UNIVERSITY OF CALIFORNIA
RIVERSIDE

System-Level Thermal Modeling and Management for Multi-Core and 3D Microprocessors

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Zao Liu

June 2014

Dissertation Committee:

Professor Sheldon X.-D. Tan, Chairperson
Professor Yingbo Hua
Professor Kambiz Vafai

The Dissertation of Zao Liu is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I owe deep gratitute to my adviser Dr. Sheldon X.-D. Tan for his guidance, understanding and most importantly his invaluable advices that help and encourage me to move forward in the right directions in my dissertation research. His mentorship provides me invaluable experience that helps me to set my long-term career goals. I sincerely appreciate his kindness to me as the most important research mentor in my life. I would also like to thank all the committee members Prof. Yingbo Hua and Prof. Kambiz Vafai for their beneficial direction, discussion and advices for improving my research.

Also, I would like to extend my special thanks to all my lab members, Duo Li, Ruijing Shen, Hai Wang, Xuexin Liu, Sahana Swarup, Eric Mullinar, Santiago, Rodriguez, Adair Palm, Xin Huang, Kai He for all kinds of supports and help, which not necessarily limits to research projects. A good team environment is indispensible to overcome challenges and make good progress in research.

Last but not least, I would like to thank my parents, my related family branches across US and China, my friends all over the world, for their constant support and encouragement, which always inspires me to move forward. Also, I would like to express my special thanks to my loved one, Ying Zheng, for her deep understanding of my academic and career pursuit. Closely bonded relationship with them is an invaluable resource for my every success.

ABSTRACT OF THE DISSERTATION

System-Level Thermal Modeling and Management for Multi-Core and 3D Microprocessors

by

Zao Liu

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, June 2014
Professor Sheldon X.-D. Tan, Chairperson

The continuously scaling down of CMOS technology inevitably increases the power density for high performance microprocessors, which makes thermal effects and related problems urgent and challenging. Unpredicted thermal behavior and on-chip thermal hot spots could lead to performance degradation of microprocessor chips, incurring reliability issues. Hence, it is becoming increasingly important to develop thermal modeling methods to predict the thermal behavior of microprocessor chips, and thermal management techniques to control the on-chip thermal hot spots and thermal related reliability issues.

This research focuses on system level thermal behavior modeling and management methods to enable the thermal aware design of high performance microprocessor chips and packages, which also considers the thermal related reliability issues. First, at chip level, a new compact lateral thermal resistance model is proposed to model the lateral thermal behavior of through silicon vias (TSVs), which was largely ignored previously. The proposed lateral thermal model is fully compatible with the existing thermal modeling method of TSVs, and could be integrated into finite difference (FD) simulation to improve the accuracy. Second, targeting at the package level modeling of thermal behavior for microprocessor chip package, the top-down approach building the thermal behavioral models from the given accurate temperature and power information by means of the subspace identification method (SID) is systematically explored in this dissertation research. Power map based approach and piecewise linear modeling method are developed to improve the accuracy of the identified model in presence of thermal nonlinearity and correlated power traces. Third, a more effective architecture level distributed thermal management method is developed to balance the on-chip temperature distribution in this dissertation research. A new temperature metric called *effective initial temperature* that incorporates both initial temperature and other transient thermal effects is proposed to make optimized task migration decisions in the new distributed thermal management method, leading to more effective reduction of thermal hot spots. The last but not least, since temperature imposes exponential influence on the reliability of the chip, this research also proposes a new system level reliability model derived from fundamental physics principles, in which reliability is modeled as life time resources that are to be consumed as the chip works. Based on this model, a dynamic management method is proposed to effectively balance and compensate the life time resources across all the cores in a multi-core processor system, preventing the chance of early failure of heavily loaded cores.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter gives an overview of thermal related modeling and management in today's multi-core microprocessor chip. The thermal and its related issues are discussed to explain the motivation of the researches in this field. Also, the existing works in these fields are reviewed, and the potential challenges are identified. The object and the organization of the dissertation are also given in this chapter.

1.1 Thermal Related Issues for High Performance Microprocessors

The continuously technology scaling down offers benefits of large scale integration and high clock rate for processor designs. Billions of transistors are integrated in a single processor chip using 14 nm silicon fabrication technologies. Due to the complexity of today's sophisticated chip, huge amount of design efforts are put to guarantee that the processor chip achieves varieties of specifications in different aspects. Various of design tools are used in the design flow to help engineers to analyze and understand the intricacies of the systems.

The design tools for digital design aims at simulation and optimization of logical design. Various of digital design tools are specialized for different design stage to evaluate and optimize the logic design. Successful examples that are extensively used by semiconductor industry are VCS simulator, Synopsis design compiler, PrimeTime, et. al. Also, the design tools for analog design focus on analyzing electrical behavior of the circuits based on physical circuit models or empirical data from lab characterization. Different tools are developed to model and simulate the analog system in different abstract levels (system level, transistor level, physical level). However, the design tools used for thermal behavior analysis are largely under-developed. Nevertheless, the increased power density has caused the on-chip temperature to become a major concern for high performance microprocessor design and package as more devices are integrated on the processor chips that aim at working clock frequency of Gigahertz range. Exacerbated temperature would result in slow transistors, increased interconnect delays, and potentially increases power consumptions due to leakage current, degrading the performance of the chip and incurring reliability issues [1, 2].

Hence, thermal management and related design problems continue to be identified by the Semiconductor Industries Association Roadmap [3] as one of the five key challenges during the next decade for achieving the projected performance goals of the industry. Accurate and efficient system level thermal modeling and analysis is vital for the thermal-aware chip and package design to improve performance, reliability, power reduction, and online temperature regulation techniques [4–6]. As the thermal effect becomes top constraint for the performance of high speed processor chip, more efficient thermal modeling and analysis technology at different levels (chip level, package level, and board level) need to be developed for thermal-aware chip and package design, and more efficient architecture level thermal management techniques are also required to effectively control the on-chip temperature, and thermal related reliability issues that are becoming increasingly challenge also needs to be addressed. Thus, to effectively enable thermal-aware chip and package design and alleviate thermal and related issues of the processor chip, this dissertation focuses on:

- chip level thermal modeling methodology using FDM (Finite Difference Method) that considers TSV thermal behavior for 3D stacked IC chips.
- package level top down thermal modeling methodology using subspace identification to extract thermal behavior model of a processor package.
- architecture level thermal management techniques that efficiently control and regulate the on-chip temperature to alleviate thermal issues for multi-core processors.
- system level reliability modeling and management method that effectively control the life time resources of different cores in multi-core processors to prevent the chance of early failure of the cores.

1.2 Finite Difference Thermal Modeling Method for 3D IC Chips

Thermal issues tend to be more exacerbated in three dimensional (3D) stacked integrated circuits (ICs), which necessitates the chip level thermal behavior model to predict thermal hot spots. The major factors driving the thermal issues in 3D ICs are increased thermal resistance along the primary heat transfer path and high power density in 3D integration. Through silicon vias (TSVs) or thermal TSVs (TTSVs) are introduced to facilitate the heat removal inside 3D chips. Essentially, TSVs are vertical vias in 3D ICs, which can effectively convey heat from multiple layers to the heat sink. TSVs have shown promise in alleviating the thermal problem seen in 3D stacked ICs [7, 8]. For thermal modeling of 3D chip structures, it is important to incorporate TSVs into the thermal model because it can significantly change the thermal profile of the 3D chip and TSV models are important for many thermal-aware physical optimizations [9–14].

Traditional approaches treat TSVs as a vertical lumped thermal resistor in each physical plane and its resistance value is proportional to the length and inversely proportional to the diameter of the TTSV [15, 16]. The TSV as a one dimensional (1D) network implies heat flows only in the vertical direction towards the heat sink of the system. This method is shown to be insufficient in capturing the thermal behavior of the TSVs since the lateral heat transfer through these structures is neglected. Recent study shows that the lateral thermal effects due to TSVs or TSV array or farms can have a significant impact on the thermal profile of the 3D chips [17]. Alternately, accurate numerical approaches such as finite difference (FD) and finite element methods (FEM) can be applied to build the thermal model of the chip package. However, to capture the small feature size of TSVs such as the diameter of TSV and thickness of the insulation layer of TSVs, very fine mesh grids are required, which significantly increase the model complexity and thus the cost of simulation.

For finite difference based analysis especially in system level or full chip levels, large grid sizes may be used in order to reduce the computation cost of thermal analysis of 3D stacked chips with TSVs. In this case, we may have one or more TSVs in one grid and it becomes important to drive the equivalent lumped models (in both vertical and lateral directions) for such TSV-bearing grids. In addition to fast finite difference analysis, a very compact and accurate TSV model which offers insight about the thermal properties of TSV structures and links the heat transfer process with the physical parameters will be very useful for architecture level TSV planning and inter-layer level thermal design for 3D ICs.

Recently, a compact thermal TSV model for 3D stacked ICs was proposed in [18], in which the lateral thermal resistance is considered. It is an initial effort to address this important thermal modeling problem. This method, however, mainly studied the stacked thermal TSVs across many active layers. It only considered the lateral thermal resistance caused by the insulation liners. Furthermore, their thermal model works only for a single TSV, not for an array of TSVs. As we can see that the array of TSVs cannot be simply constructed by this individual TSV models. To mitigate the model accuracy issues, a larger model (called distributed model) is used in [18]. This in turn will impact the analysis efficiency.

Chapter 3 of this dissertation explores a new compact physics-based lateral thermal resistance model for both a single TSV and a TSV array. It first focuses on developing a physics-based analytical expression for the lateral thermal resistance of TSVs and TSV arrays in terms of physical and material parameters. The proposed model targets at estimating the lateral thermal resistance considering TSV arrays, and the model also takes into the account of lateral resistance changes caused by the bending of isothermal curves due to the interplays of TSVs. Second, the method also integrates this lumped lateral model into finite difference thermal modeling code of 3D microprocessor package, and demonstrates the improved modeling accuracy by comparing the simulated results against the results from COMSOL, and the one from the existing method [18].

1.3 System Identification based Top-down Thermal Modeling Method for Microprocessor Packages

The traditional bottom-up approaches including FEM (finite element method), FDM (finite difference method), and computational flow dynamics (CFD) based methods were widely used in thermal modeling and analysis in the past [19–21]. For compact thermal modeling at package level, many existing approaches try to use thermal resistors and capacitors with fixed topology networks subject to different thermal boundary conditions [22–24]. However, the accurate RC values of package elements, especially for complex geometries and boundary conditions of the processor package, are difficult to determine, and the calibration against the numerical field solvers or analytical results [25, 26] and measured data are usually required [27].

For thermal modeling at package levels, many works have been proposed targeting at different applications in the past. An excellent survey on recent works can be found in [28]. In general, existing approaches can be classified into bottom-up white-box approaches, which are based on the thermal dynamic physics and approximation techniques, the top-down black-box behavioral modeling approaches and the third modeling approaches, which are something in between the two approaches [28].

Existing work on HotSpot [6,29] attempts to solve this problem by generating the compact thermal model in a bottom-up manner based on processor and package structures. However, such white-box models may suffer from accuracy issues for complicated structures and boundary conditions, which are not properly modeled in the starting models. For instance, complicated package design may require exploration of packages with different structures and materials and boundary conditions for their thermal performance in the industry setting.

Recently, top-down black-box behavioral thermal modeling methods have been proposed to extract thermal behavior model from high level abstraction. Many proposed models target at the on-line temperature regulation applications, in which compact thermal models can estimate or predict the thermal behaviors of a real systems and they can be built in a dynamic way based on the online thermal sensor reading and online power estimation techniques [30–32]. For example, a distributed online thermal model was proposed and validated on a realistic many-core system [33–35].

Another important development for the top-down black-box thermal modeling is for building more accurate and even parameterized thermal models for architecture or package level thermal-aware design and optimization. The input power signal and output thermal temperatures for learning or training are assumed to be measured from off-line complicated equipments in the lab. Existing works consist of the matrix pencil method [36] and the subspace identification method [37–39]. The major advantage of such pure behavioral modeling methods is their flexibility and easy to use as no physical restrictions and assumptions are made or required for the models. They are also very accurate as the training is based on measured data. However, stability and other properties of thermal systems need to be enforced explicitly.

Recently, Beneventi et al. [28] proposed a hybrid (or *gray-box*) identification method, in which a pre-structured compact model under physical constraints is built via an optimization approach. The main advantage of such models is that many physical properties such as stability and passivity can be satisfied automatically. But such models will be less flexible for different architectures and structures as the thermal models or topologies are based on specific architectures. Also, all the existing thermal behavioral modeling methods assume that the thermal systems are linear, which may not be the case for many practical thermal systems as we show in this work, and they have difficulty to deal with varying thermal boundary conditions.

In this dissertation, we still focus on the black-box based thermal modeling scheme based on the subspace identification method. We consider practical measured power maps, which can be obtained from thermal lab based on the test thermal vehicle (testing thermal chips). We first observe that the subspace identification method may suffer from the lack of predictability problems in general [40, 41], especially when the input power is given as series of 2-dimensional power distributions (called *power maps*) in which the input signal is highly spatially correlated. Power map-based thermal characterization is widely used in industry for thermal characterizations of package design as power maps can be easily obtained (measured or computed) practically. However, the spatially correlated power signals in the power map make the system identification process more difficult. The reason is that it is difficult to distinguish the contribution from specific input when all the inputs have the same or similar transient waveforms. Also, compact behavioral thermal modeling for changing boundary conditions still remain to be a challenging problem.

Chapter 4 of this dissertation presents a new behavioral thermal modeling technique considering more practical power maps and nonlinear effects of thermal systems for package level design space exploration of high-performance microprocessors, using subspace identification method. In the proposed new method, the thermal system model is trained by more practical power maps to improve the predictability. Also, piecewise linear techniques are used to handle the thermal nonlinearity of the processor package to improve the model accuracy without increasing the model order, which leads to a compact and accurate thermal model for package level fast thermal simulation and analysis.

1.4 Architectural Level Thermal Management Techniques

Multi-core and upcoming many-core architectures are the trend for current and future microprocessor designs. With the scaling of CMOS devices, the chip multiprocessor is progressing from the *multi-core* era to *many-core* era [42–45] where thousands of cores are predicted to be integrated on a single chip connected by a network-on-chip (NoC) in the near future. However, the increasing chip complexity and power envelope elevate the peak temperature of chips and increase the on-chip thermal gradients. Thermal constraints are the major driving force for wide adoption of multi/many core architectures. As the power density continues to increase, the excessively high temperature spots would adversely lead to performance degradation, increased cooling costs, reduced reliability and aging issues for those high-performance chip multiprocessors [6, 46]. Thermal-induced long-term reliability issues such as electro-migration, thermal stress migration, time-dependent dielectric breakdown (TDDB) and thermal cycling are especially a growing concern for today's microprocessors as the mean time to failure (MTTF) depends exponentially on the chip temperature due to Arrhenius relationship [47]. The semiconductor industry faces the challenges of maintaining reliability due to the continued increase in die size and number of transistors as well as the continuous scaling of transistors for increasing performance [48]. For multi/many core reliability systems, the power consumptions and the resulting temperatures of cores highly depend on the tasks or loads. As a result, effectively regulating the on-chip temperatures and optimizing related reliability issues of these chips becomes very important and imperative.

Dynamic thermal management (DTM) techniques have been proposed in the past to keep the temperature to stay below a limit [5, 6, 49]. These techniques, which typically consist of dynamic voltage and frequency scaling (DVFS), task throttling and clock gating, were first developed for single-core microprocessors. Recently, those techniques have been extended for multi-core architectures and MPSoCs. They include control theory-based frequency-control method [50], the combined DVFS and task migration methods [51, 52], the predictive control method [53, 54], and task migration based methods [53, 55, 56].

Among these thermal management methods, task scheduling/migration technique serves as a viable way to control the temperature of the cores [53, 55, 56] while maximally maintaining the performance of the processor system. The general idea of these methods is to migrate the heavy loaded task away from an overheated core to a cooler core. The traditional approach like [55], typically migrate the heaviest task (with largest power consumption) to the coolest cores to balance the temperature profile of all the cores. However, such an intuitive decision may lead to sub-optimal result, because it does not take into account of the influence from the neighboring cores as well as transient heat removal effect due to heat conduction of the package. To improve the thermal profile, recent work [56] proposed a new thermal management approach that used a temperature incremental factor to count the heat flux from the neighboring cores. But this approach is more of an ad-hoc way than a systematic way to solve this problem.

Also, most existing approaches [30, 53, 55, 56] are centralized control approaches, which requires a Centralized Manager (CM) to monitor the temperature and power of each core in the many-core microprocessor, and globally reallocate the resources and schedule the tasks. The centralized control method, however, will not be scalable in the context of future many core processors with hundreds and even thousands of cores [42, 44]. It will suffer high computational costs inside the CM, single point of failure, and large volume of monitoring and communication traffics around CM. To mitigate these problems of the centralized control approach, the distributed control approaches have been proposed [57–59]. In [57], an agent-based distributed power/thermal management method was proposed, using dynamic voltage and frequency scaling technique. The thermal management is done via localized agent negotiations and each agent (or the core it represents) acts independently and is reactive to its neighbors only. But this method only considers the current steady-state temperatures of each core for each agent to make the decision. In [58, 59], a distributed task migration technique, called *DTB-M*, was proposed. DTB-M monitors steady-state temperature and power in each local core, and performs task migration only between two neighboring cores. However, DTB-M aims at reducing average temperature based only on steady-state temperature information, and thus may be sub-optimal in terms of reducing on-chip thermal hot spots and transient temperature variations.

We notice that considering the thermal transient effects can be very straightforward: one can perform the full-blown transient thermal analysis inside the internal loops of those thermal optimization and scheduling algorithms. However, the major issue for such a simple approach is the prohibitive cost. Some existing works for thermal-aware task scheduling for real-time systems still try to avoid the transient analysis in the scheduling method based on the mixed-integer linear programming method [60]. However, it only allows the transient analysis on the relaxed heuristic method. In Chapter 5 of this dissertation, we propose a new task-migration based distributed thermal management scheme to reduce the on-chip temperature variance and the occurrence of hot spots by considering package related transient thermal effects without incurring the high computation costs of transient thermal analysis.

1.5 System Level Reliability Modeling and Management Techniques

High on-chip temperature could exacerbate reliability issues, which are becoming a limiting constraint in high-performance microprocessor designs due to the high failure rates in deep submicron and nanoscale devices. The increase in failure rates is caused by high integration levels and higher power densities, which leads to excessive on-chip temperatures. The introduction of new materials, processes and devices, coupled with voltage scaling limitations and increasing power consumption will impose many new reliability challenges. The semiconductor industry faces the challenges to maintaining reliability such as the continued increase in die size and number of transistors and the constant scaling of transistors for performance [48]. Increasing transistor density and thus power density is causing higher temperatures on chip, resulting in failure acceleration. Scaling to smaller transistors increases failure rates by shrinking the thickness of dielectrics. This has led the International Technology Roadmap for Semiconductor (ITRS) to predict the onset of significant reliability problems in the future, and at a pace that has not been seen in the past [61].

Some initial efforts have been carried out for system level reliability analysis for SoCs (system-on-a-chip). RAMP [62] is the first architecture level tool for modeling the long-term processor reliability of microprocessors at the design stage. The follow-up work by the same authors proposed a dynamic reliability management (DRM) concept by dynamic voltage and frequency scaling (DVFS) [63]. It showed that it is not sufficient to just manage the temperature or power from the reliability perspective. Method in [64] shows that the power/performance and reliability are intrinsically conflicting metrics and have strong interactions on SoC designs, and proposes a joint policy optimization method. Another dynamic reliability management method was proposed in [65], in which a simple PID based run-time control was applied to optimize the performance subject to the long-term reliability constraints. Recently, DVFS techniques considering negative bias temperature instability (NBTI) effects were proposed for microprocessors [66]. A supply voltage scheduling technique was proposed for optimizing energy subject to NBTI constraints [67].

Despite of these early efforts, the research on system or architecture-level reliability analysis and optimization is still in its early stage. For electromigration related reliability effects, all the early efforts at architecture and system level are based on the simple semi-empirical Black's equation shown below to estimate the mean time to failure (MTTF) of interconnect wires and simplified series reliability and constant failure rate models [62, 68].

$$MTTF = Aj^{-n}exp\{E_a/kT\} \tag{1.1}$$

Here, j is the current density, k is the Boltzmann's constant; T is the absolute temperature; E_a is the EM activation energy. However, the current density exponent, n, and Activation energy E_a in (1.1) actually depends on current density and temperature [69, 70]. Typically the values of the two parameters were obtained at the stressed (accelerated) conditions. However, value of n actually depends on the current densities and residual stress and E_a is the function of temperature. As a result, the Black's equation will not be accurate for normal operational conditions of chips. Second, the Black's equation ignores the impacts of existing thermal stress and residual stress caused by the chip manufacture process and chip-package interactions. Third, existing approaches employed high pessimistic series reliability model to compute the reliability of interconnect wires. For practical mesh-structured power grid networks, which are more susceptible to EM reliability issues, there exists high-level redundancy as the failure of some of wire segments will not be necessary to result in the voltage drops below the critical threshold, which really determine the failure of the power grid networks.

Chapter 6 of this dissertation first presents a new life time resource based reliability model derived from fundamental physics principles instead of black equation, and then proposes a dynamic management method for multi/many core processors to effectively balance and compensate the consumption of the life time resources across the cores so that early failure of a particular core could be prevented.

1.6 High Lights and Contributions

To enable thermal aware chip and package design and alleviate the on-chip thermal stress and related issues, new thermal modeling and management methods to analyze and control on-chip temperature at different abstraction levels are studied in this dissertation, thermal related reliability problems are also addressed with new prospectives, and the major contributions of this dissertation work are explained and summarized as the following different aspects:

- A compact closed form thermal model of TSV (Through silicon via) is proposed based on physics principles for chip level thermal behavior modeling of 3D ICs. The proposed model considers both insulation and the filling core of TSV because the thermal resistivity of copper is approximately one third that of silicon and cannot be ignored. We show that the space between TSVs has a significant impact on the lateral thermal resistance as TSVs change the isothermal profiles of each other, and the models for TSV array or farms must be TSV space dependent. Model calibration is usually required to take into account this variance. The research shows that the thickness of the insulation layer has limited influence on the accuracy of the proposed model in many practical cases and does not need to be calibrated. The proposed model is fully compatible with existing TSV modeling approaches, and could be integrated into a finite difference thermal model code to improve the accuracy of package level thermal simulation.

- Top down thermal modeling flow based on subspace identification method is proposed to extract the package level thermal model for microprocessors package thermal analysis. We observe that the subspace identification method may suffer predictability problem when power maps are given where power inputs are spatially correlated. For instance, the busy ALU will be likely to have frequent memory accesses and many instruction fetching activities. As a result, the corresponding function units will have power increases or decreases at the same or similar times. Such correlated input signals pose difficulty for the subspace identification method and will easily lead to loss of predictability as it is more difficult to distinguish the contributions from specific inputs when all the inputs have the same or similar transient waveforms. In this dissertation, we show that the input power signal needs to meet some independence requirements to ensure model predictability (rank of input power maps or their power signal matrix needs to meet certain requirements). A new algorithm, *ThermSubCP*, can generate independent power maps to meet the spatial rank requirement and can also automatically select the order of the resulting thermal models for the given error bounds.

- In package level thermal modeling, we also show that thermal package systems are fundamentally nonlinear. One important example is that thermal conductivities of silicon and package materials are temperature dependent. Another example is the changing thermal boundary conditions due to different fan speeds. To mitigate this problem, in the proposed subspace identification based top-down modeling flow, we apply the piecewise linear (PWL) scheme to characterize the nonlinear thermal behavior under these conditions. Our experiments show that the nonlinear effects in the thermal systems are typically mild and weak but are still significant enough to warrant the PWL modeling. However, nonlinearity due to boundary conditions can be very significant. PWL can deal with both mild and hard nonlinearities. We observe that the PWL method can lead to smaller models and reduced modeling costs compared to high order model approximation. This is important as the costs of identifying and simulating the reduced models will grow at least quadratically, it is critical to reduce the model order to maintain the efficiency gain from the reduced order modeling. The new modeling algorithm, *ThermSubPWL*, partitions the nonlinear ranges (due to temperature or boundary condition changes) into a number of small ranges and performs the modeling on each range using the previously proposed *ThermSubCP* method. A linear transformation method, which avoids the existing multi-transition requirement, is proposed to transform the identified linear local-models to the common state basis to build the continuous piecewise linear model. To the best knowledge of the authors, the proposed method is the first work addressing the nonlinear thermal modeling problem.

- To achieve more efficient architecture level thermal management, unlike the existing approaches, which use the steady state temperature, we propose a new temperature metric, called *effective initial temperature*, to guide the task migration process. The new temperature metric is based on frequency domain moment matching analysis technique. It automatically and systematically considers the impacts of neighbor cores. Furthermore, it also considers the thermal conductivity and thermal capacity effects of each individual core and rest of other cores. We show that by considering the dominant temperature moment component, the resulting algorithm can lead to more reduction of thermal hot spots without full transient thermal simulation. Furthermore, we show how the *effective initial temperature* of each core can be computed in a localized or distributed way for fully distributed thermal management. With locally computed *effective initial temperature* of each core as a thermal metric, the proposed distributed task migration algorithm can still perform very effective task scheduling in a distributed way compared to the centralized one but with much better scalability to future many-core systems with hundreds or thousands of cores.

- To effectively quantize and control electro-migration induced reliability (EM-reliability) of the system that is strongly influenced by temperature, the EM induced mean time to failure (MTTF) at the system level is modeled as a resource, which is abstracted from a more physics-based EM model at the chip physical level. Such resource-based EM models allow more flexible dynamic EM-reliability management for multi/many-core systems. With the insight of the new EM model, we propose a new reliability resource based task migration method to explicitly balance the consumption of EM life time resources for all the cores. The new method will lead to the equal chance of failure of these cores, which will maximize the life time of the whole multi/many core system by eliminating early failures. With balanced life time among all the cores, low power mode control is enabled to compensate the excessively consumed life time of all the cores when the chip is assigned with heavy tasks for a certain period of time. In this way, the MTTF of all the cores could be compensated to satisfy the requirement, giving the multi/many core system more flexibility to handle heavy task assignment when needed.

1.7 Organization

The rest of the dissertation is organized as the following: In Chapter 2, the fundamental concepts and methods of thermal modeling and management are introduced. In Chapter 3, a physical lateral thermal model of TSV and TSV arrays is proposed, and incorporated into finite difference method for chip level thermal behavior modeling and simulation of 3D stacked ICs. In Chapter 4, the proposed top-down methodology for package level thermal behavior modeling are presented, and piecewise linear method to improve model accuracy in presence of thermal nonlinearity is discussed. In Chapter 5, an architecture level distributed thermal management method that considers package induced transient thermal effect to make optimized task migration decision is presented. In Chapter 6, a system level EM-reliability model is proposed, which models EM-reliability as life time resources, and a dynamic reliability management method is developed to balance and compensate the life time resources. Chapter 7 concludes this dissertation.

Chapter 2

Fundamentals of Thermal Modeling and Management Methods

This chapter gives an overview of the basic concepts and methods used in thermal modeling and related management methods. It also presents a detailed discussion of the mathematics that will be used in the later chapters of this dissertation.

2.1 General Thermal Modeling Concepts and Approaches

This section gives a brief introduction to the basical concepts in thermal modeling and simulation, and outlines fundamental approaches used in this area.

2.1.1 Deriving Thermal Behavior Model from its Physical Principle

The fundamental physics principle of heat transfer and dissipation is given by the 3-D parabolic partial differential equation [71]. Using $T(\vec{r}, t)$ to represent the temperature at location $\vec{r}$ and time point t, the 3-D heat equation could be written as the following:

$$\rho C_p \frac{\partial T(\vec{r}, t)}{\partial t} = \nabla \cdot [\kappa(\vec{r}, T) \cdot \nabla T(\vec{r}, t)] + g(\vec{r}, t), \tag{2.1}$$

which subjects to Robin's boundary condition

$$\kappa(\vec{r}, T) \frac{\partial T(\vec{r}, t)}{\partial n} = h(T(\vec{r}, t) - T_{amb}). \tag{2.2}$$

In (2.1), ρ (Kg/m^3) denotes the density of the material that conducts the heat flow, C_p $(J/Kg{\cdot}K)$ represents the thermal capacity of the material, κ $(W/m{\cdot}K)$ represents the thermal conductivity of the material, and g (W/m^3) represents energy generation rate inside the material. In the boundary condition in (2.2), n represents a unit vector that is normal to the boundary surface, h (W/m^2K) represents the heat-transfer coefficient for the convective surface, and T_{amb} is the temperature of the ambient environment that surrounds the thermal system.

In order to compute the item $\kappa(\vec{r}, T) \cdot \nabla^2 T(\vec{r}, t)$ in (2.1), denote κ_x, κ_y, and κ_z as the effective thermal conductivities along the x, y, and z axes, respectively. Let κ_{x+}, κ_{x-}, κ_{y+}, κ_{y-}, κ_{z+}, and κ_{z-} represent the effective thermal conductivities in the $x+$, $x-$, $y+$, $y-$, $z+$, and $z-$ directions to model the inhomogeneous of the heat conduction media, respectively, i.e., we have the following relations

$$\kappa_x = \begin{cases} \kappa_{x+}, & x \rightarrow x+, \\ \kappa_{x-}, & x \rightarrow x-, \end{cases}$$

$$\kappa_y = \begin{cases} \kappa_{y+}, & y \rightarrow y+, \\ \\ \kappa_{y-}, & y \rightarrow y-, \end{cases}$$

$$\kappa_z = \begin{cases} \kappa_{z+}, & z \rightarrow z+, \\ \\ \kappa_{z-}, & z \rightarrow z- . \end{cases}$$

Using the finite difference method, derivatives at discretized mesh grid (i, j, k) are given by

$$\begin{aligned} \kappa_x \frac{\partial^2 T}{\partial x^2} &= \kappa_x \frac{T_{i+1,j,k} - 2T_{i,j,k} + T_{i-1,j,k}}{\Delta x^2} \\ &= \frac{\kappa_{x+}}{\Delta x^2}(T_{i+1,j,k} - T_{i,j,k}) + \frac{\kappa_{x-}}{\Delta x^2}(T_{i-1,j,k} - T_{i,j,k}), \\ \kappa_y \frac{\partial^2 T}{\partial y^2} &= \kappa_y \frac{T_{i,j+1,k} - 2T_{i,j,k} + T_{i,j-1,k}}{\Delta y^2} \\ &= \frac{\kappa_{y+}}{\Delta y^2}(T_{i,j+1,k} - T_{i,j,k}) + \frac{\kappa_{y-}}{\Delta y^2}(T_{i,j-1,k} - T_{i,j,k}), \\ \kappa_z \frac{\partial^2 T}{\partial z^2} &= \kappa_z \frac{T_{i,j,k+1} - 2T_{i,j,k} + T_{i,j,k-1}}{\Delta z^2} \\ &= \frac{\kappa_{z+}}{\Delta z^2}(T_{i,j,k+1} - T_{i,j,k}) + \frac{\kappa_{z-}}{\Delta z^2}(T_{i,j,k-1} - T_{i,j,k}), \end{aligned}$$

where $T_{i,j,k}$ represents the temperature at node (i, j, k); Δx, Δy, and Δz are the discretized steps along the x, y, z directions, respectively. Thus, the discretized form of (2.1) is

$$\begin{aligned} \rho C_p \frac{dT}{dt} =& -(\frac{\kappa_{x+}}{\Delta x^2} + \frac{\kappa_{x-}}{\Delta x^2} + \frac{\kappa_{y+}}{\Delta y^2} + \frac{\kappa_{y-}}{\Delta y^2} + \frac{\kappa_{z+}}{\Delta z^2} + \frac{\kappa_{z-}}{\Delta z^2})T_{i,j,k} \\ &+ \frac{\kappa_{x+}}{\Delta x^2}T_{i+1,j,k} + \frac{\kappa_{x-}}{\Delta x^2}T_{i-1,j,k} + \frac{\kappa_{y+}}{\Delta y^2}T_{i,j+1,k} \\ &+ \frac{\kappa_{y-}}{\Delta y^2}T_{i,j-1,k} + \frac{\kappa_{z+}}{\Delta z^2}T_{i,j,k+1} + \frac{\kappa_{z-}}{\Delta z^2}T_{i,j,k-1} + g_{i,j,k,t}, \end{aligned}$$

where $g_{i,j,k,t}$ is the heat generation at node (i, j, k). Furthermore, the above discretized heat equation for the mesh grid (i, j, k) can be rewritten as follows

$$\begin{aligned} \rho C_p \Delta V \frac{dT}{dt} =& -(G_{x+} + G_{x-} + G_{y+} + G_{y-} + G_{z+} \\ &+ G_{z-})T_{i,j,k} + G_{x+}T_{i+1,j,k} + G_{x-}T_{i-1,j,k} + G_{y+}T_{i,j+1,k} \\ &+ G_{y-}T_{i,j-1,k} + G_{z+}T_{i,j,k+1} + G_{z-}T_{i,j,k-1} + \Delta V g_{i,j,k,t}, \end{aligned} \tag{2.3}$$

where $\Delta V = \Delta x \Delta y \Delta z$, $G_{x+} = \frac{\kappa_{x+}\Delta y \Delta z}{\Delta x}$, $G_{x-} = \frac{\kappa_{x-}\Delta y \Delta z}{\Delta x}$, $G_{y+} = \frac{\kappa_{y+}\Delta x \Delta z}{\Delta y}$, $G_{y-} = \frac{\kappa_{y-}\Delta x \Delta z}{\Delta y}$, $G_{z+} = \frac{\kappa_{z+}\Delta x \Delta y}{\Delta z}$, and $G_{z-} = \frac{\kappa_{z-}\Delta x \Delta y}{\Delta z}$ give the physics definitions of the thermal conductance along $x-$,$x+$,$y-$,$y+$,$z-$ and $z+$ directions. Since the method of calculating thermal conductance in $z-$ and $z+$ directions have been studied [16], this dissertation will be focusing on the way to calculate the lateral thermal conductance in $x-$,$x+$,$y-$ and $y+$ direction, and apply it to finite difference thermal analysis through (2.3) .

Note that the thermal conduction inside the material is now modeled by (2.3), however, at the convective surface, the boundary condition (2.2) governs the thermal transfer, which could be discretized along x, y, and z directions as

$$\begin{aligned}
\kappa \frac{T_{x0} - T_{x1}}{\Delta x} &= h_x (T_{amb} - T_{x0}) \\
\kappa \frac{T_{y0} - T_{y1}}{\Delta y} &= h_y (T_{amb} - T_{y0}) \\
\kappa \frac{T_{z0} - T_{z1}}{\Delta z} &= h_z (T_{amb} - T_{z0})
\end{aligned} \tag{2.4}$$

in which T_{x0}, T_{y0}, and T_{z0} are the temperature grids at the convective boundary, and T_{x1}, T_{y1}, and T_{z1} are the temperature the adjacent temperature grids of those at the convective boundary inside the material. Using the thermal conductance G_x defined in (2.3), the thermal boundary equations could be written as

$$\begin{aligned}
(T_{x0} - T_{x1}) G_x &= \frac{h_x \Delta x G_x}{\kappa} T_{amb} - \frac{h_x \Delta x G_x}{\kappa} T_{x0} \\
(T_{y0} - T_{y1}) G_y &= \frac{h_y \Delta x G_y}{\kappa} T_{amb} - \frac{h_y \Delta y G_y}{\kappa} T_{y0} \\
(T_{z0} - T_{z1}) G_z &= \frac{h_z \Delta x G_z}{\kappa} T_{amb} - \frac{h_z \Delta z G_z}{\kappa} T_{z0}
\end{aligned} \tag{2.5}$$

Hence, in this way, given the chip and package properties as well as the convective cooling conditions, the thermal model of the chip and package could be built through finite difference method based on the physics principles of heat conduction equation (2.1).

2.1.2 Equivalent Thermal Circuit Modeling Approach

The discretized heat equation (2.3) has the same format as that of electrical network. With temperature analogous to voltage, and energy source analogous to current source, the discretized heat equation (2.3) could be represented by an equivalent circuit model, which is called thermal circuit to distinguish it with the conventional electrical circuit. Assuming the model has totally n temperature node, it could be proved that, the discretized heat equation (2.3) could be written as the following matrices format [71]:

$$GT(t) + C\frac{dT(t)}{dt} = BU(t), \tag{2.6}$$

in which $T(t) \in \mathbb{R}^n$ is the temperature vector with the temperature value $T(i,j,k)$ as its element for each grid (i,j,k), $G \in \mathbb{R}^{n\times n}$ is the thermal conductance matrix with G_x, G_y, and G_z in (2.3) as its elements, $C \in \mathbb{R}^{n\times n}$ is the thermal capacitance matrix with ρC_p as its elements in diagnal, $U(t) \in \mathbb{R}^n$ is the energy source vector with $\Delta V g(i,j,k,t)$ as its elements for each grid (i,j,k) at time instance t, and B is the position matrix that defines the position of the energy sources. A node of the equivalent thermal circuit is shown in Fig.2.1 to represent the thermal model described by the discretized heat equation (2.3). The temperature node connects to its 6 adjacent nodes through thermal resistors, and is connected to ground by a heat capacitor and a current source that mimic the energy source inside the mesh grid the node represents. In this way, thermal model is abstracted into an equivalent circuit network with resistors and capacitors, and the circuit analysis techniques could be directly applied to thermal analysis, providing insights for intuitive understanding of thermal behavior of the chip and its package.

2.1.3 State Space Thermal Modeling Approach

In general, multiple-input and multiple-output dynamic systems could be modeled by state space equations as the following

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{2.7}$$

where $A \in \mathbb{R}^{l \times l}$ is a state matrix, l is the number of states. $B \in \mathbb{R}^{l \times p}$, $C \in \mathbb{R}^{q \times l}$, and $D \in \mathbb{R}^{q \times p}$. The input vectors $u(t) \in \mathbb{R}^{p \times 1}$ are the measured power input traces and output vectors $y(t) \in \mathbb{R}^{q \times 1}$ are the temperature responses. The thermal system could be perceived as a multiple-input and multiple-output dynamic system with power as inputs $u(t_i)$ and temperature as outputs $y(t_i)$. The benefit of representing the thermal system in this way is that subspace identification method could be directly used to identify the state space matrices A, B, C, and D based on the measured inputs and outputs of the system [41]. Once these matrices are determined, the state space equation could be used to predict the temperature response under any power profiles. Note that given the discrete system matrices, the continuous system matrices can be obtained using methods such as Zero-Order-Hold, Impulse Invariance and Tustin Approximation [72]. Hence, this research uses subspace identification method to find the state matrices for discrete time model.

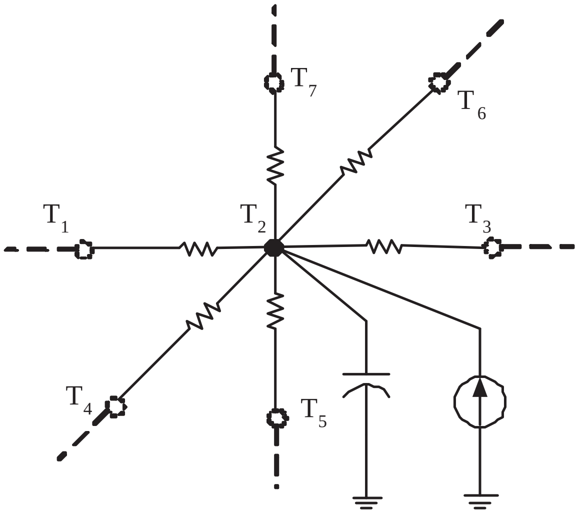


Figure 2.1: A node of the equivalent thermal circuits, which represents a temperature point connecting to all its adjacent nodes through thermal conductance.

2.2 Review of Subspace Identification Method

This section reviews an important techniques – subspace identification method that will be using for thermal modeling in this dissertation. That is, given the time domain waveform of input $u(t)$ and output $y(t)$, subspace identification method identifies the state matrices A, B, C, and D of (2.7). The subspace identification basically tries to first identify the system states first (Kalman states), then the state matrices will be obtained by the least square based optimization method [41]. For subspace method, there are several implementations such as the Ho-Kalman' method, the MOSEP method and the N4SID (Numerical algorithms for Subspace System Identification) method [40]. In this research, we apply the widely used N4SID method for the thermal system identification problem.

2.2.1 Introduction to N4SID Method

Before we present the N4SID algorithm, some notations are introduced first. Given integers a and b with $a < b$ and N is an arbitrary number representing time points, l is the order of the desired system, we define the *input Hankel matrix* as

$$U_{a|b} := \begin{bmatrix} u(a) & u(a+1) & \cdots & u(a+N-1) \\ u(a+1) & u(a+2) & \cdots & u(a+N) \\ \vdots & \vdots & \vdots & \vdots \\ u(b) & u(b+1) & \cdots & u(b+N-1) \end{bmatrix} \tag{2.8}$$

$$\in \mathbb{R}^{(b-a+1)p \times N}$$

and the *output Hankel matrix* $Y_{a|b}$ is defined accordingly as

$$Y_{a|b} := \begin{bmatrix} y(a) & y(a+1) & \cdots & y(a+N-1) \\ y(a+1) & y(a+2) & \cdots & y(a+N) \\ \vdots & \vdots & \vdots & \vdots \\ y(b) & y(b+1) & \cdots & y(b+N-1) \end{bmatrix} \tag{2.9}$$

$$\in \mathbb{R}^{(b-a+1)p \times N}$$

We further define the *state sequence* according to a given number a and the arbitrary number N as

$$X(a) := [x(a), x(a+1), \ldots, x(a+N-1)] \in \mathbb{R}^{l \times N} \tag{2.10}$$

Based on the previous definition, we can further define the *past* input, output Hankel matrices and state sequence as

$$U_p := U_{0|k-1}, \quad Y_p := Y_{0|k-1}, \quad X_p := X(0) \tag{2.11}$$

Using the same definition, the *future* input, output Hankel matrices and state sequence could be defined as

$$U_f := U_{k|2k-1}, \quad Y_f := Y_{k|2k-1}, \quad X_f := X(k) \tag{2.12}$$

and *past* data matrix and *future* data matrix as

$$W_p := \begin{bmatrix} U_p \\ Y_p \end{bmatrix}, \quad W_f := \begin{bmatrix} U_f \\ Y_f \end{bmatrix} \tag{2.13}$$

The numbers k and N which determine the row and column size of the input and output Hankel matrices are determined by user according to the number of input samples available along the time axis. Also, $k > l$ should be satisfied given that l is the order of the system.

Additionally, the extended observability matrix $\mathcal{O}_k$ is defined as follows

$$\mathcal{O}_k := \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{k-1} \end{bmatrix} \in \mathbb{R}^{kp \times l} \tag{2.14}$$

where p is the number of input and output ports.

In N4SID algorithm, an important property which can be proved is

$$\mathcal{P}_{U_f}(Y_f, W_p) = \mathcal{O}_k X_f \tag{2.15}$$

where $\mathcal{P}_B(A, C)$ represents an oblique projection of the row space of A onto the row space of C along row space of B [40, 41, 73].

By applying Singular Value Decomposition (SVD) on the left hand side of (2.15)

$$\mathcal{P}_{U_f}(Y_f, W_p) = [U_1 \; U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_1 V_1^T \tag{2.16}$$

the extended observability matrix $\mathcal{O}_k$ and the future state sequence X_f are readily to be identified as

$$\mathcal{O}_k = U_1 \Sigma_1^{1/2} \tag{2.17}$$

$$X_f = \Sigma_1^{1/2} V_1^T \tag{2.18}$$

Now the state sequence $X_f = X(k) = [x(k), x(k+1), \cdots, x(k+N-1)]$ is identified, we can proceed to determine the system matrices A, B, C, D. Specifically, we first define the following $(N-1)$-column matrices (compared to the previously defined N-column matrices) as

$$\bar{X}_{k+1} := [x(k+1), x(k+2), \ldots, x(k+N-1)] \tag{2.19}$$

$$\bar{X}_k := [x(k), x(k+1), \ldots, x(k+N-2)] \tag{2.20}$$

$$\bar{U}_{k|k} := [u(k), u(k+1), \ldots, u(k+N-2)] \tag{2.21}$$

$$\bar{Y}_{k|k} := [y(k), y(k+1), \ldots, y(k+N-2)] \tag{2.22}$$

All the four matrices are known or identified already. Then, the system matrices A, B, C, and D are solved as a least square problem directly from

$$\begin{bmatrix} \bar{X}_{k+1} \\ \bar{Y}_{k|k} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \bar{X}_k \\ \bar{U}_{k|k} \end{bmatrix} \tag{2.23}$$

2.2.2 Input signal requirements

The "quality" of input signal is very important in subspace methods. In order to get a "good" system, the input signal should satisfy the so-called *persistently exciting* condition. That is, for an l-order system and $l < k$, $\operatorname{rank}(U_{0|2k-1}) = 2pk$ for a p-port system assuming that the number of columns of the Hankel matrix N, is sufficiently large in the N4SID method [41, 73].

Generally speaking, the input signal is *qualified* if a unique l-th order system can be accurately identified from the given input and output data. Consider the FIR (finite impulse response) model

$$y(t) = \sum_{i=0}^{2k-1} q_i u(t-i) \tag{2.24}$$

where $q_0, q_1, \cdots, q_{2k-1} \in \mathbb{R}^{p\times p}$ are the system impulse responses. Denote

$$Q_{2k-1} := \left[\begin{array}{c} q_{2k-1}, q_{2k-2}, \ldots, q_0 \end{array} \right] \tag{2.25}$$

and the system identification comes down to determining Q_{2k-1} using the input and output data.

From (2.24), we readily get

$$Y_{2k-1|2k-1} = Q_{2k-1} U_{0|2k-1} \tag{2.26}$$

where $U_{0|2k-1} \in \mathbb{R}^{2Nk\times N}$. This is a least square problem and Q_{2k-1} can be uniquely solved when $U_{0|2k-1}$ has full row rank [74].

$$\text{rank}(U_{0|2k-1}) = 2pk \tag{2.27}$$

Thus, we concluded in this section that in order to uniquely and accurately identify an l-th order system, it is necessary to have the input Hankel matrices full row rank.

2.2.3 A special case - baseline model for system identification

In this subsection, we consider a special case when any type of input would satisfy the persistently exciting method, and define it as our baseline model that we will be using in our model evaluation for our proposed method in Chapter 4. This is a single input and multiple output system (SIMO). As the input number $p = 1$, according to equation (2.27)

$$\text{rank}(U_{0|2k-1}) = 2k \tag{2.28}$$

is always full rank and thus satisfies the persistently exciting condition, which implies that the model would always be able to be identified if the transient inputs and outputs are known. Thus, it is reasonable to use this model as a baseline model even though it is not quit realistic in our thermal modeling since we always have multiple related input. Isolation one input from others is not practical, but it could give us an ideal baseline that could be used to evaluate the model identified by our proposed method with practical inputs. So, in this research, an ideal baseline model is identified with a step input signal applying at only 1 input port, and all the other input ports being set to zero. We will use this baseline model to help validate the frequency response of the model identified by our proposed method under realistic power inputs.

2.3 Architectural Level Thermal Management Methods

As the power density continues to increase, the excessively high temperature spots, called *thermal hot spot* would adversely lead to performance degradation, increased cooling costs, reduced reliability and signal integrity issues for multi-core microprocessors [6, 56, 75, 76]. Thus, thermal management and related cooling techniques has been identified by ITRS [61] as one of the five key challenges during the next decade for achieving the projected performance goals of the industry.

The goal of thermal management is to keep the multi-core microprocessor system working below a safe temperature level while maintaining the efficient task execution performance. This section gives a brief introduction to the on-chip thermal management methods.

2.3.1 Introduction to Dynamic Voltage and Frequency Scaling Methods

One popular type of methods for runtime thermal management is involved with a dynamic power reduction technique, called *Dynamic Voltage and Frequency Scaling* (DVFS) [6, 77, 78]. This type of methods dynamically control the on-chip temperature profile through controlling the on-chip dynamic power, which is given by

$$P = CV^2 f \tag{2.29}$$

in which P represents the dynamic power, V represents the supply voltage, and f represents the frequency of switching activity. Thus, this type of methods could quadratically reduce the dynamic power through reducing the supply voltage and operating frequency, which translates to reduced data rate. In this way, the DVFS based thermal management method achieves temperature reduction at the expense of reduced performance. Many modern processor has included DVFS feature by implementing different performance states [79], which essentially use different voltage and frequency settings of the processor to either boost performance or reduce power.

2.3.2 Introduction to Task migration Methods

In order to maximumly maintain the performance of the processor system, task scheduling/migration technique is widely adopted as an alternative to control the temperature of the multi-core processor cores [53, 55, 56]. During the task executions, some cores might be continuously assigned with heavy tasks, while some cores might be idle or loaded with lighter tasks, the region of the cores that continuously takes on heavy tasks tend to become overheated if no action is taken to alleviate the ever increasing temperature of these cores. To handle this situation, the task migration thermal management methods detect these overheated cores and migrate the heavy loaded task away from an overheated core to a cooler core. The traditional approach like [55], typically migrate the heaviest task (with largest power consumption) to the coolest cores to balance the temperature profile of all the cores. This intuitive method could effectively alleviate the thermal emergence of these overheated cores. However, such an intuitive decision may lead to sub optimal task migration decision, because it does not take into account of the influence from the neighboring cores. In fact, the temperature of a microprocessor core is determined by the work load of the core itself, as well as the work load of its adjacent cores and transient thermal effect of the package system. To improve the thermal profile, recent work [56] proposed a new thermal management approach that used a temperature incremental factor to count the heat flux from the neighboring cores.

Task migration based thermal management methods target at achieving more balanced temperature profile through task redistribution across the chip. The optimized task configurations are critical to balance the on-chip temperature profile during run time. Traditional centralized ask migrations could possibly lead to large data traffic and overheads in multi-core processor systems.

2.4 Summary

This chapter has presented the fundamental concepts and ideas of thermal modeling and management for microprocessor systems. The three types of thermal modeling methods have been briefly introduced: physics-based modeling based on heat equation, equivalent thermal circuit modeling approach, and state space modeling approach. The urgent needs for on-chip thermal management is explained, and the two basic types of thermal management methods are reviewed: DVFS-based methods and task migration methods. The following chapters will present the proposed thermal modeling approaches at chip level and package level, and also discuss the proposed architectural level thermal management methods in details. In addition, to address thermal related reliability issues, this dissertation will also discuss how the concept of task migration in thermal management be extended to implement a new reliability management based on the proposed system level reliability modelt to balance and compensate the life time resources of multi-core processors.

Chapter 3

Compact Lateral Thermal Resistance Model of TSVs for Fast Finite Difference Thermal Analysis of 3D Stacked Chips

This chapter studies the chip level thermal modeling method that considers thermal behavior of TSV and TSV arrays for 3D stacked ICs. Thermal issue is the leading design constraint for 3D stacked ICs, and through silicon vias (TSVs) are used to effectively reduce the temperature of 3D ICs. Normally, TSV is considered as a good thermal conductor in its vertical direction, and its vertical thermal resistance has been well modeled. However, lateral heat transfer of TSVs, which is also important, was largely ignored in the past. In this chapter, we propose an accurate physics-based model for lateral thermal resistance of

TSVs in terms of physical and material parameters, and study the conditions for model accuracy. For TSV arrays or farm, we show that the space or pitch between TSVs has a significant impact on TSV thermal behavior and should be properly considered in the TSV models. The proposed lateral thermal resistance model is fully compatible with the existing modeling approaches, and thus we could build a more accurate complete TSV thermal model. The new TSV thermal model can be easily integrated into a finite difference based thermal analysis framework to improve analysis efficiency. The accuracy of the model is validated against a commercial finite element tool - COMSOL. Experimental results show that the improved TSV thermal model (with proposed lateral thermal model) could greatly improve the accuracy of finite difference method in thermal simulation comparing with the existing method.

The chapter is organized as follows: Section 3.1 discusses the problem formulation for lateral thermal modeling. Section 3.2 describes the mathematical formulation of the closed form expression of the lateral thermal resistance of the TSVs, and outlines the method to integrate the proposed model into 3D finite difference thermal code. The experimental results are discussed in Section 3.4. Section 3.5 concludes this chaper.

3.1 Problem Formulation

Finite difference based method is a popular approach for thermal analysis [80]. This numerical approach uses mesh grids to discretize the partial differential equations from the heat diffusion dynamics. For thermal modeling of 3D chip structure, it is important to incorporate TSVs into the thermal model as they significantly change the thermal profile of the 3D chip [15, 17, 81]. However, to capture the small feature size of the TSV, like diameter of TSV and thickness of the insulation layer (liner) of TSV, very fine mesh grids are required. This significantly increases the model complexity and the cost of simulation. For example, the radius of a TSV typically ranges from $10\mu m$ to $100\mu m$ while the thickness of the liner ranges from 2% to 10% of the radius of the TSV [18, 82]. As a result, to make the thermal grids as small as a few μm can lead to very huge thermal equations, which typically is not necessary. Therefore, large grid sizes can be used for the finite difference process resulting in a few TSVs inside a grid or cell. One such example is shown in Fig. 3.1 in which a TSV is placed in one cell.

The vertical resistance for the TSV model can be computed by:

$$R_V = \frac{L}{kA} \tag{3.1}$$

where L is the TSV length in the vertical direction, $A = \pi r^2$ is the cross sectional area of the TSV cylinder in the vertical direction and k is the thermal conductivity of the TSV material. In this chapter, we focus on developing the analytical expression for the thermal resistance in the lateral direction in terms of physical and material parameters. For a TSV array or farm, the vertical resistance can be computed as $R_{array,V} = \frac{L}{k_{eff}A}$, where k_{eff} is the effective thermal conductivity, which depends on the ratio of the TSV areas versus the total areas considered [16].

The structure view of a TSV cell is shown in Fig. 3.1. It consists of a TSV in the center of the cell surrounded by silicon region on either side. We assume the TSV is made of copper and is surrounded by a layer of silicon dioxide which acts as the insulation material. In the vertical direction, the copper core serves as a good thermal conductor for

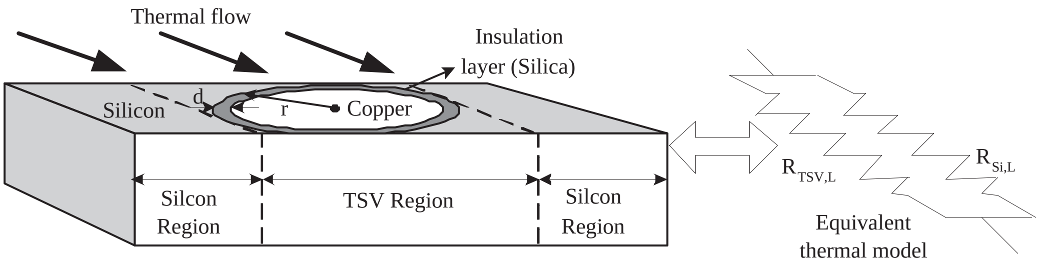


Figure 3.1: A structure view of TSV cell with copper core and insulation layer

heat transfer from the inside. However, the lateral direction of the insulation layer, although very thin, can have a large influence on the thermal resistance.

Let $R_{TSV,L}$ and $R_{Si,L}$ represent the lateral thermal resistances of the TSV region and the silicon region respectively. The lateral thermal resistance of the TSV cell can be expressed as a parallel combination of the two resistances as shown in Fig. 3.1.

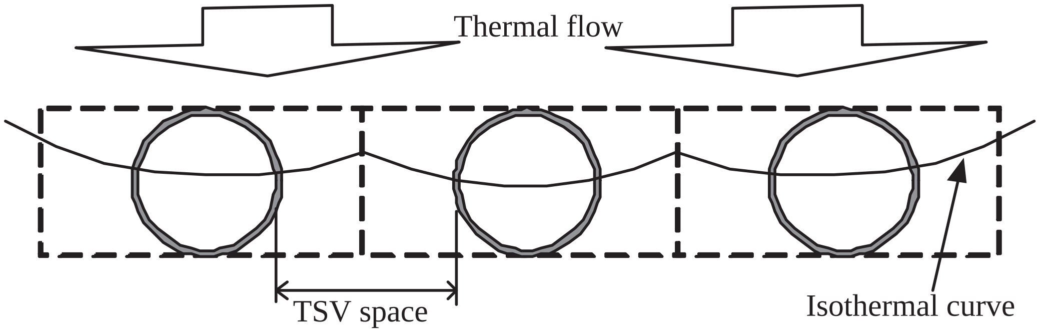


Figure 3.2: Top view of TSV array built using individual TSV cells

The TSV cell shown in Fig. 3.1 can be used to build any TSV array as shown in Fig. 3.2. The space between two TSVs (TSV space) in the array is the total width of the silicon region of each TSV cell. The lateral thermal resistance model of one TSV cell can thus be extended to compute the lateral thermal resistance of a TSV array of any scale.

The remaining part of this chapter discusses the closed form expression for the lateral thermal resistance of the TSV, and the compact finite difference thermal modeling method that incorporates the lateral thermal resistance effect.

3.2 Analytical expression for lateral thermal resistance of TSVs

3.2.1 Mathematical derivation of the closed form expression

The heat transfer follows the fundamental heat diffusion equation (2.1). For many heat transfer problems with unknown boundary conditions, closed form expression cannot be found [83].

To determine the closed form expression for the lateral thermal resistance of the TSV region, we start with the following three major assumptions:

1. Inside the insulation layer, majority of the isothermal curve is parallel to the insulation layer, which means that the lateral thermal flow is always vertical to the insulation layer as shown in Fig. 3.3.
2. The lateral thermal resistance of the silicon region contributed by the four corners of the TSV region is small compared to the TSV and is not considered while formulating the closed form expression.
3. Inside the copper core, the heat flow is straight as shown in Fig. 3.3, and the detour of the heat flow close to the insulation layer is negligible.

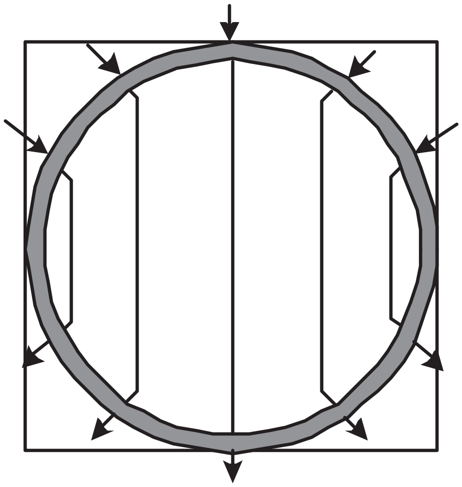


Figure 3.3: Direction of lateral thermal flow (arrows) across the TSV region

With these assumptions, we can calculate the resistance across a small portion of the insulation layer of the TSV as shown in Fig. 3.4. Considering the symmetry of the entire structure, the thermal resistance going through the insulation layer is [83]:

$$R_{ins,L} = 2\int_{r-\delta}^{r} \frac{\rho_{ins} dl}{\pi l h} \tag{3.2}$$

where r is the radius of cross section of the TSV, δ is the thickness of the insulation layer, h is the height of the cell, l is the position variable across the insulation layer, ρ_{ins} is the thermal resistivity of the insulation material and $R_{ins,L}$ is the calculated lateral thermal resistance of the insulation layer.

Using $\alpha = \delta/r$ as the ratio of the insulation layer thickness to the TSV radius, the equation (3.2) reduces to a closed form expression:

$$R_{ins,L} = -\frac{2\rho_{ins} ln(1-\alpha)}{\pi h} \tag{3.3}$$

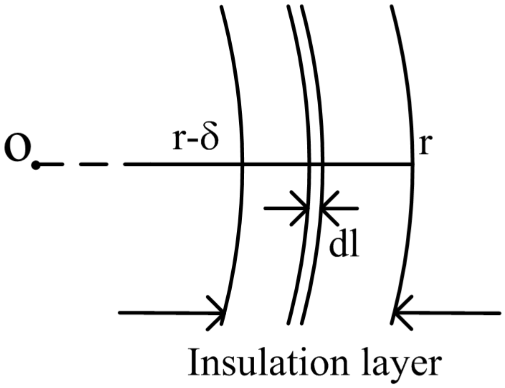


Figure 3.4: Thermal resistance calculation across the insulation layer

Assuming the heat flow is straight as shown in Fig. 3.3 and considering the symmetry of the structure, the lateral thermal resistance across the copper core as shown in Fig. 3.5 is given by:

$$R_{cu,L} = \int_{0}^{r-\delta} \frac{\rho_{cu} dl}{h\sqrt{(r-\delta)^2 - l^2}} \tag{3.4}$$

where ρ_{cu} is the thermal resistivity of the copper core of the TSV, and $r - \delta$ is the radius of the copper core. Considering the symmetry of the entire structure, the lateral thermal resistance of the copper core is:

$$R_{cu,L} = \frac{\rho_{cu}\pi}{2h} \tag{3.5}$$

Therefore, from assumptions 1-3, the lateral thermal resistance in the TSV region in a TSV cell can be written as:

$$R_{TSV,L} = -\frac{2\rho_{ins} ln(1-\alpha)}{\pi h} + \frac{\rho_{cu}\pi}{2h} \tag{3.6}$$

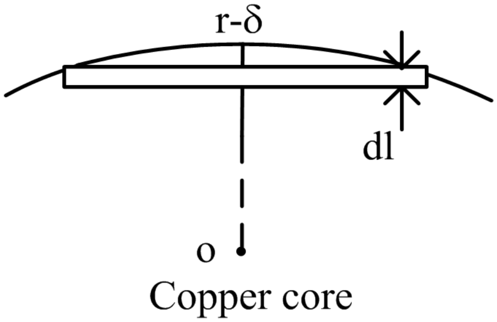


Figure 3.5: Thermal resistance calculation across the TSV core

Equation (3.6) shows that the value of lateral thermal resistance in the TSV region is proportional to $ln(1-\alpha)$ and $1/h$. This implies that the lateral thermal resistance is directly proportional to α and inversely proportional to the thickness.

The lateral thermal resistance of the entire TSV cell is the parallel combination of the lateral thermal resistance of $R_{TSV,L}$ and $R_{Si,L}$. This equivalent circuit model is shown in Fig. 3.1. As a result, for $R_{Si,L}$, we have

$$R_{Si,L} = \frac{2\rho_{si}r}{dh} \tag{3.7}$$

where d is the width of side silicon region. Since the width of the side silicon region orthogonal to the thermal flow is $d = rP$ where P is the ratio of the TSV space to TSV radius r, neglecting the bending of isothermal lines close to TSVs, the lateral thermal resistance of the side silicon region can also expressed as:

$$R_{Si,L} = \frac{2\rho_{si}}{Ph} \tag{3.8}$$

Therefore, the total lateral thermal resistance of the TSV cell is:

$$R_{TSVcell,L} = R_{TSV,L}//R_{Si,L} \tag{3.9}$$

Here, we assume that we have identical two side silicon regions with same width and the TSV is in the middle.

3.2.2 Model accuracy analysis and calibration

The derived analytical expression given in (3.9) is based on some assumptions and therefore will have some inaccuracies in some practical cases.

1. The lateral thermal resistance in equation (3.9) is based on the assumption that the isothermal curve is not bent in the silicon region in the TSV cell. However, due to the presence of TSVs, the isothermal curve has to be bent as shown in Fig. 3.2. Depending on the ratio of TSV space to TSV radius (P), the isothermal curve could be bent differently. Thus, the value of $R_{Si,L}$ depends on the ratio of TSV space to TSV radius. This has not been captured by the closed form model.
2. The thickness of the insulation layer can influence model accuracy. If the insulation layer is too thick, the isothermal profile in the insulation layer may change, and majority of the heat flow will no longer be vertical to the insulation surface. Thus, the thermal flow in the tangential direction inside the insulation layer will not be negligible. On the other hand, if the insulation layer is too thin, the model accuracy may go down. As the insulation layer gets thinner (for instance, as α tends to 0), the lateral thermal resistance from the silicon part in the TSV region becomes significant and will need to be accounted for.

We first consider the impacts due to the spaces between two TSVs. This space is normally 2 to 6 times the radius of the TSV (the ratio of TSV space to TSV radius

$P \in [2, 6]$) [7, 82, 84]. The isothermal curve is strongly influenced by this variation. Thus, to capture the influence of the space between the TSV arrays, model calibration based on numerical data is required to enhance the model accuracy. The modified lateral thermal resistance can be written as:

$$R_{TSVcell,cal,L} = R_{TSVcell,L}\theta \tag{3.10}$$

where θ, the modification factor, can be written as a linear function of the ratio of TSV space to TSV radius (P).

$$\theta = \beta_1 P + \beta_2 \tag{3.11}$$

Currently, detailed numerical analysis using COMSOL or measurements on some specially design chips will still be required to obtain the two fitting parameters β_1 and β_2 in the calibration formula.

But our experiments show that the calibration formula does not need to be re-calibrated again if the TSV structure is similar and the liner thickness is within a range as already shown by Fig. 3.11. Only if material changes, the model need to be re-calibrated. But more research to find the closed form expressions for β_1 and β_2 in terms of TSV structure and material parameters will be still interesting and this can be future investigation.

We remark in our proposed model in (3.9), the lateral thermal resistance $R_{TSVcell,L} = R_{TSV,L}//R_{Si,L}$ has two components, the resistance from TSV itself $R_{TSV,L}$ and the resistance from the two side silicon regions: $R_{Si,L}$. The space between TSV here is also the half width of the two side silicon region orthogonal to the heat flow direction as shown in Fig. 3.1. As a result, with larger distance, or larger P, the smaller the resistance of the side silicon will be as shown in (3.8). The total lateral resistance $R_{TSVcell,L}$ hence will go down with the increasing distance as shown Fig. 3.11.

But according to (3.10), the modification of $R_{TSVcell,cal,L}$ will grow as P increase. However, this is not the case practically. If we have TSVs which are far away, then the resistance of the side silicon will dominate the total lateral resistance as we can see. The resistance contribution of the TSV itself becomes less significant and less relevant. From practical point of a view, this will not happen. First, such TSV with very long side silicon region will not exist in the finite difference analysis as such TSV will be decomposed into several smaller grids or cells. The grid with TSVs will have small TSV distance, thus small P. As a result, a TSV, which is far away from other TSVs, essentially corresponds to condition for a small or zero-valued width d of the side silicon region and $P = 0$. The meaningful values for P and thus width of the side silicon region d are typically small for the practical finite difference analysis framework.

Although the thickness of the insulation layer affects the model accuracy, its influence is limited to 2% to 10% of the TSV radius, the range of practical thickness of the insulation layer [18, 82]. Thus, (3.6) is valid since our assumptions for deriving it are not violated for the practical thickness of the insulation layer within this range. Experimental results will show that the calculated thermal resistance error of the TSV cell due to the variation of the insulation layer thickness in this range is acceptable and does not need further calibration.

TSV arrays of any dimension could be constructed by extending the fundamental TSV cells periodically in lateral directions. Hence, having established the lateral thermal model of each fundamental TSV cell, and assuming a TSV array has M TSVs in L direction and N TSVs in the direction that is perpendicular to it, the lateral thermal resistance in L direction could be calculated by using parallel and serial connection theory as

$$R_{TSVarray,L} = R_{TSVcell,cal,L}\frac{M}{N} \tag{3.12}$$

Thus, once the lateral resistance of each TSV is calculated, estimating the lateral resistance of any TSV array is straightforward.

We expect that the proposed closed form model is a compact and robust one, the calibrated parameter β_1 and β_2 in (3.11) applies to TSV of any arbitrary radius in the silicon layer with different thickness. We will testify this in our experiments.

3.3 Compact thermal model with closed form lateral thermal resistance

By now, we have presented our proposed method of using closed form expression to calculate the thermal resistance of TSV and TSV arrays in lateral direction, which could be used to calculate the total lateral thermal resistance of a silicon structure with TSV array in it as

$$R_{tot,L} = R_{TSVarray,L} + R_{Si,L} \tag{3.13}$$

in which $R_{TSVarray,L}$ is the lateral thermal resistance contributed by the TSV array, and $R_{Si,L}$ is the lateral thermal resistance of the bulk silicon area with no TSVs and could be easily calculated by using

$$R_{Si,L} = \rho_{Si}\frac{L_\sigma}{A_\sigma} \tag{3.14}$$

in which L_σ is the length of the silicon in the direction of the heat flow, and A_σ is the cross area perpendicular the direction of the heat flow. In vertical direction, the calculation of thermal resistance including the TSV and TSV arrays is more straightforward, which has already been discussed in many previous works like [16, 18]. Using these approaches, the thermal resistance in vertical direction that considers the copper core and insulation liner in TSV array could be expressed as

$$R_V = R_{cu,V}//R_{ins,V}//R_{Si,V} \tag{3.15}$$

in which $R_{cu,V}$ represents the thermal resistance of the copper core of the TSV array, $R_{ins,V}$ represents the vertical thermal resistance of the insulation liner of the TSV array, and $R_{Si,V}$ represents the portion of the vertical thermal resistance from the remaining silicon bulk. Using the material and geometry parameters defined in Section 3.2, assuming the TSV array is $M \times N$ (M TSVs in x direction and N TSVs in y direction) and the chip area that contains the TSV array is S, the vertical thermal resistance due to copper core $R_{cu,V}$ of the TSV array can be calculated as

$$R_{cu,V} = \rho_{cu} \frac{h}{\pi r^2 (1-\alpha)^2 MN} \tag{3.16}$$

and the $R_{ins,V}$ is

$$R_{ins,V} = \rho_{ins} \frac{h}{\pi r^2 (1-(1-\alpha)^2) MN} \tag{3.17}$$

and the vertical thermal resistance due to the remaining bulk silicon $R_{Si,V}$ is

$$R_{Si,V} = \rho_{Si} \frac{h}{S - \pi r^2 MN} \tag{3.18}$$

Similarly, using the definition of thermal capacitance in [85], the thermal capacitance of the chip area that contains the TSV array could be expressed as an added summation of thermal capacitance contributed by different structures formed by different materials as

$$\begin{aligned} C_p = {} & c_{p,cu}h(\pi r^2(1-\alpha)^2 MN) \\ & + c_{p,ins}h(\pi r^2(1-(1-\alpha)^2)MN \\ & + c_{p,Si}h(S-\pi r^2 MN) \end{aligned} \tag{3.19}$$

In which $c_{p,cu}$ is the specific heat of the copper core in the TSV array, and $c_{p,ins}$ is the specific heat of the insulation liner, and $c_{p,Si}$ is the specific heat of the bulk silicon.

Hence, in the Cartesian coordinate, using R_{x+},R_{x-},R_{y+},R_{y-} to denote the thermal resistance in lateral direction, and R_{z+}, R_{z-} to denote the thermal resistance in vertical direction, the complete compact model of a silicon structure that contains the TSV array, like the structure in Fig. 3.6 (a), can be represented by complete thermal model shown in Fig. 3.6 (b). In this model, the lateral thermal resistance contributed by TSV arrays are considered using (3.12) and (3.13). For example, in $x+$ direction, the lateral thermal resistance R_x+ could be conveniently calculated as

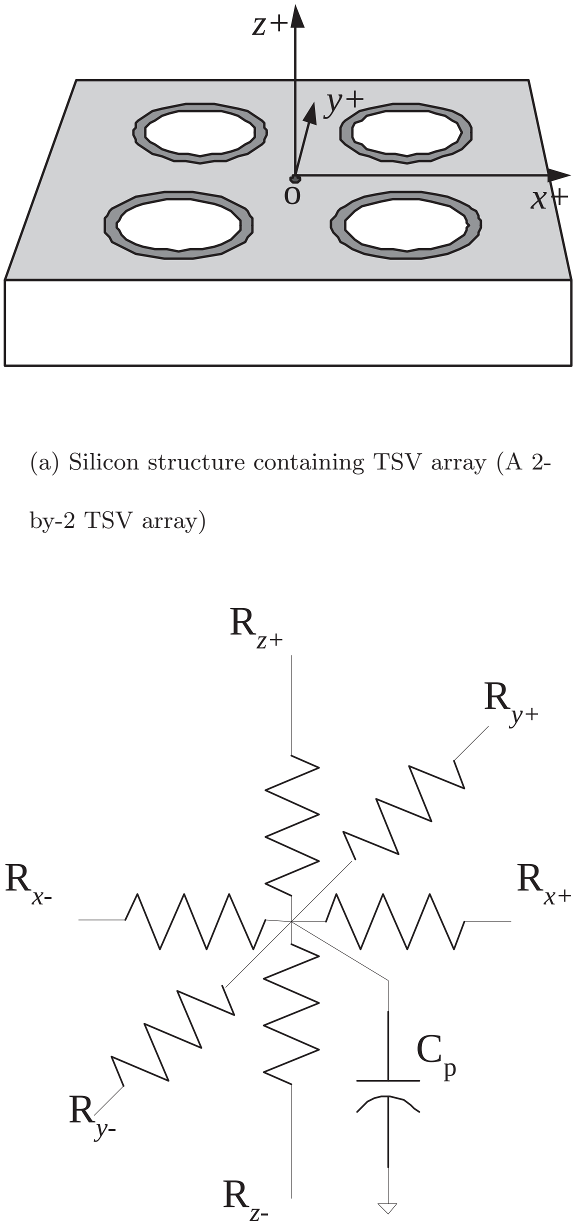
$$R_{x+} = R_{TSVarray,x+} + R_{Si,x+} \tag{3.20}$$

in which $R_{TSVarray,x+}$ represents the lateral thermal resistance across the TSV array in $x+$ direction and could be calculated using (3.12); $R_{Si,x+}$ represents the total lateral thermal resistance of the remaining silicon bulk region that is equivalent to serial connection with the TSV array in $x+$ direction. Therefore, in (2.3), we have $G_{x+} = \frac{1}{R_{x+}}$ in $x+$ direction, and $G_{x-} = \frac{1}{R_{x-}}$, $G_{y+} = \frac{1}{R_{y+}}$, $G_{y-} = \frac{1}{R_{y-}}$, $G_{z+} = \frac{1}{R_{z+}}$, $G_{z-} = \frac{1}{R_{z-}}$ in all other directions, and the lateral thermal resistance in these direction could be obtained in similar way as that of R_{x+}.

We remark that we distinguish the lateral resistance in two x and two y directions (for instance $x+$ and $x-$ directions for x). Such anisotropic effect in the x and y directions actually comes from the observation that thermal behavior of TSV will affect each other when they are placed in an array or a farm form as we showed in the chapter. For instance, for a finite difference grid n, in its $x+$ direction, it has TSV arrays adjacent to it, but in its $x-$ direction, it has no TSV arrays adjacent to it. In this case, the lateral thermal resistance would be different in $x+$ direction and $x-$ direction.

Since the closed form expression of the lateral thermal resistance of the TSV array is used in the thermal modeling, the accuracy improvement in lateral direction is achieved without increasing computation cost. The final lumped thermal model is now shown in Fig. 3.6 (b), which can be easily used by the finite difference method as one finite difference grid that contains one TSV or more TSVs. Such compact models will allow more efficient finite difference analysis without significant loss of temperature accuracy at both vertical and lateral directions for a 3D ICs.

For practical TSV with bonding technology, a thermal TSV, fabricated by 3D technology may span through several layers. For instance, a typical three-layer TSV structure and the corresponding thermal model (just the x and z directions and thermal capacitance is not shown here) is shown in Fig. 3.7. This TSV spans through three layers describing the silicon substrate (Si), the inter layer dielectric (ILD) and metal interconnects (i.e. the back end of line or BEOL), and the bonding layer, respectively [18]. Our proposed lateral thermal model is fully compatible with the existing layer by layer thermal modeling approach by just replacing the lateral thermal resistance computed in the silicon layer using



(a) Silicon structure containing TSV array (A 2-by-2 TSV array)

(b) The complete compact thermal model of the silicon structure with TSVs

Figure 3.6: Silicon structure containing TSV array and its compact thermal model considering thermal resistance of TSVs in 3D Cartesian coordinate

the proposed method. For layers like ILD and bonding layer that primarily use insulation materials, traditional thermal modeling techniques that do not consider lateral thermal resistance could still be applied, since the lateral thermal resistance in these layers is orders of magnitude higher than that in the vertical direction of TSVs.

We also remark that the TSVs shown in Fig. 3.7 are actually close to via-last TSVs. This work focuses on modeling the behavior of general TSVs and it can be applied to TSVs built before transistors (via first) or after the transistors (via-last). Also, inside silicon layer, both via-first and via-last TSVs have the same structure - copper core and insulation liner. Our work focuses on how to develop a compact lateral thermal resistance model for such a structure. We target at general TSVs, no matter they are via-first or via-last. The two types of TSVs may have differences in vertical ILD layers. As discussed already, we do not need to apply lateral thermal resistance model in ILD layer. Instead, the traditional vertical-only TSV thermal models should work in this case for the TSV portion in the ILD layers.

3.4 Numerical Results and Discussion

Experiments are performed using COMSOL 4.1 software on a Linux server with 1.6GHz quad-core CPU and 16GB memory and the results are compared against our proposed model.

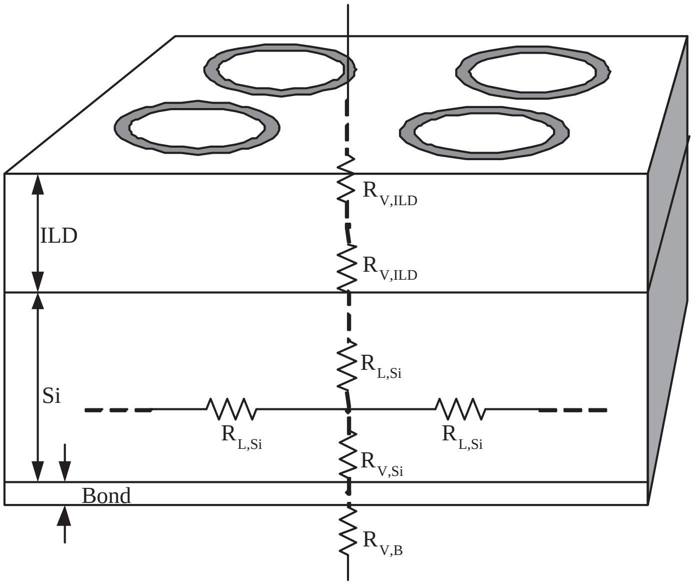


Figure 3.7: Extending the proposed lateral model into more complicated TSV structures

3.4.1 Experimental results of a TSV cell

Experimental setup

To measure the lateral thermal resistance using COMSOL, we define a structure as shown in Fig. 3.8. A power source of $10^{-4}W$ is on the front face (top) of the structure and the convection surface with heat exchanging rate of $1000W/(m^2K)$ is on the back face (bottom). The back face is the only face that exchanges heat with the surroundings. All the other faces (left and right faces) have no heat exchange with the environment, and thus the lateral thermal resistance $R_{Therm,L}$ of the entire structure in the direction of thermal flow can be measured by:

$$R_{Therm,L} = \frac{\Delta T}{q} \tag{3.21}$$

where ΔT is the temperature difference across the structure (from the front face to the back face), and q is the corresponding power flow.

The structure can be divided into 3 blocks where block1 and block3 are the silicon blocks, and block2 is the TSV cell block sandwiched between block1 and block3 as shown in Fig. 3.8. The figure also shows the isothermal curves from COMSOL simulation where the direction of thermal flow is represented by short arrows perpendicular to the isothermal curves. It can be seen that the isothermal curves are bent close to the location of TSV and the majority of the isothermal curves are at the location of the TSV overlapping with the insulation layer. This shows that most of the heat flow is perpendicular to the insulation layer which is in line with the assumptions made in the Section 3.2. In our experiments, the radius of the TSV (from the center to the outer interface between silicon and insulator) is fixed at $r = 25um$ and the height of the structure is fixed at $h = 50um$. The thickness of the insulation layer varies from $0.5um$ to $2.5um$. The width of the silicon region, denoted by rP (r is the TSV radius, P is the ratio of TSV space to TSV radius) in Fig. 3.8, varies from 2 to 6 times the TSV radius. Physically, the value of P represents the space between the TSVs in a TSV array built with this TSV cell. We study how the variation of P influences the accuracy of the proposed model for a TSV cell.

The lateral thermal resistance of the TSV cell can be measured as:

$$R_{TSVcell,M,L} = \frac{\Delta T}{q} - R_{SerSi} \tag{3.22}$$

where $R_{TSVcell,M,L}$ is the measured lateral thermal resistance of the TSV cell, and R_{SerSi} is the lateral thermal resistance of the silicon region in the thermal flow direction. This thermal resistance is in series with the thermal resistance of the TSV cell. Here, $R_{SerSi} = 2\rho_{Si}/W$

represents the series connection of the two silicon blocks where the thermal resistance of each one is denoted as ρ_{Si}/W. W is the width of the blocks in the thermal flow direction and this is shown in Fig. 3.8. In the following subsection, the calculated lateral thermal resistance is compared against the measured data based on FEM results to validate the proposed closed form thermal resistance model.

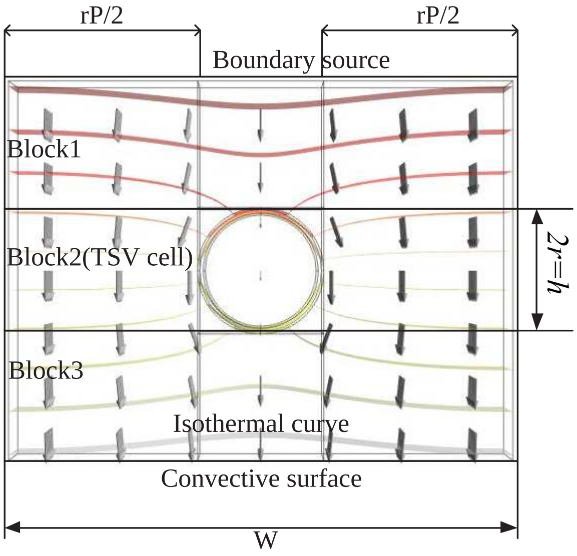


Figure 3.8: Isothermal curve of the TSV cell under test

Result analysis and discussion

We first simulate the steady state temperature response of the structure using COMSOL. The lateral thermal resistance can be measured using equation (3.22). The modeled lateral thermal resistance before calibration can be calculated directly based on the material and geometry of the structure using equation (3.9). To determine the coefficient of the calibration factor (β_1 and β_2), we need to use the measured thermal resistance data of at least two TSV cells with different P. The values of P we choose for model calibration are $P_1 = 2$ and $P_2 = 6$. To reduce the calibration error under different values of insulation layer thickness ranging from $0.5um$ to $2.5um$, we choose the measured data of the structure with insulation layer thickness as $1.5um$, which is in the middle of the range: $0.5um$ to $2.5um$. Thus, the calculated coefficients of the linear function θ is computed as: $\beta_1 = 0.0279$, $\beta_2 = 0.956$.

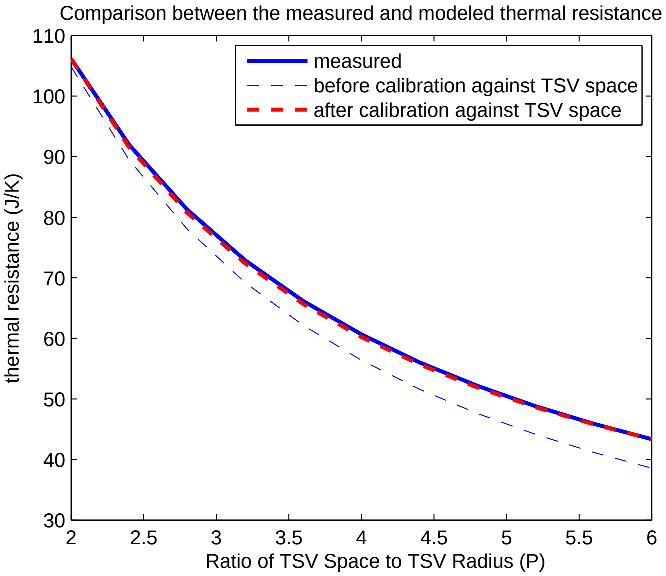


Figure 3.9: The measured data used for calibration (the TSV with the insulation linear to be $1.5um$ vs the proposed model before and after calibration

In both Fig. 3.11 (a) and (b), the solid line shows measured lateral thermal resistance, the finer dash line shows the calculated result from the model before calibration, and the bond dash line shows the calculated result from the calibrated model. Our observations are as follows:

1. The model before calibration deviates from the measured data because of the modified isothermal lines introduced by the presence of TSVs.

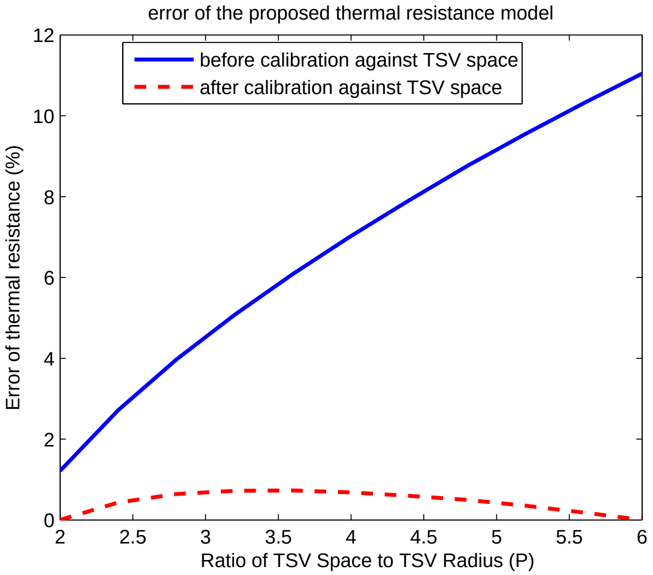
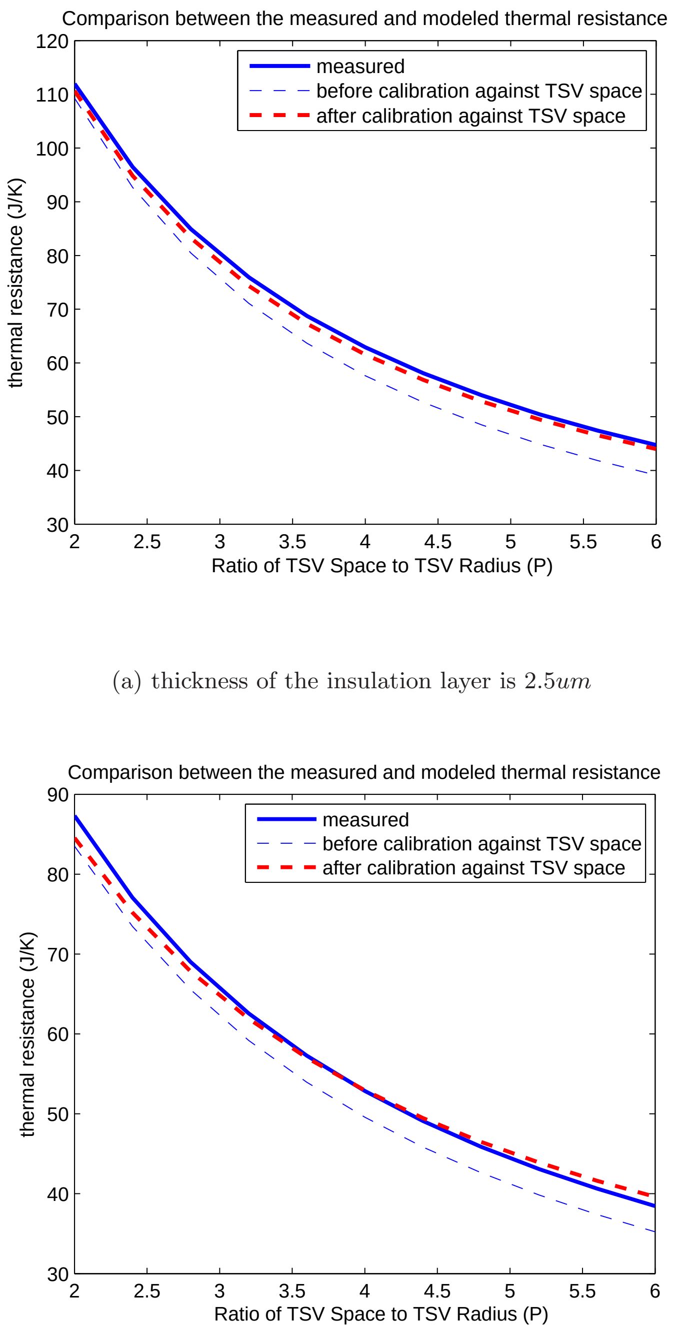


Figure 3.10: Error of the proposed model (before calibration and after calibration) for the TSV with the insulation linear thickness to be $1.5um$

2. Model calibration significantly increases the accuracy of the modeled thermal resistance in the targeted range of TSV space.
3. Both the thickness of the insulator liner and the space between TSVs influence the lateral thermal resistance. Thicker the insulator liner leads to higher lateral thermal resistance in general, and larger space between TSVs causes lower lateral thermal resistance.

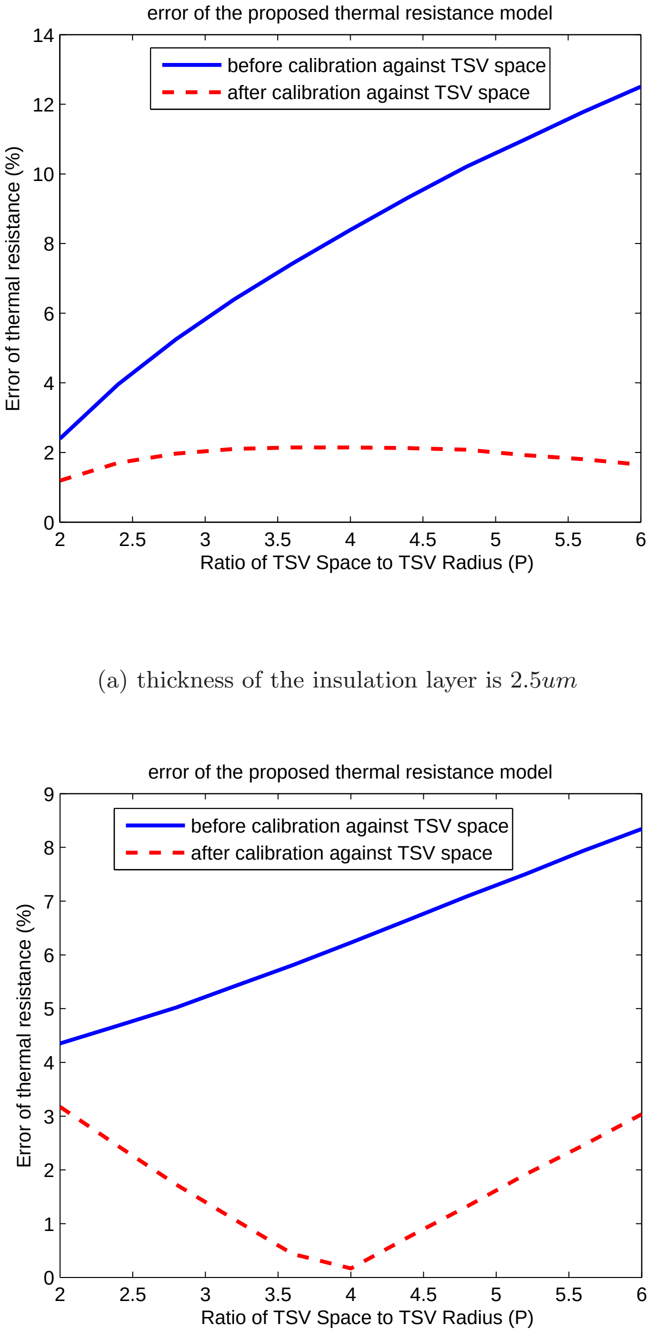
The relative model error with different TSV space and different insulation liner thickness is shown in Fig. 3.12. It indicates that without calibration, the error keeps increasing as the space between the TSVs increases. After applying the calibration, the model error reduces significantly and the smallest error is found for the structure that has insulation layer thickness of $1.5um$. This is because the calibration is directly applied to the measured data of this type of structure.



(a) thickness of the insulation layer is $2.5um$

(b) thickness of the insulation layer is $0.5um$

Figure 3.11: Comparison of the proposed model and the measured data in different insulation thickness



(a) thickness of the insulation layer is $2.5um$

(b) thickness of the insulation layer is $0.5um$

Figure 3.12: Error of the proposed model (before calibration and after calibration)

For structures with different insulation layer thickness on TSVs, the error increases and the maximum error is found to be 2% to 3%, which is an acceptable error margin. This result confirms that variations in the insulation layer thickness of TSVs (of 2% to 10% of TSV radius) does not bring in significant error to the model. Thus, it is not necessary to calibrate the model for various value of this parameter in most practical cases. Hence, once the lateral thermal resistance model of a TSV cell is calibrated with different P, it can be reused to build the model for any TSV array.

In addition, we remark that the existing lumped modeling approach presented in [18] does not consider the space between two TSVs and thus the model accuracy compromises under the cases when the space between two TSV changes. Fig.3.13 shows a comparison of the proposed modeling method with the existing lumped modeling method, which clear illustrates this problem.

3.4.2 Experimental results for TSV arrays

Experimental setup

Having validated the lateral thermal resistance model for a TSV cell in the previous section, we now compute the lateral thermal resistance in the thermal flow direction of a TSV array. For this purpose, we consider two examples: a 1D TSV array and a 2D TSV array as shown in Fig. 3.14 and compare the measured lateral thermal resistance against the modeled value for both the arrays. The heat source and the convection boundary is set up in the same way as before: $10^{-4}W$ at the front face and $1000W/(m^2K)$ at the back

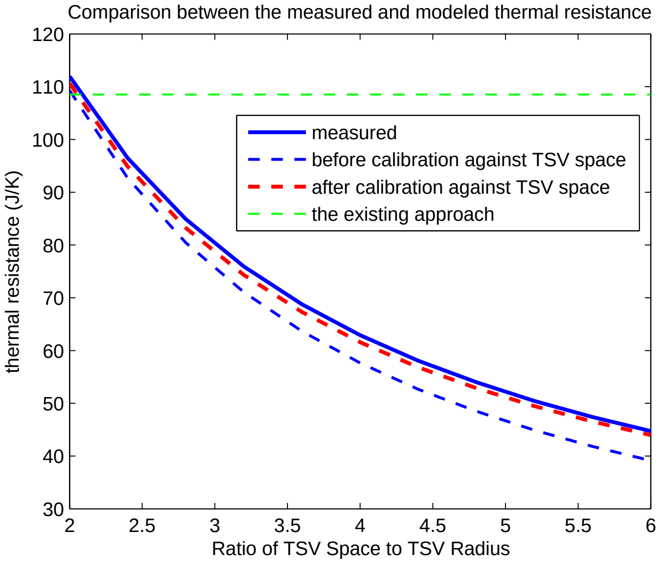


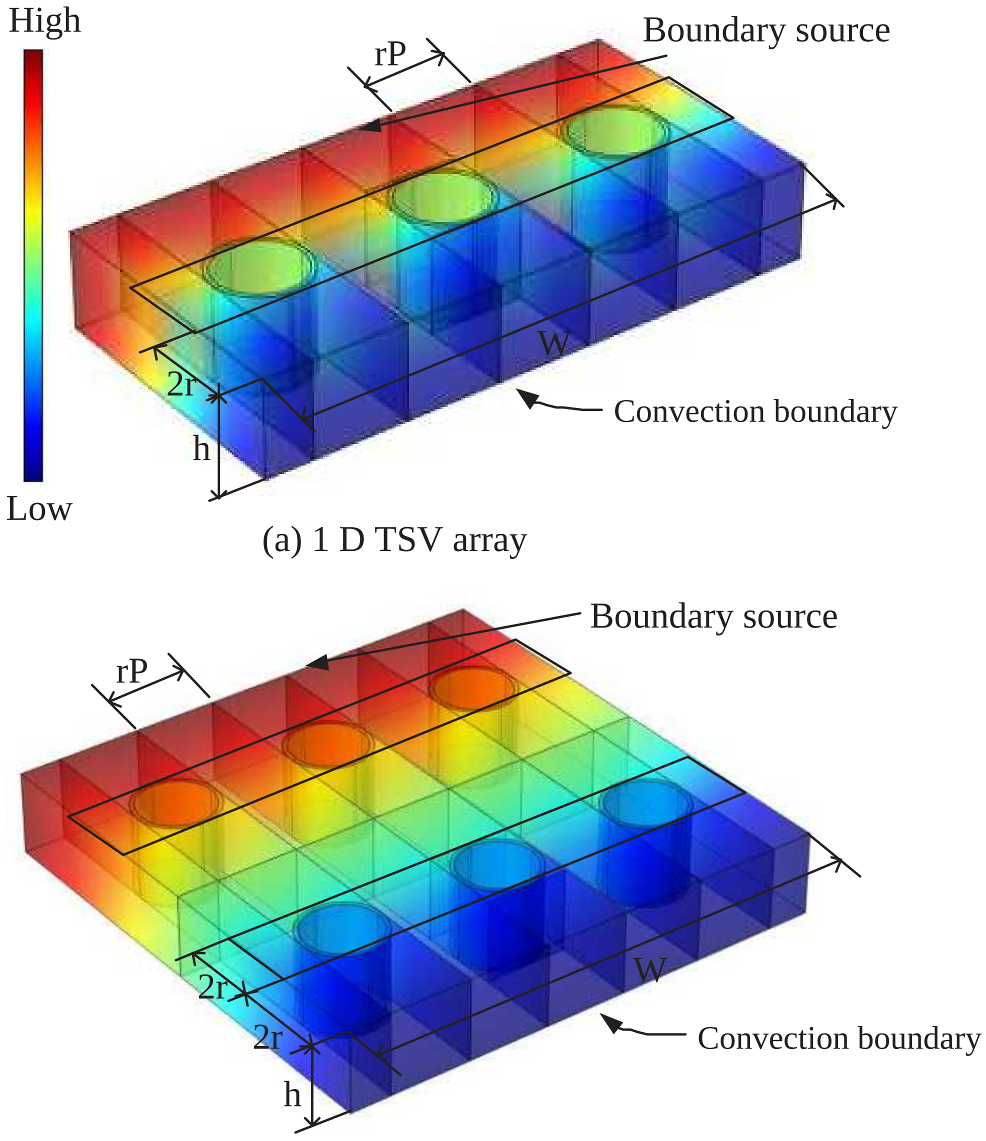
Figure 3.13: Comparison of the proposed modeling method with the existing lumped modeling method

face. COMSOL 4.1 is used to simulate the temperature profile of the TSV array and that is compared against the lateral thermal resistance modeled by (3.22).

Result analysis and discussion

The results for the TSV array are obtained for a TSV radius of $r = 25um$, insulation layer thickness of $\delta = 2.5um$ (10% of the TSV radius) and a block height of $h = 50um$. The results for the 1D and 2D arrays are summarized in Table 3.1 and Table 3.2 respectively. The lateral thermal resistance is measured under different scenarios by varying the space between TSVs from $50um$ to $150um$. This corresponds to the ratio of TSV space to radius (denoted as P) of 2 to 6.

For the 1D TSV array shown in Fig. 3.14 (a), Table 3.1 shows the measured lateral thermal resistance. The lateral thermal resistance is modeled using equation (3.22)



(a) 1 D TSV array

(b) 2 D TSV array

Figure 3.14: TSV array structures used for lateral thermal resistance validation

with $R_{Si,L} = 2\rho_{Si}/W$, where W is the width of the TSV array as indicated in Fig. 3.14 (a). Since the TSV array is a parallel combination of 3 TSV cells, the modeled lateral thermal resistance for this array can be calculated as: $R_{TSVcell,cal,L}/3$ where $R_{TSVcell,cal,L}$ is calculated using equation (3.10). The results in Table 3.1 show that the modeled thermal resistance closely matches the measured data for the 1D TSV array.

Table 3.1: Comparison of modeled and measured lateral thermal resistance of 1D TSV array

P	Measured(K/J)	Modeled(K/J)	Error
2	37.30	36.87	1.15%
4	20.96	20.66	1.43%
6	14.91	14.66	1.68%

Table 3.2 shows the measured total lateral thermal resistance of the 2D array as indicated by the enclosed region in Fig. 3.14 (b). The lateral thermal resistance is modeled using equation (3.22) where $R_{Si,L} = 4\rho_{Si}/W$ and W is the width of the TSV array as indicated in Fig. 3.14. The space between the two 1D TSV arrays in Fig. 3.14 lines up with the direction of the thermal flow and it does not influence the lateral thermal resistance because it does not change the lateral isothermal lines that are perpendicular to the thermal flow. Hence, the modeled resistance for the 2D array is $2R_{TSVcell,cal,L}/3$, as it is a series combination of two 1D arrays shown in Fig. 3.14 (a). The results show that the modeled thermal resistance closely matches the measured data for the 2D TSV array as well.

Table 3.2: Comparison of modeled and measured lateral thermal resistance of the 2D TSV array

P	Measured(K/J)	Modeled(K/J)	Error
2	74.62	73.74	1.18%
4	41.35	41.32	0.07%
6	28.81	29.32	1.68%

As mentioned before, we expected that our proposed model is applicable to TSV of any arbitrary radius in silicon layer of different thickness. To test this, we perform the following changes to the array structure in Fig. 3.14.

1. Change the radius of TSV (r) to 10 um, and the thickness of the insulation liner (δ) to 1 um.
2. Change the thickness of silicon layer (h) to 10um, and keep the ratio of TSV space to TSV radius to be 3, 4, and 5 (set the space between 2 adjacent TSVs to be 30 um, 40 um 50 um) respectively.

We repeat the experiment to measure the lateral thermal resistance contributed by the 2D TSV array, and compare the data with the calculated model. In the procedure of calculating the modeled lateral thermal resistance, we use the same calibration parameter β_1 and β_2 obtained from our previous experiments to test their validity in a new structure. The result summarized in Table 3.3 confirms that the calculated thermal resistance closely matches its measured counterpart obtained from fine finite element simulation, which indicates the robustness of our proposed closed form model.

Table 3.3: Comparison of modeled and measured lateral thermal resistance of the modified 2D TSV array

P	Measured(K/J)	Modeled(K/J)	Error
3	269.1	261.7	2.75%
4	207.72	205.20	1.2%
5	168.94	170.30	0.8%

3.4.3 Experimental results of finite difference simulation

Finally, we demonstrate how the proposed model be used in finite difference thermal modeling method to improve modeling accuracy without incurring computation cost. As explained before, fine mesh grid modeling method is impractical for the thermal modeling of large scale TSV arrays. However, to demonstrate the accuracy improvement of our proposed method, it is very necessary to have golden results to compare with. In our experiment, we use COMSOL to build a scaled down chip structure with reduced number of TSV structures on it so that the fine grid finite element simulation tool could handle it and produce the thermal data that we could compare with. In this section, we use COMSOL to simulate two-layer stack structures connected by TSV arrays with different array dimensions.

Temperature responses of a 4×8 TSV array

The first structure used for temperature response test is shown in Fig. 3.15, which represents part of a 3-D stacked chip structure that could be built and simulated by COMSOL without incurring memory issue. Both the lateral dimensions of Si layer and the Al layer are 200 um by 500 um, and the thickness of Si layer is 10 um, while the thickness of Al layer is 20 um. The two layers are connected by a TSV array consists of 32 TSVs (4×8 array) in total as shown in Fig. 3.15. The radius of each TSV is 10 um, the insulation liner thickness is 1 um, and the space between two adjacent TSVs is 30 um. The convective coefficient of the top surface of the Al layer is set to be 10000 $W/(m^2K)$ and 50000 $W/(m^2K)$ as shown in Fig. 3.15 to model the different convective cooling effects at different location of the chip package. We assume that the majority of the power sources should be located outside the TSV area, while inside the TSV area, the power density is relatively lower, creating energy flow in lateral direction that will be manifested by the lateral thermal resistance across the TSV arrays. In our setup, the step power sources of 3e6 W/m^2 are placed at the bottom of the silicon layer where there is no TSV arrays, and 1e6 W/m^2 power sources are placed at the area inside the TSV array as shown in Fig. 3.15. In this way, we could testify the lateral thermal effect of the TSV arrays in 3D finite difference simulation using this reduced structure.

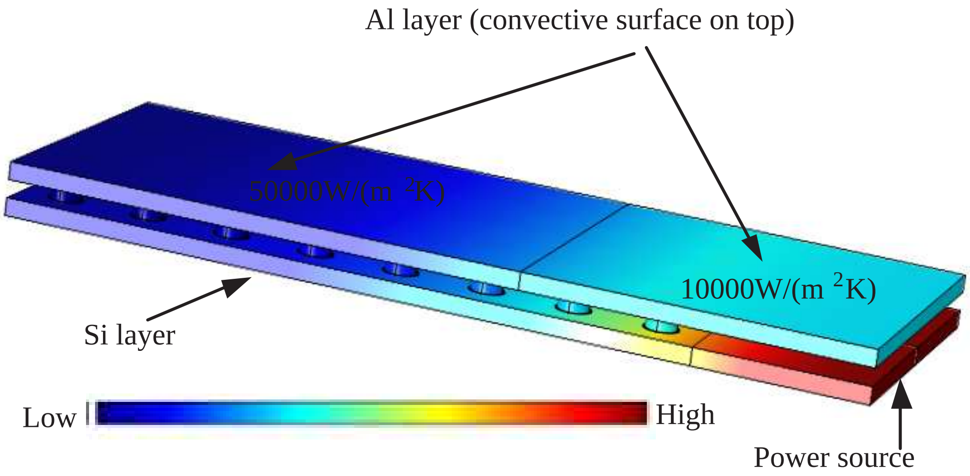


Figure 3.15: Two layer chip structure with 4×8 TSV array and the temperature distribution simulated by COMSOL

In this experiment, we compare the simulated temperature response from COMSOL with the one predicted by our proposed model, and we also compare the results from our proposed model against the existing one in [18]. In the finite difference method, coarse mesh grids are used to enable fast simulations, the bottom view of the 2 by 5 mesh grid is shown in Fig. 3.16. We remark that fine grid simulation could capture every details of the temperature gradient in the structure, however, the coarse grid method could only capture the average temperature of one mesh grid that has no TSVs in it, and capture the average temperature of TSV cores for the grid with TSVs located inside. Normally, we are concerned with predicting the temperature of the locations where no TSVs present, because these are the locations where the temperature tends to exceed the alarming rate due to the less effective heat removal in these areas. Thus, we compare the temperature responses of the silicon mesh grid.

The transient step temperature responses of one silicon mesh grid from different methods are plotted in Fig. 3.17. Clearly, we could see that the proposed model yield more accurate results because of its more accurate lateral thermal resistance model. The existing model does not consider the space between TSVs and thus overestimates the lateral thermal blockage effect, leading to 3.08 Celsius higher temperature prediction, while the proposed approach leads to only 0.6 Celsius deviation above the COMSOL simulation results.

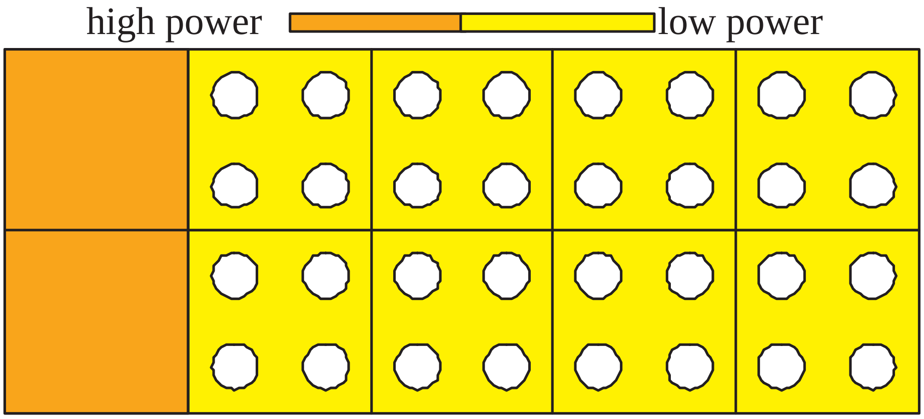


Figure 3.16: 2D bottom View of the 2 × 5 meshed chip and its power source configuration

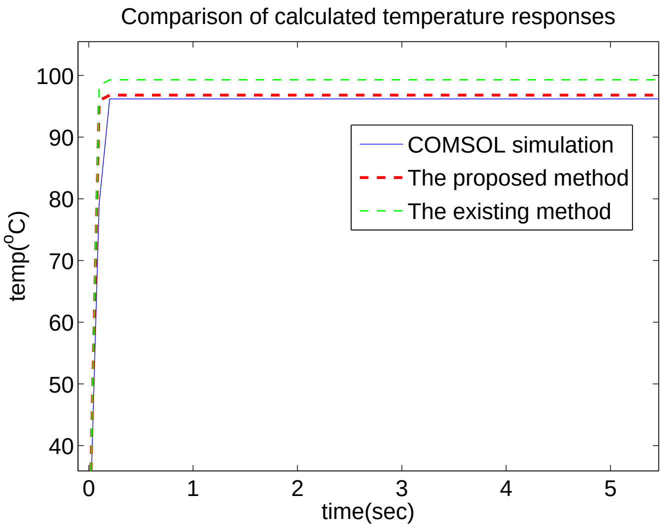


Figure 3.17: Transient temperature responses of the two layer structure with 4×8 TSV array

To clarify the application scope of the proposed modeling method, we remark that the lateral thermal flow is important to manifest the effect of the lateral thermal resistance. If the lateral thermal flow is not significant, the effect of lateral thermal resistance will be reduced, and the proposed model will have limited accuracy gain in this case. To proof this, we reconfigure the power density, by swapping one high power source (3e6 W/m^2) on the silicon grid with its adjacent low power source (1e6 W/m^2) on a TSV grid. In this way, the majority of the heat generated by the high power source inside the TSV grid will flow vertically, because much less lateral thermal gradient is created under this power source configuration. Fig. 3.18 shows the comparison between the existing method and the proposed method for the temperature of the TSV grid that has high power source, which confirms that the proposed method does not have much accuracy gains in cases like this.

However, normally, the transistor devices needs to be placed away from TSVs for reliability concerns, and thus, the power in the array of TSV array should be lower, in

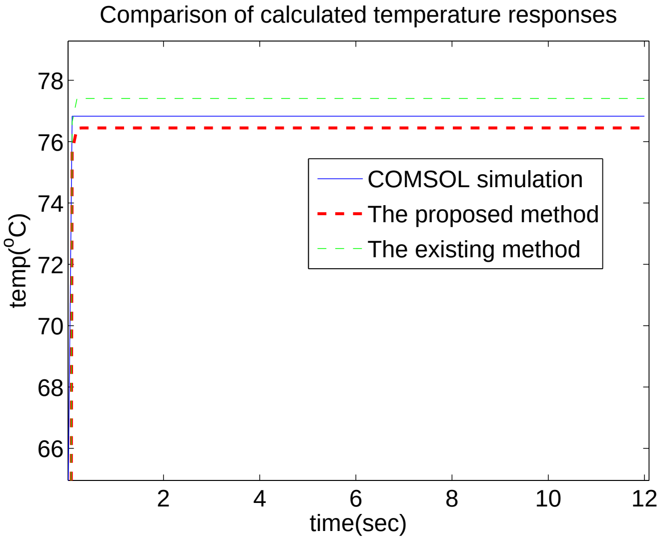


Figure 3.18: Transient temperature responses of TSV grid in the two layer structure with 4×8 TSV array under reconfigured power sources

which case large lateral thermal gradient would be formed, and the proposed model needs to be used. In some application cases, such as representing the TSV arrays for wide I/O memory in 3D ICs, it is reasonable to assume that heat sources locate at the boundary of the TSV array, but not inside the array because the power of wide I/O memory is orders of magnitude lower than that of the arithmetic processor. And this is also the case when the lateral heat flow across the TSV array is maximized, and thus the influence of the lateral thermal resistance across the TSV arrays need to be carefully modeled. To represent this case, we reuse this structure with 4×8 TSV array, and place $5 \times 10^6 W/m^2$ power sources at the bottom of the silicon area where there is no TSV arrays, while inside the silicon area of TSV arrays, the source is set to be 0 W/m^2. In this case, the transient step temperature responses of one silicon mesh grid from different methods are plotted in Fig. 3.19. The existing model leads to 4.11 Celsius higher temperature prediction comparing with the COMSOL simulation results, which clearly demonstrates increased error (comparing with error of 3.08 Celsius in the previous test case) because the accuracy of the lateral thermal resistance of the TSV array displays more significant effect due to the increased lateral thermal flow in this configuration, while the error of the proposed approach is just 0.44 Celsius.

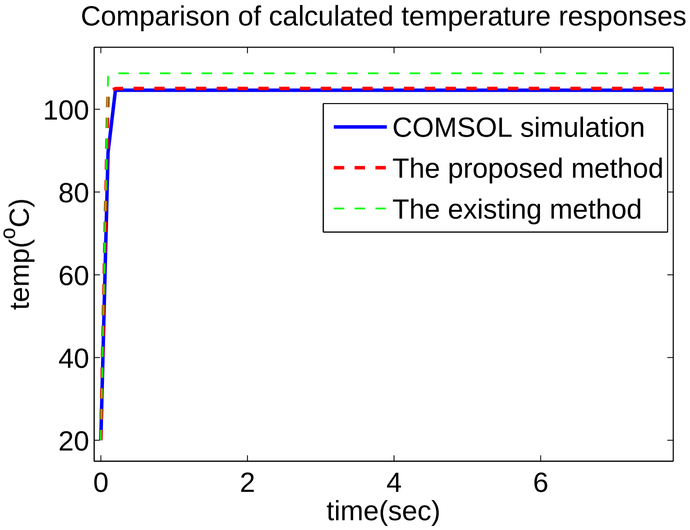


Figure 3.19: Transient temperature responses of the two layer structure with 4×8 TSV array representing wide I/O memory

In both examples, the modeling and simulation time (using 120 time steps) of the proposed compact model is only 0.79 seconds, while the COMSOL simulation takes 2468 seconds, which is more than 3000 times longer, and thus unaffordable if simulating even larger structures. Hence, building compact model using accurate closed form lateral TSV model is very important to capture the thermal response of 3D chip with large numbers of TSVs.

Temperature responses of a 4×10 TSV array

We remark that the small difference between the existing model and the proposed model on a reduced dimension structure may accumulate to more significant temperature differences in larger structure with much larger scale of TSVs and TSV arrays, which could not be built and simulated by fine grid simulation tools like COMSOL. However, to demonstrate this trend, we build a larger structure with 4×10 TSV arrays to show how the lateral thermal effect would be accumulated over the distances, and lead to increased temperature error if not properly considered. The structure built in COMSOL with 4×10 TSV array is shown in Fig. 3.20, in which the dimension of TSVs are still the same, the dimension of both Si layer and Al layer is $20um \times 60um \times 10um$ to cover the 4×10 TSV arrays.

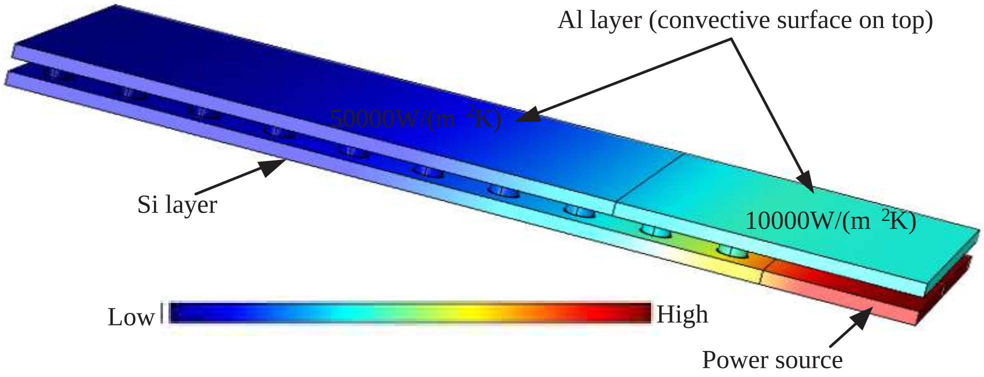


Figure 3.20: Two layer chip structure with 4×10 TSV array and the temperature distribution simulated by COMSOL

We assume that the TSV array represents low power wide I/O memory as it is the case when modeling lateral thermal resistance is more demanding. Like in the previous test case of wide I/O memory, we place $5 \times 10^6 w/m^2$ sources at the bottom silicon layer outside of the area covered by TSV array.

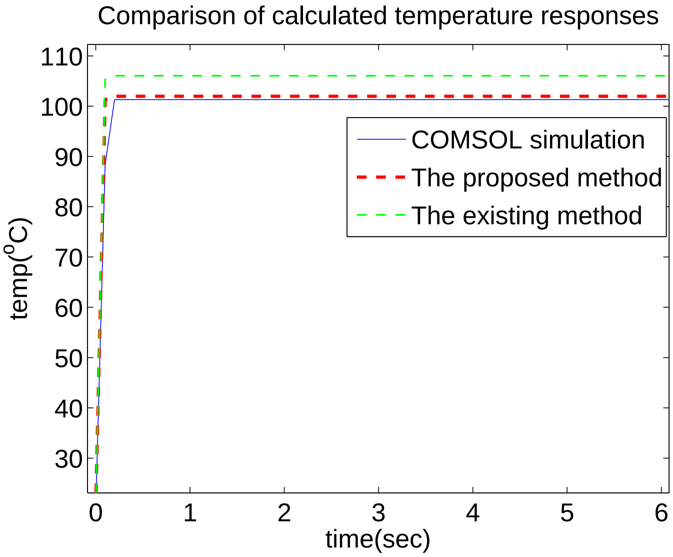


Figure 3.21: Transient temperature responses of the two layer structure with 4×10 TSV array representing wide I/O memory

The simulated temperature response is shown in Fig. 3.21. The result using the existing model shows 4.75 Celsius differences away from the COMSOL simulation, which is about 0.64 Celsius error increment comparing with the wide I/O memory testing case with 4×8 TSV arrays (4.11 Celsius error) in the previous sub-section, while the result using proposed model shows only 0.66 Celsius deviation away from the COMSOL simulation.

3.5 Summary

In this chapter, we have proposed a simple yet accurate physics-based compact thermal resistance model for lateral TSV thermal resistance. Our research shows that lateral TSV thermal resistance model is not only a function of the geometry of the TSV such as radius and thickness of liner, but also strongly depends on the space between TSVs because of changes in the isothermal curves when TSVs are placed at different locations with respect to each other. With the new lateral TSV thermal resistance model, one can build more compact grids for finite difference analysis that considers the effect of TSVs. The compact models can be also used for chip level thermal analysis and optimization. Experimental results show that the proposed TSV model, which is a function of the TSV geometry and space between TSVs, can lead to very small errors compared to the detailed numerical analysis, and thus improve the accuracy of finite difference thermal simulation.

Chapter 4

Subspace-based Compact Nonlinear Thermal Modeling of Packaged Integrated Systems

In this chapter, a new behavioral thermal modeling technique for high-performance microprocessors at package level is proposed. Firstly, the new approach applies the subspace identification method, which is reviewed in section 2.2 of Chapter 2, with the consideration of practical power maps with correlated power signals. We show that the input power signal needs to meet an independence requirement to ensure the model predictability and propose an iterative process to build the models with given error bounds. Secondly, we show that thermal systems fundamentally are nonlinear and then propose a piecewise linear (PWL) scheme to deal with nonlinear effects. The new piecewise linear models can model thermal behaviors over wide temperature ranges or over different thermal boundary convective conditions due to different fan speeds. Further, the PWL modeling technique can lead to much smaller model order without accuracy loss, which translates to significant savings in both the simulation time and the time required to identify the reduced models compared to the simple modeling method by using the high order models.

Experimental results validate the proposed method on a realistic packaged integrated system modeled by the multi-domain/physics commercial tool, COMSOL, under practical power signal inputs. The new piecewise linear models can model thermal behavioral over wide temperature ranges and different thermal boundary convective conditions due to different fan speeds. Further, the PWL modeling technique can lead to much smaller model order without accuracy loss, which translates to significant simulation time reduction and about 10X less time to identify the reduced models compared to the simple modeling method by using the high order models in our examples.

This chapter is organized as follows: Section 4.1 presents the thermal modeling problem we are trying to solve. Section 4.2 presents the new power-map based thermal modeling technique. Section 4.3 gives the new nonlinear thermal modeling technique. Finally, section 4.4 presents the experimental results of both *ThermSubCP* and *ThermSubPWL*, with section 4.5 concluding the chapter.

4.1 Thermal modeling problem considering power maps

We first present how the power inputs are modeled in our problem. A microprocessor chip is partitioned into $p = n \times m$ power grids as shown in Fig. 4.1, where each square power grid has a power source as an input and the measured temperatures at its adjacent 4 corners as outputs. We can abstract this power grid model into a discrete multi-input and multi-output (MIMO) system with $p = n \times m$ power inputs and q temperature outputs as shown in Fig. 4.2. The $n \times m$ power input distribution at one time instance is defined as a *power map*, which can be measured or obtained by simulations or other methods practically.

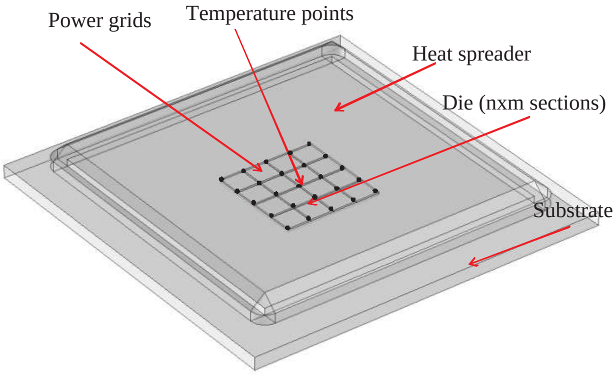


Figure 4.1: Meshed chip and package

In general, the abstracted p-input and q-output thermal system could be represented as

$$\begin{aligned} x(t+1) &= F(x(t), u(t)), \\ y(t) &= G(x(t), u(t)), \end{aligned} \tag{4.1}$$

where $F(x)$ and $G(x)$ both are nonlinear vector functions of state variable vector $x(t)$ and input signal vector $u(t)$. In our problem, the input vectors $u(t) \in \mathbb{R}^p$ are the measured power input traces and output vectors $y(t) \in \mathbb{R}^{q1}$ are the temperature responses.

Existing approaches typically assume that thermal system in Fig. 4.1 is linear. As a result, (4.1) can be rewritten as the standard linear state transition form:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{4.2}$$

where $A \in \mathbb{R}^{l\times l}$ is a stable matrix, l is the number of states. $B \in \mathbb{R}^{l\times p}$, $C \in \mathbb{R}^{q\times l}$, and $D \in \mathbb{R}^{q\times p}$. With p inputs $u(t_i)$ and q outputs $y(t_i)$ where $i = 1, 2, \ldots, s$, the problem now becomes finding state matrices A, B, C, and D, where D is typically considered as a matrix of zeros. $x(t)$ are the Kalman state vector and it does not have different physical meaning.

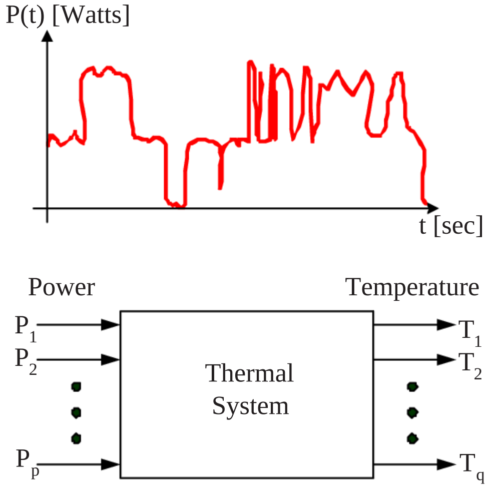


Figure 4.2: The abstracted model system and correlated power inputs

Also, the $n \times m$ power inputs may be highly correlated as mentioned before. In an extreme case, all the power input waveforms are exactly the same and they are only different in magnitude. Fig. 4.2 (top) shows a typical power input waveforms. Their spatial difference in magnitude essentially is described by the *power map* of the chips. The magnitude (power map) distribution can be defined by a function and applied in a practical setting to an artificial testing package (called *testing vehicle*). Such magnitude distribution is called *power map configuration* in this chapter.

Such highly correlated power inputs, however, will lead to poor predictability when using the subspace identification method [41, 73]. Fig. 4.3 shows the waveforms produced by the model identified with highly correlated power inputs (one power map configuration), where the results from model and from original temperature does not match well. To overcome this problem, subspace identification procedure using multiple power map configurations is proposed to identify the thermal package model [38], and it works well when the system is linear and can be described by (4.2).

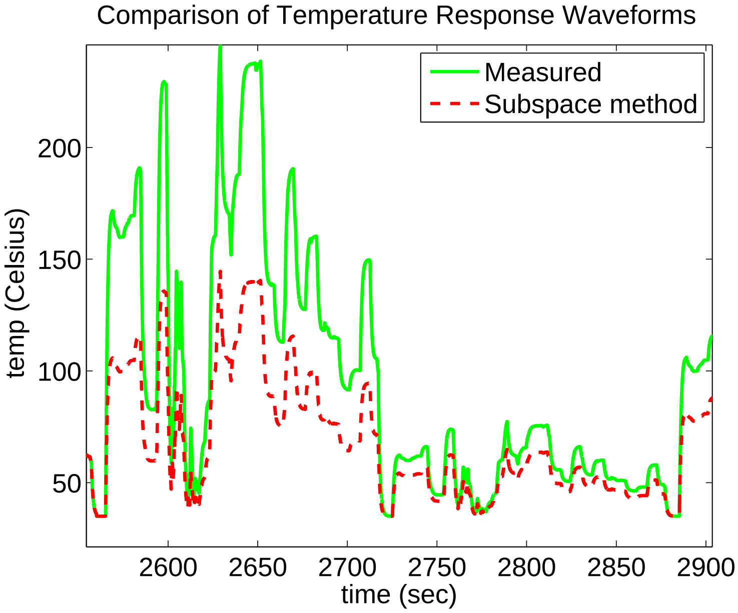
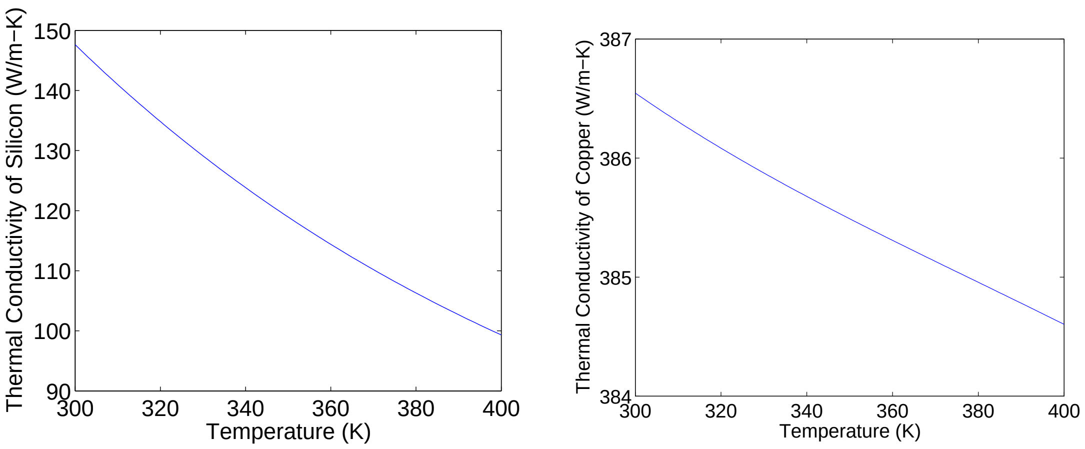


Figure 4.3: Transient temperature response of the system identified with one highly correlated power inputs

The second issue we are facing is that thermal behavior of packaged electronic systems will show weakly nonlinear effects due to the temperature-dependent properties of the packaging materials [86]. Fig. 4.4 shows the temperature dependence of thermal conductivity of Si and Cu. Fig. 4.5 shows if we excite a chip package system shown in Fig. 4.1 with a sinusoid power input, we can clearly observe the harmonic components in the discrete Fourier transform of the transient temperature response, which evidently indicates the nonlinearity of the underlying thermal system although the nonlinear components are mild and weak. Such mild nonlinear behaviors, however, can still lead to significant loss of accuracy as shown in Fig. 4.6 if typical low order is used. In addition, the potential change of external cooling condition need to be modeled as variable thermal resistance, and its variation possibly could lead to even stronger nonlinearity, making pure linear modeling approach more difficult.

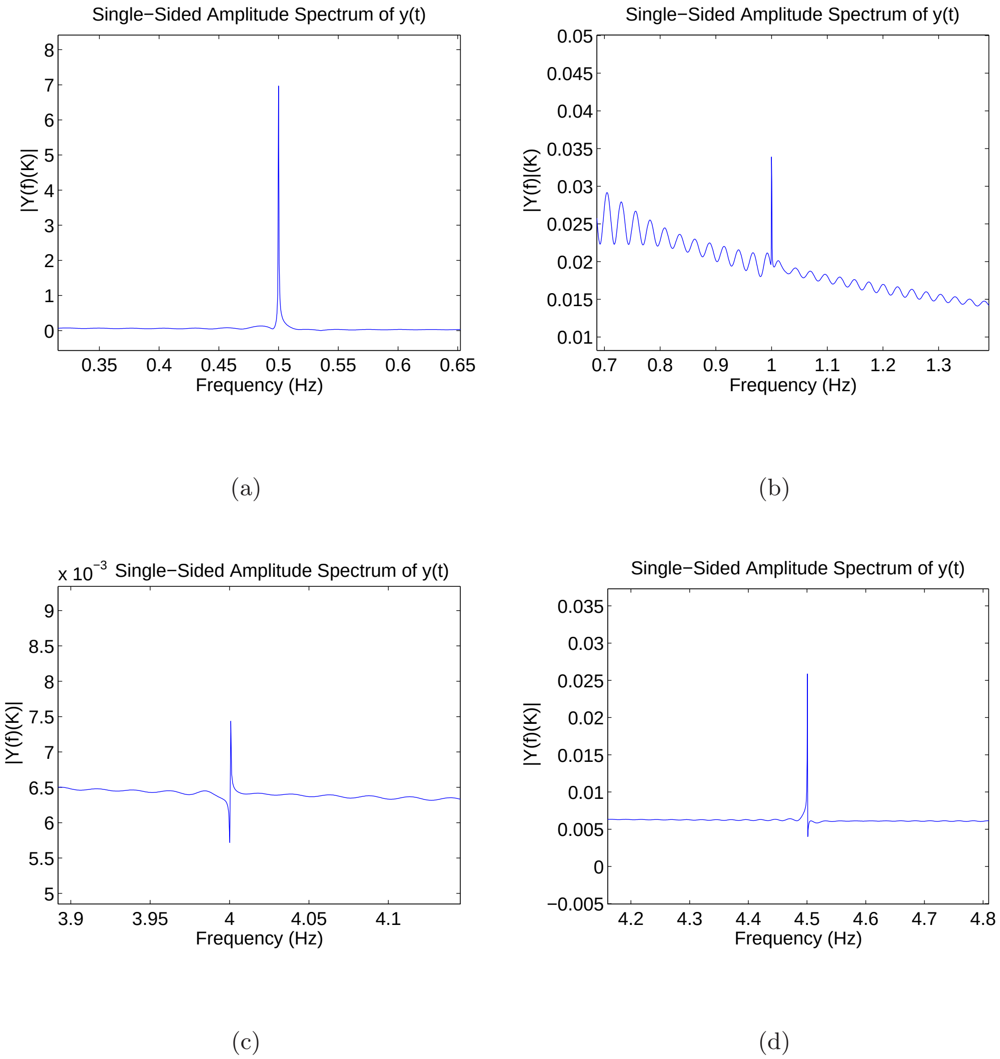


(a)

(b)

Figure 4.4: Temperature dependence of the thermal conductance (a) Silicon (b) Copper

To mitigate this problem, we propose to use linear models to represent the thermal behavior of the packaged electronic systems under different ranges of thermal conditions (piecewise linear model approach). As a result, significant accuracy improvement is achieved by using just low order models that would significantly save the computation cost and facilitate fast thermal analysis.



(a)

(b)

(c)

(d)

Figure 4.5: Frequency domain response of the thermal system under sinusoid input with frequency of 0.5HZ (a) baseband spectral (b) 1st order harmonics (c) 2nd order harmonics (d) 3rd order harmonics

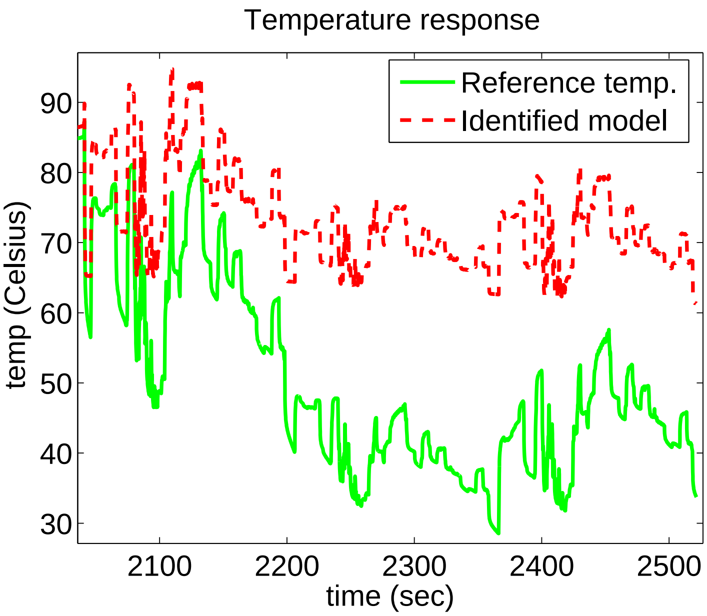


Figure 4.6: Accuracy loss of the temperature response of the identified 4-th order linear model

4.2 Power map based thermal modeling method for correlated power inputs

4.2.1 Spatial rank requirement

In general, the *persistently exciting*, or PE condition can be easily satisfied for a MIMO dynamic system when all the transient input signals are uncorrelated spatially. However, if those signals are highly correlated spatially as in the case of power maps obtained from the measurement, the PE condition may not be easily satisfied, which leads to poor predictability of the resulting models as shown in Fig. 4.3.

Consider a 2-input system as an example. We assume that all the inputs have exactly the same time domain waveform and denote it as $f(t)$. The differences in magnitude are represented by another spatial function $g(x)$ in 1-D space (x-axis) for simplicity, which represents the 1-D power map configuration. The i-th input sample for such 2-input system is

$$u(i) = \begin{bmatrix} g(x_0)f(t_i) \\ g(x_1)f(t_i) \end{bmatrix}. \tag{4.3}$$

We further define the i-th block row in the input Hankel matrix as

$$U_i = \begin{bmatrix} g(x_0)f(t_i) & g(x_0)f(t_{i+1}) & \cdots & g(x_0)f(t_{i+N-1}) \\ g(x_1)f(t_i) & g(x_1)f(t_{i+1}) & \cdots & g(x_1)f(t_{i+N-1}) \end{bmatrix}. \tag{4.4}$$

The input Hankel matrix can be expressed as

$$U_{0|2k-1} = \begin{bmatrix} U_0 \\ U_1 \\ \vdots \\ U_{2k-1} \end{bmatrix}. \tag{4.5}$$

The *persistently exciting* condition is satisfied when $U_{0|2k-1}$ has full row rank, that is $\text{rank}(U_{0|2k-1}) = 4k$ for this 2-input case. However, it is clear that the two rows in U_i are linearly dependent, which makes $\text{rank}(U_{0|2k-1}) = 2k$ and fails to satisfy the *persistently exciting* condition.

In order to make the input Hankel matrix $U_{0|2k-1}$ full row rank, we need to make the i-th block row U_i full row rank, assuming $N \gg k$. For this 2-input example case, we can achieve this by simply introducing another power map configuration. Now we have two configurations, g_1 and g_2. The i-th block row U_i is shown in (4.6), where $i < m < i + N - 1$.

$$U_i = \begin{bmatrix} g_1(x_0)f(t_i), ..., g_1(x_0)f(t_{i+m}), g_2(x_0)f(t_{i+m+1}), .., g_2(x_0)f(t_{i+N-1}) \\ g_1(x_1)f(t_i), ..., g_1(x_1)f(t_{i+m}), g_2(x_1)f(t_{i+m+1}), .., g_2(x_1)f(t_{i+N-1}) \end{bmatrix} \tag{4.6}$$

The two dimensional case can be generalized into higher dimensions with g_i as a function of two spatial variables x and y. By calling the row rank of i-th block row U_i the *spatial rank* of input signals and assuming we have sufficient samples (N is large enough), we have the following proposition.

Proposition 1 *For p-input MIMO dynamic systems with correlated input signals, the spatial rank of input signal matrix, U_i, must be equal to p to satisfy the persistently exciting condition in the subspace identification method.*

We remark that for real microprocessor with package, we can feed the microprocessors with different programs, and those programs may generate different power maps to meet the PE conditions. For thermal characterization based on *testing vehicle*, we have the flexibility to generate different power maps on the testing chip. In this case, we propose to automatically generate multiple *independent* power map configurations to meet such spatial rank requirement for input signals of subspace identification method. in the next part, we show how this can be achieved.

4.2.2 Orthogonal set of power map configurations

In this subsection, we show how to automatically generate independent power map configurations to meet the PE requirement as mentioned in the previous subsection. This process is useful since the number of inputs can be large, and it also provides a guideline for generating practical power maps for the proposed top-down method.

Take the 1-D example again, if $x \in [0, L]$, a set of orthogonal functions over the interval $[0, L]$ can serve as the systematic solution for the independent and robust configurations. By definition of orthogonality, the 1-D orthogonal function set $\{g_1, g_2, \ldots, g_N\}$ satisfies

$$\int_0^L g_i(x)g_j(x)\,dx = \begin{cases} 0 & i \neq j \\ \\ \|g_i(x)\|^2 & i = j \end{cases}. \tag{4.7}$$

Note that one is free to choose any set of orthogonal functions for automatic power map generation. In this chapter, We arbitrarily choose orthogonal power map configurations generated by the 2-D function set

$$g_{mn}(x, y) = sin(m\pi x/L_x)sin(n\pi y/L_y), \tag{4.8}$$

in which m and n are the indices starting from 1; x and y are the position variables; L_x and L_y are the size of the chip in the direction of x and y axis respectively.

We remark that users may not have the luxury to generate the orthogonal spatial functions as they are limited by specific functional logics in practical chips or artificial power patterns in testing chips. The user could use any power map (spatial configurations) as long as it satisfies the input rank requirement. Nevertheless, the proposed power map generation method can provide some general guidelines for practical model generations.

4.2.3 Thermal modeling algorithm - ThermSubCP

Now we are ready to introduce our linear thermal modeling algorithm considering the highly correlated power inputs – *ThermSubCP*, which stands for Thermal modeling using the Subspace Identification method for Correlated Power maps.

Once we generate all the independent power map configurations, we need to generate two transient power sequences – one for model identification and one for validation. For each power map configuration, we basically divide the given transient power input waveform into two parts in this approach. The first part will be used for the model identification and the second part will be used for the validation. To test the predictability of the models, we will also add some additional power maps, which are not used in model identification. For instance, suppose we have a 4-input MIMO system, then we need 4 independent power maps with transient power inputs denoted as P_1, P_2, P_3, P_4. Then we split $P_1 = [P_{11}, P_{12}]$ into two parts in time scale. We do the same thing for the other 3 power inputs. Then the identification sequence will be $[P_{11}, P_{21}, P_{31}, P_{41}]$, while the validation sequence will be $[P_{12}, P_{22}, P_{32}, P_{42}, P_{a1}, P_{a2}]$, where P_{a1}, P_{a2} are the additional power inputs in power maps not used for identification.

ThermSubCP method also tries to automatically select the proper order of the models to satisfy the given error bounds by gradually increasing the order of the models until the accuracy in the validation phase is met. In our implementation, we use the maximum of the mean errors over all the transient responses for all the outputs as the error criteria. The proposed ThermSubCP flow is presented in Fig. 4.7.

```
ALGORITHM: THERMSUBCP
Input: $p$ power map configurations (power inputs and output responses)
Output: thermal model with proper order to meet the error bound

1. Start with order one and use the $p$ model identification configurations to generate thermal models.
2. Use the validation configurations to generate the output of the subspace model.
3. Compute the average error for each output.
4. If the error criteria is not satisfied, increase the model order and goto step 1. Otherwise, return the models obtained so far and stop.
```

Figure 4.7: The new ThermSubCP algorithm

4.3 Piecewise linear thermal modeling approach – ThermSub-PWL

The second issue we try to mitigate for thermal modeling is to deal with nonlinear effects of packaged microprocessor thermal systems. In this section, we present the new nonlinear thermal modeling technique, *ThermSubPWL*, on top of the linear modeling techniques we have discussed.

4.3.1 Local models for partitioned linear ranges

As shown in Fig. 4.5, thermal systems for packaged microprocessors show weakly nonlinearity, which may come from temperature dependent material properties, as well as different thermal boundary conditions such as variant convective coefficients for heat sinks cooled by fans with different speeds. Practically, we may have to build thermal models for different temperature ranges and boundary conditions, which is very important modeling objective for many thermal component modeling problems [26,27]. In this case, the thermal systems become naturally nonlinear as the temperature dependent material properties will lead to variable thermal system matrices, and the variant boundary conditions will lead to variable thermal resistances to the ambient for the thermal circuits.

If we still use the linear models to characterize the system, we observed that we have to use higher order to get good approximation since the nonlinearity is mild in many cases. Of course, such approximate models will not show any nonlinear effects, but they will use more poles or states to emulate the effects of nonlinearity on thermal responses of those systems. So we will end up with much higher orders for the thermal models even for linear approximation, which will hurt the performance of the thermal analysis. In addition, such analysis also loses the nonlinear effects of the original systems.

To mitigate this problem and reduce the model order and modeling errors, we propose to partition the temperature range or each fixed boundary conditions into a number of sub-ranges and then we build the state space linear models for each sub-ranges by the *ThermSubCP* method, and these local models are then used to build piecewise linear thermal model for the whole thermal system.

In the following, we build PWL model for different temperature ranges as a driving example. Similar strategy can be apply to build PWL model for different boundary conditions, with results shown in the experimental result section.

We notice that one issue with such a piecewise linear thermal modeling scheme is that temperature is location dependent across a whole package. There may be temperature gradients among different locations. To mitigate this problem, we use the average temperature of any instance time to guide model switching. The thermal gradients in a well-designed chip are typically well managed and reduced by the online thermal management techniques [30, 53]. Even with some degrees of thermal gradients, the local models should still be valid as it is a localized model and should be valid for a temperature range as long as the average temperature is still representative for the overall temperature of the processor die.

In order to obtain local models for different temperature ranges, we use stair-like power-temperature training sets to identify these models as Fig. 4.8 shows. For example, the model M_i is identified during time interval $[t_{i-1}, t_{i+1}]$, which corresponds to the temperature range of $[T_{i-1}, T_{i+1}]$. Since model M_i is identified with the temperature data ranging from $[T_{i-1}, T_{i+1}]$, the correct using of the subspace identification method guarantees that the identified model is valid for this temperature range.

In order to avoid the predictability issue and improve the accuracy of the subspace identification method, we use independent power map configurations as given by (4.8) to identify each local model for the corresponding temperature range.

By using the stair-like input-output data, the linear models of the subsystems in different temperature ranges could be accurately identified via the *ThermSubCP* method. Note that, all the pairs of the two adjacent models, like M_i and M_{i+1}, are identified with a shared portion of data, which makes both models valid for the same temperature range, like $[T_i, T_{i+1}]$ shown in Fig. 4.8. The reason is that the transition from one thermal model to another thermal model is gradual, and this shared portion can facilitate determination of model transformation matrices as will be discussed below.

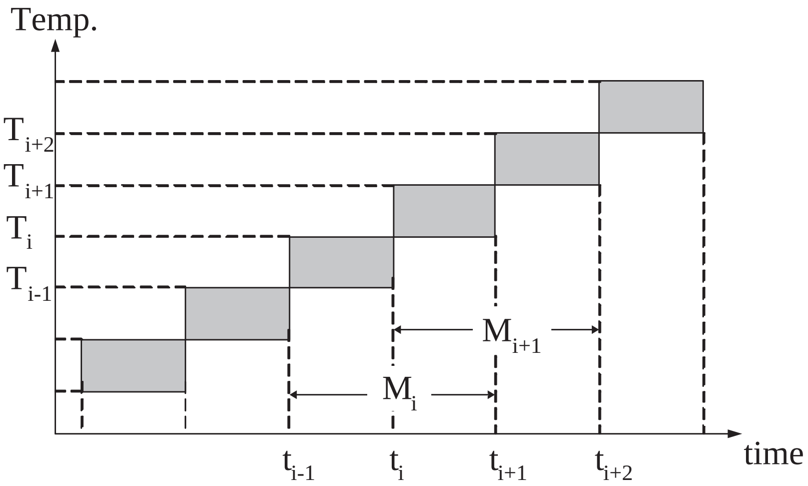


Figure 4.8: Identification of linear subsystems for different temperature ranges

4.3.2 Determination of model transitions

The linear models for each subsystem could not be directly combined to build the piecewise linear model for the thermal system of the microprocessor package because these identified models are not built on the same state variable basis. Hence, it is desirable to have a common state basis for all the local models instead of using different states for different models. This requires linear transformation to transform all these models onto the same basis. In [87], the transitions are assumed to be known at each time instance. Assuming the model transition is abrupt at the transition time instance t_i as shown in Fig. 4.9, it can be proved that the state of model M_a and state of model M_b are differed by a linear transformation T_{ba} as

$$x_{M_b}(t_i) = T_{ba} x_{M_a}(t_i), \tag{4.9}$$

where $x_{M_a}(t_i)$ is the state of model M_a at the transition time instance t_i, and $x_{M_b}(t_i)$ is the state of model M_b at the same transition time. Hence, to determine the linear transformation matrix T_{ba}, multiple transitions are required to solve the linear equations (4.10) in the sense of least squares as shown in [87].

$$[x_{M_b}(t_1), x_{M_b}(t_2), ...,] = T_{ba}[x_{M_a}(t_1), x_{M_a}(t_2), ...,]. \tag{4.10}$$

However, in our thermal system modeling, if we have specific temperature value for transitions between two models, we have to excite the states of the two models such that we have many independent states from the two models and transitions happens between the two models with those states. This will lead to much larger training tasks for the modeling process.

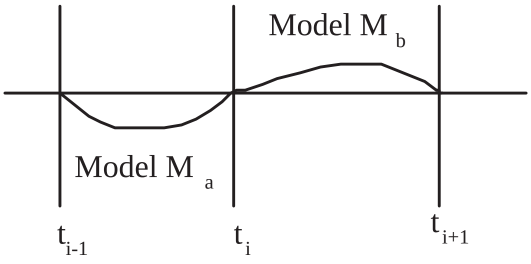


Figure 4.9: Abrupt model transition at known time instance

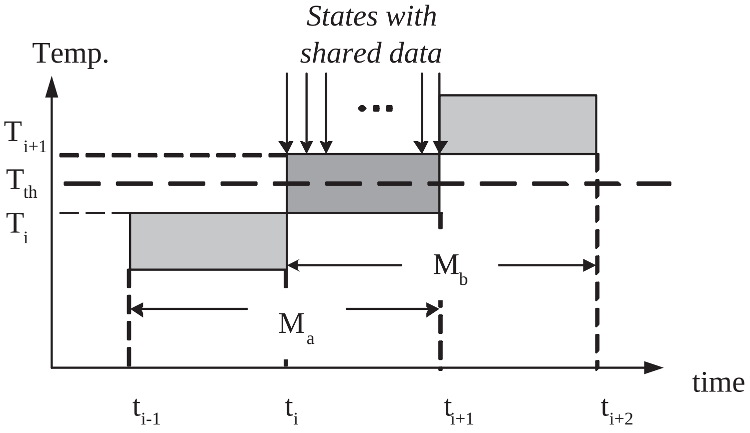


Figure 4.10: Model transition from M_a to M_b

To mitigate this problem, we propose a *transition region* concept in this chapter. We observe that the temperature transition from one model to another model is in general a gradual process as indicated in the time interval from t_i to t_{i+1} shown in Fig. 4.10, instead of an abrupt one that happens at a specific time instance. We define a transition region as shown in Fig. 4.10 in which both local models are valid (in other words, a state of model M_a will switch to a corresponding state of model M_b at any given time in this region).

As discussed before, the subspace identification method guarantees that any two adjacent models are valid for a portion of shared data sets from t_i to t_{i+1} (even though in the simulation we specify arbitrary T_{th} to determine the abrupt transition from M_a to M_b as shown in Fig. 4.10), thus, the relationship of the states for these two models within the range of the shared data set could be written as (4.11)

$$x_{M_b}(t_i : t_{i+1}) = T_{ba} x_{M_a}(t_i : t_{i+1}), \tag{4.11}$$

in which the matlab-like notation $x_{M_a}(t_i : t_{i+1})$ represents the states of model M_a during t_i to t_{i+1}, and $x_{M_b}(t_i : t_{i+1})$ represents the states of model M_b during t_i to t_{i+1} as Fig. 4.10 shows.

Hence, in this way, instead of requiring multiple model transitions as in [87], as long as we have enough independent states in the gradual transition region to make the state matrix $x_{M_a}(t_i : t_{i+1})$ full row rank, we could compute the transformation matrix T_{ba} by solving (4.11) in a least square sense. By using T_{ba}, we could transform model M_b to the basis of model M_a through (4.9).

Following this method, we could calculate the transformation matrices between any two adjacent thermal models using (4.11), and transform the model basis with (4.9). In this way, it is straightforward to transform all the identified local linear models to the common basis. We illustrate this by a 3-local system that has model M_a, M_b and M_c. It is straightforward to transform other model states into common model basis of M_a using

$$\begin{aligned} x_{M_b}(t_i) &= T_{ba} x_{M_a}(t_i), \\ x_{M_c}(t_i) &= T_{cb} T_{ba} x_{M_a}(t_i). \end{aligned} \tag{4.12}$$

In other words, we can just use the common state x_{M_a} as the local model states for all the local models. With the linear local model built from different temperature range or different linear ranges onto the same state basis through linear transformations, the piecewise linear model could smoothly switch from one model to another, which benefits the simulation accuracy.

4.4 Numerical results and discussions

The proposed method has been implemented in Matlab. We show results on a practical package modeled by commercial modeling tool COMSOL V 4.1 [88]. We first use a linear modeling method, then we employ the proposed piecewise linear method to reduce the model order and improve the simulation efficiency.

The packaged microprocessor design used in this study is shown in Fig. 4.11, where the convective boundary on the top of heat sink models the convective cooling from the fan placed above the processor. The aluminum heat sink is glued to the copper integrated heat spreader (IHS) that is attached to silicon die through a thin layer of thermal interface material (TIM). The materials and geometries of the major parts of the package are shown in Table 4.1, and we partition the die area into 4×4 power grids as shown in Fig. 4.12, and each grid represents a different function block.

Table 4.1: Material and geometry of the microprocessor package

Parts	Material	Dimensions (mm)
Die	Silicon	$10 \times 10 \times 0.7$
TIM	Thermal grease	$10 \times 10 \times 0.2$
IHS	Copper	$31 \times 31 \times 1.5$
Heat sink	Aluminium	$64 \times 64 \times 6.3$
Substrate	FR4	$37.5 \times 37.5 \times 1.3$

To model the power consumption of these function blocks, the input power sources are placed in these power grids and we measure the temperature at the adjacent 4 points of each square power grid, and thus we could have 25 observable temperature points starting from the up left corner of grid P1(1st temperature point) to the down right corner of grid P16 (25th temperature point) as shown in Fig. 4.12. As a result, we end up with a 16-input and 25-output thermal system. The convection coefficient is set to 450 $(W/(m^2 \cdot K))$ to model the convective air cooling effect from the cooling fan on top of the chip package.

To build a more realistic package with right dimension and materials, we applied COMSOL V4.1 [88] to build the package structures with on-chip power waveforms as inputs. The time step is set to 0.1 second for the transient simulation, and the thermal response

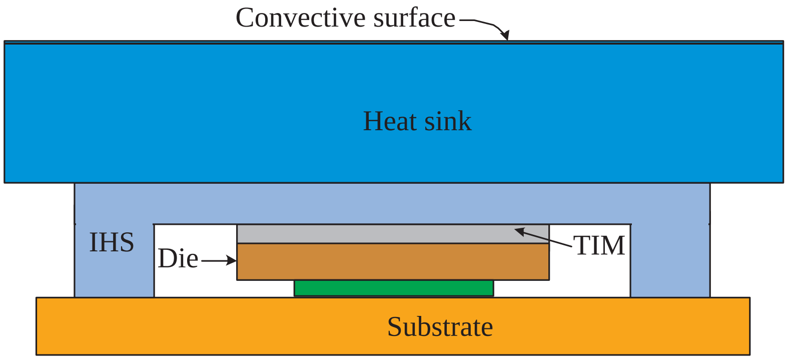


Figure 4.11: Microprocessor chip package

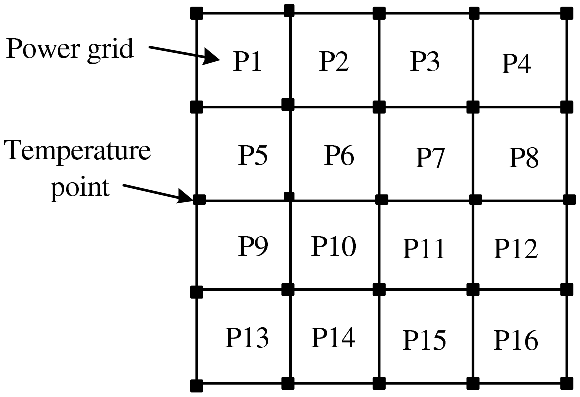


Figure 4.12: Top view of partitioned die area with power grids and temperature points (heat sink removed)

could be obtained by COMSOL V4.1 using the finite element method under the input power maps we generated. Fig. 4.13 shows the transient power waveform from our industry partner (The magnitude is determined by the specific power map).

Fig. 4.14 shows the steady state temperature distribution under a given power input on the constructed package and the chip. The colors in Fig. 4.14 indicate different temperatures, the red color represents hotter part of the package, in this case, it is at the center of the die, and the blue part represents the cooler part of the package like heat spreader. The temperature goes from high to low, and the color turns from red to blue. At the edge of the die, it is hotter than the heat sink and cooler than the die center, so it displays orange or yellow.

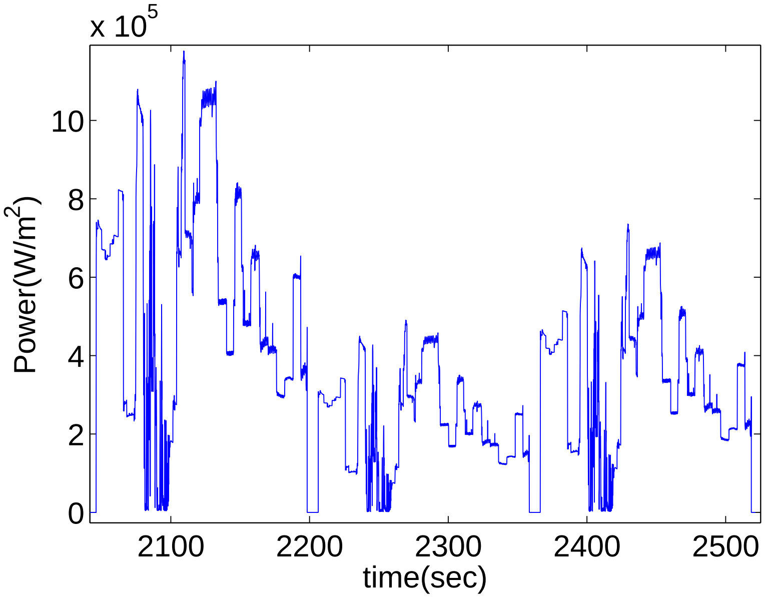


Figure 4.13: Transient on-chip power waveform from industry

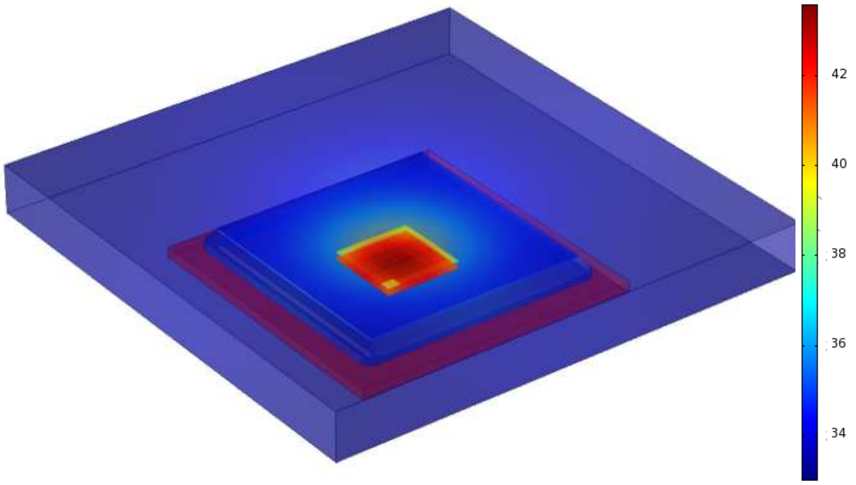


Figure 4.14: Steady state temperature distribution simulated by COMSOL 4.1

4.4.1 Multi-configuration model identification and validation for the package

Using the proposed method in section 4.2.2, 16 orthogonal power map configurations are generated by the 2-D orthogonal sine sets. The system is identified with these 16 input power map configurations, which covers 12800 transient time steps (about 800 time steps per power map). But we can use less number of time steps per power map. In the validation phase, in addition to the 16 automatically generated power map configurations, new configurations from M_1 to M_4 are introduced and their spatial distributions are defined in Table 4.2.

A 15th-order thermal model is obtained. The matched time domain response is shown in Fig. 4.15 (a) and frequency domain response is shown in in Fig. 4.16 for this 15th-order thermal model. The baseline results are obtained by using step power input with only one port is excited at a time and it can be viewed as the golden of the transfer

Table 4.2: Additional input power map configurations for validation

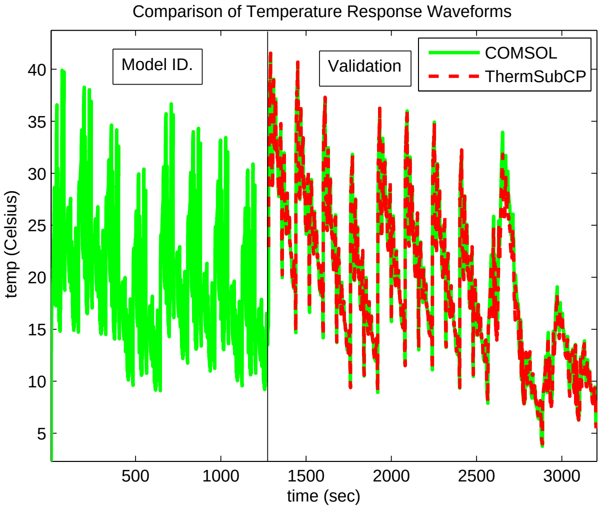
(Range of position variables: $0 < x < 4 \quad 0 < y < 4$)

Configuration Num.	Spatial Distribution of the Input
M_1	$P = P_0(cos(\frac{x+y-2}{4}) + \frac{xy^3}{ln(x+2)}$
	$+e^{y+2} + \frac{1}{3}\frac{\sqrt{xy}ln(4y+x-4)}{5x^2\sqrt{y+1}\sqrt{\sqrt{x}y}})$
M_2	$P = P_0(e^{-2x-y} + \frac{x}{4})$
M_3	$P = P_0(e^{-xy} + \frac{ln(\sqrt{x}y^2+1)}{x})$
M_4	$P = P_0\frac{(x+y+e^{-x+2}+xy^3e^{-x}+1)}{5y\sqrt{x}}$

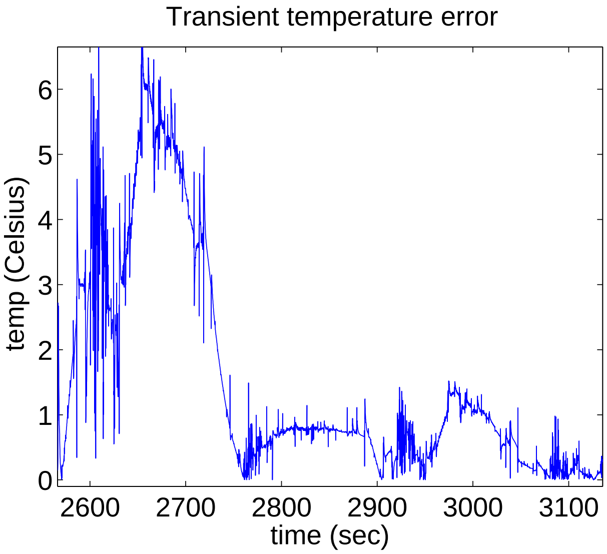
function. More detailed description about the baseline model could be found in Chapter 2. From these figures, we can clearly see that the proposed method gives very accurate thermal models compared to the golden. We could clearly see the filtering effect of the thermal system, which actually makes good physic sense. The power, analogous to the electrical current in a resistor, could be changed instantly, while the temperature changes of the system, analogous to the voltage changes on a cap, usually takes some time to happen due to the heat capacitance of the system.

Table 4.3 clearly shows the trade-off between the model accuracy and the identification time for the same amount of identification data; the cost of model identfication for more accuracy model with higher order is signficantly higher than low order compact models.

The pole-zero analysis indicates that the system is stable, and Fig. 4.17 shows that the poles of one transfer function of the 15th-order model ('$\times$' represents poles while '$\circ$' represents zero) are within a unit cycle. We have similar observation for other transfer functions and model of different orders. Also, Fig. 4.15 (b) plots the zoomed-in temperature



(a) On-chip temperature responses at the 1st temperature point



(b) Absolute errors of on-chip temperature responses at the 1st temperature point

Figure 4.15: Transient temperature responses and the temperature errors of the identified model (order:15)

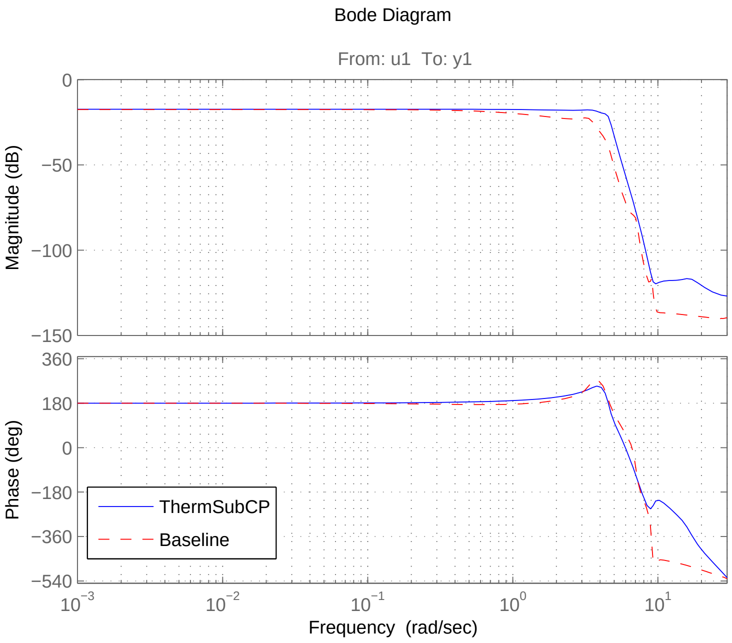


Figure 4.16: Bode diagram of the transfer function from input u_1 to output y_1

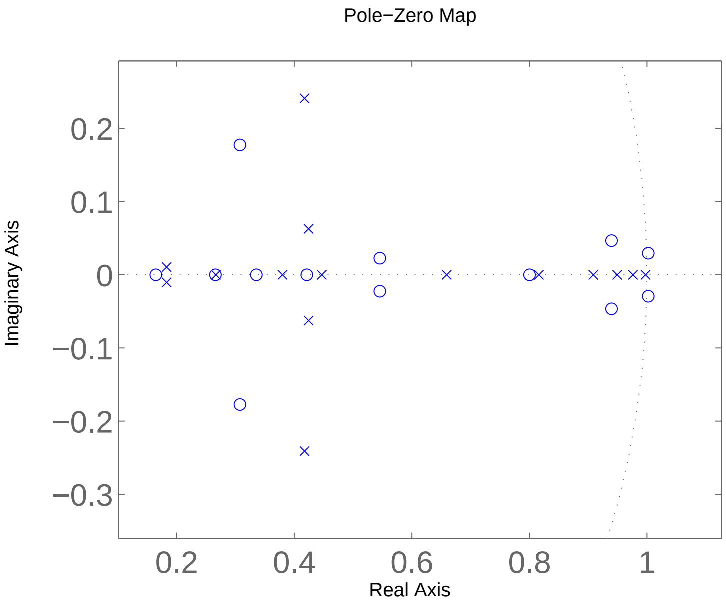


Figure 4.17: Poles and zeros distribution of the 15th-order model

absolute errors in the transient response, which gives the worst errors in entire validation period. The errors at some worst time points are relatively larger (about 15% or 6 to 7 Celsius), but over all the time steps, they are quite small. The output error statistics of the identified systems with different order is summarized in Table 4.3, where we list the maximum of the relative mean errors among all the ports over the entire transient simulation period(*Max Mean error*), and *Model ID time* is the time (in minutes) required to identify the model.

Table 4.3: Model identification CPU times and model errors

ThermSubCP order	35	25	15
Model ID time (min)	25.0	15.5	4.83
Max mean error	1.17%	2.58%	4.91%

4.4.2 PWL-based nonlinear modeling and validation

In this section, we use PWL modeling approach to build a compact thermal model of the packaged microprocessor system. To obtain the identification data of different local models for different temperature ranges, PRBS (Pseudo Random Binary Sequence) signals with stair-like envelops shown in Fig. 4.18 are used as inputs to characterize the system parameters; and in the validation phase, the transient waveform in Fig. 4.13 is used. PRBS signal has the white-noise like spectrum so that it can excite all the thermal system states, and we expect that it could potentially lead to better and more accurate identification results [35].

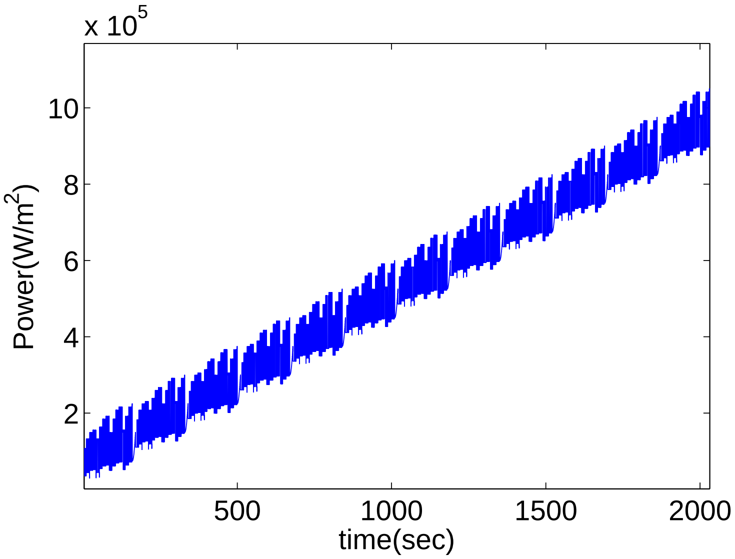


Figure 4.18: The stair-like PRBS input signal in model identification phase

PWL-based nonlinear modeling for different temperature ranges

As discussed before, temperature dependent material properties could lead to nonlinear behavior of the thermal system, thus we use piecewise linear method to model the thermal behavior in different temperature ranges. In this case, the stair-like envelop contains 12 "steps" that corresponds to 12 different ranges of input power intensity as shown in Fig. 4.19. We could arbitrarily partition the data and attribute them to different linear models that need to be identified with these data. At the beginning, we use "scheme-1" shown in Fig. 4.19 (a) to identify the linear models. In this scheme, each model is identified based on two consecutive data sets, and the adjacent models are built with one shared data set. In this way, 11 models will be identified in total, given the 12 data sets, and the piecewise linear model is to be built with these 11 local models. In order to avoid the predictability issue as discussed before, for each range of the input power, 16 orthogonal configurations are generated by $g_{mn}(x, y)$ as defined in (4.8).

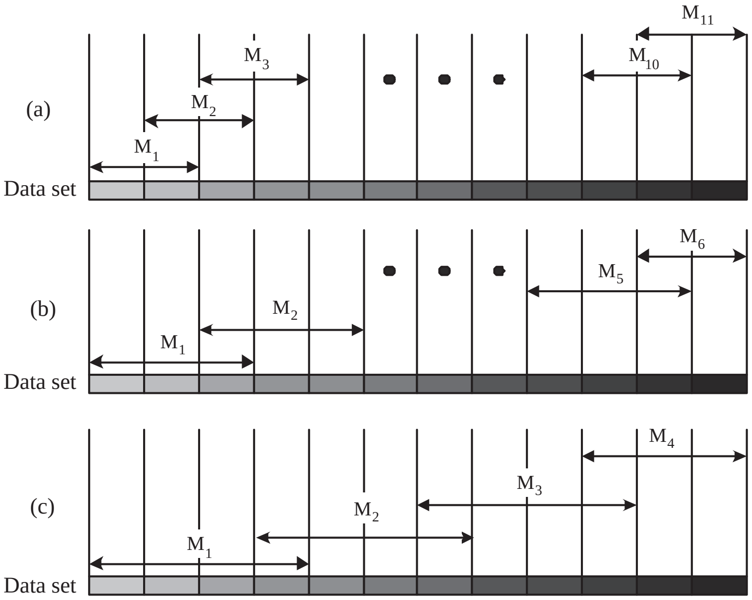


Figure 4.19: Data partition schemes for model identification (a) scheme-1(11 local models) (b) scheme-2(6 local models) (c) scheme-3 (4 local models)

By choosing 4th order model, and using the subspace identification method, all the 11 linear models could be identified. Applying the proposed method to determine the transformation matrices, all the linear models could be transformed to the same basis. Since the piecewise linear model built up in this way contains multiple local models, it is reasonable to partition the overall temperature range into the sub-ranges that the local models correspond to. The simulation result in Fig. 4.20 confirms that the temperature value predicted by the output of the identified piecewise model (dash line) closely matches the reference data (solid line).

In comparison, we also use different schemes of data partition. By using "scheme-2" and "scheme-3", we end up building the piecewise linear models with 6 local models

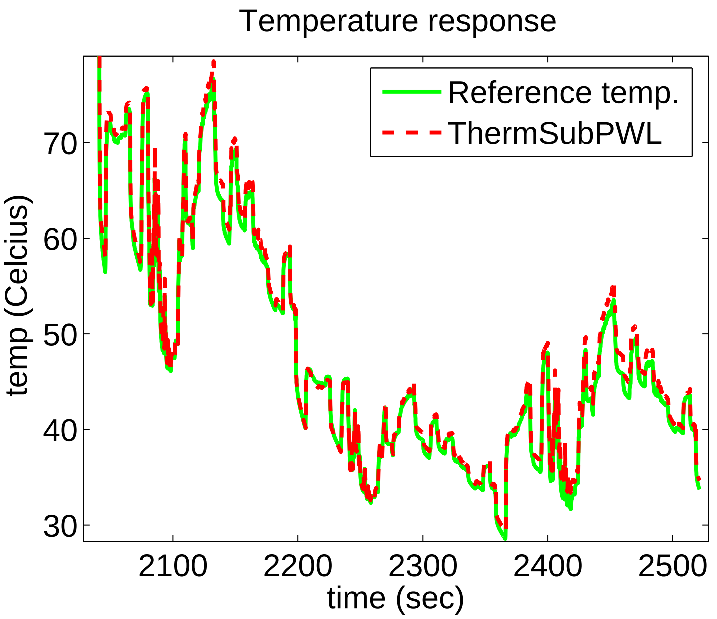


Figure 4.20: Transient view of the on-chip temperature response (at the 1st temperature point) of the piecewise linear models (PLM) built with 11 local models

and 4 local models respectively, and the simulation results are presented in Fig. 4.21 and Fig. 4.22. From the transient simulation results in Fig. 4.20, Fig. 4.21, Fig. 4.22, and the temperature error of the transient response Fig. 4.23, Fig. 4.24, Fig. 4.25, it clearly demonstrated the performance improvement as more linear local models are used. We could clearly observe that, for the same order, the error reduces as the number of the linear models in use increases, which shows a compelling evidence of using piecewise linear model for compact thermal behavior modeling and simulation. The output error statistics of the identified system is summarized in Table 4.4, where we also list the maximum of the mean errors (*Max Mean error*) among all the ports over the entire transient simulation period. We have also confirmed that each local models are stable, and as an example, a pole-zero configuration (poles marked with '$\times$' and zeros marked with '$\circ'$) of one local linear model is shown in Fig. 4.26, which shows that all the poles are inside the unit circle.

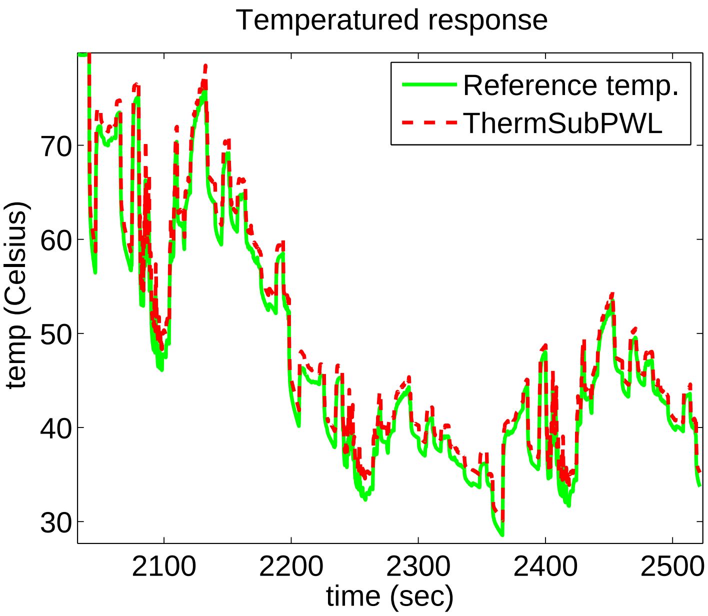


Figure 4.21: Transient view of the on-chip temperature response (at the 1st temperature point) of the piecewise linear models (PLM) built with 6 local models

On the other hand, the linear model with order 4 suffers from huge loss of accuracy as shown in Fig. 4.6 before. To make the linear model achieve comparable accuracy with the piecewise linear model, high order model needs to be chosen. In our experiment, we used 20412 transient time points to identify the model. As summarized in Table 4.5, the time required to identify (ID time) the high order linear model (LM) is 627.1 seconds, while on the other hand, the time required to identify all the piecewise linear models (PLM) is 63.8 seconds. Hence, the speedup factor for model identification is 9.8 comparing with linear model. Also, we used 25412 time points in transient simulation, and the high order linear model uses 22.2 seconds to conclude the simulation, and the piecewise linear model uses only 7.88 second to conclude the simulation, which is approximately 35% of the simulation time of the high order linear model.

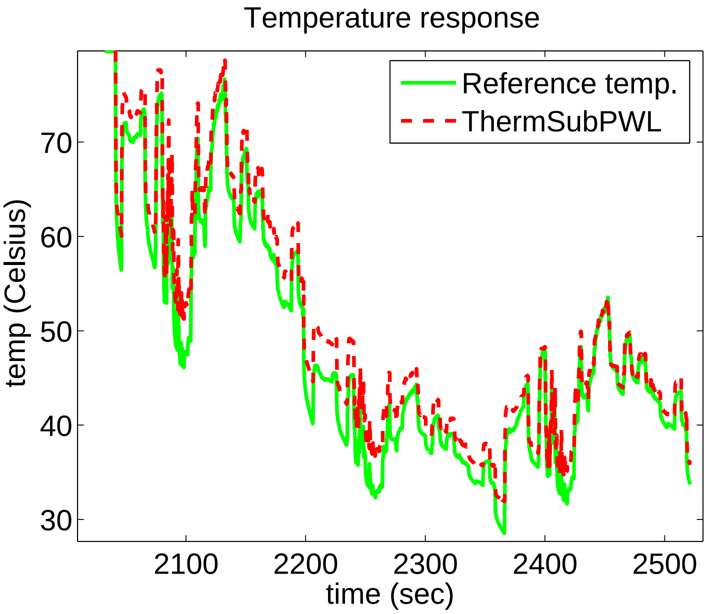


Figure 4.22: Transient view of the on-chip temperature response (at the 1st temperature point) of the piecewise linear models (PLM) built with 4 local models

We remark that although in our case, order 15 of the linear model is not significantly higher than the order 4 in a sense, yet, the time required to identify the state space model through subspace identification method increases significantly because a large amount of input and output data are required to identify the state space model accurately. As a result, choosing low order model to identify the targeted dynamic system leads to substantial savings in subspace identification method, which is important in the process of building and calibrating a dynamic model in a dynamically changing environment. Also, piecewise linear model achieves substantial savings in simulation time because the lower order model is used in simulation.

Table 4.4: Model accuracy comparison with different identified models (order: 4)

Num of linear models in use	11	6	4
Max mean error	2.1%	3.9%	5.9%

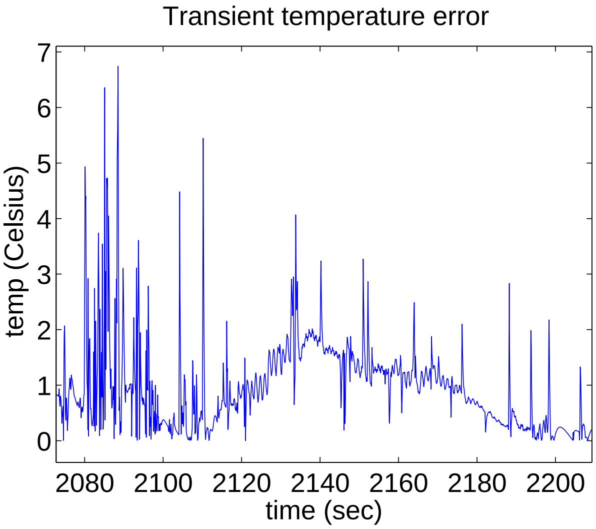


Figure 4.23: Absolute errors of the on-chip temperature response (at the 1st temperature point) of the piecewise linear models (PLM) built with 11 local models

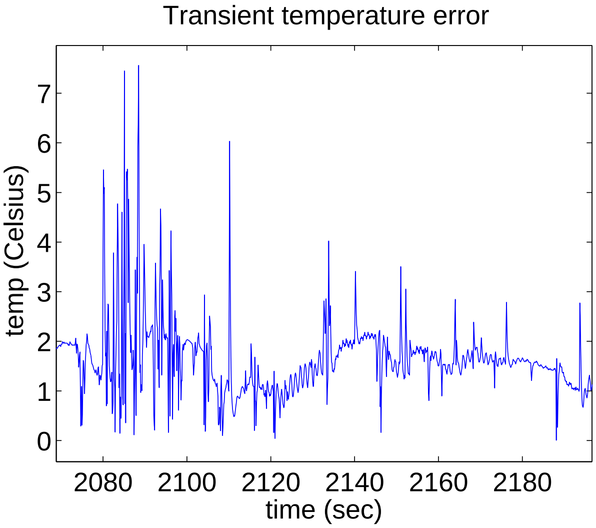


Figure 4.24: Absolute errors of the on-chip temperature response (at the 1st temperature point) of the piecewise linear models (PLM) built with 6 local models

Table 4.5: Comparison of model accuracy and CPU times

Comparison Items	Error	ID time	simulation time
PLM (order:4)	2.1%	63.8 sec	7.88 sec
LM (order:15)	2.3%	627.1 sec	22.2 sec

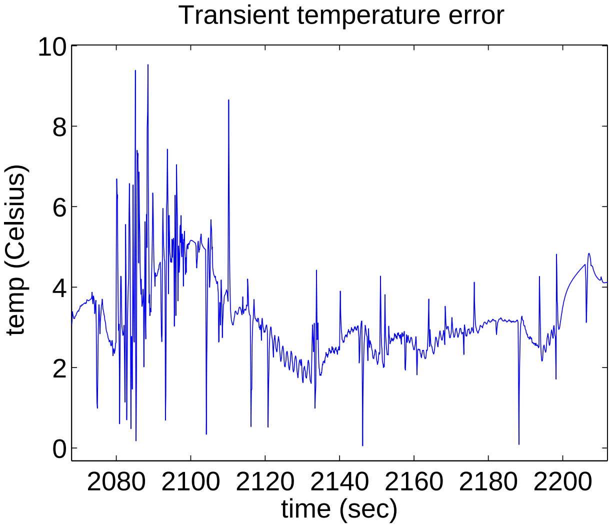


Figure 4.25: Absolute errors of the on-chip temperature response (at the 1st temperature point) of the piecewise linear models (PLM) built with 4 local models

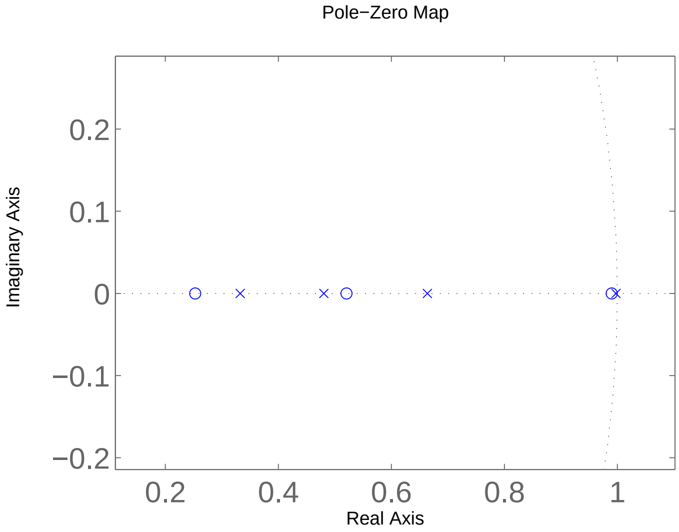


Figure 4.26: Pole-zero configuration of a transfer functions in an identified local model

PWL-based nonlinear modeling for different convection rates

Another potential thermal nonlinearity could be caused by the changing speed of the cooling fan, which translates to changing (nonlinear) boundary conditions for the thermal circuits. In this case, our proposed piecewise linear model offers a viable solution to build a model working for the changing boundary conditions.

In package level thermal modeling, different fan speeds could be translated to different convection rates at the top surface of the heat sink. Hence, to model various fan speeds, in our experiment, we linearly change the convection rates of the top surface of the heat sink as shown in Fig.4.27, and use the model identification routine with 16 orthogonal power map configurations to identify each local model for a certain range of the varying convection rate in Fig. 4.27. Each two adjacent models covers a common range of the convection rate, which is served as model transition region to determine the transformation matrices that transform all the local models to a common state base using the procedure in section 4.3.2. The identified model is validated under different ranges of convection rate in the validation phase. The power trace in the experiment is shown in Fig. 4.28, in which PRBS signal is used for model identification and while the power signals from our industry partner are used for model validation.

By choosing 4th order model, and following the same procedure used before, all the 4 local models could be identified. The simulation result in Fig. 4.29 (a) confirms that the piecewise linear model could effectively predict temperature waveforms in different ranges of convection rate. As a comparison, if we use a single 4th order linear model, the predicted transient response sometimes could be significantly deviated away from the

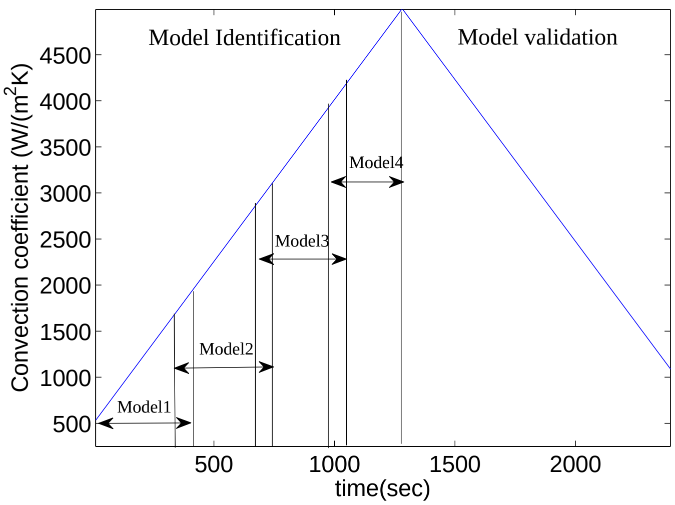


Figure 4.27: Different convection rates of the heat sink

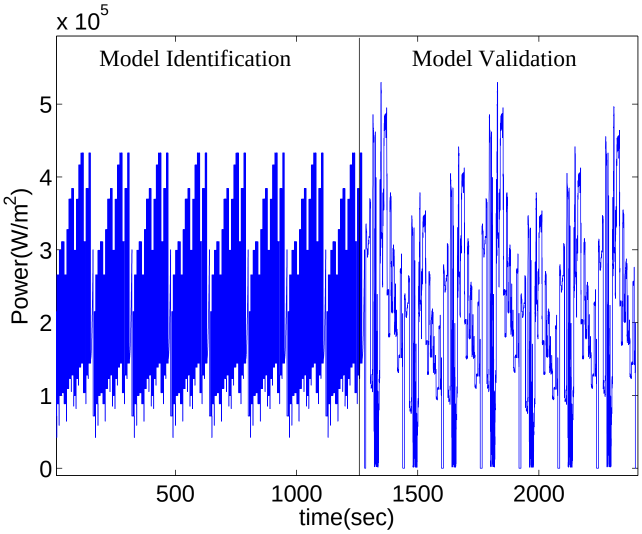
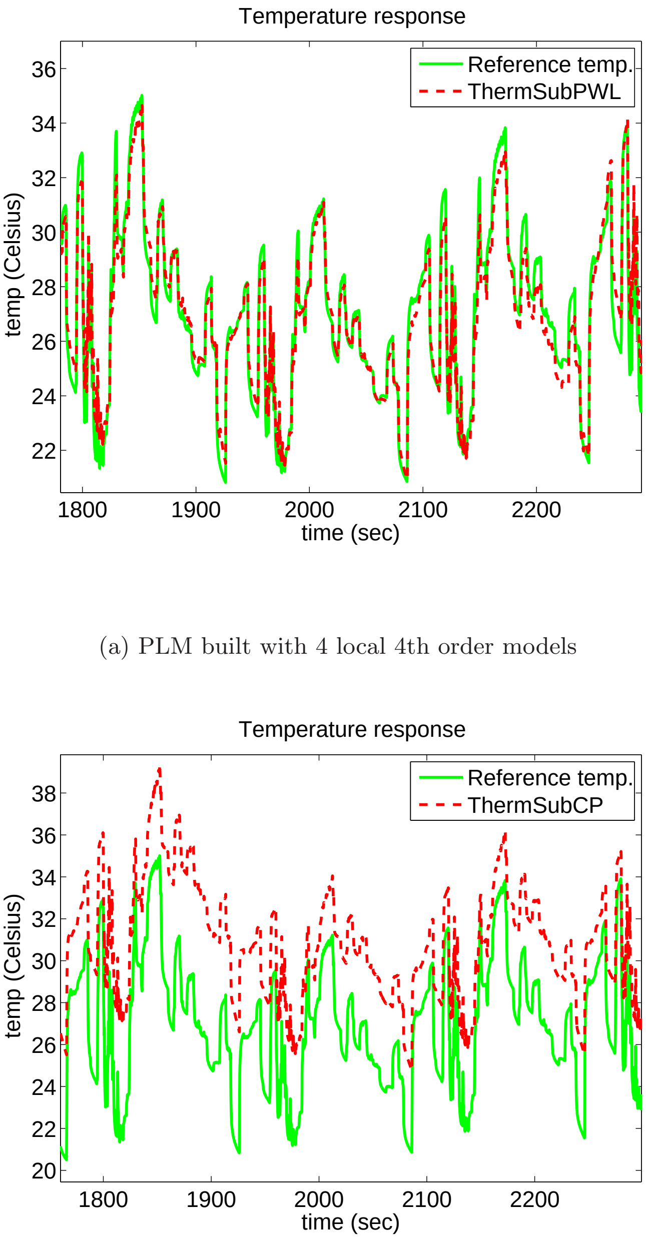


Figure 4.28: Power traces used to build and validate the piecewise linear model for different convection rate

reference temperature data as Fig. 4.29 (b) shows. The errors of the transient waveform by the piecewise linear model and the corresponding linear model with the same order are shown in Fig. 4.30 (a) and (b) respectively.

In addition, we could anticipate that the model identified under higher convection rate should be a faster model because the package could reach steady state quicker with higher convection rate, which could be confirmed by investigating the pole-zero configuration shown in Fig. 4.31. We could find that the dominant pole (0.9879 in z-plane) of one transfer function in Model2 (identified with convection rate from 1625 $W/(m^2K)$ to 2750 $W/(m^2K)$) is farther away from the unit circle than the dominant pole (0.9920 in z-plane) of the corresponding transfer function in Model1 (identified with convection rate varying from 500 $W/(m^2K)$ to 1725 $W/(m^2K)$), and it is also much closer to a zero, which partially cancels the effect of the dominant pole. Hence, this result of pole-zero configuration suggests that Model2 has faster response as theoretically anticipated, which also indicates that each local model is identified to take account of the significantly changed convection rate overtime.

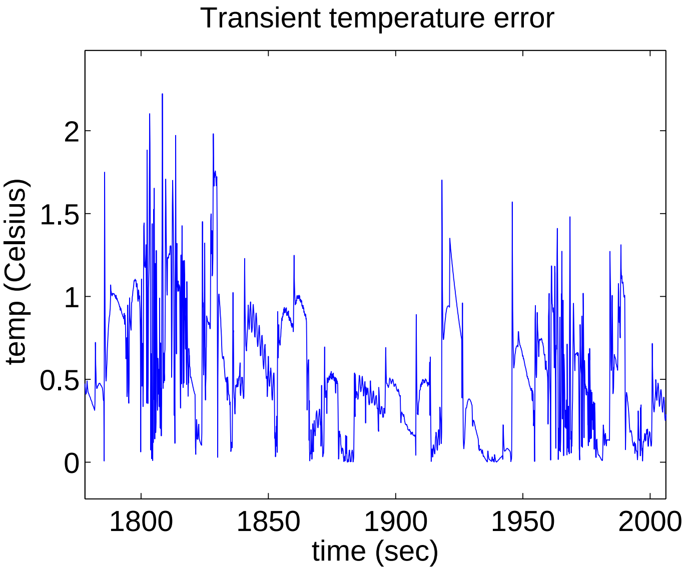
Hence, in this way, we applied the proposed method to model the chip package with variant convection rate, and the experiment result of the piecewise linear model shows significant accuracy improvement over the conventional linear modeling, which confirms the validity of the proposed methodology in thermal modeling. By investigating the resulting pole-zero configuration of the identified model under different convection rate, we also confirmed that each local model is identified to consider the potential huge variation of the fan cooling effects during the chip operation.



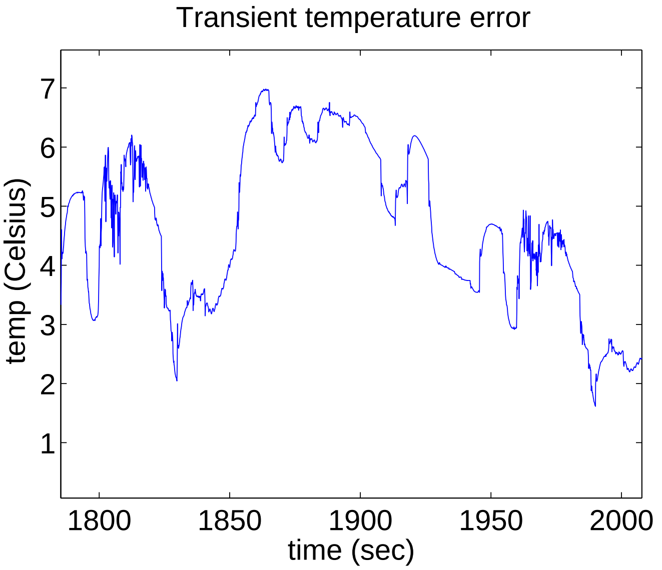
(a) PLM built with 4 local 4th order models

(b) A simple linear 4th order model

Figure 4.29: Comparison of the on-chip temperature response (at the 1st temperature point) predicted by the piecewise linear models (PLM) with the one predicted by a linear model under variant convection rate

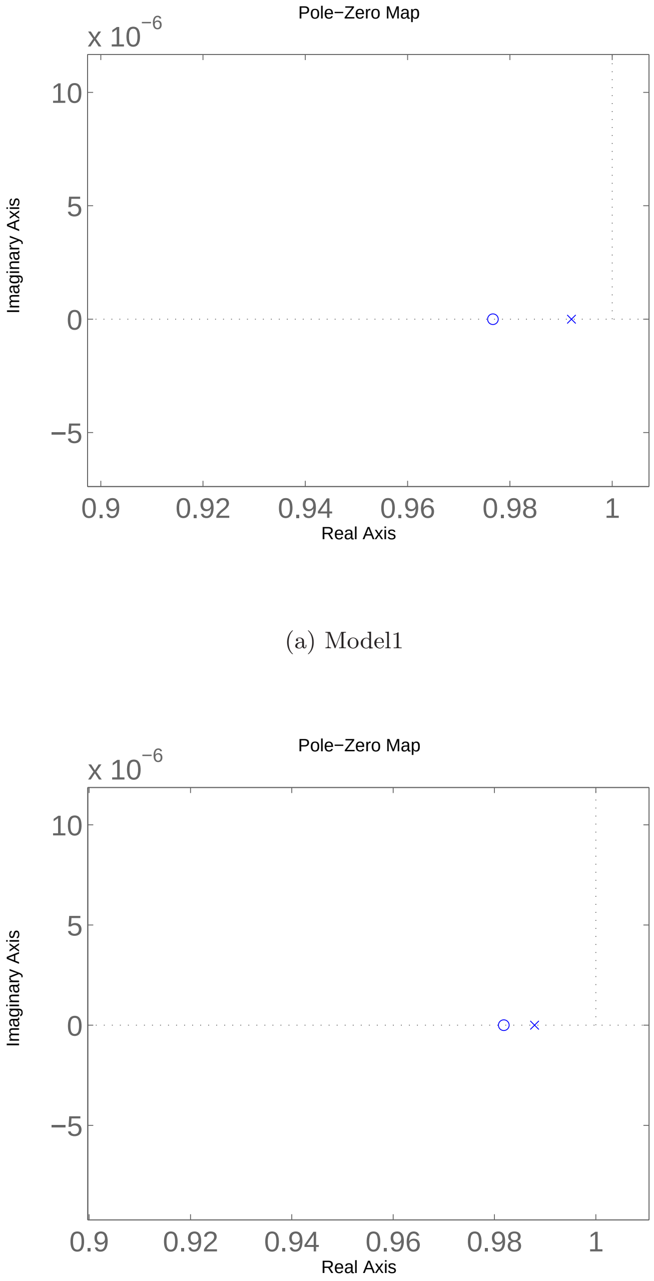


(a) Piecewise linear model



(b) Linear model

Figure 4.30: Absolute error of one on-chip temperature response at the 1st temperature point



(a) Model1

(b) Model2

Figure 4.31: Comparison of dominant pole-zero of the transfer functions in Model1 and Model2 (from input 1 to output 1)

4.5 Summary

In this chapter, we have first proposed a new package level thermal modeling technique considering practical power maps with highly correlated power inputs. We have used orthogonal sets to generate independent power maps to meet the spatial rank requirement in presence of highly correlated input signals. The new method, *ThermSubCP*, can also automatically select the order of the thermal models for the given error bounds. Secondly, we have also proposed piecewise linear (PWL) modeling approach for modeling nonlinear effects and various boundary conditions. The piecewise version of the new modeling algorithm, *ThermSubPWL*, partitions the nonlinear ranges (due to temperature or boundary condition changes) into a number of small ranges and performs modeling in each range. Experimental results have validated the proposed method on a practical microprocessor package modeled by commercial multi-domain/physics tool, COMSOL V4.1, under practical power signal inputs. By using multiple independent power map configurations in model identification, the predictability of the identified model is significantly improved. The new piecewise linear models can model thermal behavior over wide temperature ranges or over different thermal boundary conditions due to different cooling configurations. Further, the PWL modeling technique can lead to much smaller model order without accuracy loss, which translates to significant simulation time reduction and about 10X less time to identify the reduced models compared to the simple modeling method by using the high order models in our examples.

Chapter 5

Task Migration based Distributed Thermal Management Considering Transient Effects for Many-Core Microprocessors

In this chapter, a new architecture level task-migration based distributed thermal management scheme for many-core microprocessors is proposed to reduce the on-chip temperature variance and the occurrence of hot spots by considering more transient thermal effects. First, unlike the existing approaches which use the steady state temperature, the new method uses a new temperature metric called *effective initial temperature* to guide the task migration process. The new temperature metric based on moment matching based frequency domain analysis technique incorporates both initial temperature and other transient effects to make optimized task migration decision. The resulting schedule algorithm leads to more reduction of hot spots without full transient thermal simulation. Second, we show how the *effective initial temperature* can be computed in a localized way for distributed thermal management for multi-core and many core systems. Experimental results on a 100-core microprocessor demonstrate that the proposed method reduces 21% more thermal hot spots compared to the alternative method that does not use *effective initial temperature*, and reduces 44 % more thermal hot spots compared to an existing distributed thermal management method (DTB-M) [58], leading to more balanced temperature distribution of many-core microprocessor chips. Also, comparing with the centralized counterparts, the proposed method shows acceptable loss of performance, and is more practical for thermal management in many-core systems.

The rest of this chapter is organized as follows. In section 5.1, we present a general outline of the thermal management problem and the motivation to solve it. In section 5.2, we present the proposed method with the algorithm flow. In section 5.3, the proposed methods are evaluated and their significance is discussed. Section 5.4 concludes this chapter.

5.1 Problem formulation

In this work, we aim at minimizing the transient temperature variations across the whole chip by properly assigning tasks among different cores in a distributive way. Specifically, we assume a multi-core microprocessor consists of N tasks of different load, denoted as $\mathbb{P} = [P_1, P_2, \cdots, P_N]$. M processor cores are involved in executing the tasks. All of these tasks do not have the sequential dependency. Each task has a different load or power consumption (assume we have homogeneous cores) when they are executed on specific core of the many-core processor. Since the loads of the tasks are different, the power intensity varies across all the cores, resulting in non-uniformly distributed temperature profile. Using C_p (Unit:J/K) to denote heat capacitance, $P(t)$ to represent the transient power generated in a core, which is related to its load, and $\dot{Q}(t)$ to represent the transient heat transfer rate to and from a core, the transient temperature of a core $T(t)$ can be formulated as

$$T(t) = \frac{1}{C_p}\int_{t_0}^{t}(P(t) - \dot{Q}(t))dt + T(t_0) \tag{5.1}$$

Note that the heat transfer rate $\dot{Q}(t)$, which influences the temperature of the core, depends on the thermal characteristics and material properties of the many-core systems and the final temperature $T(t)$ is also dependent on the thermal capacitance of the systems. If $T_i(t)$ and $T_j(t)$ represent the temperatures for core i and j, then the goal of the new task migration method is

$$\text{Minimize } \Delta T_{ij}(t) = T_i(t) - T_j(t), i \in M, j \in M, i \neq j \tag{5.2}$$

The optimization is achieved by properly assigning the N tasks into the M cores in a periodic and distributed manner.

Similar to [57,58], in our distributed control framework, we assume that each core can only talk to its adjacent cores during this task migration process. Each core has the knowledge of its temperature and power information of its current task, as well as those of its adjacent cores. As a result, the task migration could only be performed between two adjacent cores based on their local information of temperature, power and package properties.

5.2 The proposed distributed thermal management method

In this section, we present a new distributed thermal management based on the task migration scheme to balance the on-chip temperature profile of the many-core microprocessors. The task migration is carried out in a distributed way, and the migration happens only between two adjacent cores, which are floor-planned as meshed tile structure.

As mentioned before, the new task migration method is based on the following two new techniques: First, a new thermal metric, called *effective initial temperature*, denoted as T_{eff}, is used to guide the task migration process. T_{eff} allows the consideration of temperature influence from other part of the multi-core chip through transient effects and heat removal capability of each core due to the chip package. Then, we extend the T_{eff} calculation in a localized way to make it suitable for distributed thermal management, which optimally considers the power and the temperature of each core, as well as the thermal influence from its adjacent area through package induced transient thermal effect.

In the following subsections, we first introduce the new thermal metric T_{eff} and provide its physics insights in details before presenting the overall task migration algorithm flow.

5.2.1 Effective initial temperature for task scheduling

Traditional task migration methods typically assign heavy tasks to the cores with low steady-state temperature to balance the on-chip temperature profile [53, 55, 59]. Due to the transient temperature effects, such simple strategy will not work well as the transient temperature depends on not only the thermal conductance G, but also thermal capacitance

C of the whole processor (including package and heat sinks etc.). One can always perform full-blown transient thermal simulations to predict the transient temperature responses given task loads of cores. However, such simulation will be very expensive or even prohibitive in the optimization loop.

In this work, to mitigate this problem, we propose a new temperature metric, called *effective initial temperature*, for task migration without full transient simulation. In the following part, we give the definition of the *effective initial temperature*, and then justify it as a better temperature metric to predict the transient temperature behaviors for task migration based thermal management.

0th-moment temperature based indicator

Moment matching based analysis has been used for fast estimation of interconnect delays in the past [89,90]. In this work, we apply moment matching based analysis to derive a dominant temperature metric (*effective initial temperature*) to guide the task migration. This scheme is motivated by the observation that the 0th moment component of the power traces of the microprocessor is dominant compared to high frequency components. Fig. 5.1 shows the Fourier transform of the power trace from a benchmark, which clearly indicates the dominance of the 0th moment component. As a result, the corresponding 0th moment of the transient temperature responses can be used as a good indicator of the temperature for each core.

Mathematically, for equivalent thermal circuit using G as the thermal conductance matrix and C as the thermal capacitance matrix, we can apply Modified Nodal Analysis

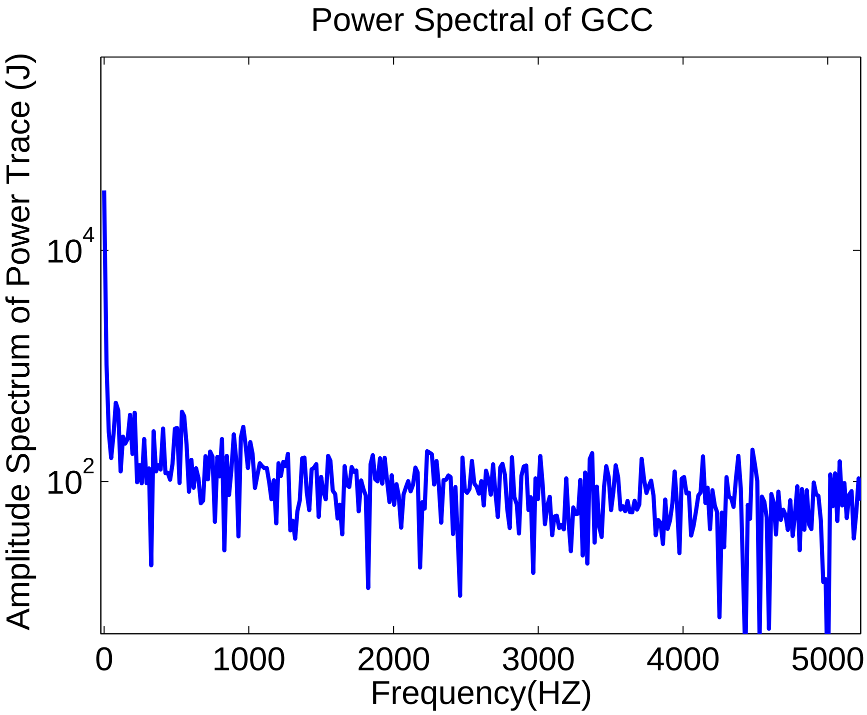


Figure 5.1: Discrete Fourier transform of the power trace for running GCC benchmark

(MNA) to formulate the thermal circuit

$$GT(t) + C\dot{T}(t) = u(t) \tag{5.3}$$

where $T(t)$ is the time-domain temperature response, and $u(t)$ is the power trace of the processor.

Assume $U(s)$ as the Laplace transform of the power trace $u(t)$ of that processor, and $T(s)$ as the corresponding temperature response, in frequency domain, we could have

$$GT(s) + sCT(s) - CT(t_0) = U(s) \tag{5.4}$$

where $T(t_0)$ is the initial temperature vector of all the nodes at starting time t_0, or alternatively, the temperature vector at the end of the previous simulation cycle.

To find out the contribution of the transient temperature in terms of frequency moments, we expand $T(s)$ and $U(s)$ using Taylor's series at $s = 0$ to have

$$\begin{aligned} G(T_0 + T_1 s + T_2 s^2 + \cdots) + sC(T_0 + T_1 s + T_2 s^2 + \cdots) \\ -CT(t_0) = U_0 + U_1 s + U_2 s^2 + \cdots \end{aligned} \tag{5.5}$$

Thus, each temperature moment could be calculated in a recursive way as follows:

$$\begin{aligned} T_0 &= G^{-1}(U_0 + CT(t_0)) \\ T_1 &= G^{-1}(U_1 - CT_0) \\ T_2 &= G^{-1}(U_2 - CT_1) \\ &\vdots \\ T_m &= G^{-1}(U_m - CT_{m-1}) \end{aligned} \tag{5.6}$$

Since the 0th moment of the power trace U_0 is dominant, the 0th moment component of the transient temperature response T_0 is also dominant. Thus, it makes a good sense to use the 0th moment component of the response temperature (called *0th moment temperature*) as an on-chip temperature indicator. In this way, we could conveniently use (5.7) to calculate the dominant 0th temperature T_0 given the 0th moment component of power trace U_0 for the future task and the initial temperature $T(t_0)$ at the current time.

$$T_0 = G^{-1}U_0 + G^{-1}CT(t_0). \tag{5.7}$$

The term T_0 is not traditional 0th moment response from the resistor-only thermal systems. It actually contains transient effects (G and C) of the thermal system, which will be explored by this work. As we can see, the 0th moment temperature consists of two components, $G^{-1}U_0$, which actually is the steady state temperature responses considering resistances only of the thermal circuits. The second term, $G^{-1}CT(t_0)$, relates to the initial temperature condition $T(t_0)$, as well as the thermal capacitance and conductance of the whole system.

Physical insight derived from effective initial temperature

We now define a new term called *effective initial temperature* as

$$T_{eff} = G^{-1}CT(t_0). \tag{5.8}$$

T_{eff} has temperature unit in frequency domain. The term $CT(t_0)$ can be viewed as the initial condition due to energy source stored from the previous task executions. The thermal systems obtained from the finite difference method can be remodeled as equivalent RC networks due to the similarity between the thermal diffusion equations and the Maxwell wave equations. In such networks, each node has a lumped thermal capacitor connecting to ground [71]. The resulting matrix C is typically a diagonal matrix. On the other hand, G^{-1} is a dense or full matrix. This means that the real temperature of a specific core (or element in the vector T_{eff}) will not only depend on itself but also on other cores of the chip as well. Given T_{eff}, we can rewrite (5.7) as

$$T_0 = G^{-1}U_0 + T_{eff} \tag{5.9}$$

Physically, T_0 represents the frequency-domain solution of the thermal system (5.4) when $s = 0$ or $t = \infty$, or the transient stable response. So if we can minimize T_0, we actually minimize the transient temperature response. Now let us look at the two terms in T_0. The first term $G^{-1}U_0$ represents the steady state temperature responses on the resistance-only thermal network due to the dominant power inputs. The second term T_{eff} represents the response due to the initial temperature $T(t_0)$ under a specific chip structure (including the package) defined by thermal matrices G and C, which is analogous to zero input ($U_0 = 0$) response of the thermal system.

As a result, to minimize the T_0 differences among different cores, we should assign heavy load task (with large U_0) to the core with low T_{eff} (versus to the core with low $T(t_0)$). The idea can be better illustrated by the following example. Fig. 5.2 to Fig. 5.4 give an intuitive illustration of the physical meaning of T_{eff} and the task scheduling scheme based on T_{eff} in a 4-core microprocessor, where $T(t_0)$ in Fig. 5.3 is the initial temperature distribution across the cores of the process chip at time t_0, and the effective initial temperature T_{eff} in Fig. 5.4 represents the steady state response of $G^{-1}C$ system due to $T(t_0)$ as illustrated by Fig. 5.2 . The powers of the tasks (Pa, Pb, Pc, Pd) in Fig. 5.4 are sorted from low to high. The bottom of Fig. 5.4 shows that the tasks are assigned to the different cores based on T_{eff} to better balance on-chip temperature profile. In comparison, the traditional method just looks at $T(t_0)$ (the input of $G^{-1}C$ system), and assign the heavy task to the core with lowest temperature.

In short, the new thermal metric, T_{eff}, provides a very compact and simple way to consider the transient thermal effects of each core and impact from other cores as well, which leads to more effective task scheduling to reduce on-chip hot spots as will be shown in the experiments.

5.2.2 Distributed calculation of effective initial temperature

As clarified before, the definition of T_{eff} introduces a new way of task scheduling method for thermal balancing of the processor chip. However, according to (5.8), the calculation of each element in T_{eff} requires all elements in $T(t_0)$, which is the global distribution

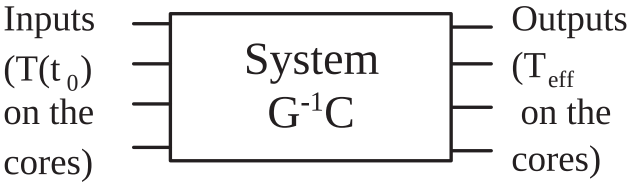


Figure 5.2: The transient thermal system and input and output

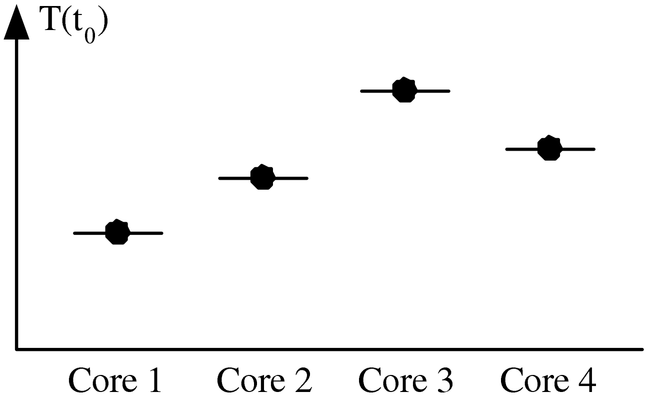


Figure 5.3: The initial temperature distribution at time $T(t_0)$

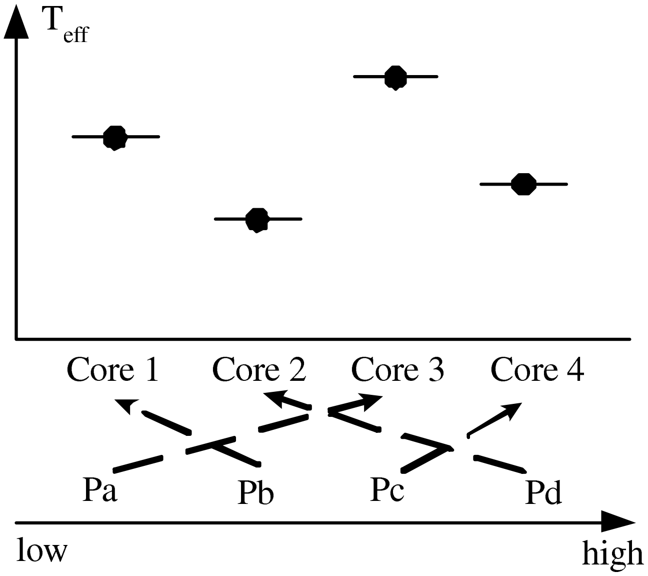


Figure 5.4: The resulting temperature distribution T_{eff} under $U_0 = 0$ and the resulting task migration scheme

of the temperature values across the whole chip. Thus, we need to estimate a distributed T_{eff} in a distributed way so that it could be used in the proposed method.

The distributed version of T_{eff} could be obtained in the following way: Instead of using the global temperature distribution and global package model as suggested by (5.8), we could use a local package model with local temperature values from the current core and its adjacent cores to calculate the distributed effective initial temperature for the local chip region as

$$T_{eff,N(i)} = R_{N(i)} C_{N(i)} T_{N(i)}(t_0) \tag{5.10}$$

in which $T_{eff,N(i)}$ contains the effective initial temperature values of core i and its adjacent cores in the local region as shown in Fig. 5.5. In this figure, $T_{N(i)}$ contains the real temperatures of core i and its adjacent 8 cores in the same local region:

$$T_{N(i)} = [T_a, T_b, T_c, T_d, T_e, T_f, T_g, T_h, T_i]. \tag{5.11}$$

$R_{N(i)}$ and $C_{N(i)}$ are the local thermal resistance matrix and thermal capacitance matrix constructed from the elements from row a to i and column a to i from original G^{-1} and C matrices.

We remark that $T_{eff,N(i)}$ is the localized version of T_{eff} for localized core area as shown in Fig. 5.5. In this way, the effective initial temperature can be calculated in a distributed manner in each local region, using the thermal parameters of each local core area of the chip package and the local temperature distribution of each core i and its neighborhood. In other words, each core can systematically consider the impact from its immediate neighbors, which is in contrast with existing ad-hoc based approaches for considering neighbor effects [56].

The number of such local regions equals the number of cores in the whole chip and each core can compute its $T_{eff,N(i)}$ independently. The detailed procedure about the proposed distributed task migration using distributed effective initial temperature will be explained later.

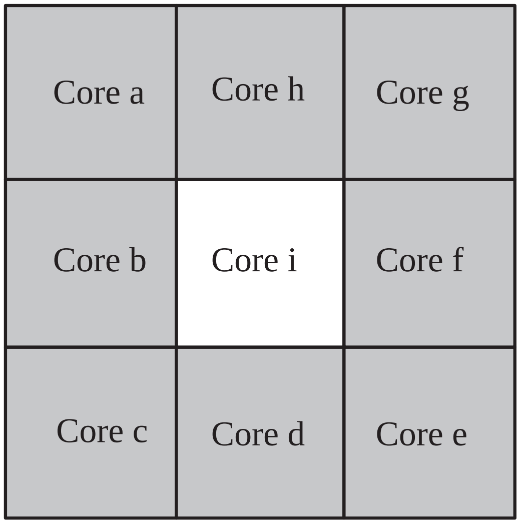


Figure 5.5: The local region used to calculate the distributed $T_{eff,N(i)}$ in core i and its adjacent cores

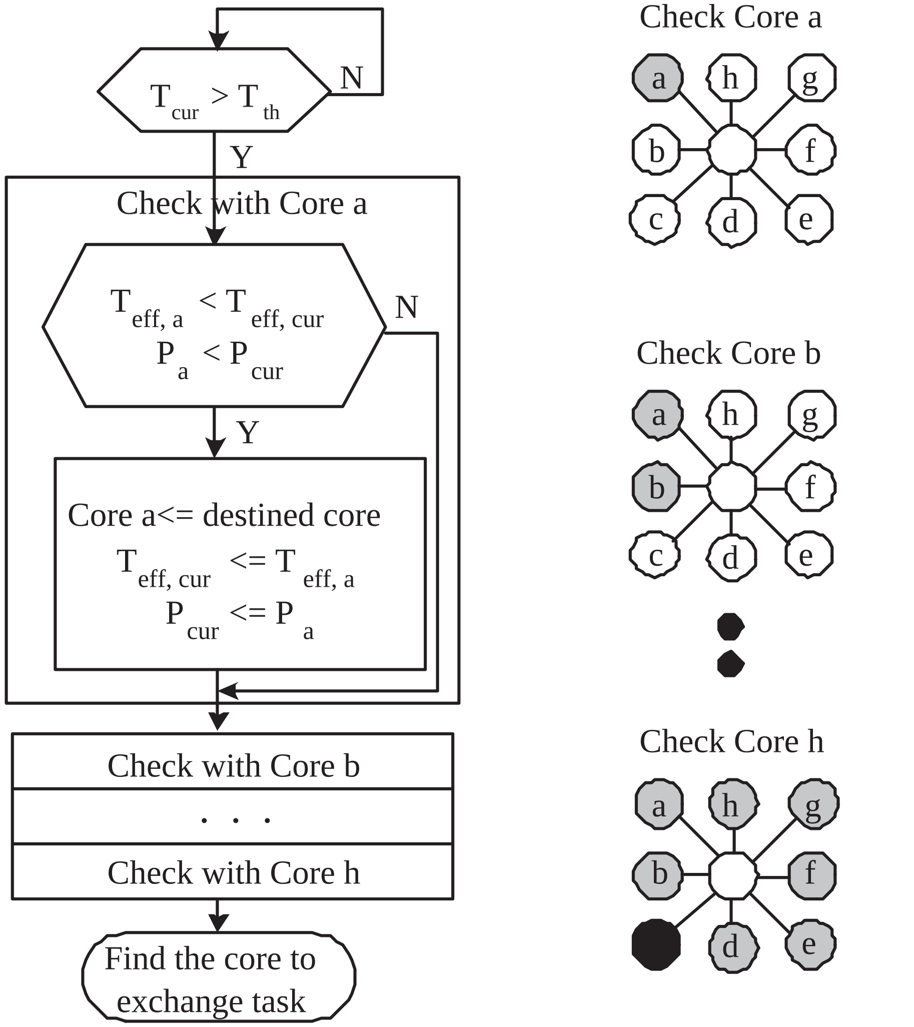
5.2.3 The proposed new distributed task migration method

Task migration policy is used to balance the temperature profile of multi/many-core microprocessors. In our proposed approach, we use distributed effective initial temperature as a new thermal metric that takes account of transient thermal effect to guide task migration decision. The new approach is distributed in the sense that task migrations happen only between two adjacent cores as shown in Fig. 5.5, where core a-h are the adjacent cores of core i. The proposed distributed task migration method can be outlined as the following: we define the current core as the one that the task migration policy is applied to, the destined core as the adjacent core that could potentially exchange task with the current core if the following two task migration criteria are satisfied:

1. $T_{eff,des} < T_{eff,cur}$, where $T_{eff,des}$ is the effective initial temperature of the destined core and $T_{eff,cur}$ is that of the current core.
2. $P_{des} < P_{cur}$, where P_{des} is the task load of the destined core in the new execution cycle, and P_{cur} is the counterpart of the current core.

The idea behind such a task migration policy is to ensure that we will likely to have a gain in terms of reduced temperature variance across the cores after the task exchange, which lead to more balanced on-chip temperature distribution across all the cores and more reduction of thermal hot spots.

The details of the distributed task migration method flow are illustrated in Fig. 5.6 (a). The task migration policy will be activated if the temperature of the current core exceeds a certain threshold temperature T_{th}. The current core communicates with its adjacent cores to check if the migration criteria (1) and (2) are satisfied. The flow starts with core a (the upper left adjacent core). If the criteria are satisfied, core a is selected as the destined core for task migration; at the same time, the thermal and load parameters of the current core $(T_{eff,cur}, P_{cur})$ are updated by the thermal and load parameters of core a $(T_{eff,a}, P_{a},)$. Then the flow continues to check with core b, if the migration criteria are satisfied, it indicates that core b is a better migration choice with even lower $(T_{eff,b},\ P_b)$ than the previous migration choice of the destined core, and thus will be selected to replace the previous choice as the destined core for task migration. The same procedure will be repeated from core c to core h. The flow continues to update the choice and will finally find



(a) task migration method flow
(Applicable to any arbitrary core)

(b) A graphic example
(the proper core is chosen)

Figure 5.6: The proposed distributed task migration method

the best choice for task migration among the 8 adjacent cores, which has the lowest value of effective initial temperature (T_{eff}) and load (P). An example flow is shown in Fig. 5.6 (b) in which the circle nodes represent the cores, the solid grey nodes indicate the checked cores, and the solid black node indicates the chosen core for task migration at the end of the task migration method flow. In this example, the new method checks with all of the 8 adjacent cores (stamped with grey) and finally selects core c as the destined core (stamped with black) to migrate the task to.

The resulting distributed thermal management method can take account of the task load, the thermal influence from each core and its adjacent region, and the thermal transient effect due to the package properties to more effectively reduce thermal variance and balance the temperature profile across all the cores.

5.2.4 The overall run-time thermal management scheme

Having discussed the distributed task migration policy with the proposed effective initial temperature as an on-chip thermal metric, it is time to present the proposed overall thermal management scheme.

Like existing works such as [58], we also assume that each task occupies an equal slice of execution time (one execution cycle), and power traces for each task could be obtained from OS or predicted from the history of the power traces using some estimation methods like time series prediction methods. We assume that the multi/many-core microprocessor is executing different sets of tasks in different execution cycles (denoted as EXE cycle) as shown in Fig. 5.7. At the beginning of each execution cycle, the proposed distributed task migration policy is activated in each core if the temperature of that core T_{core}

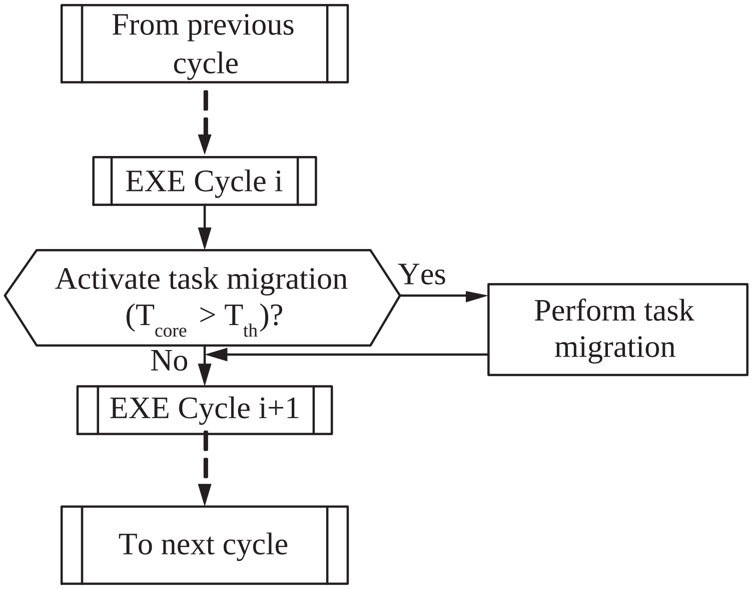


Figure 5.7: The run-time thermal management framework

exceeds a certain threshold (T_{th}), and task migration between two adjacent cores is performed if the task migration criteria discussed in section 5.2.3 is satisfied. In this way, the tasks are scheduled to balance the temperature profile of the multi/many-core microprocessors in the new execution cycle so that the on-chip temperature gradient and occurrence of thermal hot spots during the task execution is reduced. The algorithm flow of the proposed thermal management scheme is summarized in Algorithm 1.

In this framework, we focus on developing new criteria of the task assignment scheme that could more effectively balance the temperature profile in the multi-core system from algorithm level prospective. Thus, the resulting temperature profiles given by different task migration schemes are the primary concerns of this study, like [56, 60]. And also, as in [59], we assume that the migration overhead is small comparing with the task execution cycle, and thus the overheads between two task execution cycles are negligible.

```
Algorithm 1 Distributed task migration flow
Input: Task loads, many-core processor configuration
Output: Optimized temperature distribution
  Start simulation at initial package temperature
  for each execution cycle do
    1. Simulate power traces under different task loads.
    2. Obtain temperature responses of the many-core microprocessor (via thermal sensors
    or estimation).
    if migration criteria is met then
      Perform distributed task migration using the proposed scheme in Fig. 5.6 core by
      core.
    end if
  end for
```

5.3 Numerical results

5.3.1 Experiment setups

The proposed thermal management method is implemented using Matlab 7.0, and Hotspot [6] is used to build the thermal model based on the configuration of the many-core microprocessor and simulate the temperature responses.

A 100-core processor system with 10 × 10 configuration as shown in Fig. 5.8 is used to evaluate the proposed method. Each core of the processor has geometry of 4 mm × 4 mm, and the thickness of the processor chip is 0.15 mm. More detailed thermal package structure and material properties are summarized in Table 5.1, in which k denotes thermal conductivity, and c denotes specific heat. The heat convection capacitance of the heat sink of the 100 core processor is set to 4600 J/K in model configuration of Hotspot with boundary convection rate of 10 $W/(m^2K)$, and the starting temperature of the chip package is set to be 60 ^oC. The sampling interval of the thermal data of Hotspot is set to be 30 μs to preserve simulation accuracy.

Table 5.1: The package structure and thermal properties of the many-core microprocessor.

Components	Chip	Heat Spreader	Heat Sink
Thickness(mm)	0.15	1.00	6.90
k ($W/(mK)$)	100.0	400.0	400.0
c ($J/(m^3K)$)	1.75×10^6	3.55×10^6	3.55×10^6

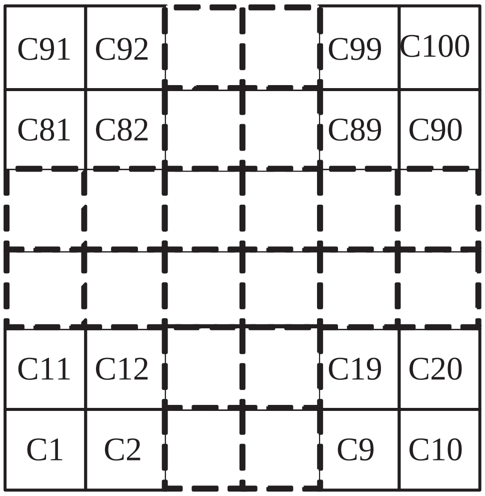


Figure 5.8: The configuration of the 100-core microprocessor die.

We used Wattch as the architecture level power analysis tool [91], and simulate the detailed transient power traces using 16 different dynamic workloads (ammp, apsi, bzip2, equake, galgel, gcc, lucas, mesa, parser, twolf, vpr, applu, art, crafty, fma3d, gap) from SPEC2000 benchmarks. During the simulation, the proposed distributed task migration policy will be applied to each core to optimize the task assignment at the beginning of each execution cycle (2048 time steps), and the simulation consists of 20 task execution cycles in total.

To evaluate the proposed thermal management method, we first look at the variations of on-chip temperature under different thermal management methods using both real temperature and effective initial temperature. Then we look at the numbers of thermal hot spots given by the proposed method and compare it against the existing and alternative methods.

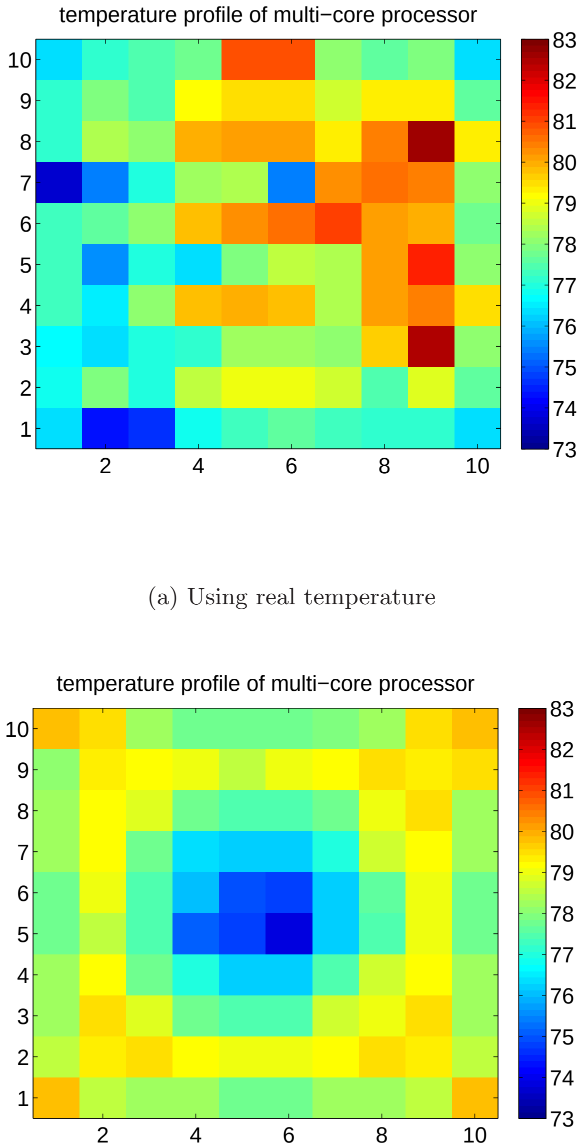
5.3.2 Task scheduling based on the new effective initial temperature

Now we first demonstrate the effectiveness of the task scheduling technique based on the proposed *effective initial temperature* metric.

The 16 benchmark tasks with different dynamic workloads are randomly assigned to the cores of the 100-core chip in each cycle of task execution, and we set the threshold temperature $T_{th} = 70^oC$, and the dynamic thermal management would be activated once the temperature of the cores exceeds this threshold value.

We first show that by using the newly proposed temperature metric, the resulting scheduling scheme can lead to more reduction of the temperature variance. Fig. 5.9 shows temperature differences (in ^oC) across all the 100 cores under these two methods at the end of 20th task execution cycle. It clearly shows less temperature divergence among all the cores by using the proposed effective initial temperature.

The resulting temperature variances (TempVar) and maximum temperature differences (MaxTempDiff) at the end of the 20th task execution cycle under different task scheduling scheme are also summarized in Table 5.2, in which the notations 'Centralize-Teff'



(a) Using real temperature

(b) Using the proposed effective initial temperature

Figure 5.9: On-chip temperature distribution across processor cores at the end of 20th task execution cycle

represents the centralized method using the proposed thermal metric T_{eff}; 'Centralize' represents the conventional centralized method using real temperature.

Table 5.2: temperature variance and maximum temperature difference

Methods	TempVar	MaxTempDiff
Centralize-Teff	1.242 $^oC^2$	5.961 ^oC
Centralize	1.666 $^oC^2$	8.858 ^oC

In addition to showing the temperature variation reduction, it is interesting to note that the new method automatically tries to find cores with better heat removal capability for heavier tasks. Fig. 5.10 shows snapshots of the task configurations across the 100-core under different centralized task scheduling scheme at the beginning of the 20th task execution cycle with the thermal management being activated. The tasks with light or heavy workloads, are configured across the 100 cores to balance the on-chip temperature profile based on different task migration criteria. Fig. 5.10 (a) is the task profile under the conventional centralized task scheduling method using real temperature while Fig. 5.10 (b) is the task profile under the centralized task scheduling method using the proposed effective initial temperature. We can see that the on-chip task configuration differs significantly. Under the conventional method, the task assignment depends only on the temperature of that core, and we can observe that first, the heavy tasks are getting too concentrated, which will lead to elevated temperature in the new execution cycle. Second, many of these heavy tasks are not scheduled to the cores that have more efficient heat removal capability, like the cores at the corner of the core region of the chip.

However, when looking at the task configuration under the centralized task scheduling method using the proposed effective initial temperature shown in Fig. 5.10 (b), we could clearly observe the following, which is almost opposite to the case in Fig. 5.10 (a): First, the heavy tasks are separated. Second, the heavy tasks are moved away from the center to the corner of the core region, where the heat removal is more efficient as those corner regions have shorter thermal paths to heat sink and ambient and less thermal capacity to store the thermal energy for a given initial temperature.

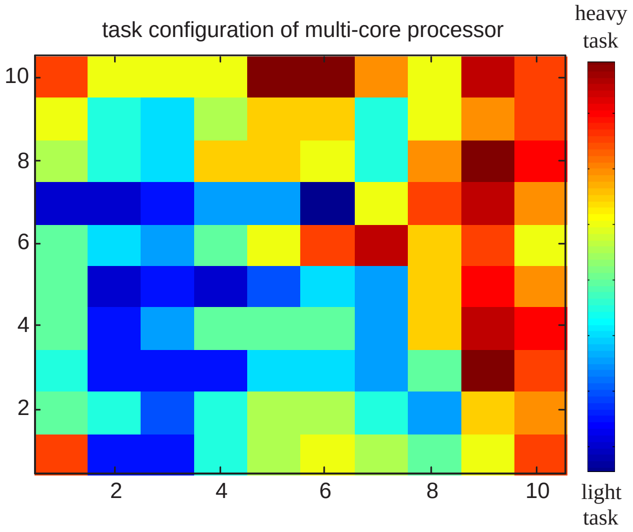
We remark cores at the corner may not alway have better heat removal capability than the middle ones in real chips. For instance, for emerging 3D stacked ICs, the heat removal capability of a core will also depends on the locations of (thermal) through silicon vias in a layer. The results shown in Fig. 5.10 (b) actually depends on the specific structure and boundary conditions used for the many-core chip structure we developed. However, what we want to stress is that the proposed method can automatically consider the transient effects in terms of the initial temperature, thermal conductance and capacitance of the cores to make better scheduling decision.

5.3.3 Performance evaluation of distributed thermal management

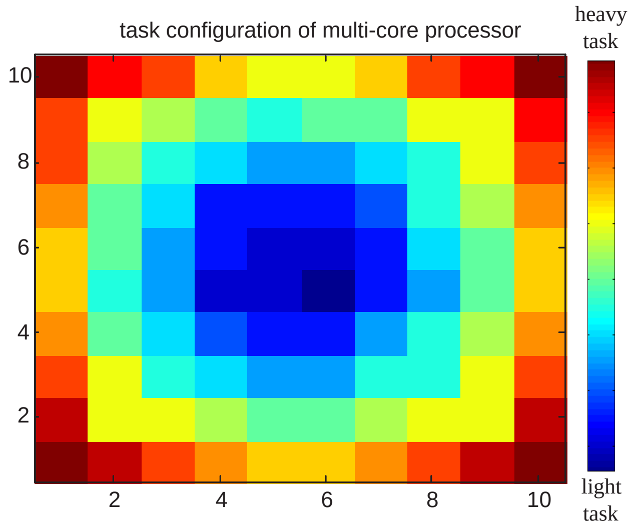
Now we show the results from the proposed distributed task migration based on the proposed T_{eff}. Since the T_{eff} now is computed distributively in localized package region, it will give more optimal task scheduling results comparing with the alternative distributed method that directly use local temperature.

First, we will compare different thermal management methods in terms of the temperature variance and number of thermal hot spots during the task executions.

We compare the result of the proposed task migration schemes against the conventional ones and the recently reported DTB-M in [58] using the same 16 dynamic workloads



(a) Using real temperature



(b) Using the proposed effective initial temperature

Figure 5.10: On-chip task configurations at the beginning of the 20th task execution cycle under centralized task scheduling methods

of SPEC2000 benckmarks on the 100 core processor platform as in the previous section 5.3.2. Fig. 5.11 shows the comparison of transient temperature variance, in which the notation 'Centralize', and 'Centralize-Teff' are the same as in the previous section 5.3.2, and 'Dist-Teff' represents the distributed method using effective initial temperature, which is also our proposed distributed thermal management method shown by the task scheduling framework in Fig. 5.6; 'Dist' represents the distributed method using the same framework in Fig. 5.6, except it uses real temperature instead of the proposed effective initial temperature; 'DTB-M' is the existing distributed method in [58].

The centralized method using T_{eff} achieves the best results in term of reducing the on-chip temperature variances among all these methods, while the proposed distributed thermal management method using T_{eff} shows comparable performance with the centralized thermal management method using real temperature. Among the three distributed thermal management method, the proposed one that uses T_{eff} for task migration leads to less temperature variance than the counterpart that uses real temperature, and both are more efficient than the existing DTB-M in terms of temperature variance reduction. In general, comparing with centralized methods, we observe reasonable performance degradation of the distributed counterparts in terms of optimizing on-chip temperature variance. However, distributed method is more practical and affordable method in multi-core systems since it would significantly reduce the communication costs to the control center.

The reduction of on-chip temperature variance could help to reduce the thermal hot spots. Fig. 5.12 shows the statistics of thermal hot spot occurrence above different temperature levels during the task execution. The centralized method using T_{eff} achieves

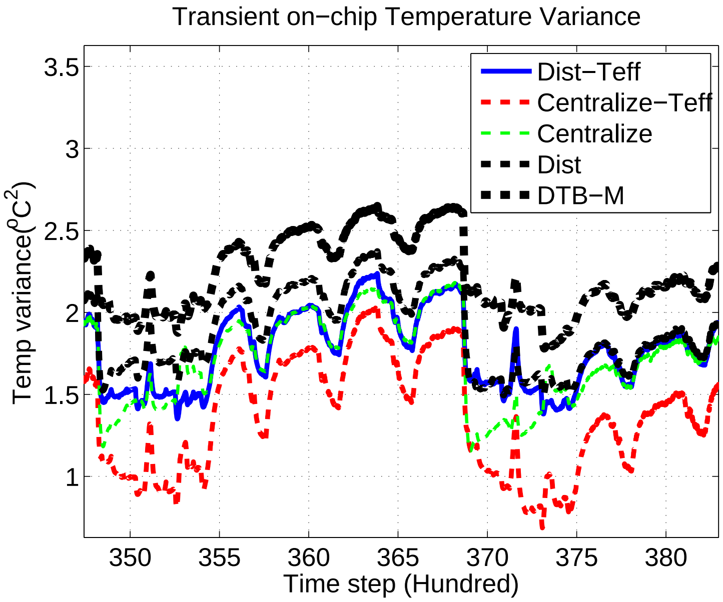


Figure 5.11: Comparison of the transient temperature variance under different thermal management policy

the best thermal hot spot reduction, and the proposed distributed method shows comparable reduction of thermal hot spot comparing with the centralized method that uses real temperature to guide task migrations. Comparing with the alternative distributed counterpart method that uses real temperature, the proposed method leads to 21% reduction of thermal hot spot occurrence during task executions. Comparing with the existing DTB-M, the number of thermal hot spots encountered during task executions is reduced by 44% under the proposed distributed task migration method. Thus, as a distributed task migration method, our proposed method can remove on-chip thermal hot spots more effectively comparing with other distributed methods.

Fig. 5.13 shows the comparison of average temperature of the chip employing different thermal management methods. We can observe that the centralized method with T_{eff} gives the best results. Among the three distributed methods, the DTB-M method does reach the lowest average temperature among the distributed thermal management method, the improvement on the average temperature over other task migration methods, however, is marginal. The result showing in this figure confirms that improving the average temperature may not always lead to better reduction of temperature variations and related thermal hot spots as shown in Fig. 5.11 and Fig. 5.12.

Hence, these results clearly demonstrate that the effective initial temperature T_{eff} is a better thermal metric comparing with real temperature, and the proposed distributed task migration method would lead to reduced temperature variance and reduced number of thermal hot spot during the task execution, yielding more balanced on-chip temperature distribution across all the cores comparing with other distributed methods. Comparing with the existing method that targets at reduction of average temperature, it is more important to balance the temperature and reduce the temperature variations to effectively eliminate thermal hot spots.

5.4 Summary

In this chapter, a new distributed task migration technique for thermal management is proposed to reduce the on-chip temperature variance and thermal hot spots during the task execution for many-core microprocessors. First, we introduce a new temperature metric, which considers the transient thermal effects based on dominant frequency domain moment matching techniques, to guide the task migrations. Second, we show how the new temperature metric can be computed in a localized way for distributed thermal management. Experimental results on a 100-core microprocessor shows that the proposed

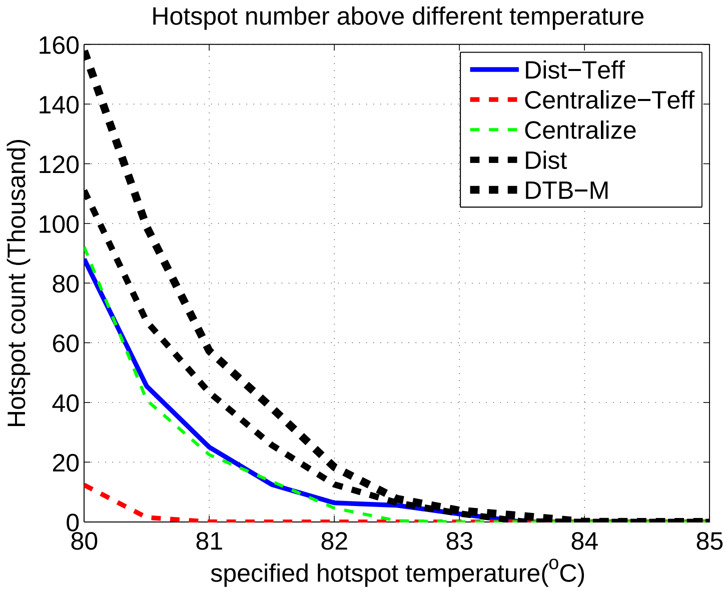


Figure 5.12: The occurrence of thermal hot spots above different temperature levels during task execution.

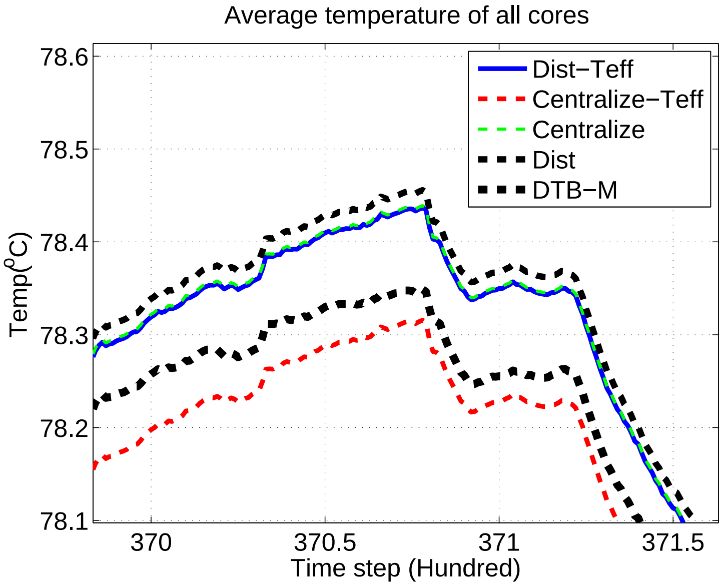


Figure 5.13: Comparison of on-chip average temperature during task execution

distributed method works more effectively to reduce the number of thermal hot spots (21% more thermal hot spot reduction comparing with the alternative approach under the same distributed framework, and 44% more thermal hot spot reduction comparing with the existing distributed thermal management methods). The results also suggest that the proposed method leads to smaller temperature variations across the many-core microprocessor, which indicates that more balanced temperature profile has been achieved. Also, we observe comparable performance when comparing our proposed method with the centralized control method that optimizes task scheduling based on real temperature information.

Chapter 6

Dynamic Reliability Management Considering Resource-Based EM Modeling for Multi-Core Microprocessors

Thermal issue could exacerbate reliability problems, and make it more challenging than ever before as technology node scales down. This chapter presents a new approach for dynamic management – a system-level reliability management technique for multi/many core microprocessors. In the new approach, the electro-migration (EM) induced mean time to failure (MTTF) at the system level is modeled as a resource, which is abstracted from a more physics-based EM model at the chip physical level. A core of multi-core systems can spend the MTTF resources at different rates specified by the temperature and the related power consumption. Such resource-based EM models allow more flexible EM-reliability management for multi/many-core systems. On top of the new EM model, we borrow the concept of task migration from thermal management method, and propose a reliability guided task migration method to explicitly balance the consumption of EM resources for all the cores. The new method will lead to the equal chance of failure of these cores, which will maximize the life time of the whole multi/many core system. With balanced MTTF among all the cores, low power mode control is enabled to compensate the excessively consumed life time of all the cores when the chip is assigned with heavy tasks for a certain period of time. In this way, the MTTF of all the cores could be compensated to satisfy the requirement, giving the multi/many core system more flexibility to handle heavy task assignment when needed.

Experimental results on a 36-core processor platform shows that the proposed task migration scheme could balance the life time consumption of all the cores, and maintain evenly-distributed MTTF slacks across different cores, while the traditional temperature based task migration scheme leads to diverged MTTF consumption. The results also show that the balanced MTTF consumption could be easily compensated by switching to a low power mode, which could not be achieved if MTTF consumption diverges across different cores.

The rest of this chapter is organized as follows. In section 6.1, the problem of reliability modeling and the resource based reliability management is presented. In section 6.2, the new EM reliability model based on strain analysis is presented. In section 6.3, the proposed reliability resource management scheme is presented, and its results are evaluated and discussed in 6.4. Section 6.5 concludes this chapter.

6.1 Problem formulation and reliability modeling

As the reliability and performance are intrinsically conflicting factors, one has to consider them jointly at system level optimization as shown in the existing works [62,64,65]. To consider and model the reliability effects due to various failure mechanisms, many existing works use the so-called sum of failure rate model (SOFR) [92] to compute MTTF of a whole system from its components. SOFR model consists of the competing risk model which estimates the failure rate of each component and the series model which estimates the failure rate of the system based on failure rate of each component. Such SOFR models only work when the following conditions are met: first, each failure mechanism proceeds independently (they do not affect each other). Second, the whole system fails when the first of its component fails. For EM-related reliability on a practical power grid network, unfortunately, neither conditions are met. In this work, we try to mitigate those problems by using a new physical-based EM model and redundancy-aware analysis techniques to compute the MTTF of a power grid for given current sources, supply voltage and temperatures.

For multi-core microprocessors, the optimization could be achieved through proper reliability management of resources and tasks. In this work, we treat MTTF as a reliability resource that could be consumed and controlled during task executions. For optimization purpose, instead of dealing with the difficult trade-off between performance and reliability in a given period, in this work, we target at finding a way to compensate the life time if it is excessively consumed during a certain period when the processor is loaded with heavy tasks.

6.2 New physics-based EM modeling and analysis

EM is a physical phenomenon of the migration of metal atoms along a direction of applied electrical field. Atoms (either lattice atoms or defects/impurities) migrate toward the anode end of metal wire along the trajectory of conducting electrons. Over time, the lasting unidirectional electrical load increases these stresses, as well as the stress gradient along the metal line. In some cases, usually when a line is long, this stress can reach a critical level, resulting in a void nucleation at the cathode and/or hillock formation at the anode end of line.

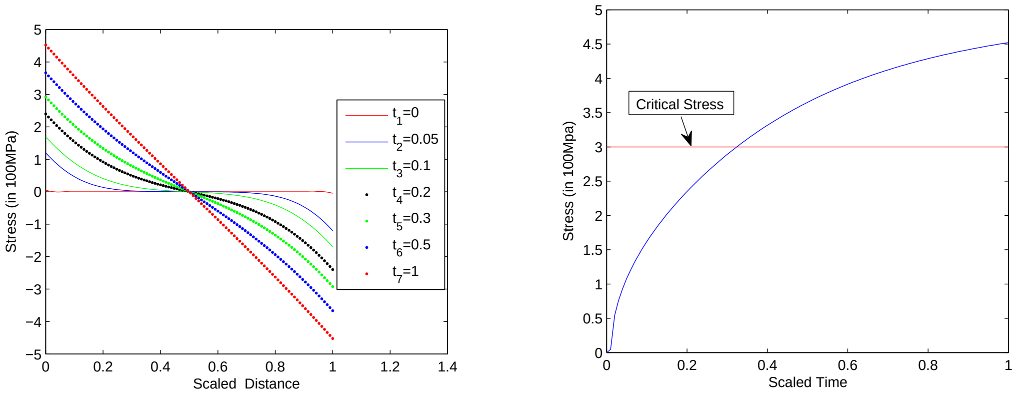
Degradation of the electrical resistance of interconnect segments, caused by void nucleation and growth, can be derived from the solution of kinetics equation describing the time evolution of stress in the interconnect segment [93–96]. Indeed, the obtained instant in time when stress reaches the critical value $\sigma = \sigma_{crit}$, indicates the void nucleation time; extracted kinetics of the void volume evolution governs the evolution of the segment electrical resistance. While being successful in simulating the EM related physics in the frame of the finite-element analysis (FEA) [96], this type of modeling cannot be employed for the purpose of the analysis of stress evolution caused by the electrical load in hundreds of millions interconnect segments. A reason is the enormous size of the computational problem. To address this problem the physics-based analytical compact model like the description of the void nucleation time and kinetics of void size evolution should be developed.

6.2.1 Void dynamics: Nucleation and growth phases

Development of such analytical formulation was proposed by Korhonen [93] and further developed by other researchers; see for example [94–96]. Since the atomic flux divergence results in the volumetric strain, the derived one dimensional diffusion-like equation for the hydrostatic stress field $\sigma(x,t)$, takes the form, [93]:

$$\frac{\partial \sigma}{\partial t} = \frac{\partial}{\partial x}\left[\kappa\left(\frac{\partial \sigma}{\partial x} + \frac{eZ\rho j}{\Omega}\right)\right] \tag{6.1}$$

Here, $\kappa = D_a B\Omega/kT$, where D_a is the atomic diffusivity, and B is the bulk modulus, Ω is the atomic volume; e is the electron charge, eZ is the effective charge of the migrating atoms, ρ is the wire electrical resistivity. Solution of this initial-boundary value problem is the infinite series [93].



(a)

(b)

Figure 6.1: (a) EM-stress distribution change over time in a simple metal wire. (b) EM-stress evaluation versus time.

Fig. 6.1(a) shows the EM-induced stress development for a single metal wire over time from the finite element analysis for a given current density and temperature setting.

Fig. 6.1(b) shows the stress evaluation over time. Each time unit here is 10^7 seconds. During this process, tensile stress (positive stress) will be developed at the cathode side (the left node), while compressive stress (negative stress) will be generated at the anode side of the wire (right node). When the tensile stress hits critical stress, void will be generated, which is called nucleation time. After this, the void will grow, which will increase the resistance of the wires until the growth stops. In contrast, existing EM model only consider one of such two processes modeled by Black's equation. Also when the wire hits its MTTF, the wire is treated as open circuit, which will over-estimate EM-impacts in circuit reliability.

To mitigate those mentioned problems, a more physics-based EM model has been proposed recently for full-chip reliability analysis [97,98], which is the basis for the proposed work.

In this new EM model, the EM development process consists of two phases - the nucleation phase and the growth phase. In the first nucleation phase, a closed-form expression to compute the nucleation time (t_{nuc}) is given, which is a function of current density, temperature, the residual stress of the wire due to thermal and other effects as well as other wire geometry and material parameters. Dependence of t_{nuc} on grain size allows one to introduce a simple statistical model for void nucleation at the line cathode edge. Employment of the lognormal distribution as the experimentally proven distribution of the grain size in the polycrystalline metal line provides different t_{nuc} for the geometrically identical metal lines loaded with the same electrical current densities.

The second phase is the void size growth: voids are formed at t_{nuc} and grow at $t > t_{nuc}$. The wire resistance starts to increase over the time in the growth phase. As a result, the p/g network becomes a time-varying network and its voltage drops will keep changing over the time.

6.2.2 EM assessment at power grid level

Because of the concern with the long-term average effects of the current, in EM related work a DC model of the power grid is generally assumed [99]. As a result, we consider only the EM-induced kinetics of the power grid network resistances. In our problem formulation, each mortal wire, which subjects to the EM impact, will start to change its resistance value upon achieving the nucleation time. As a result, we end up with the power grid systems, which is a linear, time-varying and driven by the DC effective currents, which is modeled as $G(t)v(t) = I_{eff}$, where, $G(t)$ a $n \times n$ time-varying conductance matrix; I_{eff} is the effective DC current source vector; $v(t)$ is the corresponding vector of nodal voltages and n is the size unknown voltages. In our problem, the time scale is the EM time scale, which can be months or years.

In the new EM-induced reliability analysis algorithm for p/g networks, we compute the voltage drops of the grids at fixed EM time step. The resistance of one or more wires begins to change (increase) starting with their nucleation times. At each time step, we collect new wires whose nucleation times were reached, and compute the new resistance for existing wires in the growth phases and corresponding voltage drops of the whole grids. This process is repeated until the voltage drop of one or more nodes exceeds the critical voltage drops allowed (say 10% of Vdd).

6.2.3 System level EM-reliability resource consumption model

Given the new physics-based EM model, we now introduce our system level EM-reliability resource consumption model. Our observation is based on the fact that EM-induced stress development process can be viewed as source consumption process, in which, the difference between the current stress and the critical stress, or the stress slack is the source. Once electrical current starts to flow through the wire, the EM process starts to spend the resource at a rate, which is the function of the temperature and current density. We notice that treating the EM as a resource was first introduced in [100]. But this work is still based on the traditional Black's equation.

At the system level, instead of using stress directly, we treat the life time of the processor specified by MTTF as a resource that could be consumed as the chip works. We define the specified MTTF as a nominal value, denoted as $MTTF_N$, which is the intended or required life of the chip under a typical temperature and power setting for a core or system. For example, a processor has nominal MTTF of 10 years under temperature of 70^oC and working power of $20W$ as a specification. However, in reality, MTTF varies under different temperature and power settings. Hence, we denote the real MTTF as $MTTF_R$.

According to [100], the life time of the chip due to EM could be expressed as

$$lifetime = \frac{1}{(\sum_{k=1}^{n}(\Delta t_k \frac{1}{MTTF_{R,k}}))/T} \tag{6.2}$$

where $MTTF_{R,k}$ is the actual MTTF under the k-th power and temperature settings for Δt_k period, assuming the chip works through n different power and temperature settings and $T = \sum_{k=1}^{n} \Delta t_k$. As a result, we propose a new EM-reliability resource consumption model. In this model, we treat the nominal MTTF ($MTTF_N$) as the resource to consume, which actually is the *life time* defined in (6.2), and we define an average rate to consume $MTTF_N$ as the amount of life time the working chip consumes during each unit time interval. In the nominal case, the chip is working under its specified temperature and power setting, and it has life time given by $MTTF_N$. Hence, the amount of life time consumed by the chip in each second is 1 EM second, that is to say, the nominal average consumption rate is $r_N = 1$. In reality, depending on different physical settings, the average consumption rate could be either higher or lower than its nominal rate, and we define average consumption rate as

$$r_R = \frac{MTTF_N}{MTTF_R} \tag{6.3}$$

in which the life time in real case ($MTTF_R$) could be estimated by the new proposed reliability model in the previous sub-sections. With this definition, we could see that if $MTTF_R > MTTF_N$, then $r_R < r_N$, which indicates that the chip is consuming its nominal life time at lower rate, and thus the real life time is longer than the nominal one. Conversely, if $MTTF_R < MTTF_N$, then $r_R > r_N$, which indicates that the chip is consuming its nominal life time at higher rate, and thus the real life time is shorter than the nominal one. Hence, instead of saying MTTF changes, we perceive MTTF as a constant resource, which is given by $MTTF_N$, and it is the consumption rate (r_R) of MTTF that determines the real life time of the chip. If the time integration of EM slacks over a period is zero, then the life time or MTTF of the chip during that period will the $MTTF_N$ as predicted by (6.2).

6.3 EM-reliability management method

6.3.1 EM-reliability resource based task migration

First, we present the new task migration method to balance the EM-reliability resources, which is different from the conventional task migration method that targets at improving on-chip temperature profile. According to the definition of average MTTF consumption rate defined by (6.3), if $r_R > r_N$ persistently, it will introduce excessive consumption of MTTF, which would possibly lead to early failure of the chip if no compensation is made. In real application, it is common that $r_R > r_N$ during the period when heavy tasks are assigned to the chip, and the life time is excessively consumed during this period, while on the other hand, when light tasks are assigned to the chip, $r_R < r_N$, and less life time is consumed during this period. We define *MTTF resource slack* as the accumulative MTTF consumption difference between real case and nominal case over all different task execution periods, which is calculated through

$$S_d = \sum_{k=0} (r_N - r_R(k)) \Delta T \tag{6.4}$$

In which $r_R(k)$ is the average consumption rate during the k-th execution cycle, ΔT is the unit interval (UI) of the execution cycle, and $k = 0$ indicates that the MTTF resource slack is accumulated from the very beginning when the chip first gets powered-on, and (6.4) illustrates the followings:

- If $S_d = 0$, it indicates the overall consumption of the chip would lead to its intended MTTF. It is easy to verify $lifetime = MTTF_N$ by using (6.2) in this case.

- If $S_d < 0$, it indicates that the life time is excessively consumed for the past execution periods, and it requires compensations in future to avoid early failure.
- If $S_d > 0$, it indicates that the life time is consumed less than its nominal rate for the past execution periods, and it allows increased consumptions in future without causing early failure.

In a multi-core system, for each core i, we could calculate MTTF resource slack for core i at the end of each task execution cycle, and denote it as $S_d(i)$. We could also characterize the average power of the tasks in the coming execution cycle for each core. Assuming that the multi-core processor has N cores, and the average power of the tasks on each core are denoted as P_1,P_2,...,P_N, and the MTTF resource slack for each core are denoted as $S_d(1)$, $S_d(2)$,...,$S_d(N)$. To balance the EM-reliability of all the cores, we sort out the order of power consumptions and that of the MTTF resource slack, and assign the highest power to the core with highest value of S_d, and assign the second highest power to the core with the second highest value of S_d, and so on. The overall task migration scheme is shown in Fig. 6.2, in which S_d is calculated based on (6.4) and (6.3), using the estimate $MTTF_R$ through our proposed reliability model. In this way, the MTTF consumption of different cores could be balanced, which means that all the cores will be targeting at similar length of life time, avoiding early failure of some cores due to continuously heavy load assignment.

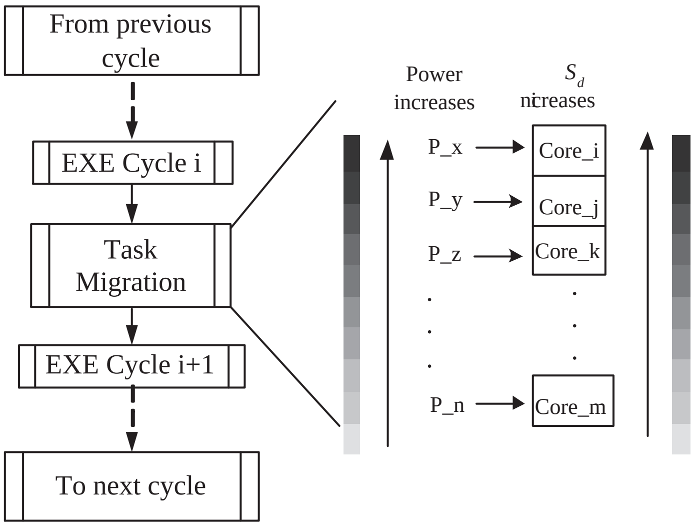


Figure 6.2: The proposed reliability resource based task migration scheme

6.3.2 EM-reliability resource based low power control

With the proposed task migration scheme, the MTTF consumption of different cores is balanced so that all the cores would have comparable life time. However, task migration would not be able to compensate the excessively consumed MTTF if all the cores are loaded with heavy tasks. Hence, low power mode needs to be enabled to compensate the overly consumed MTTF later on, so that the chip could maintain its intended life time. In this application, low power mode could be implemented by a P-state with reduced supply voltage and operation frequency, which is used to control the power of the system. With MTTF consumption getting balanced across all the cores, low power mode could effectively balance the life time of the cores as will be discussed later in this sub-section. Here, we first introduce the overall concept of the low power mode control to compensate the excessively consumed MTTF of a single core as illustrated in Fig. 6.3.

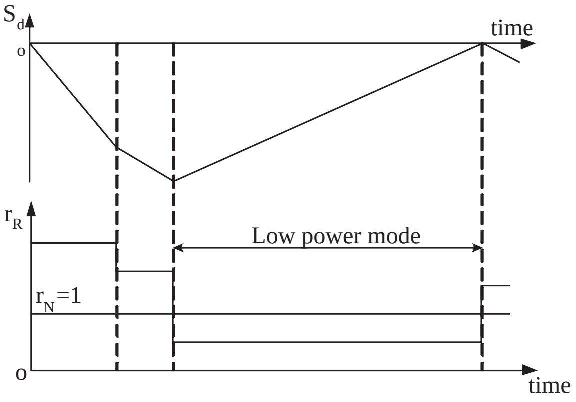
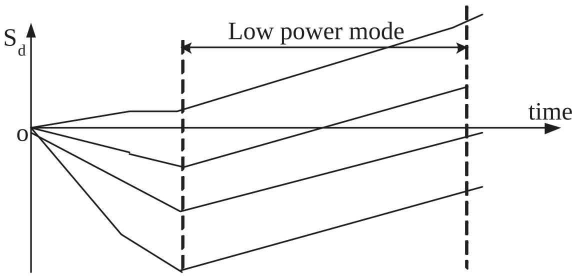
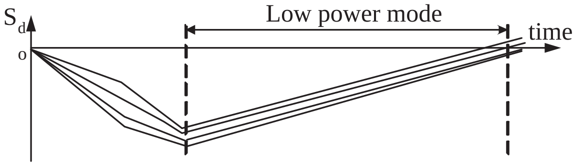


Figure 6.3: Low power mode compensation scheme (one core)



(a)



(b)

Figure 6.4: Low power mode compensation scheme for a 4-core system with (a) imbalanced MTTF consumption (b) balanced MTTF consumption

From this Fig. 6.3 and (6.4), it is clear that S_d will start negative accumulation when $r_R > r_N$, which indicates faster consumption of MTTF comparing with nominal case. However, once the chip switches to low power mode with $r_R < r_N$, the excessively consumed MTTF starts to get compensated, and eventually, the consumed MTTF gets compensated to the nominal consumption of MTTF when $S_d = 0$ over time.

Since the task migration scheme has succeeded to balance the MTTF consumption of all the cores, low power mode in this case could effectively compensate the excessively consumed life time of all the cores. Fig. 6.4 shows the examples of a 4-core system to illustrate the importance of balancing MTTF consumption of all cores before low power mode could be used to effectively compensate the excessively consumed MTTF of the muti-core systems. Fig. 6.4 (a) shows a 4-core system with imbalanced MTTF consumption in which some of the cores are excessively consuming MTTF while other cores are not. In this case, the low power mode would not be able to compensate the consumption of all the cores. But on the other hand, in Fig. 6.4 (b), the low power mode with $r_R < r_N$ could effectively compensate all the excessively consumed MTTF, and make the MTTF resource slack to be zero ($S_d = 0$), which regulates the MTTF of all the cores at its nominal value.

Since the MTTF consumption could be compensated by low power mode with the proposed task migration scheme, it clearly implies the following:

- Excessive life time consumption is not necessarily causing early failure as long as it could be compensated.
- The processor could 'borrow' some life time from future and use it to fulfill the completion of heavy task assignment in a certain period.

Hence, in terms of MTTF resource compensation, we propose the following scheme to trade off heavy task execution and MTTF requirement.

- When working in high performance mode, the multi-core processor continues to keep this mode for N cycles after over 80% of cores have $S_d < 0$, and then switch to low power mode starting from the $N + 1$ cycles.
- When working in low power mode, the multi-core processor continues to keep this mode for M cycles after over 80% of cores have $S_d > 0$, and then switch to high performance mode starting from the $M + 1$ cycles.

In reality, the number N and M could be specified by the user based on the needs of handling heavy load and compensating life time. In this way, the required MTTF of all the cores could be maintained through low power mode compensation, while the processor has the flexibility to handle heavy task assignment when needed.

6.4 Numerical results and discussions

The proposed reliability model is implemented in C++, and the task migration and low power mode control framework is built in Matlab environment. Hotspot [6] is used to build the thermal model based on the configuration of a 36 core processor, and Wattch [91] is used as the architecture level power simulation tool. In this work, we extend its functionality to calculate power under different supply voltage and working frequency. The dynamic workloads from spec2000 benchmarks (ammp, apsi, bzip2, equake, galgel, gcc, lucas, mesa, parser, twolf, vpr, applu, art, crafty, fma3d, gap, gzip, mcf, mgrid, swim vortex) are used as tasks to simulate power traces.

6.4.1 A walk-through example for the proposed reliability estimation

In this section, we walk through an example to illustrate the proposed method. We first set the supply voltage to be 2.65V and operation frequency to be 950MHZ for a single core processor under test. The task “ammp” is used to generate power and thermal traces. The supply voltage of this single core processor chip, average power and temperature of each function unit over the transient simulation period could be obtained and used as the input of the proposed reliability model. The new reliability model estimates that the nucleation time (the time when the first void is formed in power grid networks) to be 11.22 years and the MTTF (the time when the chip fails) to be 19.31 years. The degradation of the IR drop is shown in Fig. 6.5. The supply voltage of the function units maintains to be 2.65 V as the value of its initial setup with the IR drop remaining to be zero before nucleation time when the first void is formed. Once it reaches the nucleation time, the formation of the void increases the resistance of the interconnect, resulting in increased IR drop in the power delivery networks. The IR drop continues to increase as the number of voids in power grids increases over years, which eventually leads to the failure of the chip. MTTF is the time when the IR drop increases to 10% of the original supply voltage (0.265V in this case).

In fact, the MTTF really depends on the executing tasks, that is heavier tasks lead to shorter life time and lighter tasks lead to longer life time. As an example, the estimated values of MTTF under different task executions are plotted in Fig. 6.6, in which MTTF shows significant different values from one task to another.

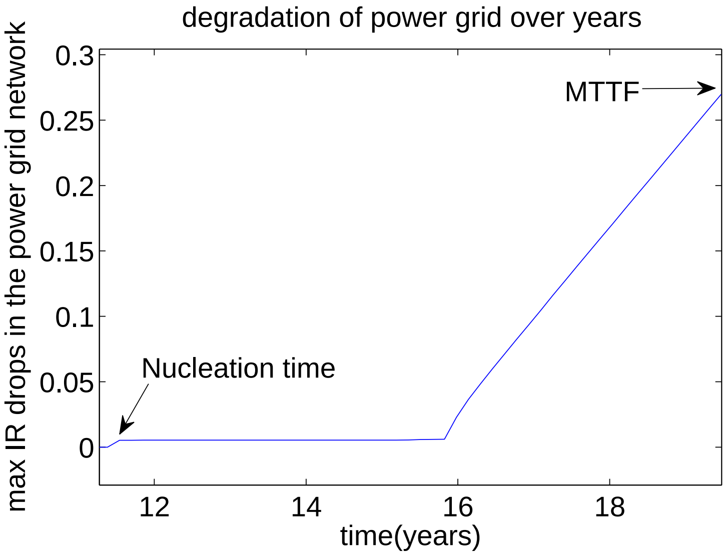


Figure 6.5: The increase of IR drop in power grid over years

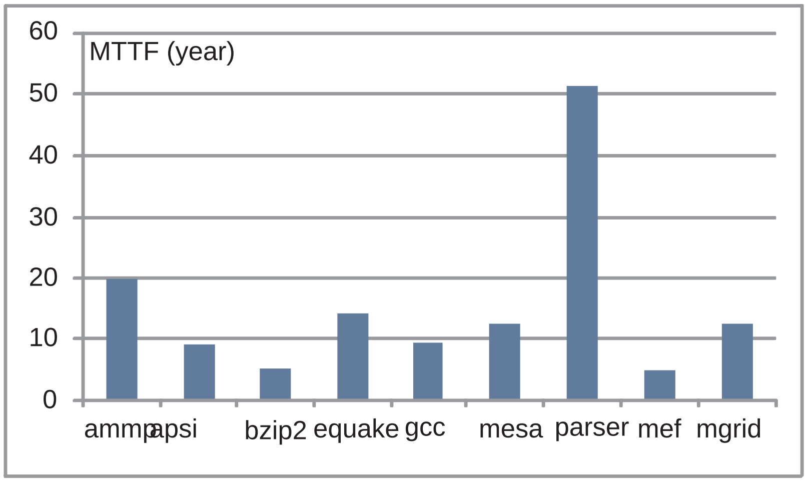


Figure 6.6: The different MTTF values under different task loads

6.4.2 Evaluation of the proposed methods

In this part, we will be testing how the MTTF of the cores gets accumulatively consumed during the task executions. We remark that the task migration for reliability balancing management could occur for relatively long period of time (hours, days or even weeks) in practical cases. Hence, in practice, the migration overhead is just negligible, and thus we do not need to consider any types of migration overhead in our test. Also, since it is impractical to run software based testing framework for task execution length over days and years, we have to 'scale down' the time frame in our testing environment. To do this, we scale down the length of task execution interval, and perform task migration at the end of each interval. In this way, we could find out how the MTTF are consumed for each core over several task execution intervals. In our testing environment, we keep each time step to be 30 us for power simulation and thermal simulation, and specify each task execution interval to be 362 time steps. The nominal MTTF for each core is set to 15 years, and the 36 core processor is used as testing vehicle. In our framework, our processor has two different P-states, one is high performance P-state, with 1 G of frequency and 1.4 V of supply voltage; and the other is low power P-state, with 800 M of frequency and 1.12 V of supply voltage.

First, we disable low power mode control and use the proposed task migration method to balance the reliability across all the cores. The experimental results of the MTTF resource slack as reflected by S_d is shown in Fig. 6.7, in which the unit of S_d is normalized to UI (unit interval of the task execution as defined by ΔT in (6.4)), and the unit of time is measured by task execution cycles. We could clearly see that MTTF consumption is balanced across all the cores, and thus, the MTTF is consumed in similar rate, which indicates that all the cores are regulated to have similar life time.

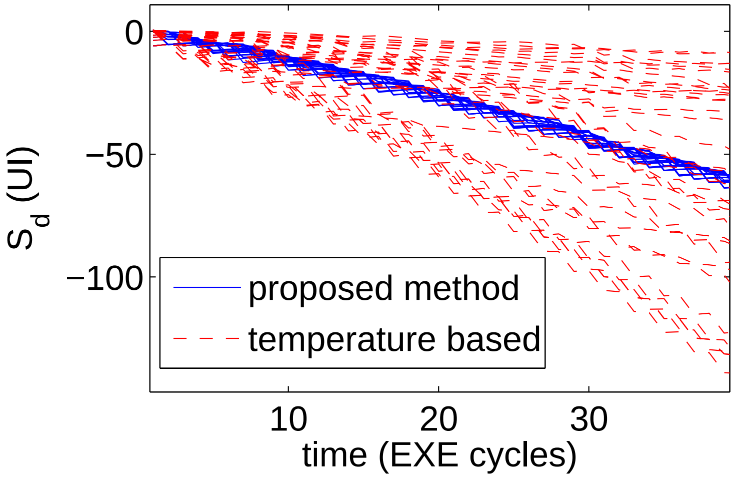


Figure 6.7: MTTF resource slack (represented by S_d) under different task migration schemes

In addition, as a comparison, we also implement temperature based task migration scheme that migrates the heaviest tasks to the cores with lowest temperature and testify its performance in terms of MTTF resource slack. As demonstrated by Fig. 6.7, the temperature oriented task migration could not balance MTTF consumption, and we could clearly observe that some of the cores are consuming significantly more MTTF than others, leading to imbalanced MTTF consumption.

Since the result in Fig. 6.7 has confirmed that the proposed task migration scheme could balance the MTTF consumption across all the cores, making the cores targeting at similar life time, the low power control could now effectively compensate the MTTF consumption by switching the processor to low power mode. In this part of experiment, the low power control mode is setup as the following: In high performance state, if over 80% of the cores have $S_d < 0$ for over 10 task execution cycles ($N = 10$), the processor switches to low power mode. In low power mode, if over 80% of cores have $S_d > 0$ for over 1 cycle ($M = 1$), the processor switches back to high performance state. After enabling the low power control, as Fig. 6.8 shows, the processor starts with high performance mode, in which the MTTF is excessively consumed for all the cores, and S_d of all the cores are decreasing simultaneously. After 10 task execution cycles after 80% of the cores have $S_d < 0$, the processor switches to low power mode. Once the processor switches to low power mode, values of S_d start to accumulate in positive direction because all the cores are now consuming MTTF at lower rate than the nominal rate, and we could clearly observe that S_d values calculated from different cores get effectively compensated to around 0 as the task runs under the proposed task migration scheme, which indicates that the overall MTTF consumption is close to the nominal case and the cores are targeting at achieving their required MTTF. The calculated standard deviation of MTTF resource slack by the end of the 40 execution cycle is 2.27 UI, which is converged and would not keep increasing as more tasks get executed.

On the other hand, if we use temperature based migration, low power mode could not be used to compensate the MTTF consumption, as Fig. 6.8 shows, because the MTTF consumption for different cores are completely different, and the values of S_d diverge as task runs, which indicates the life time of different cores diverges, and some cores would likely to have early failure if tasks are executed under this scheme. The calculated standard deviation of the MTTF resource slack by the end of the 40 execution cycle is 41.08 UI, which is around 18 times larger than that of the standard deviation using the proposed method. And this standard deviation will increase as more tasks are executed.

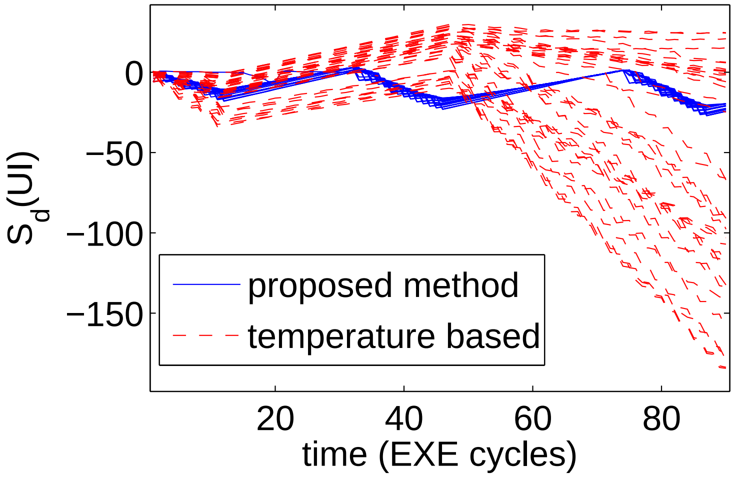


Figure 6.8: MTTF resource slack (represented by S_d) compensation using low power mode under different task migration schemes

The result from Fig. 6.8 also suggests that, the processor could be assigned with heavy load for a certain period under the proposed migration scheme. Because, as long as the cores could be balanced to have comparable MTTF, their life time consumption could be effectively compensated by low power mode.

6.5 Summary

This chapter presents a new reliability management method to balance and control the life time of multi-core processor chip due to electromigration (EM) process. A more physics-based EM model was used for more accurate prediction of the MTTF without using empirical solutions. The proposed reliability management treats the MTTF as a resource to consume during task execution, and uses task migration to balance the MTTF consumption across different cores, which leads to comparable life time among different cores to maximize the life time of the whole system. With the balanced MTTF consumption, a low power control mode can effectively compensate the MTTF consumption of all the cores after executing heavy task loads, which gives the processor the flexibility to handle heavy load when needed since the excessively consumed MTTF could be compensated later on. The experimental results confirm that, comparing with the proposed approach, the traditional temperature-based task migration approach could not balance MTTF consumption, and the low power mode could not effectively compensate the life time of all the cores.

Chapter 7

Conclusion

Driven by Moore's law, the continuously increased complexity of large scale integration inevitably leads to increased power density, under which thermal behavior modeling and management techniques become necessary and challenging to enable thermal aware chip design and alleviate on-chip thermal and related reliability constraint. This chapter concludes this dissertation with highlighted achievements, which covers four aspects in this field: (1) compact chip level finite difference 3D thermal modeling method considering lateral thermal resistance of TSVs. (2) subspace identification based top-down thermal package modeling technique in presence of highly correlated on-chip power waveforms. (3) architectural level distributed on-chip thermal management method considering package induced transient thermal effects. (4) system level reliability model and management method to balance and control life time resources in multi-core systems.

In Chapter 3, a compact thermal resistance model is proposed to model the lateral thermal resistance of TSV and TSV arrays. Both the analytical approach and the experimental results confirm that the lateral thermal resistance is a function of the geometry of each TSV (radius and thickness of insulation linear) and the space between two adjacent TSVs. The new lateral TSV thermal resistance model is fully compatible with the existing thermal modeling approaches, and could be integrated into finite difference method for thermal analysis. The experimental results show that the proposed TSV model has improved accuracy in modeling lateral thermal resistance comparing with the existing one, and it could be used to capture the lateral thermal effect using compact grids for finite difference chip level thermal analysis method, leading to improved accuracy without increase in computational complexity.

In Chapter 4, a top-down thermal modeling approach for microprocessor package thermal behavior analysis in presence of highly correlated on-chip power is proposed. The proposed method, *ThermSubCP*, uses power map approach to meet the spatial rank requirement in presence of highly correlated power inputs, improving the predictability of subspace identification method. Also, to tackle the nonlinear effects and various boundary condition of the thermal package system, piecewise linear modeling approach is proposed to increase the accuracy without increasing the model order. The piecewise modeling approach, *ThermSubPWL*, partitions the thermal nonlinearity ranges, either due to temperature or varying boundary conditions into a number of small ranges, and uses power map based subspace identification approach (*ThermSubCP*) to build thermal model for each range. The proposed method is validated on a practical microprocessor package built by commercial multi-domain/physics tool, COMSOL V4.1, driven by highly correlated on-chip power signal. The experiment results show that multiple power map configurations could significantly improve the predictability of the identified model. The new piecewise linear modeling approach can be used to model thermal behavior over wide temperature ranges or over different thermal boundary conditions due to different cooling configurations. Further, the PWL modeling technique can lead to much smaller model order without accuracy loss, which translates to significant simulation time reduction and about 10X less time to identify the reduced models compared to the linear modeling method trying to improve accuracy through using the high order models in experiments.

In Chapter 5, a new architectural level distributed task migration method is proposed as a thermal management scheme for multi-core microprocessor. The proposed method uses a new temperature metric called *effective initial temperature* to guide the task migration processes. Since *effective initial temperature* incorporates the initial temperature condition of the chip and other thermal transient effect of the package, the proposed method could make more optimized task migration decisions comparing with the conventional ones, leading to more balanced on-chip temperature distribution. In the experiment on a 100-core microprocessor, comparable performance is observed when comparing the proposed method with the centralized control method that optimizes task scheduling based on real temperature information. The experiment results show that the proposed distributed method works more effectively to reduce the number of thermal hot spots (21% more thermal hot spot reduction comparing with the alternative approach under the same distributed framework, and 44% more thermal hot spot reduction comparing with the existing distributed thermal management methods). These results demonstrate that the proposed method leads to smaller temperature variations across the many-core microprocessor comparing with other distributed method, which indicates that more balanced temperature profile has been achieved.

In Chapter 6, a new dynamic reliability management method is developed to balance and control the life time of multi-core processor chip caused by electromigration (EM). A system level life time resource based EM model is developed from physics-based derivations. The proposed reliability management treats the MTTF as a resource to consume during task execution, and uses task migration to balance the MTTF resource consumption across different cores, which leads to comparable life time among different cores to prevent early failure of heavily loaded cores. With the balanced MTTF consumption, a low power control mode is used to effectively compensate the MTTF consumption of all the cores after executing heavy task loads, which gives the processor the flexibility to handle heavy load when needed since the excessively consumed MTTF could be compensated later on. The experimental results confirm that, the proposed method could lead to balanced MTTF consumption, and effectively compensate the life time of all the cores by switching to low power mode, while the traditional temperature-based task migration approach could not.

Bibliography

[1] A. H. Ajami. Thermal integrity: a must for low-power-IC digital design. In *Electrical Design, www.elecdesign.com*, Oct. 2005.

[2] Nicholas Allec, Zyad Hassan, Li Shang, Robert P. Dick, and Ronggui Yang. ThermalScope: Multi-scale thermal analysis for nanometer-scale integrated circuits. In *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, pages 603–610, Piscataway, NJ, USA, 2008. IEEE Press.

[3] International technology roadmap for semiconductors (ITRS), 2009 update. http://public.itrs.net.

[4] M. Pedram and S. Nazarian. Thermal modeling, analysis, and management in VLSI circuits: Principles and methods. *Proc. of the IEEE*, 94(8):1487–1501, Aug. 2006.

[5] D. Brooks and M. Martonosi. Dynamic thermal management for high-performance microprocessors. In *Proc. of Intl. Symp. on High-Performance Comp. Architecture*, pages 171–182, 2001.

[6] K.Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *International Symposium on Computer Architecture*, pages 2–13, 2003.

[7] J. Burns. TSV-based 3D integration. In A. Papanikolaou, D. Soudris, and R. Radojcic, editors, *Three Dimensional System Integration*, pages 13–22. Springer, November 2010.

[8] Robert Patti. 3D integration: New opportunities for advanced packaging. In *Electrical Performance of Electronic Package and Systems (EPEPS)*, pages 1–41, October 2011.

[9] P. Benkart, A. Kaiser, A. Munding, M. Bschorr, H. J Pfleiderer, E. Kohn, A. Heittmann, H. Huebner, and U. Ramacher. 3d chip stack technology using through-chip interconnects. *Design Test of Computers, IEEE*, 22(6):512–518, 2005.

[10] G. H. Loh, Y. Xie, and B. Black. Processor design in 3d die-stacking technologies. *Micro, IEEE*, 27(3):31–48, 2007.

[11] B. Kim, C. Sharbono, Tom Ritzdorf, and D. Schmauch. Factors affecting copper filling process within high aspect ratio deep vias for 3d chip stacking. In *Electronic Components and Technology Conference, 2006. Proceedings. 56th*, pages 6 pp.–, 2006.

[12] M. Motoyoshi. Through-silicon via (tsv). *Proceedings of the IEEE*, 97(1):43–48, 2009.

[13] Jingsheng Cong, Guojie Luo, and Yiyu Shi. Thermal-aware Cell and Through-Silicon-Va Co-Placement for 3D ICs. In *Proc. Design Automation Conf. (DAC)*, pages 670–675, 2011.

[14] M.P.D. Sai, Hao Yu, Yang Shang, Chuan Seng Tan, and Sung Kyu Lim. Reliable 3-d clock-tree synthesis considering nonlinear capacitive tsv model with electrical-thermal-mechanical coupling. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 32(11):1734–1747, Nov 2013.

[15] Ankur Jain, Robert E. Jones, Ritwik Chatterjee, and Scott Pozder. Thermal modeling and design of 3D integrated circuits. In *11th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, pages 1139–1145, May 2008.

[16] José L. Ayala, Arvind Sridhar, and David Cuesta. Thermal modeling and analysis of 3D multi-processor chips. *Integration, the VLSI Journal*, 43:327–341, September 2010.

[17] Yibo Chen, Eren Kursun, Dave Motschman, Charles Johnson, and Yuan Xie. Analysis and mitigation of lateral thermal blockage effect of through-silicon-via in 3d ic designs. In *Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design*, ISLPED '11, pages 397–402, Piscataway, NJ, USA, 2011. IEEE Press.

[18] Hu Xu, F. Vasilis Pavlidis, and D Giovanni Micheli. Analytical Heat Transfer Model for Thermal Through-Silicon Vias. In *Proc. European Design and Test Conf. (DATE)*, pages 1–6, 2011.

[19] Siva P. Gurrum, Yogendra K. Joshi, William P. King, Koneru Ramakrishna, and Martin Gall. A compact approach to on-chip interconnect heat conduction modeling using the finite element method. *Journal of Electronic Packaging*, 130:031001.1–031001.8, September 2008.

[20] A. M. Sridhar, A. Vincenzi, et al. 3D-ICE: Fast compact transient thermal modeling for 3D-ICs with inter-tier liquid cooling. In *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, pages 463–470. IEEE Press, 2010.

[21] Sarang Shidore. Compact thermal modeling in electronics design. *Electronics Cooling*, 13(2):22–25, 2007.

[22] C. Lasance, H Vinke, H Rosten, and K.-L. Weiner. A novel approach for the thermal characterization of electronic parts. In *Proceedings of the IEEE 11th Annual Semiconductor Thermal Measurement and Management Symposium*, pages 1–9, 1995.

[23] Filip Christiaens, Bart Vandevelde, Eric Beyne, Robert Mertens, and Jan Berghmans. A generic methodology for deriving compact dynamic thermal models, applied to the PSGA package. *IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part A*, 21(4):565–576, December 1998.

[24] A. Augustin, B. Maj, and A. Kostka. A structure oriented compact thermal model for multiple heat source ASICs. *Mircroelectronics Journal*, 36(8):700–704, August 2005.

[25] Y. C. Gerstenmaier and G. Wachutka. Rigorous model and network for transient thermal problems. *Mircroelectronics Journal*, 33:719–725, September 2002.

[26] Heinz Pape, Dirk Schweitzer, John H. J. Janssen, Arianna Morelli, and Claudio M. Villa. Thermal transient modeling and experimental validation in the european project PROFIT. *IEEE Tran. on Components and Pacakaging Technologies*, 27(3):530–538, September 2004.

[27] M. Rencz, G. Farkas, V. Székely, A. Poppe, and B. Courtois. Boundary condition independent dynamic compact models of packages and heat sinks from thermal transient measurements. In *Proceedings of the 5th Electronics Packaging Technology Conference*, pages 479–484, 2003.

[28] Francesco Beneventi, Andrea Bartolini, Andrea Tilli, and Luca Benini. An Effective Gray-Box Identification Procedure for Multicore Thermal Modelling. *IEEE Transactions on Computers*, pages 1–1, 2013.

[29] W. Huang, M. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy. Compact thermal modeling for temperature-aware design. In *Proc. Design Automation Conf. (DAC)*, pages 878–883, 2004.

[30] Ayse Kivilcim Coskun, Tajana Simunic Rosing, and Kenny C. Gross. Utilizing predictors for efficient thermal management in multiprocessor SoCs. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 28:1503–1516, October 2009.

[31] Ryan Cochran, Abdullah Nama Nowroz, and Sherief Reda. Post-silicon power characterization using thermal infrared emissions. In *Proc. Int. Symp. on Low Power Electronics and Design (ISLPED)*, pages 331–336, New York, NY, USA, 2010. ACM.

[32] Da-Cheng Juan, Huapeng Zhou, D. Marculescu, and Xin Li. A learning-based autoregressive model for fast transient thermal analysis of chip-multiprocessors. In *Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, pages 597–602, 2012.

[33] Andrea Bartolini, Matteo Cacciari, Andrea Tilli, and Lucas Benini. A distributed and self-calibrating model-predictive controller for energy and thermal management of high-performance multicores. In *Proc. European Design and Test Conf. (DATE)*, pages 1–6, 2011.

[34] Andrea Bartolini, Matteo Cacciari, Andrea Tilli, and Lucas Benini. Thermal and energy management of high-performance multicores: Distributed and self-calibrating model-predictive controller. *IEEE Trans. on Computers*, 24:170–183, 2013.

[35] Andrea Bartolini Andrea Tilli Francesco Beneventi Roberto Diversi and Luca Benini. SCC Thermal Model Identification via Advanced Bias-Compensated Least-Squares. In *Proc. European Design and Test Conf. (DATE)*, pages 1–6, 2013.

[36] Duo Li, Sheldon X.-D. Tan, Eduardo H. Pacheco, and Murli Tirumala. Parameterized architecture-level dynamic thermal models for multicore microprocessors. *ACM Trans. Des. Autom. Electron. Syst.*, 15(2):1–22, 2010.

[37] T. Eguia, S. X.-D. Tan, R. Shen, D. Li, E. H. Pacheco, M. Tirumala, and L. Wang. General parameterized thermal modeling for high-performance microprocessor design. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 2011.

[38] Z. Liu, S. X.-D. Tan, H. Wang, R. Quintanilla, and A. Gupta. Compact thermal modeling for package design with practical power maps. In *1st International IEEE Workshop on Thermal Modeling and Management: Chips to Data Centers, (TEMM)*, July 2011.

[39] Z. Liu, S. X.-D. Tan, H. Wang, A. Gupta, and S. Swarup. Compact nonlinear thermal modeling of packaged integrated systems. In *Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, pages 157–162, January 2013.

[40] P. V. Overschee and B. D. Moor. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.

[41] Tohru Katayama. *Subspace Methods for System Identification.* Springer, 2005.

[42] S Borkar. Thousand core chips: a technology perspective. In *Proc. Design Automation Conf. (DAC)*, pages 746–749, 2007.

[43] Tilera's 64-core Processor: TILEProf64. `http://www.tilera.com/products/processors/TILE-Gx_Family`.

[44] Intel's 48-core Single-Chip Cloud Computer. `http://www.intel.com/content/www/us/en/research/intel-labs-single-chip-cloud-computer.html`.

[45] The Intel Xeon Phi Coprocessor (60 cores and 240 threads and 1teraflops). `http://www.intel.com/content/www/us/en/high-performance-computing/high-performance-xeon-phi-coprocessor-brief.html`.

[46] Wei Huang, Shougata Ghosh, Siva Velusamy, Karthik Sankaranarayanan, Kevin Skadron, and Mircea R. Stan. HotSpot: A compact thermal modeling methodology for early-stage VLSI design. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 14(5):501–513, May 2006.

[47] Failure Mechanisms and Models for Semiconductor Devices. In JEDEC Publication JEP122-A, Jedec Solid State Technolgy Association, 2002.

[48] Critical Reliability Challenges for The International Technology Roadmap for Semiconductors (ITRS). In International Sematech Technology Transfer Document 03024377A-TR, 2003.

[49] S.H. Gunther, F. Binns, D.M. Carmean, and J.C. Hall. Managing the impact of increasing microprocessor power consumption. In *Intel Technology Journal*, First Quarter 2001.

[50] F. Zanini, D. Atienza, and G. De Micheli. A control theory approach for thermal balancing of mpsoc. In *Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, pages 37–42, 2009.

[51] James Donald and Margaret Martonosi. Techniques for multicore thermal management: Classification and new exploration. In *Proceedings of the 33rd annual international symposium on Computer Architecture*, ISCA '06, pages 78–88, Washington, DC, USA, 2006. IEEE Computer Society.

[52] Fabrizio Mulas, David Atienza, Andrea Acquaviva, and Salvatore Carta. Thermal Balancing Policy for Multiprocessor Stream Computing Platforms. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 28(12):1870–1882, 2009.

[53] Inchoon Yeo, Chih Chun Liu, and Eun Jung Kim. Predictive dynamic thermal management for multicore systems. In *Proc. Design Automation Conf. (DAC)*, DAC '08, pages 734–739, New York, NY, USA, 2008. ACM.

[54] F. Zanini, D. Atienza, L. Benini, and G. De Micheli. Multicore thermal management with model predictive control. In *Proc. 19th European Conference on Cicuit Theory and Design*, pages 90–95, Piscataway, NJ, USA, 2009. IEEE Press.

[55] Michael Powell, Mohamed Gomaa, and T N Vijaykumar. Heat-and-run: leveraging smt and cmp to manage power density through the operating systems. In *ACM SIGPLAN NOTICES*, volume 39, pages 260–270, 2004.

[56] Guanglei Liu, Ming Fan, and Gang Quan. Neigbor-Aware Dynamic Thermal Management for Multi-core Platform. In *Proc. European Design and Test Conf. (DATE)*, pages 187–192, 2012.

[57] Thomas Ebi, Mohammad Abdullah Al Faruque, and Jörg Henkel. TAPE: thermal-aware agent-based power economy for multi/many-core architectures. In *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, ICCAD '09, pages 302–309, New York, NY, USA, 2009. ACM.

[58] Yang Ge, P Malani, and Qinru Qiu. Distributed task migration for thermal management in many-core systems. In *Proc. Design Automation Conf. (DAC)*, pages 579–584, 2010.

[59] Y Ge, Q Qiu, and Q Wu. A Multi-Agent Framework for Thermal Aware Task Migration in Many-Core Systems. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 20(10):1758–1771, October 2012.

[60] T Chantem, X. S Hu, and R. P Dick. Temperature-Aware Scheduling and Assignment for Hard Real-Time Applications on MPSoCs. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 19(10):1884–1897, 2011.

[61] International technology roadmap for semiconductors (ITRS), 2012 update, 2012. `http://public.itrs.net`.

[62] J Srinivasan, SV Adve, P Bose, and J Rivers. Ramp: A model for reliability aware microprocessor design. *IBM Research Report*, 2003.

[63] J Srinivasan, S V Adve, P Bose, and J A Rivers. The case for lifetime reliability-aware microprocessors. In *Computer Architecture, 2004. Proceedings. 31st Annual International Symposium on*, pages 276–287, 2004.

[64] Tajana Simunic, Kresimir Mihic, and Giovanni Micheli. *Optimization of Reliability and Power Consumption in Systems on a Chip*, volume 3728 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[65] E Karl, D Blaauw, D Sylvester, and T Mudge. Reliability modeling and management in dynamic microprocessor-based systems. In *Design Automation Conference, 2006 43rd ACM/IEEE*, pages 1057–1060, 2006.

[66] Mehmet Basoglu, M Orshansky, and M Erez. NBTI-aware DVFS: A new approach to saving energy and increasing processor lifetime. *Low-Power Electronics and Design (ISLPED), 2010 ACM/IEEE International Symposium on*, pages 253–258, 2010.

[67] A Calimera, E Macii, and M Poncino. Energy-optimal SRAM supply voltage scheduling under lifetime and error constraints. In *Design Automation Conference (DAC), 2013 50th ACM / EDAC / IEEE*, pages 1–6, 2013.

[68] D Brooks, R. P Dick, R Joseph, and L Shang. Power, Thermal, and Reliability Modeling in Nanometer-Scale Microprocessors. *Micro*, 2007.

[69] M Ohring. *Reliability and Failure of Electronic Materials and Devices - Milton Ohring - Google Books*. Academic Press, San Diego, 1998.

[70] M Hauschildt, C Hennesthal, G Talut, O Aubel, M Gall, K B Yeap, and E Zschech. Electromigration early failure void nucleation and growth phenomena in Cu and Cu(Mn) interconnects. In *2013 IEEE International Reliability Physics Symposium (IRPS)*, pages 2C.1.1–2C.1.6. IEEE, 2013.

[71] Y.-K. Cheng, C.-H. Tsai, C.-C. Teng, and S.-M. Kang. *Electrothermal Analysis of VLSI Systems*. Kluwer Academic Publishers, 2000.

[72] Gene F. Franklin, Michael L. Workman, and Dave Powell. *Digital Control of Dynamic Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.

[73] Peter Van Overschee and Bart De Moor. *Subspace Identification for Linear Systems, Theory - Implementation - Applications*. Kluwer Academic Publishers, 2006.

[74] G. H. Golub and C. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, 3rd edition, 1996.

[75] Michael Hertl, Dianel Weidmann, and Alex Ngai. An advanced Reliability Improvement and Failure Analysis Approach to Thermal Stress Issues in IC Packages. In *35th International Symposium for Testing and Failure Analysis, ISTFA*, pages 28–32, 2009.

[76] A. Chakraborty, K. Duraisami, A. Sathanur, P. Sithambaram, A. Macii, E. Macii, and M. Poncino. Implementation of a thermal management unit for canceling temperature-dependent clock skew variations. *Integration, the VLSI Journal*, 41(1):2–8, 2008.

[77] Rajarshi Mukherjee and Ogrenci Seda Memik. Physical aware frequency selection for dynamic thermal management in multi-core systems. In *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, pages 547–552, 2006.

[78] S. Herbert and D. Marculescu. Analysis of dynaic voltage/frequency scaling in chip-multiprocessors. In *Proc. Int. Symp. on Low Power Electronics and Design (ISLPED)*, pages 38–43, 2007.

[79] H Hanson, S W Keckler, S Ghiasi, K Rajamani, F Rawson, and J Rubio. Thermal response to DVFS: analysis with an Intel Pentium M. *Low Power Electronics and Design (ISLPED), 2007 ACM/IEEE International Symposium on*, pages 219–224, 2007.

[80] M. Necati. Ozisik. *Finite Difference Methods in Heat Transfer.* Taylor & Francis, Inc., 1994.

[81] H Qian, H Liang, C H Chang, W Zhang, and H Yu. Thermal Simulator of 3D-IC with Modeling of Anisotropic TSV Conductance and Microchannel Entrance Effects . In *Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, 2013.

[82] Zhaohui Chen, XiaoBing Luo, and Sheng Liu. Thermal analysis of 3D packaging with a simplified thermal resistance network model and finite element simulation. In *Electronic Packaging Technology and High Density Packaging*, pages 737–741, 2010.

[83] Theodore Bergman, Adrienne Lavine, Frank. P. Incropera, and David P. DeWitt. *Fundamentals of Heat and Mass Transfer.* John Wiley & Sons, New York, 7th edition, 2011.

[84] Jianyong Xie, Biancun Xie, and Madhavan Swaminathan. Electrical-thermal modeling of through-silicon via (tsv) array in interposer. *International Journal of Numerical Modeling: Electronic Networks, Devices and Fields*, 2012.

[85] Yi-Kan Cheng, Ching-Chi Teng, Sung-Mo Kang, and Ching-Han Tsai. *Electrothermal analysis of VLSI systems.* Cambridge University Press, New York, NY, USA, 2000.

[86] Marta Rencz and Vladimir Szekely. Studies on the nonlinearity effects in dynamic compact model generation of packages. *IEEE Transactions on Components, and Packaging Technologies*, 27(1):124–130, March 2004.

[87] Vincent Verdult and Michel Verhaegen. Subspace identification of piecewise linear systems. In *Proc. 43rd IEEE Conference on Decision and Control (CDC)*, pages 3838–3843, 2004.

[88] www.comsol.com. COMSOL Mutiphysics: User Guide. *Version 4.1.*

[89] L. T. Pillage, R. A. Rohrer, and C. Visweswariah. *Electronic Circuit and System Simulation Methods.* McGraw-Hill, New York, 1994.

[90] Sheldon X.-D. Tan and L. He. *Advanced Model Order Reduction Techniques in VLSI Design.* Cambridge University Press, 2007.

[91] David Brooks, Vivek Tiwari, and Margaret Martonosi. Wattch: A framework for architectural-level power analysis and optimizations. In *Proc. Int. Symp. on Computer Architecture (ISCA)*, pages 83–94, 2000.

[92] *Assessing Product Reliability*, chapter Chapter 8. NIST/SEMATECH e-Handbook of Statistical Methods. `http://www.itl.nist.gov/div898/handbook/`.

[93] M. A. Korhonen, P. Borgesen, K. N. Tu, and C. Y. Li. Stress evolution due to electromigration in confined metal lines. *Journal of Applied Physics*, 73(8):3790–3799, 1993.

[94] J. J. Clement. Reliability analysis for encapsulated interconnect lines under dc and pulsed dc current using a continuum electromigration transport model. *Journal of Applied Physics*, 82(12):5991–6000, 1997.

[95] M. E. Sarychev, Yu. V. Zhitnikov, L. Borucki, C.-L. Liu, and T. M. Makhviladze. General model for mechanical stress evolution during electromigration. *Journal of Applied Physics*, 86(6):3068–3075, 1999.

[96] V Sukharev, A Kteyan, E Zschech, and W D Nix. Microstructure Effect on EM-Induced Degradations in Dual Inlaid Copper Interconnects. *IEEE Transactions on Device and Materials Reliability*, 9(1):87–97, 2009.

[97] X. Huang, T. Yu, V. Sukharev, and S. X.-D. Tan. Physics-based electromigration assessment for power grid networks. In *Proc. Design Automation Conf. (DAC)*, June 2014.

[98] Valeriy Sukharev. Beyond Black's Equation Full-Chip EM/SM Assessness In 3D IC Stack. pages 1–29, April 2013.

[99] Sandeep Chatterjee, Mohammad Fawaz, and N Farid Najm. Redundancy-Aware Electromigration Checking for Mesh Power Grids. In *IEEE International Conference On Computer-Aided Design (ICCAD)*, 2013.

[100] Zhijian Lu, Wei Huang, J. Lach, M. Stan, and K. Skadron. Interconnect lifetime prediction under dynamic stress for reliability-aware design. In *Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on*, pages 327–334, Nov 2004.