fast, automated thermal simulation of three-dimensional integrated circuits. In *Thermal and Thermomechanical Phenomena in Electronic Systems, 2004. ITHERM'04. The Ninth Intersociety Conference on*, pages 706–713. IEEE, 2004.

[51] T. Y. Wang and C. C. P. Chen. Thermal-ADI: a linear-time chip-level thermal simulation algorithm based on alternating-direction implicit (ADI) method. *IEEE Trans. Very Large Scale Integr. Syst.*, 11(4):691–70, Aug. 2003.

[52] P. Liu, H. Li, L. Jin, W. Wu, X.D.T. Sheldon, and J. Yang. Fast thermal simulation for runtime temperature tracking and management. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 25(12):2882–2893, 2006.

[53] T.Y. Wang and C.C.P. Chen. Spice-compatible thermal simulation with lumped circuit modeling for thermal reliability analysis based on modeling order reduction. In *Quality Electronic Design, 2004. Proceedings. 5th International Symposium on*, pages 357–362. IEEE, 2004.

[54] P. Li, L. T. Pileggi, M. Asheghi, and R. Chandra. IC thermal simulation and modeling via efficient multigrid-based approaches. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 25(9):319–26, 2006.

[55] Y. Yang, Z. Gu, C. Zhu, R. P. Dick, and Li Shang. ISAC: Integrated space-and-time-adaptive chip-package thermal analysis. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 26(1):86–99, Jan. 2007.

[56] K. Skadron, M.R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan. Temperature-aware microarchitecture: Modeling and implementation. *ACM Transactions on Architecture and Code Optimization (TACO)*, 1(1):94–125, 2004.

[57] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan. HotSpot: A compact thermal modeling methodology for early-stage VLSI design. *IEEE Trans. Very Large Scale Integr. Syst.*, 14(5):501–13, May 2006.

[58] W. Huang, K. Sankaranarayanan, K. Skadron, R.J. Ribando, and M.R. Stan. Accurate, pre-rtl temperature-aware design using a parameterized, geometric thermal model. *IEEE Transactions on Computers*, pages 1277–1288, 2008.

[59] W. Huang, K. Skadron, S. Gurumurthi, R.J. Ribando, and M.R. Stan. Differentiating the roles of ir measurement and simulation for power and temperature-aware design. In *Performance Analysis of Systems and Software, 2009. ISPASS 2009. IEEE International Symposium on*, pages 1–10. IEEE, 2009.

[60] L. Zhang, W. Chen, Y. Hu, J. A. Gubner, and C. C. P. Chen. Correlation-preserved statistical timing with a quadratic form of Gaussian variables. *IEEE Trans. Comput.-Aided Des. Integr. Circuit Syst.*, 25(11):2437–49, Nov. 2006.

[61] H. Chang and S. Sapatankar. Statistical timing analysis under spatial correction. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 24(9):1467–82, Sept. 2005.

[62] J. Cong, J. Wei, and Y. Zhang. A thermal-driven floorplanning algorithm for 3d ics. In *Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, pages 306–313. IEEE Computer Society, 2004.

[63] Z. Li, X. Hong, Q. Zhou, S. Zeng, J. Bian, H. Yang, V. Pitchumani, and C.K. Cheng. Integrating dynamic thermal via planning with 3d floorplanning algorithm. In *Proceedings of the 2006 international symposium on Physical design*, pages 178–185. ACM, 2006.

[64] M.C. Tsai, T.C. Wang, and T.T. Hwang. Signal through-silicon via planning in 3d fixed-outline floorplanning. In *Green Circuits and Systems (ICGCS), 2010 International Conference on*, pages 584–588. IEEE.

[65] M.D. Mikhailov and M.N. Ozisik. *Unified analysis and solutions of heat and mass diffusion*. Dover Publications, 1994.

[66] N.Y. Olcer. On the theory of conductive heat transfer in finite regions. *International Journal of Heat and Mass Transfer*, 7(3):307–314, 1964.

[67] MD Mikhailov. General solutions of the heat equation in finite regions. *International Journal of Engineering Science*, 10(7):577–591, 1972.

[68] J. Parry, H. Rosten, and G.B. Kromann. The development of component-level thermal compact models of a c4/cbga interconnect technology: The motorola powerpc 603 and powerpc 604 risc microprocessors. *Components, Packaging, and Manufacturing Technology, Part A, IEEE Transactions on*, 21(1):104–112, 1998.

[69] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. Numerical recipes in c++. 2002.

[70] X. Lu, P. Tervola, and M. Viljanen. A novel and efficient analytical method for calculation of the transient temperature field in a multi-dimensional composite slab. *Journal of Physics A: Mathematical and General*, 38:8337, 2005.

[71] M. Frigo and SG Johnson. Fftw manual version 3.1–the fastest fourier transform in the west. *Massachusetts: Massachusetts Institute of Technology*, 2004.

[72] W. Liao, L. He, and K. Lepak. Temperature-aware performance and power modeling. *UCLA, Los Angeles, CA, Tech. Rep. UCLA Eng*, pages 04–250, 2004.

[73] F. Lallement, B. Duriez, A. Grouillet, F. Arnaud, B. Tavel, F. Wacquant, P. Stolk, M. Woo, Y. Erokhin, J. Scheuer, et al. Ultra-low cost and high performance 65nm cmos device fabricated with plasma doping. In *VLSI Technology, 2004. Digest of Technical Papers. 2004 Symposium on*, pages 178–179. IEEE, 2004.

[74] S. Mukhopadhyay, A. Raychowdhurym, and K. Roy. Accurate estimation of total leakage current in scaled cmos logic circuits based on compact current modeling. *Proc. Des. Autom. Conf.*, pages 169–174, June 2003.

[75] D. Lee, D. Blaauw, and D. Sylvester. Gate oxide leakage current analysis and reduction for vlsi circuits. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 12(2):155–166, 2004.

[76] R. Shen, S.X.D. Tan, and J. Xiong. A linear algorithm for full-chip statistical leakage power analysis considering weak spatial correlation. In *Proceedings of the 47th Design Automation Conference*, pages 481–486. ACM, 2010.

[77] Y. Liu, R.P. Dick, L. Shang, and H. Yang. Accurate temperature-dependent integrated circuit leakage power estimation is easy. In *Proceedings of the conference on Design, automation and test in Europe*, pages 1526–1531. EDA Consortium, 2007.

[78] V. De and S. Borkar. Technology and design challenges for low power and high performance. In *Proceedings of the 1999 international symposium on Low power electronics and design*, pages 163–168. ACM, 1999.

[79] L. Cheng, P. Gupta, C.J. Spanos, K. Qian, and L. He. Physically justifiable die-level modeling of spatial variation in view of systematic across wafer variability. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 30(3):388–401, 2011.

[80] S. Bhardwaj, S. Vrudhula, P. Ghanta, and Y. Cao. Modeling of intra-die process variations for accurate analysis and optimization of nano-scale circuits. In *Proceedings of the 43rd annual Design Automation Conference*, pages 791–796. ACM, 2006.

[81] B. Cline, K. Chopra, D. Blaauw, and Y. Cao. Analysis and modeling of cd variation for statistical static timing. In *Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, pages 60–66. ACM, 2006.

[82] C. Schwab and R. A. Todor. Karhunen-Loève approximation of random fields by generalized fast multipole methods. *J. of Comput. Phys.*, 217(1):100–22, Sep. 2006.

[83] J. Xiong, V. Zolotov, and L. He. Robust extraction of spatial correlation. *IEEE Trans. Comput.-Aided Des. Integr. Circuit Syst.*, 26(4):619–31, Apr. 2007.

[84] M. Gao, Z. Ye, D. Zeng, Y. Wang, and Z. Yu. Robust spatial correlation extraction with limited sample via l1-norm penalty. In *Proceedings of the 16th Asia and South Pacific Design Automation Conference*, pages 677–682. IEEE Press, 2011.

[85] F. Liu. A general framework for spatial correlation modeling in VLSI design. In *Proc. Des. Autom. Conf.*, pages 817–22, Jun. 2007.

[86] R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: A Spectral Approach, revised edition*. Springer-Verlag, 2003.

[87] D. Zhang and Z. Lu. An efficient, high-order perturbation approach for flow in random porous media via Karhunen-Loève and polynomial expansions. *J. of Comput. Phys.*, 149(2):773–94, Mar. 2004.

1896

[88] S. Reda, R. Cochran, and A. Nowroz. Improved thermal tracking for processors using hard and soft sensor allocation techniques. *Computers, IEEE Transactions on*, (99):1–1, 2011.

[89] X. Zhou, J. Yang, M. Chrobak, and Y. Zhang. Performance-aware thermal management via task scheduling. *ACM Transactions on Architecture and Code Optimization (TACO)*, 7(1):5, 2010.

[90] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer. Statistical timing analysis: From basic principles to state of the art. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 27(4):589–607, 2008.

[91] X. Ye, P. Li, and F.Y. Liu. Exact time-domain second-order adjoint-sensitivity computation for linear circuit analysis and optimization. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 57(1):236–248, 2010.

[92] Z. Feng, P. Li, and Z. Ren. Sice: design-dependent statistical interconnect corner extraction under inter/intra-die variations. *Circuits, Devices & Systems, IET*, 3(5):248–258, 2009.

[93] X. Ye, P. Li, and F. Liu. Practical variation-aware interconnect delay and slew analysis for statistical timing verification. In *Proc. Int. Conf. on Comput.- Aided Des.*, pages 54–9, Nov. 2006.

[94] N. Mi, J. Fan, S.X.D. Tan, Y. Cai, and X. Hong. Statistical analysis of on-chip power delivery networks considering lognormal leakage current variations with spatial correlation. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 55(7):2064–2075, 2008.

[95] P.Y. Huan, J.H. Wu, and Y.M. Lee. Stochastic thermal simulation considering spatial correlated within-die process variations. In *Proceedings of the 2009 Asia and South Pacific Design Automation Conference*, pages 31–36. IEEE Press, 2009.

[96] S. A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR*, pages 240–243, 1963.

[97] V. Barthelmann, E. Novak, and K. Ritter. High dimensional polynomial interpolation on sparse grids. *Advan. Comput. Math.*, 12(4):273–88, Mar. 2000.

[98] G. M. Phillips. *Interpolation and Approximation by Polynomial*. Springer-Verlag, 2003.

[99] X. Li, J. Le, P. Gopalakrishnan, and L. T. Pileggi. Asymptotic probability extraction for non-normal distributions of circuit performance. In *Proc. Int. Conf. on Comput.- Aided Des.*, pages 2–9, Nov. 2004.

[100] B. Tutuianu, F. Dartu, and L. Pileggi. An explicit RC-circuit delay approximation based on the first three moments of the impulse response. In *Proc. Des. Autom. Conf.*, pages 611–6, Jun. 1996.

[101] A. Azzalini. The skew-normal distribution and related multivariate families. *Board of the Foundation of the Scandinavian Journal of Statistics*, 32:159–88, Jun. 2005.

[102] J. Bienacel, D. Barge, M. Bidaud, N. Emonet, D. Roy, L. Vishnubhotla, I. Pouilloux, and K. Barla. Anticipation of nitrided oxides electrical thickness based on XPS measurement. *Materials Science in Semiconductor Processing*, 7(4–6):181–3, 2004.

[103] M. Healy, M. Vittes, M. Ekpanyapong, C. S. Ballapuram, S. K. Lim, H. H. S. Lee, and G. H. Loh. Multiobjective microarchitectural floorplanning for 2-d and 3-d ICs. *IEEE Trans. Comput.Aided Des. Integr. Circuits Syst.ems*, 26(1):38–52, 2007.

[104] A. B. Kahng. Classical floorplanning harmful? In *Proc. Int. symp. Phys. Des.*, pages 207–13, 2000.

[105] S.N. Adya and I.L. Markov. Fixed-outline floorplanning: Enabling hierarchical design. *IEEE Trans Very Large Scale Integ. Syst.*, 11(6):1120–35, 2003.

[106] J. Z. Yan and C. Chu. DeFer: deferred decision making enabled fixed-outline floorplanning algorithm. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 29(3):367–81, 2010.

[107] G. Luo J. Cong and Y. Shi. Thermal-aware cell and through-silicon-via co-placement for 3D ICs. In *Des. Autom. Conf.*, pages 670–9, 2011.

[108] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proc. ACM National Conference*, pages 517–24, 1968.

[109] C. T. Lin, D. M. Kwai, Y. F. Chou, T. S. Chen, and W. C. Wu. CAD reference flow for 3D via-last integrated circuits. In *Proc. Asia and South Pacific Des. Autom. Conf.*, pages 187–192, 2010.

[110] HotSpot 5.0, http://lava.cs.virginia.edu/HotSpot/HotSpot-HOWTO.htm.

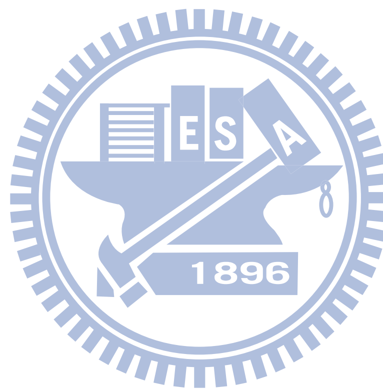
[111] H. Qian and S.S. Sapatnekar. Stochastic preconditioning for diagonally dominant matrices. *SIAM Journal on Scientific Computing*, 30(3):1178–1204, 2008.

[112] R.S. Varga. Matrix iterative analysis, 2000.

[113] D. S. Bernstein. *Matrix mathematics: theory, facts, and formulas with application to linear systems theory*. Princeton University Press, 2005.

Biography

Pei-Yu Huang received the B.S. degree in electrical engineering from the National Taiwan University of Science and Technology, Taiwan, in 2004. From 2004, he is pursuing the Ph.D. degree in the Department of Communication Engineering. His research interests include computer-aided design of integrated circuits, thermal analysis, thermal optimization technique, and power grid analysis.



Publication List

Journal:

[1] Pei-Yu Huang and Yu-Min Lee, "Full-chip thermal analysis for the early design stage via generalized integral transforms", IEEE Transactions on Very Large Scale Integration Systems, vol. 17, no. 5, pp. 613–626, 2009.

[2] Pei-Yu Huang and Yu-Min Lee, "Hierarchical Power Delivery Network Analysis via Bipartite Markov Chains", International Journal of Electrical Engineering (IJEE), vol. 16, no. 2, pp. 121-132, 2009.

International Conference:

[1] Pei-Yu Huang, Chi-Wen Pan and Yu-Min Lee, "On-Chip Statistical Hot-Spot Estimation Using Mixed-Mesh Statistical Polynomial Expression Generating and Skew-Normal Based Moment Matching", Accepted by Asia South Pacific Design Automation Conference (ASPDAC), 2012.

[2] Huai-Chung Chang, Pei-Yu Huang, Ting-Jung Li, and Yu-Min Lee, "Statistical Electro-Thermal Analysis with High Compatibility of Leakage Power Models", International SoC Conference (SOCC), pp. 139-144, Sept. 2010.

[3] Pei-Yu Huang, Jia-Hong Wu and Yu-Min Lee, "Stochastic Thermal Simulation Considering Spatial Correlated Within-Die Process Variations", Asia South Pacific Design Automation Conference (ASPDAC), pp. 31-36, 2009.

[4] Shih-An Yu, Pei-Yu Huang and Yu-Min Lee, "A Multiple Supply Voltage Based Power Reduction Method In 3-D ICs Considering Process Variations And Thermal Effects", Asia South Pacific Design Automation Conference (ASPDAC), pp. 55-60, 2009.

[5] Pei-Yu Huang, Chih-Kang Lin, and Yu-Min Lee, "Full-Chip Thermal Analysis for the Early Design Stage via Generalized Integral Transforms", Asia South Pacific Design Automation Conference (ASP-DAC), pp. 462-467, 2008.

[6] Pei-Yu Huang, Chih-Kang Lin, and Yu-Min Lee, "Hierarchical Power Delivery Network Analysis Using Markov Chains", International SoC Conference (SOCC), pp.

283-286, 2007.

[7] Pei-Yu Huang, Huan-Yu Chou and Yu-Min Lee, “An Aggregation-based Algebraic Multigrid Method for Power Grid Analysis”, International Symposium on Quality Electronic Design (ISQED), pp. 159-164, 2007.

[8] Pei-Yu Huang,Yu-Min Lee, Jeng-Liang Tsai, and Charlie Chung-Ping Chen, “Simultaneous area minimization and decaps insertion for power delivery network using adjoint sensitivity analysis with IEKS method”, International Symposium on Circuits and Systems (ISCAS) , pp. 1291-1294, 2006

Domestic Conference:

[1] Pei-Yu Huang, Chi-Wen Pan and Yu-Min Lee, "Thermal Analysis for Early Physical Design Stages of 3-D ICs using Look-Up Table Techniques", VLSI Design/CAD Symposium, 2011.

[2] Pei-Yu Huang and Yu-Min Lee, "Statistical Hot-Spot Identification Using On-Chip Thermal Yield Profile", VLSI Design/CAD Symposium, 2011.

[3] Pei-Yu Huang, Jia-Hong Wu and Yu-Min Lee, and Huai-Chung Chang,“Stochastic Thermal Simulation Considering With-in Die Process Variations”, VLSI Design/CAD Symposium, 2008.

[4] Shih-An Yu, Pei-Yu Huang and Yu-Min Lee,“Power Optimization in 3D ICs Considering Process Variations and Thermal Effect” , VLSI Design/CAD Symposium, 2008.

International Workshop:

[1] Pei-Yu Huang, Chih-Kang Lin, and Yu-Min Lee, “Full-Chip Thermal Analysis via Generalized Integral Transforms”, Workshop on Synthesis And System Integration of Mixed Information Technologies (SASIMI), 2007.

[2] Yih-Lang Lin, Pei-Yu Huang, Chih-Hong Hwang, and Yu-Min Lee, “Performance- and congestion-driven multilevel router”, Workshop on Synthesis And System Integration of Mixed Information Technologies (SASIMI), 2006.

[3] Pei-Yu Huang, Chih-Hong Hwang, Po-Han Lai, and Yu-Min Lee, “Hierarchical Power Deliver Network Analysis Via Bipartite Markov Chain”, Workshop on Synthesis And System Integration of Mixed Information Technologies (SASIMI),

2006.

[4] Cheng-Hsuan Chiu, Yu-Chan Chang, Pei-Yu Huang, Chih-Hong Hwang, Yu-Min Lee, "Crosstalk-Driven Placement with Considering On-Chip Mutual Inductance and RLC Noise", Workshop on Synthesis And System Integration of Mixed Information Technologies (SASIMI), 2006.