UNIVERSITY OF CALIFORNIA
RIVERSIDE

Towards Addressing Thermal and Reliability Challenges in Nanometer Integrated Circuits

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

Doctor of Philosophy

in

Electrical Engineering

by

Sheriff Imthias Sadiqbatcha

December 2021

Dissertation Committee:

Dr. Sheldon X.-D. Tan, Chairperson
Dr. Daniel Wong
Dr. Nael Abu-Ghazaleh

The Dissertation of Sheriff Imthias Sadiqbatcha is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

This dissertation represents the culmination of work that would not have been possible if not for the support, mentorship, and collaboration from numerous individuals. Firstly, I would like to convey my gratitude to my PhD advisor Dr. Sheldon Tan for giving me the opportunity to pursue this program and for all his guidance and support that were invaluable to my work. I would also like to thank my committee members, Dr. Daniel Wong and Dr. Nael Abu-Ghazaleh, for their gracious and invaluable advice. Additionally, I would like to thank my fellow researchers at the VLSI Systems and Computation Lab for their collaboration, feedback, and friendship. Namely, I would like to thank Chase Cook, Han Zhou, Hengyang Zhao, Jinwei Zhang, Liang Chen, Maliha Tasnim, Mohammadamir Kavousi, Sachin Sachdeva, Shaoyi Peng, Shuyuan Yu, Taeyoung Kim, Wentian Jin, Yibo Liu, and Zeyu Sun. In addition to my mentors and colleagues at UC Riverside, I would like to thank a few additional individuals who have had an enormous impact on my personal and professional life. Namely, I would like to convey my sincere gratitude to my long time mentor and friend Dr. Saeed Jafarzadeh, who helped greatly in shaping my career. Lastly and most importantly, I want to thank my family, especially my wife Daniya, my parents Zeenath and Sadiq, and my sister Fazalath for their love and encouragement.

To my mother Zeenath, my wife Daniya, and a bit of it to my sister Fazalath.

ABSTRACT OF THE DISSERTATION

Towards Addressing Thermal and Reliability Challenges in Nanometer Integrated Circuits

by

Sheriff Imthias Sadiqbatcha

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, December 2021
Dr. Sheldon X.-D. Tan, Chairperson

On-chip power densities continue to increase in modern integrated circuits (IC) due to rapid integration and feature scaling. As a consequence, today's high-performance processors have become more thermally constrained than ever before. Increase in temperature has been shown to exponentially degrade reliability of semiconductor chips and has consequently become one of the leading concerns in the industry today. In this thesis, we present our findings and share our contributions from our research efforts in the areas of pre-silicon IC reliability analysis, post-silicon thermal estimation, and advanced microprocessor cooling. Specifically, the first segment of this manuscript will focus on a novel structure-based approach to accelerating electromigration (EM) wear-out for the purposes of post-silicon qualification and burn-in testing. The proposed approach achieves time-to-failure acceleration comparable to the existing current and temperature based stressing techniques at close to nominal operating conditions. Temperature and reliability go hand-in-hand; hence monitoring and managing the processor's temperature while it is in use is equally important in order to maximize performance while minimizing reliability impacts. Therefore, the second

segment of this thesis will present our data-driven post-silicon approach to estimating the spatial temperature distribution across the surface of the die in real time. This approach leverages the latest advancements in recurrent-neural-networks for time-series estimation. The estimated temperatures from the proposed model can then be used to supplement the temperature information sensed from the embedded thermal sensors in order to make better informed thermal and reliability regulation decisions. Lastly, the third segment of this thesis will focus on leveraging the aforementioned real-time temperature estimation technique and the emerging thermo-electric based active cooling technologies to propose an on-demand targeted cooling system for modern high-performance processors. This approach yields the sub-ambient cooling benefits of thermo-electric cooling with lower power overheads.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Rapid integration and feature size scaling continues to increase power densities in modern integrated circuits (IC). As a consequence, today's high performance processors have become more thermally constrained than ever before. Increase in temperature has been shown to exponentially degrade reliability of semiconductor chips [6,7], and has consequently become one of the leading concerns in the industry today. Hence, there has never been a more critical time for research in the area of reliability, temperature estimation, and runtime thermal management of high performance ICs. In this thesis, we present our findings and share our contributions from our research efforts in the areas of pre-silicon IC reliability analysis, post-silicon thermal estimation, and advanced microprocessor cooling.

From the reliability preservative, several dominant reliability effects affect modern technology nodes, and are therefore carefully studied, simulated and tested during the pre-silicon design verification and post-silicon validation process. These effects include negative-bias temperature instability (NBTI), time-dependent dielectric breakdown (TDDB), hot

carrier injection (HCI), and electromigration (EM). Out of these, electromigration (EM) is the top reliability concern for copper-based back-end-of-the-line (BEOL) interconnects, both in present technologies and in the foreseeable future. 2015 ITRS-Interconnect predicted that EM lifetime of interconnects in VLSI chips will be reduced by half for each generation of nodes [8]. This is primarily due to increasing current densities and shrinking wire cross-sections, which determine the critical sizes for EM effects. On the other hand, many applications, ranging from automotive, to medical devices and aerospace equipment, require a long lifetime and have very demanding reliability requirements. As a result, testing and verification of reliability, especially EM-reliability, of VLSI chips used in those applications becomes even more critical. For many practical applications, reliability of 10 years or more is typically expected [9]. However, post-silicon testing of a chip for the duration of its projected lifetime is not practical. Hence, accelerated testing and stressing-conditions are needed to shorten the validation process. The first segment of this thesis (Ch. 2) will focus on our efforts and contributions in this area.

As previously stated, temperature and reliability go hand-in-hand, and as such, thermal validation and sign-off is another crucial step in today's physical design flow. Commercial tools exist to ensure sound thermal design starting from the device level [10] to the system-on-chip (SoC) level [11]. While design time thermal considerations play a crucial role in ensuring reliability and consistent performance, monitoring and managing the processor's temperature while it is in use is equally important. This is especially a challenge for system integrators that produce thin and light mobile devices, laptops, and embedded systems where the space restrictions limit the effectiveness of traditional coolers such as

heatsinks and heatpipes. In such cases, software based thermal monitors and controllers can be used along with the external coolers to ensure proper operation. To this end, runtime power and thermal control schemes are being implemented in most, if not all new generations of devices [12, 13]. These control schemes depend on accurate real-time temperature information of, at the very least, all of the dominant hotspots but ideally of the entire die area of the processor in order to be effective [14, 15].

On-chip temperature sensors alone cannot provide the full-chip temperature information since the number of sensors that can be placed in a chip is limited due to the high design overheads that they incur. As we have shown in [16], the number of hotspots on a typical commercial microprocessor far exceeds the number of embedded temperature sensors. Consequently, the thermal and power control algorithms that solely depend on the embedded sensors become oblivious to significant temperature peaks that occur spatially distant from the sensor locations. For example, Fig. 1.1 shows a significant temperature difference observed between a hotspot and the nearest embedded sensor in an Intel Core i7-8650U processor.

Adapting smart sensor placement algorithms that aid in spatial temperature interpolation of non-sensor locations can help mitigate this issue [4, 17, 18]. However, these methods require modifications to the chip's design (i.e. adding or relocating temperature sensors) which is not a post-silicon approach that can be applied to off-the-shelf processors. Sensing the need for a viable and urgent solution to this problem, we have developed two data-driven post-silicon approaches that can be used for the purposes of estimating the temperatures of all the dominant hostpots of the processor or to estimate the full spa-

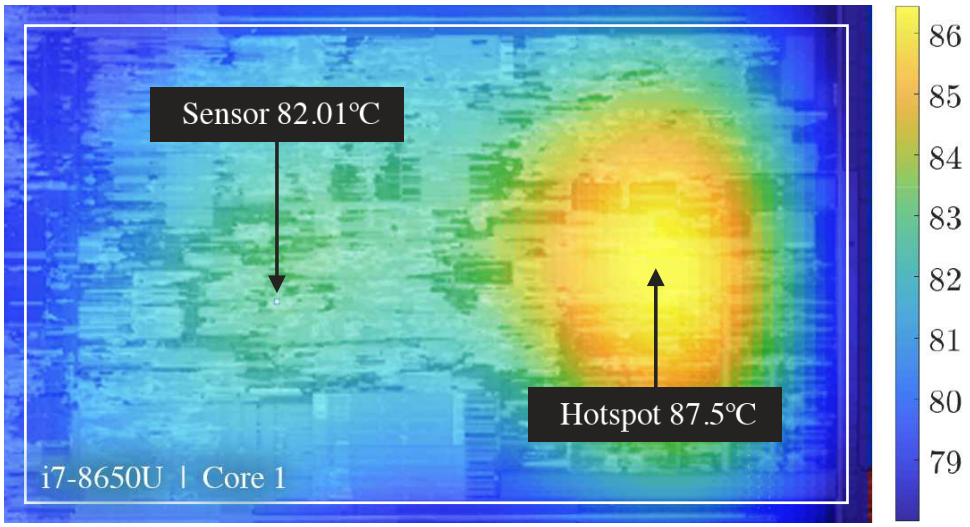


Figure 1.1: Hot-spot observed spatially distant from an embedded temperature sensor in an Intel i7-8650U

tial temperature distribution across the surface of the die in real time. This work will be presented in the second segment of this thesis (Ch. 3, 4).

Lastly, as previously mentioned, thermal and performance management algorithms, such as dynamic voltage and frequency control (DVFS) schemes, utilize the processor's run-time temperature information to dynamically control the chip in order to maximize performance while adhering to its temperature limits. While such control schemes play a major role in managing the processor's temperature while it is under load, equally as important are the cooling systems that aid in extracting heat away from the chip's surface. Such cooling systems, traditionally made up of a forced convection based fin-array heatsink coupled with a fan, are also modulated dynamically according to the chip's temperature. While our spatial temperature estimation schemes can aid these traditional cooling systems with more fine-grained real time temperature information, they can be even more advantageous when coupled with the emerging thermo-electric based cooling techniques. Hence, in the last

segment of this thesis (Ch. 5) we present our work in thermo-electric based targeted cooling that leverages the benefits of our real-time full-chip temperature estimation technique.

In the rest of this section, we further detail the motivation behind the work presented in this thesis as well as list our contributions to the respective areas of research.

1.1 Electromigration wear-out analysis

With temperature having a profound impact on all major IC reliability effects, creating burn-in conditions that accelerate EM in isolation becomes a difficult task. Hence, the primary motivation of the first segment of this thesis is to independently accelerate EM specific failures in VLSI chips under near-nominal working conditions so that EM failure effects can be fully verified and validated. If an acceleration from 10 years to a few hours is expected, one needs a time-to-failure (TTF) reduction in the order of $10^5\times$. However, this is quite challenging to achieve in reality, especially when it is required for the chip to fail exclusively under EM wear-out. Traditional acceleration methods, using high temperature and high voltage, also accelerate other reliability effects such as time dependent dielectric breakdown (TDDB) for dielectrics, biased temperature instability (BTI) and hot carrier injection (HCI) for CMOS devices. As a result, we aim to limit the testing temperature to be within the range of $-55°C$ to $125°C$ with $150°C$ being the maximum temperature limit [19, 20].

There are two main challenges. First, we need scalable EM models which are accurate both across the high stress conditions used during testing, as well as the normal use conditions. The model data from these accelerated tests can then be used to estimate the

true time-to-failure (TTF) when the chip is subjected to a normal use-case. Existing Black and Blech based empirical EM models [21, 22] are considered too conservative. Moreover, they do not fit well over a wide range of stressing conditions as the activation energy and current exponents are stress-condition dependent [23]. This will make it difficult to apply the results obtained from the high stressing conditions to the normal operating conditions. Second, once we have an accurate EM model, we need to find realistic EM stressing conditions and other acceleration techniques to achieve the desired reduction in TTF. This has to be done while ensuring that the chip fails exclusively under EM, not other reliability effects. Hence, the stressing condition should only accelerate EM, which is not the case with existing methods.

To address these challenges, we propose novel EM acceleration techniques based on stress engineering in the multi-segment interconnect wires by utilizing a recently proposed physics-based EM model [1, 24]. The significance of the proposed work is that it allows accelerated EM testing to be carried out at nominal current densities, and at a much lower temperature compared to traditional methods. The specific contributions of this segment of the thesis are as follows:

- First, we show how reservoirs can be exploited to significantly increase hydrostatic stress on a wire leading to an accelerated EM wear-out process. To leverage this effect for accelerated EM testing, we propose a novel interconnect structure with one and two reservoir segments. We show how the desired reduction in TTF can be achieved by simply configuring the geometry of the proposed structure.

- Second, we demonstrate how a similar EM acceleration effect can be achieved using

sink-based multi-segment structures. We then propose our second EM acceleration technique that leverages these atomic sinks. The sink-based structures are more challenging to design since they require several parameters to be tuned to achieve the desired reduction in TTF.

- We then propose a third EM acceleration technique based on combining both reservoir and sink segments in one hybrid structure.

- We show that, in general, 10% increase in temperature can achieve about 10× reduction in TTF. However, on-chip temperature has an upper limit for the working operations of CMOS circuits. Hence, in this work all the tests will be performed below 150°C.

- For practicality, all the structures proposed in the work will be designed to operate under two modes: normal-use and acceleration. Under normal-use, the structures will meet the lifetime requirement of at least 10 years. Under acceleration mode, the structures will fail quickly, within days or hours, while maintaining nominal current densities and stressing temperatures below 150°C.

Simulation results show that by combining temperature and the aforementioned structure based acceleration techniques, we can reduce the EM lifetime of a interconnect wire from 10 years down to a few hours (about $10^5\times$ acceleration in TTF) under the 150°C temperature limit. This, for the very first time, will afford us the ability to test EM under a controlled environment without the risk of invoking other reliability effects that are also accelerated by the traditional methods.

1.2 Real-time heat-source temperature estimation

In order to enhance reliability, many system level thermal/power regulation techniques such as clock gating, power grating, dynamic voltage and frequency scaling (DVFS) and task migration have been proposed in the past [25–28]. One critical aspect of the aforementioned algorithms is correctly estimating the full-chip temperature profile to properly guide the online thermal management schemes [14, 15]. However, accurate thermal estimation is a difficult task, especially for commercial off-the-shelf multi-core processors. Some of the existing methods depend on the on-chip temperature sensors. However, very few physical sensors are typically available, and they may not be located in close proximity to the true hot-spots on the chip, consequently misleading the temperature regulation decision [29].

Hence, the more popular solution is to supplement the on-chip sensor readings with estimated temperatures of all the prominent heat-sources or hot-spots on the chip via thermal models based on estimated power-traces. These methods offer higher spatial resolution as they allow for the temperature of all the heat-sources on the chip to be monitored in real-time [30–32].

In this area, existing approaches consist of several bottom-up numerical methods such as HotSpot [30] based simplified finite difference methods, finite element methods [33], equivalent thermal RC networks [34], and the recently proposed top-down behavioral thermal models based on matrix pencil method [35] and subspace identification method [36, 37]. However, the existing methods suffer from several drawbacks. First, most of the compact thermal models need accurate power-traces as inputs; but estimating the power of each

functional unit (FU) of a real microprocessor is not a trivial task, if not infeasible [38, 39]. On the other hand, from the system-level thermal or power management perspective, the parameters that can be easily accessed are the frequency, voltage, and many other performance metrics natively supported by most commercial processors. Thermal models which are functions of those parameters will be more desirable and practical. Second, calibration of the compact models against the actual chip temperature under different workloads and thermal boundary conditions is very difficult. The reason being, measuring the temperature profile of a working chip that is under load without the heat sink is a difficult task. Lastly, there is still a lack of an exclusively post-silicon approach to locating and estimating the temperatures of dominant heat-sources on the chip. Such a method would enable the development and deployment of more robust thermal control schemes for current and older generations of processors.

Hence, in this work, we aim to address the aforementioned issues with the novel contributions summarized below:

- First, we establish a lucid infrared (IR) thermal imaging setup with an advanced thermo-electric based rear-mounted cooling technique. This system allows us to obtain accurate online thermal maps of commercial multi-core processors while they are under load.

- Second, we propose a novel post-silicon approach to locating the prominent heat-sources on commercial microprocessors without any proprietary information about the chip's design. Our approach involves 2D discrete cosine transformation for noise reduction, and Laplacian transformation followed by K-mean clustering for heat-source identification.

• Third, we propose the use of high-level performance metrics provided by tools such as Intel's Performance Counter Monitor (IPCM), instead of low-level performance counters, as the inputs to our thermal models as they provide a comprehensive view of the processor's utilization in real-time. Moreover, they are easily accessible as most commercial processors are natively supported.

• We apply Long-Short-Term-Memory (LSTM) networks to build the system-level hybrid thermal model that is capable of highly accurate online temperature estimation. The proposed model is parameterized with IPCM metrics such as chip frequency, instruction counts, etc., and is trained and tested exclusively using thermal data measured directly from the processors under test.

• Since the model is meant to be deployed for real-time use, we explore methods of reducing the performance overhead and inference time of the model. This includes a novel power correlation based approach to identifying the thermally irrelevant IPCM metrics and eliminating them in order to reduce the input dimensionality of the model, and an analysis on network sizing to determine the ideal NN configuration that offers sufficient trade-off between accuracy and inference time.

• Lastly, we have structured the proposed framework such that it does not require any design changes and moreover does not need any information on the chip's architecture or floorplan. Hence it can just as easily be applied by the original manufacturer as well as a third-party for current and older generations of commercial processors.

Experimental results from two Intel multi-core processors (i5-3337U and i7-8650U)

show that the proposed thermal model achieves very high accuracy (root-mean-square-error: 0.55°C to 0.93°C) in estimating the temperatures of all the identified heat-sources on the chip. For i5-3337U the maximum root-mean-square-error (RMSE) is 0.76°C or 1.79%, and for i7-8650U the maximum RMSE is 0.93°C or 1.35%. Further details on this work and the results will be presented in Ch. 3.

1.3 Real-time full-chip heatmap estimation

Following the heat-source based temperature estimation work introduced in the previous section, we further present an entirely new data-driven approach, named *RealMaps*, to deriving a light-weight thermal model that is capable of real-time estimation of full-chip spatial heatmaps. The estimated heatmaps from the model can then be used to further supplement the temperature readings from the embedded temperature sensors for more effective thermal monitoring and control [40]. To our knowledge, the proposed approach is the first one that can be implemented on existing commercial multi-core processors for real-time full-chip heatmap estimation. The following is a summary of the contributions from this work.

- First, *RealMaps* can be implemented on most, if not all, existing commercial micro-processors and micro-controllers as it only uses the existing temperature sensors and workload independent utilization information. In other words, our strictly post-silicon approach does not require any modifications to the chip's design. Additionally, unlike many existing methods, it requires no proprietary design, floor-plan or process-specific information and therefore can be implemented by both the original chip manufacturer and

third-parties, such as system integrators and academic research labs, on future, current, and older generations of microprocessors alike.

• Second, our model is built based on high-level performance monitors, which are supported in most, if not all, commercial microprocessors. High-level performance monitors, unlike low-level performance counters, provide system-level utilization metrics such as the core frequency, instruction counts, cache hit/miss-rates etc rather than functional-unit-wise access rates.

• Third, *RealMaps* uses the previously mentioned advanced infrared (IR) thermography setup that enables lucid heatmaps to be recorded directly from commercial microprocessors while they are under load. This system allows us to build and validate the model using data measured directly from commercial off-the-shelf microprocessors as opposed to using simulation platforms.

• Fourth, to reduce the dimensionality of the model, 2D spatial discrete cosine transformation (DCT) is first performed on the heatmaps so that they can be expressed with just their dominant DCT frequencies. This allows for the model to be built to estimate just the dominant spatial frequencies of the 2D heatmaps, rather than the entire heatmap images, making it significantly more efficient.

• Last but not least, we once again propose the use of long-short-term-memory (LSTM) neural-networks (NN), which can discern temporal information, to build the model. This popular recurrent-neural-network (RNN) architecture is ideal for extracting features from sequential input data and therefore performs very well in the application at hand where

several time-steps of high-level performance metrics are used for each time-step of temperature inference.

Experimental results validated using measured thermal data from two commercial chips (Intel i5-3337U and i7-8650U) show that *RealMaps* can estimate the full-chip heatmaps with 0.9°C and 1.2°C root-mean-square (RMS) error with $0.4ms$ of inference time. This makes the proposed approach very desirable for online thermal estimation. Additionally, when compared to the state-of-the-art full-chip heatmap estimation method, *EigenMaps* [4], which requires *pre-silicon* design modifications, our post-silicon *RealMaps* shows similar accuracy, but with much less computational cost. Further information on this work and the detailed experimental results will be presented in Ch 4.

1.4 Thermo-electric based targeted cooling

Thermal management has become ever more challenging as newer technology nodes continue to increase power densities in modern high performance processors. As previously mentioned, managing the temperature of the processor not only affects the performance of the chip, but also its reliability. Effective and efficient cooling of high performance processors is therefore an important area of research.

Traditional cooling systems such as passive (natural convection) and active (forced convection) heatsinks have been used for many years as they offer a relatively simple, and cost-effective solution to the problem at hand. However, with the aforementioned increase in power-densities, it is becoming increasingly difficult to effectively manage the temperature of high performance processors using heatsink cooling; this is especially true for server

grade chips with high core counts. Increasing the size of the heatsink is one solution to this problem, however it is not a realistic solution due to space constraints. In addition, heatsink cooling can, at best, offer cooling up-to the ambient temperature. Hence, there is a need for more advanced, area and power efficient, cooling systems for modern high performance processors both at the consumer level and at the server/data-center level.

Thermoelectric coolers (TEC) are a promising technology that offer high cooling density with low area overheads. TECs are two-sided solid-state devices that use the Peltier effect to generate a heat flux with applied potential difference between the two terminals of the device. TECs effectively transfer heat from one side (cold-side) to the other (hot-side) in proportion to the applied voltage. Hence, the rate of heat-flow can be precisely controlled, allowing for the design of dynamically controlled cooling systems. Dynamically modulating the TEC voltage for on-demand cooling is crucial for making TECs a viable cooling method for microprocessors since TECs incur high power overheads.

Many studies have explored the use of TECs for the application of cooling high performance processors. Dousti et al. proposed a hybrid cooling system that use both TEC devices and the traditional fans with the use of an optimization algorithm to dynamically vary the TEC voltage and fan speed in accordance to the processor's operating temperature [41]. Jayakumar et al. proposed a dynamic thermal management scheme (DTM) that enables the joint control of the processor's frequency and voltage, TEC voltage and the fan speed with the aim of maximizing performance of the chip under temperature and power constraints [42]. While these methods were effective in cooling the processor, they are not energy-efficient implementations of TEC cooling as the entire die is cooled at the same rate.

Meaning, energy is wasted in cooling the areas of the chip that are consuming very little power (generating very little heat). In addition, these methods do not address the concern of high thermal gradients across the chip's surface, which over-time results in some areas of the chip aging at a faster rate than other areas.

Alternatively, Long, et al. proposed a method of embedding Thin-Film TEC devices within the chip's package where TECs are placed only over hotspots [43]. While this method offers spatially varying cooling, it does not offer temporally varying cooling, which is an important factor in reducing the power consumption of the TECs. The reason is that hotspots are not active (producing heat) at all times as they are workload dependent. Hence, Dousti et al. [44] proposed the use of TEC clusters located over the functional units that regularly produce hotspots on the chip. The TEC clusters are established using a clustering algorithm that aims to minimize power waste from unnecessary use of TECs. A current bypass switch technique is used to turn the clusters on and off as needed while the processor is under load. This method allows both spatially and temporally varying TEC cooling. However, the drawback of this method is that this system needs to be specially designed for each processor model and therefore is not a generic cooling system that can be widely adopted. Moreover, this method requires integrating the TEC clusters within the processor's package, hence requiring additional efforts in package design. There is still a lack of an externally mounted TEC-based on-demand cooling system that is generic enough to be used for a variety of microprocessors. Such a system can be quickly and easily adopted by industry with minimal overheads.

In this work, we aim to mitigate the aforementioned problems by proposing a on-

demand TEC-based active cooling technique, called *TEC-Array*, which can perform targeted cooling of active hotspots that vary spatially and temporally in multi-core processors. The proposed system consists of an array of TEC modules modulated by a power-density based control scheme. Our specific contributions are summarized below:

- First, we propose to use the current state-of-the-art power modeling techniques to estimate the full-chip power-density distribution across the processor die. The estimated power-density maps are then converted to discrete voltage maps; which are in turn used to dynamically configure the voltage settings of the TEC-array. The proposed approach allows dynamically varying targeted cooling where hotspots are cooled more intensely than non-hotspots. The proposed TEC-Array is especially beneficial for the running conditions in multi-core processors when tasks are mapped to a few cores while the others remain idle. In this scenario, only the cores that are under load can be cooled with minimal cooling applied to the idle cores which in turn leads to significant energy savings over time and a reduction of the aforementioned thermal gradients.

- Second, to validate the proposed TEC-array based active cooling and power management method, we further propose a numerical simulation framework, which employs a more accurate 3D coupled multiphysics model for TEC devices to consider Peltier, heat transfer, Joule heating and complex electro-thermal coupling effects by solving the coupled heat conduction and current continuity equations.

- Lastly, with numerical results on an Intel quad-core chip, we show that the proposed TEC-array can substantially reduce the peak temperatures compared to the traditional passive heat sink cooling method. Furthermore, compared to existing single TEC module

based cooling methods, the proposed approach can reduce both TEC power and temperature gradients across the chip under the same maximum temperature constraints, which can further reduce spatial temperature induced stress such as thermal cycling, thermo-migration, unbalanced aging, etc. As a result, the new TEC-array based cooling technique can enable more aggressive chip performance with increased thermal design power (TDP) while maintaining the chip's design lifetime.

We remark that the primary goal of this work is to allow the chip to run at higher performance modes with increased TDP without suffering the thermal gradient induced aging impacts compared to the existing heat-sink and single TEC device-based coolers. It should be acknowledged that the TEC devices do indeed incur higher power overheads compared to traditional heat-sink based coolers. On the other hand, TEC devices provide significantly higher cooling potential compared to traditional heat-sinks, hence optimizations can be carried out with the objective of maximizing the chip's performance with respect to the power consumption of the cooling system. If the total power consumption (chip power plus cooling power) is a concern, we note that the proposed technique is orthogonal to the other online chip power reduction/management schemes such as dynamic voltage and frequency scaling (DVFS), and task scheduling and migration which can be leveraged to optimize the overall power consumption of the system (chip plus cooling power) as demonstrated in many existing works. For example, in [45] Amrouch et. al. propose a thermal management scheme for Neural processing units that involves combined control of a TEC-based cooler as well as modulating the chip's frequency, and inference precision in order to find the optimal setting to maximize the chip's throughput while minimizing the overall power consumption

of the system. Similarly in [46] Lundquist et al. proposed using a TEC cooler and heatsink with a fan to dynamically control the processor cooling. Here, the TEC voltage and fan speeds are controlled adaptively to sustain the chip temperature under a predetermined threshold in order to avoid performance degradation while minimizing the power consumption of the system. Such combined control can be used in conjunction with the proposed TEC-array in order to realize further optimization to reduce the power consumption of the entire system while maximizing performance relative to the consumed power. In Ch 5 of this thesis, we will further detail the proposed framework and the associated experimental results.

The thesis is organized as follows. Ch .2 presents our novel structure based EM acceleration technique that allows accelerated post-silicon qualification tests to be conducted exclusively for EM without the risk of invoking other reliability effects in the process. Ch .3 proposes our data-driven approach to post-silicon heat source identification and thermal estimation of the identified heat-sources. Ch. 4 presents an extension of our work from Ch .3, where we propose a framework to estimate the full-chip heatmaps in real time. Ch. 5 leverages the work in Ch. 4 and the emerging thermo-electric based cooling techniques to propose an on-demand spatially-varying cooling method for modern high-performance processors. Ch. 6 concludes this article.

Chapter 2

Accelerated Electromigration Wear-out Analysis for Nanometer Integrated Circuits

2.1 Related Work and Motivation

A number of physics-based EM models and assessment techniques have been proposed in the past [1,24,47–57]. Huang *et al.* proposed a compact EM time-to-failure (TTF) model based on the approximate closed form solution of Korhonen's equation for a single wire and studied the impact that wire redundancy has on EM failure in the power grid networks [47, 50, 58]. This work has been extended to multi-segment wires [49] and time-varying current cases [53, 54]. Additionally, a numerical solution based finite difference method [55] and Krylov subspace method [55] have been explored. These EM models are

primarily based on the hydrostatic stress diffusion kinetics in confined metal wires and hence can consider stress evolution and distribution in entire interconnect structures consisting of many wire segments. This is in stark contrast with the traditional Black-Blech EM models, which can only consider isolated single segment wires. As a result, these open new ways to manipulate multi-segment interconnect structures in order to achieve the desired stress and aging effects. Recently a very accurate 3-phase EM model was proposed [1, 24], which better described the post-voiding wire resistance change behavior of copper dual damascene interconnects. We will utilize this EM model for the analysis performed in this work.

Traditionally, for accelerating the EM failure process, raising temperature has been the most effective method as the EM effects are exponentially dependent on temperature. However, elevated temperature will lead to other failures, such as time-depended-dielectric-breakdown (TDDB), very quickly. Moreover breaching the well-known thermal limits of semiconductor devices is not a reliable approach [19]. Increasing the current density is another way to accelerate EM wear-out. However, it is well-known that the impacts of current density is reversely proportional to the time-to-failure (TTF) under EM with current exponent between 1 and 2 [59]. This limits the impact of current density on acceleration. Furthermore, high current densities can lead to thermal runaway effects due to excessive Joule heating [60]. In order to ensure failure exclusively under EM, a new acceleration method is urgently needed, which can achieve the same reduction in TTF but at a much lower current density and temperature; hence the motivation behind our work presented in this chapter.

2.2 Review of EM physics and three-phase EM model

2.2.1 EM in a nutshell

EM is a physical phenomenon of the migration of metal atoms along the direction of the applied electrical field. Atoms (either lattice atoms or defects / impurities) migrate along the trajectory of conducting electrons. Under EM, hydrostatic stress is generated inside the embedded metal wires due to the momentum exchange between the electrons and lattic atoms. When the stress exceeds the so-called critical stress value, atom migration is initiated resulting in the formation of voids and hillocks at the cathode and anode terminals of the wire respectively.

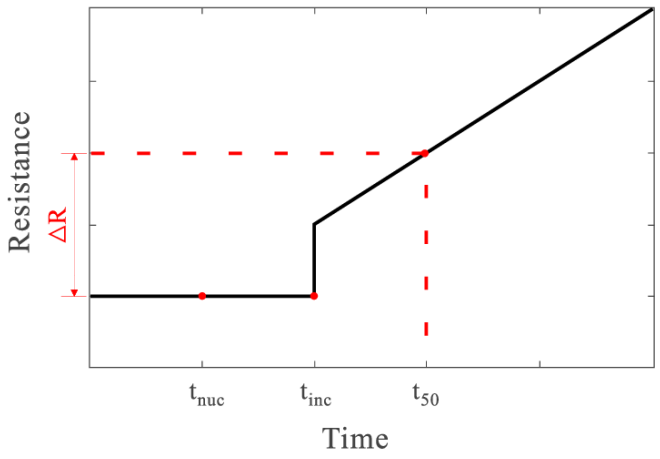
The traditional method of predicting time-to-failure (TTF) is based on approximations and statistical methods such as Black's equation [21] and Blech's limit [22]. However, they are subject to growing criticism due to their over conservativeness and lack of consideration of multi-segment interconnect wires [50]. To mitigate this problem, a number of new physics-based EM modeling techniques have been proposed [49–51, 56, 58, 61–64] based on solving the Korhonen's hydrostatic stress diffusion equation [65]. Recently a three-phase EM model [1, 66], which better represents the wire resistance change over time, was proposed. This model will be discussed in the following section.

2.2.2 The three-phase physics-based compact EM model for multi-segment wires

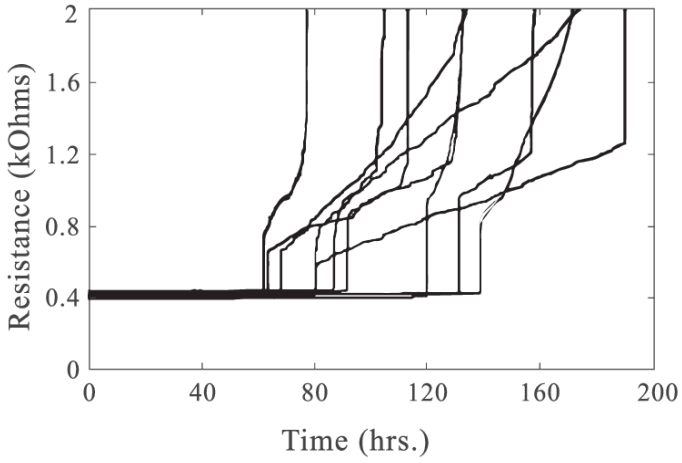
In the existing physics-based EM models, the EM failure process in general can be viewed as two phases: the nucleation phase, in which the void is generated after the critical

stress is reached, and the growth phase, in which the void starts to grow. Existing compact EM models are also versed in terms of the two phases, where each phase is described by time-to-failure (TTF) as a function of current density and other parameters [50, 67]. However, such a simple EM model ignores the fact that when the void is nucleated or formed, it will not change the wire's resistance immediately. It has been experimentally observed that there exists a so-called *critical void size* [2, 68], which is typically the via-diameter or cross section area of the interconnect wire. Since the conductivity of Cu is much higher than the barrier layers, resistance of the wire does not change until the void grows to a point where its volume equals or becomes larger than the cross-section of the via or wire. Only then will all the current start to flow over the thin barrier layer, which will lead to a very high current density and consequent joule heating. The joule heating in turn will lead to a small resistance jump, indicating the end of this phase (see Fig. 2.1(a)). Fig. 2.1(b) shows the experimentally measured resistance change over time where the small resistance jumps are clearly visible. Also, sometimes the barrier layers are not very stable, due to manufacturing process variations, causing the barrier layer to quickly burn out resulting in an open circuit as is shown in Fig. 2.1(b) [2].

Based on these observations, a three phase EM model has been proposed for a single segment wire [1, 66]. The new model has three phases as shown in Fig. 2.1(a): (1) the *nucleation phase* from $t = 0$ to t_{nuc}; (2) the *incubation phase* from t_{nuc} to t_{inc}; and (3) the *growth phase* from t_{inc} to t_{50} (or t_{gro}), together indicate the time-to-failure in statistical terms (50% of the samples fail). This model was later extended to consider more general multi-segment interconnect structures [69] (i.e. Fig. 2.2). We remark that the three-phase



(a)



(b)

Figure 2.1: (a) Illustration of resistance change over time, courtesy of [1]. (b) Experimentally measured resistance change over time, courtesy of [2]

EM model is more consistent with the measured wire resistance change over time than existing physics-based EM models such as [50,61,64], which do not consider the incubation time. The incubation time, which is the time between nucleation time and the time when the wire resistance changes, can be significant for the overall time-to-failure (TTF) analysis and is dependent on the wire structure as well. As a result, the three-phase EM model

is more accurate than other physics-based models and will be used for the proposed EM acceleration work in this article.

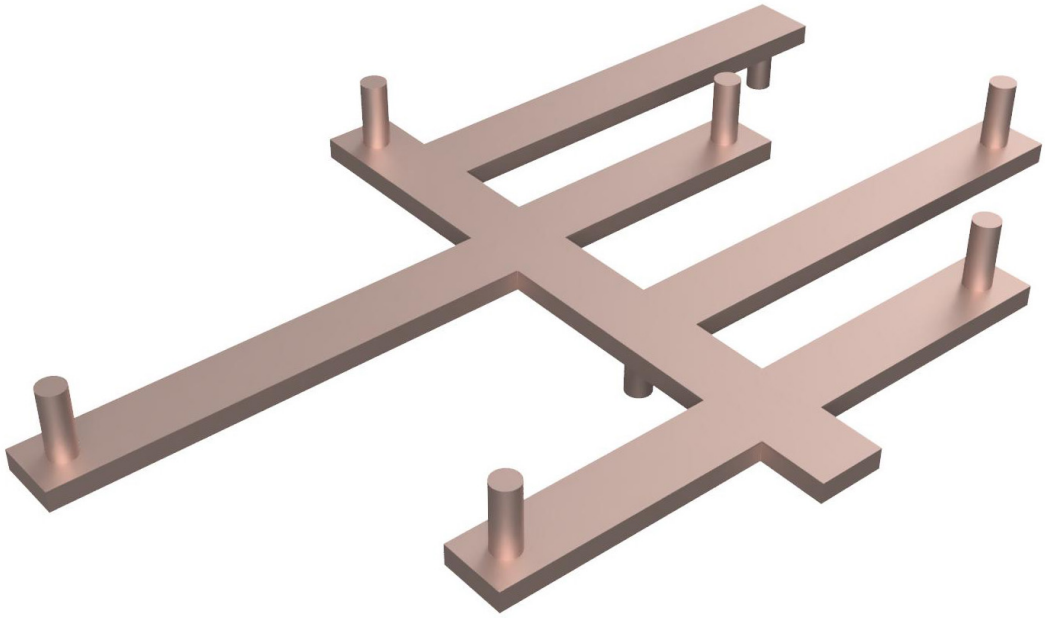


Figure 2.2: Illustration of a multi-segment Interconnect wire structure

2.2.3 Multi-mode failure scheme

The 3-phase EM model, discussed in the previous subsection, allows us to consider a robust multi-mode failure scheme. Typically, only parametric failures, or late-failures (LF), are considered for EM wear-out where, as the void grows, resistance of the wire slowly increases and reaches a point where the circuit can no longer function as intended. This type of failure occurs in the so called via-below structures as shown in Fig. 2.3(a), where the flow of electrons is from a lower metal layer to a higher one (hence this structure is also called an upstream structure). In this case, even when the void grows to its critical

size (cross sectional area or via diameter), current can still flow through the barrier layer, but, at a much higher resistance. Here the wire can be considered as failed at the end of the growth phase, which, typically, is the point where the wire's resistance increases by 10% (or other user defined criteria). However, there exists another type of failure, called early-failure (EF), which is observed in the so called via-above structures like the one shown in Fig 2.3(b). Here the electron flow is from a higher metal layer to a lower one (hence it is also called a downstream structure). Since a non-conductive capping layer is applied between the layers of metallization in the dual damascene process, once the void reaches the critical size, we instantly see an open circuit because current cannot flow though the dielectric capping layer. Therefore, in such structures the wire fails at the end of the incubation phase. This critical distinction can only be accounted for accurately using the new 3-phase EM model. More detailed study on these failure schemes can be found in [2, 68]. In this article, we present TTF results for both the via-above (TTF_{EF}) and via-below (TTF_{LF}) topologies.

2.3 Atomic reservoir and sink enhanced EM acceleration

The new EM acceleration techniques are inspired by the observation that atomic reservoir and sink segments in a multi-segment interconnect wire can have a significant impact on the hydrostatic stress evolution in the wire [3, 70]. These structures are good candidates for accelerating EM failure effects as they offer a great deal of flexibility and configurability on the EM lifetime of active interconnects without affecting their normal functionality. EM induced hydrostatic stress on the main conductive wire segments that share a terminal with the reservoir and sink structures can be drastically altered using

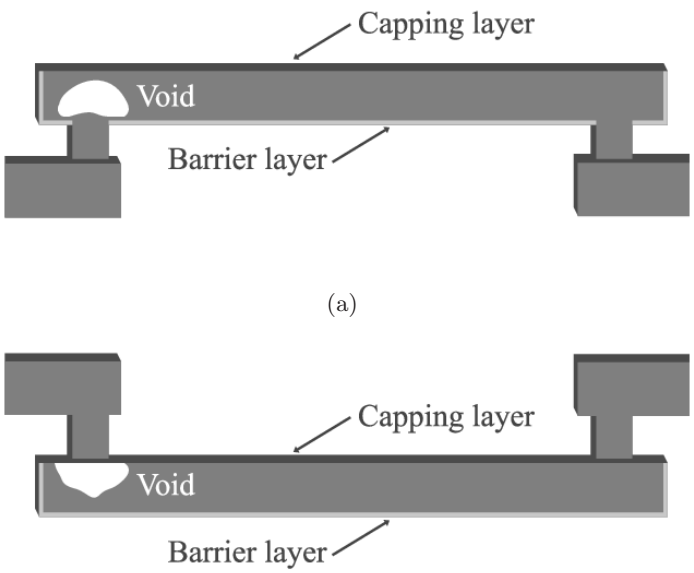


Figure 2.3: (a) Via-below or upstream and (b) Via-above or downstream wire structures showing void formations

several design parameters.

The impact of these structures is analyzed in the following subsections. The unique properties of reservoirs and sinks are then exploited in designing structures for accelerated EM testing. The proposed structures are designed to operate under two modes: normal use and acceleration. The configurable nature of these structures allow them to be designed for a lifetime of 10+years (typical lifetime requirement for ICs) under normal use, and just a few days or hours under acceleration mode (or as desired for the application at hand). Note, temperature of 353K ($\sim$80°C) is assumed for all the analysis in this section. Temperature configuration for actual EM acceleration tests will be discussed in Sec. 2.4.

2.3.1 Configurable reservoir based EM failure acceleration

Reservoir structures (passive interconnect segments) are typically added to the cathode terminal of active interconnect wires that are vulnerable to EM wear-out. These structures decrease the rate of hydrostatic stress evolution on the active wires, consequently prolonging nucleation time. Fig. 2.4 shows the measured failure distribution of a two segment wire with reservoir segments of different lengths [3]. As the results show, the lifetime of the wire increases when the reservoir is present ($L_r > 0$). The more reservoir area (longer length in this case), more the lifetime is prolonged.

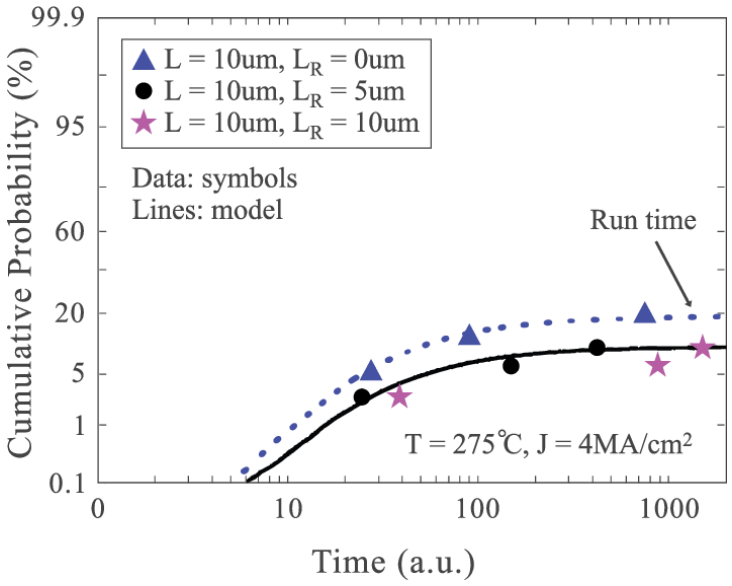


Figure 2.4: Experimentally obtained failure distribution for a two segment wire with a reservoir. L_r is the length of the reservoir segment. Courtesy of [3]

To further demonstrate the impact of reservoir segments, let us consider an active interconnect segment (main-branch), with no reservoir structure, shown in Fig. 2.5. With the previously discussed 3-phase EM model, transient stress across this wire segment can

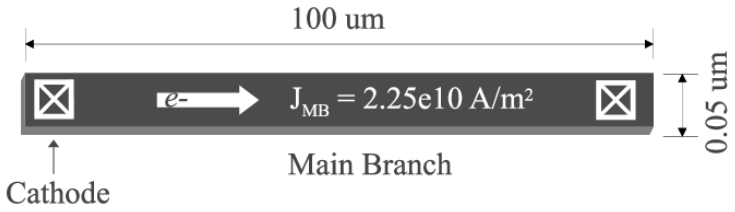


Figure 2.5: Active interconnect wire segment

be computed using a numerical approach such as finite element or finite difference. Fig. 2.8 (a) shows the hydrostatic stress evolution over time at the cathode terminal computed using finite element analysis. Only the cathode node is shown since, in most cases, void is nucleated here as the cathode end of the wire experiences the maximum tensile stress. The results for this structure show void nucleation, t_{nuc}, at $1.68 \times 10^3 hrs$. Post nucleation, the incubation, t_{inc}, and growth, t_{gro}, times can be calculated using the closed form equations discussed previously. For this structure the results are: $t_{inc} = 1.68 \times 10^4 hrs$, and $t_{gro} = 1.56 \times 10^4 hrs$. Therefore the effective TTF for early and late failure cases are: $TTF_{EF} = 1.85 \times 10^4 hrs$ and $TTF_{LF} = 3.41 \times 10^4 hrs$ for the interconnect wire shown in Fig. 2.5.

Let us now consider the effect of adding a passive reservoir segment to the cathode terminal of this wire. For now, let us arbitrarily set the reservoir to be half the length and twice the width of the active wire ($W_R = 0.1um$ and $L_R = 50um$) as shown in Fig. 2.6. Transient stress analysis at the cathode of this new structure shows nucleation delayed to $t_{nuc} = 1.29 \times 10^4 hrs$ as shown in Fig. 2.8 (b). Incubation and growth times stay the same since these depend on total atom flux at the cathode and at this point, only the main-branch is carrying current and therefore contributing to the effective flux. Nonetheless, delaying nucleation time prolongs the wire's lifetime: $TTF_{EF} = 1.72 \times 10^4 hrs$ and $TTF_{LF} =$

$3.28 \times 10^4 hrs$. This delay in nucleation time makes reservoir insertions a effective and efficient technique that can be used to mitigate EM violations in interconnect structures without the need for costly redesigns.

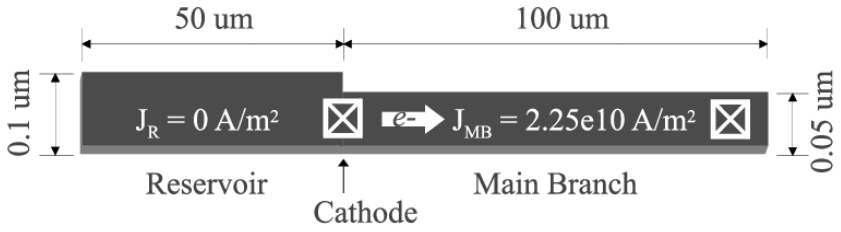


Figure 2.6: Reservoir at the cathode of an active interconnect wire

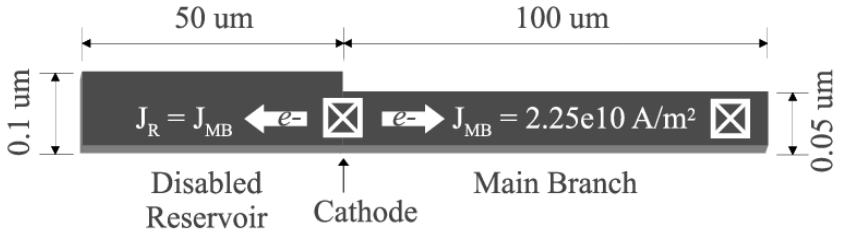


Figure 2.7: Disabled reservoir at the cathode of an active interconnect wire

Interestingly, if we design the structure shown in Fig. 2.6 such that current in the reservoir segment can be activated during runtime, we can exploit a very unique property. Let us consider the structure shown in Fig. 2.7, which is identical to the structure in Fig. 2.6 but with current flow enabled in the reservoir segment. Let us arbitrarily set the current density in the reservoir segment (Disabled Reservoir) to be the same as the main-branch, but in the opposite direction. We call this a disabled reservoir since this configuration effectively disables the benefits of the reservoir segment, shifting nucleation

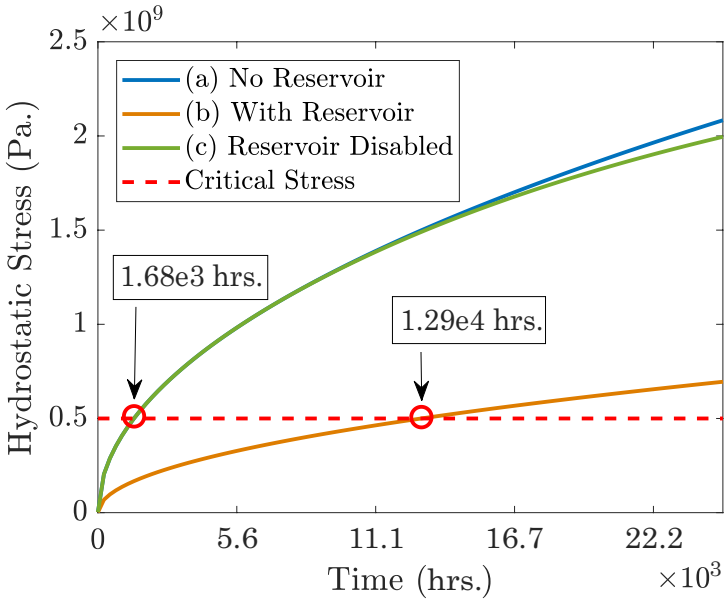


Figure 2.8: Impact of reservoir on nucleation time

time back to what was originally observed from the structure in Fig. 2.5. This acceleration in nucleation time is shown in Fig. 2.8 (c). Moreover, the additional atom flux generated by the electron flow in the reservoir segment also accelerates the incubation and growth times: $t_{inc} = 1.41 \times 10^3 hrs$ and $t_{gro} = 5.19 \times 10^3 hrs$. The effective TTF now becomes: $TTF_{EF} = 3.08 \times 10^3 hrs$ and $TTF_{LF} = 8.28 \times 10^3 hrs$. This is a significant reduction in lifetime achieved, at nominal current density and temperature, by merely switching on current flow in the reservoir segment. This critical observation is the basis for our reservoir-enhanced EM acceleration technique.

Based on this analysis, we propose a configurable two-segment interconnect structure shown in Fig. 2.9 [69]. The structure consists of a two segment wire (one reservoir and one main-branch), one MOSFET device (switch to disable the reservoir) and two resistors

R_1 and R_2 to configure the currents in the two wire segments. The bottom half of Fig. 2.9 shows the 3D view of this design. During normal use, the reservoir will remain passive (zero current density). Once acceleration ($Acc.Signal$) is activated, the current density in the reservoir will become non-zero, thus disabling the reservoir and accelerating EM wear-out.

In addition to the simple two-segment interconnect structure, we further propose the three-segment configurable interconnect structure shown in Fig. 2.10, with two reservoir segments. Here, the configurable circuity is omitted for the sake of better presentation. This multi-segment design is meant to show both the robustness of the EM model used in this study, as well as the flexibility the circuit designers have to design such reservoir structures for EM acceleration. Typically, circuit designers work under several constraints where optimization between critical parameters is crucial. The configurable nature of the proposed structure naturally allows for optimization between geometry, current density, and desired lifetime under normal use and acceleration modes.

As mentioned previously, the proposed structure will be designed to operate under two modes: normal use and acceleration. Under normal use (no current in reservoir), it is critical to ensure that the structure will have a lifetime of at least 10 years (or as needed for the given application). Under acceleration (current enabled in reservoir), we want the structure to fail quickly (typically within days or hours). Hence, the goal is to find a configuration (W_R, L_R) that will meet these requirements.

Traditionally, current density of the main branch, J_{MB}, is significantly increased to achieve EM acceleration. However, this method also accelerates other reliability effects and, above a certain threshold, leads to joule-heating causing additional problems. Hence, we

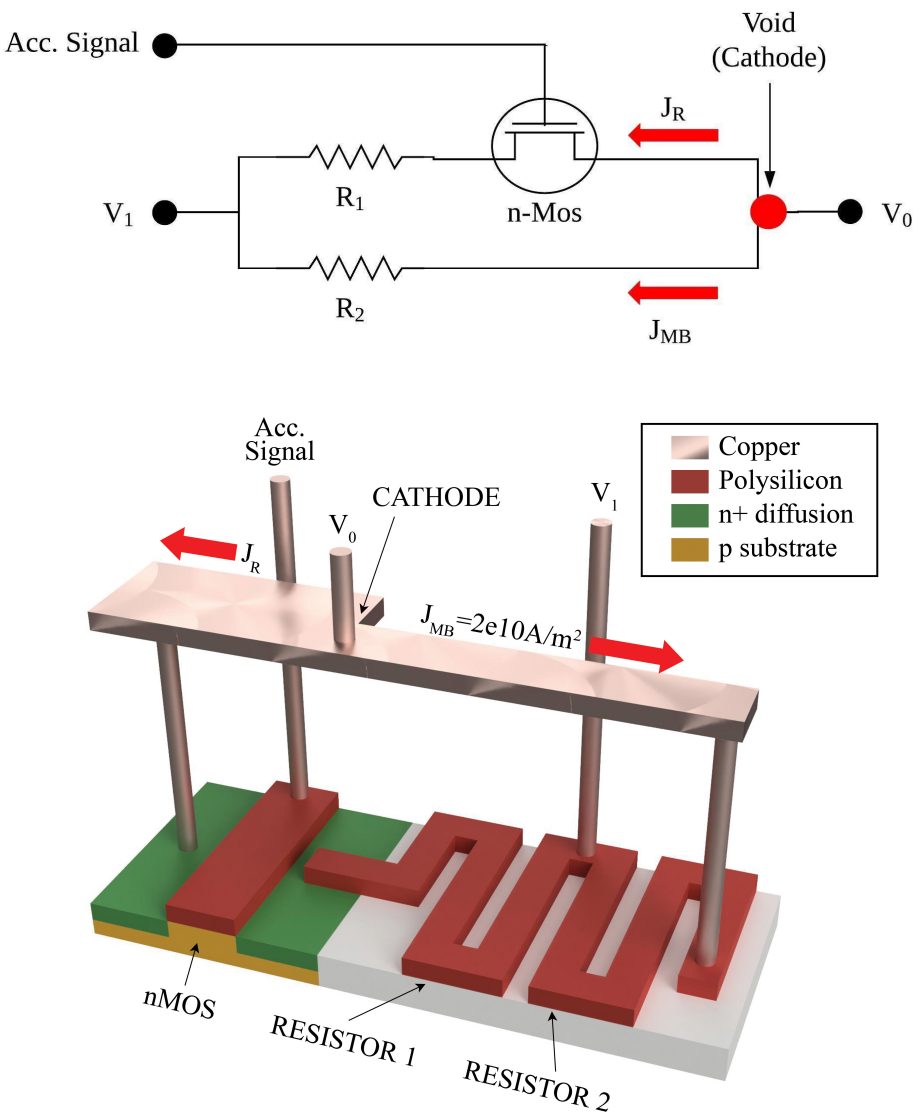


Figure 2.9: The proposed configurable reservoir-based EM wear-out acceleration circuit (For illustration only; components not drawn to scale with respect to each-other)

will fix J_{MB} to be the same under both normal use and acceleration modes. For acceleration mode we will simply activate current in the reservoir such that $J_R = 0$ becomes $J_R = J_{MB}$. Indeed, J_R can be set higher than J_{MB} as long as it abides by the design rules (i.e. Synopsys

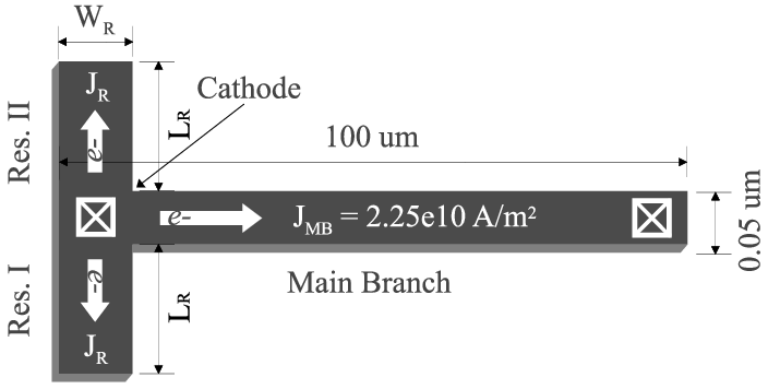
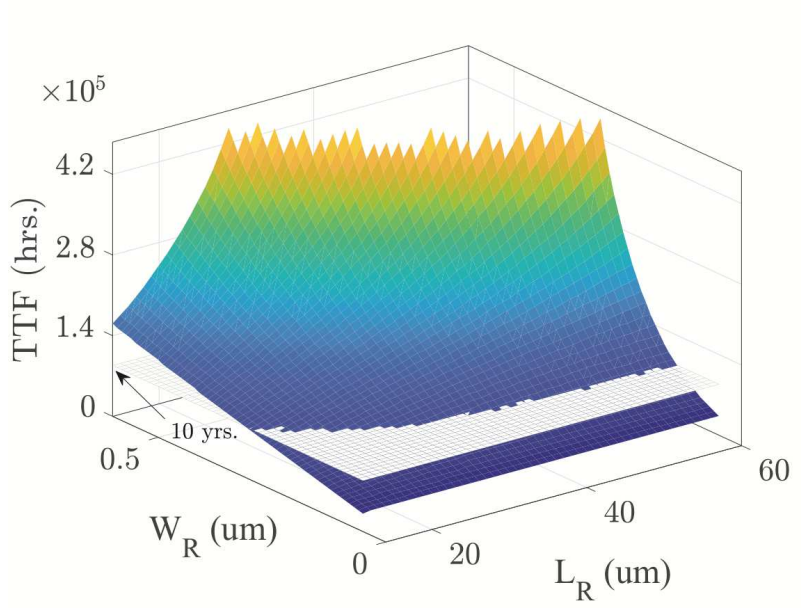


Figure 2.10: Proposed EM acceleration structure with two reservoir segments

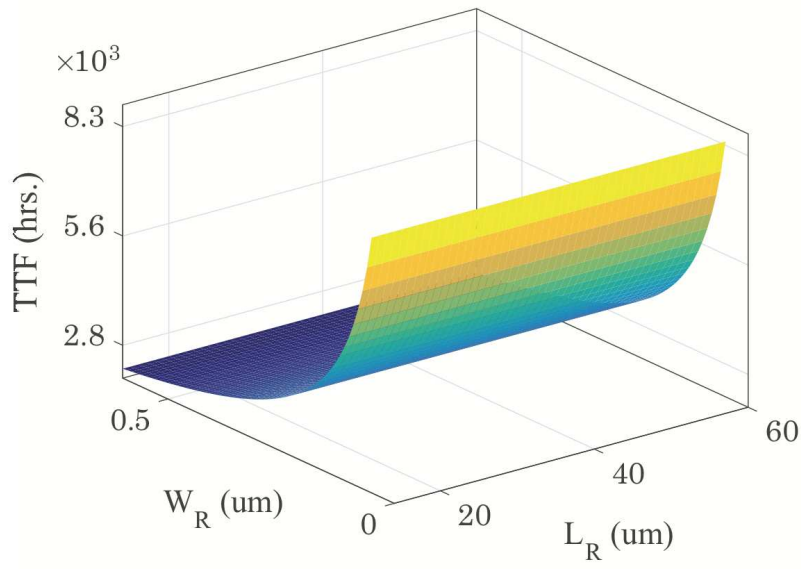
32nm PDK [71]).

As shown in Fig. 2.11, the proposed structure gives the circuit designer a great deal of flexibility in achieving the desired TTF for the application at hand. For instance the configuration $W_R = 0.3um$ and $L_R = 18um$ results in $T_{nuc} = 8.31 \times 10^4 hrs$, $T_{inc} = 4.17 \times 10^3 hrs$, and $T_{gro} = 1.56 \times 10^4 hrs$ ($TTF_{EF} \approx 10 years$, $TTF_{LF} \approx 11.7 years$) under normal use, and $T_{nuc} = 1440hrs$, $T_{inc} = 325hrs$, $T_{gro} = 1197hrs$ ($TTF_{EF} \approx 73.6days$, $TTF_{LF} \approx 123.5days$) under acceleration mode. TTF can be reduced a little further with larger reservoirs, for instance the configuration $W_R = 1um$ and $L_R = 18um$ results in $T_{nuc} = 1427hrs$, $T_{inc} = 103hrs$, $T_{gro} = 381hrs$ ($TTF_{EF} \approx 63.8days$, $TTF_{LF} \approx 79.6days$). However, bear in mind, this was achieved at a working temperature of 353K (~80°C), this structure under burn-in conditions will yield a failure time that is much lower. These testing conditions will be discussed in detail in the next section.

Note, we can make a critical observation from the results shown in Fig. 2.11. For the proposed structure operating under normal mode (Fig. 2.11(a)), the results show that



(a)



(b)

Figure 2.11: TTF_{LF} (a) Normal use (b) Acceleration mode

the TTF is a function of reservoir area. However, under acceleration mode (Fig. 2.11(b)), it is clear that TTF is a function of just the reservoir width, not its length. This is because the acceleration in nucleation time is caused by the additional atom flux through the reservoir's cross section at the cathode boundary of the active wire. The additional length of the reservoir is merely there to ensure that the structure meets the lifetime requirement for normal use (10+ years or immortality); it has no impact on acceleration. This observation will be exploited later with the hybrid structure that combines both reservoir and sink segments.

2.3.2 Configurable sink based EM failure acceleration

Atomic sinks can be passive or active interconnect structures that, when added to the anode terminal of an active interconnect wire, can significantly increase the steady state tensile stress at the cathode. Additionally, adding a sink segment can reduce the compressive stress at the anode node, hence reducing the chance of hillock formations or extrusions.

Let us consider the structure shown in Fig. 2.12(a) where $L_{MB} = 5um$, $W_{MB} = W_S = 0.05um$, $L_S = 95um$ and $J_{MB} = J_S = 2.25 \times 10^{10} A/m^2$. This structure is indeed identical to the single main-branch that was shown in Fig. 2.5, but this time split into two segments. We will now refer to the first segment as the main-branch and the second segment as the active sink. As expected, hydrostatic stress evolution over time at the cathode terminal (Fig. 2.13(a)) of the structure in Fig. 2.12(a) is identical to what was previously observed for the structure in Fig. 2.5. However, if we can design this structure

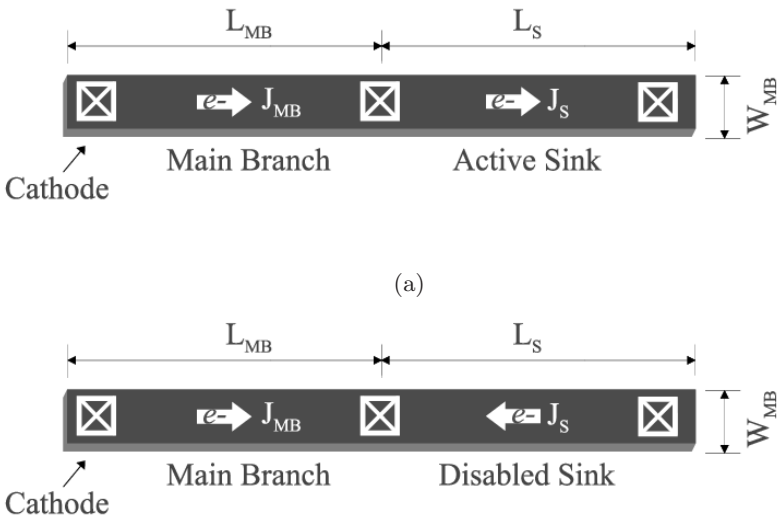


Figure 2.12: Active interconnect wire with: (a) an active sink at the anode (b) disabled sink at the anode

such that the direction of current in the sink segment can be reversed during runtime, then we can effectively disable the impact of the sink segment, hence significantly reducing the tensile stress at the cathode (Fig. 2.13(b)). Note, sink structures behave very differently than reservoir structures. While reservoirs affect both steady-state and transient stress, sink structures only affect the steady-state. This is a critical distinction that should be noted.

Previously in [72], we proposed two methods to trigger the wire to fail. In the first method, an active wire segment is converted to a passive sink, where as in the second method, a passive sink is converted to an active sink. We have revised this approach in this article, effectively combining the two methods, turning an active wire segment into an active sink directly. This technique allows us to easily control the mortality of the interconnect

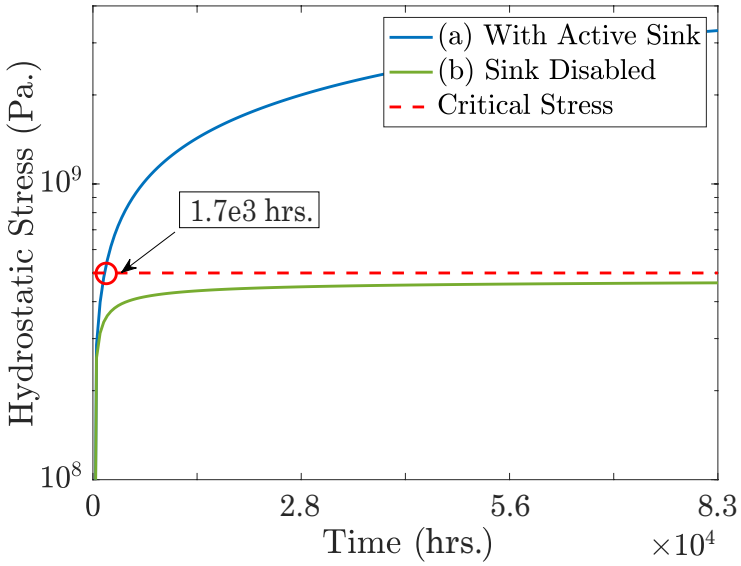


Figure 2.13: Hydrostatic stress progression at the cathode with: (a) an active sink at the anode (b) a disabled sink at the anode

structure simply by controlling the direction of current flow in the sink segment.

To take advantage of this behavior, we have to carefully design the structure such that the tensile stress at the cathode saturates above critical stress for acceleration (active sink), and below critical stress for normal use (disabled sink). When steady-state stress is below critical stress, the structure is considered immortal under EM (will never fail). Hence, unlike the reservoir based method, the sink based method requires careful tuning of three variables (main-branch length, sink length, and current density) to achieve the desired TTF under normal-use and acceleration modes. The TTF results for various configurations are shown in Table 2.1. Note, all these configurations are carefully designed so that the structure is immortal under normal use; the results presented in the table are from when the structure is operated under acceleration mode.

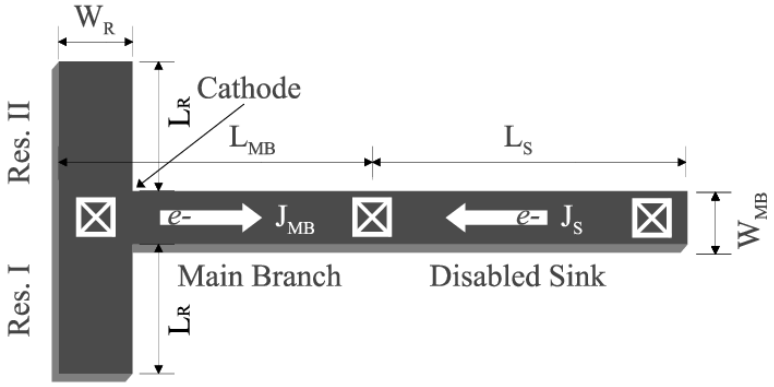
Table 2.1: TTF acceleration with various sink and main-branch configurations

$L_{MB}(um)$	$L_S(um)$	$J(A/m^2)$	$TTF_{EF}(hr)$	$TTF_{LF}(hr)$
100	60	2.90×10^9	1.15×10^5	2.36×10^5
50	30	4.10×10^9	6.75×10^4	1.1×10^5
30	20	6.80×10^9	2.92×10^4	4.47×10^4
20	15	1.20×10^{10}	1.25×10^4	1.84×10^4
15	10	1.60×10^{10}	8×10^3	1.13×10^4
10	7	3.10×10^{10}	3.42×10^3	4.53×10^3
8	6	3.80×10^{10}	2.62×10^3	3.36×10^3
7	4	5.30×10^{10}	1.79×10^3	2.25×10^3

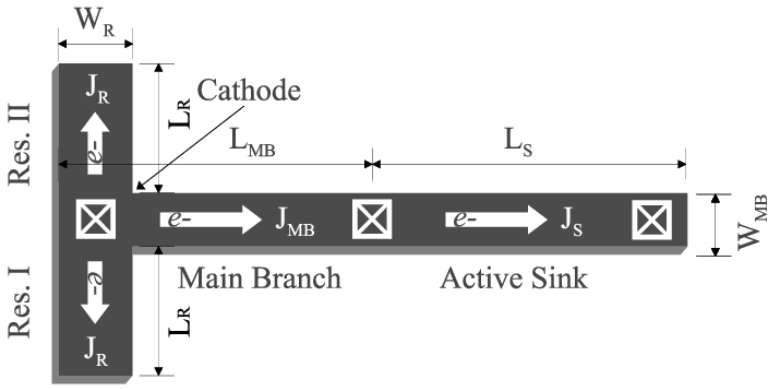
2.3.3 Hybrid EM acceleration technique combining both reservoir and sink structures

In this subsection we leverage the effects of both the reservoir and sink structures to propose a hybrid structure that achieves even better TTF acceleration. The new design, shown in Fig. 2.14, is a 4-segment interconnect structure consisting of two configurable reservoirs, one configurable main-branch, and one configurable sink. We will fix the reservoir length, L_R to be $20um$ and only adjust its width, W_R, to achieve the desired TTF. Likewise, the width of the main branch and the sink segment will be fixed at $0.05um$. In order to design the structure for the desired TTF, we will configure the length of the sink, L_S, width of the reservoir, W_R, length of the main branch, L_{MB}, and current density in the main-branch, J_{MB}. The current density in the sink (or disabled sink for normal use) will be set equal to the main-branch, $J_S = J_{MB}$.

To trigger acceleration mode, the direction of current in the disabled sink segment will be reversed, turning it into an active sink. However, the current density will be kept the same. At the same time, the current flow will be activated in the reservoir segments. Again,



(a)



(b)

Figure 2.14: The proposed hybrid EM acceleration structure with one sink and two reservoir segments under (a) normal use and (b) acceleration mode

the current density in these segments will also be the same as the main branch, $J_R = J_{MB}$. The direction of current flow during normal use and acceleration mode is illustrated in Fig. 2.14(a) and Fig. 2.14(b) respectively.

The simulation results for the proposed structure with varying configurations is shown in Table 2.2. As the results show, this hybrid structure allows us to achieve significant TTF acceleration and meet the lifetime requirement for normal use while keeping the

Table 2.2: TTF acceleration with various sink, main segment, and reservoir configurations

L_{MB} (um)	L_S (um)	W_R (um)	J (A/m^2)	TTF_{EF} (hr)	TTF_{LF} (hr)
100	60	0.3	5×10^9	3.44×10^5	3.53×10^5
50	30	0.4	1.25×10^{10}	7.33×10^3	8.67×10^3
30	20	0.5	2×10^{10}	2.1×10^3	2.52×10^3
20	15	0.6	2.75×10^{10}	1.12×10^3	1.29×10^3
15	10	0.7	3.5×10^{10}	694.44	780.56
10	7	0.8	4.25×10^{10}	472.22	513.89
8	6	0.9	5×10^{10}	344.44	372.22
7	4	1	5.75×10^{10}	261.67	277.78

reservoir length at 20um. This was possible since, unlike the reservoir-based acceleration structure where the additional length of the reservoir was needed to guarantee 10+ years of lifetime during normal use, in this hybrid structure the sink segment is designed to guarantee immortality during normal use. Hence, here the reservoir can be designed solely for enhancing the EM effects during acceleration mode which is only impacted by the reservoirs width, not its length.

With this hybrid structure, we were able to achieve a TTF of 10.9 days for the early failure case, and 11.6 days for the late failure case. However, as previously stated, this is achieved at a working temperature of $\sim$80°C, with no increase in current density, simply by triggering acceleration mode where current flow is activated in the reservoir segments, and the direction of current is reversed in the sink segment. During testing, the operating temperature will be increased to accelerate EM effects, consequently reducing TTF even further. Our goal in this study is to leverage the unique properties of the proposed structures so the desired TTF can be achieved with minimal increase in operating temperature. In the next section we will discuss these testing conditions.

2.4 Temperature-based EM acceleration

In this section we study the impact of temperature in accelerating the EM aging process. If the wire is immortal (tensile stress at the cathode saturates below critical stress), then temperature will not have any impact on EM wear-out. However, if the wire is mortal, then increasing temperature will have a significant impact in accelerating TTF under EM. This is due to the fact that EM is fundamentally an atom diffusion process which is activated only if steady-state tensile stress is above 500MPa (critical stress).

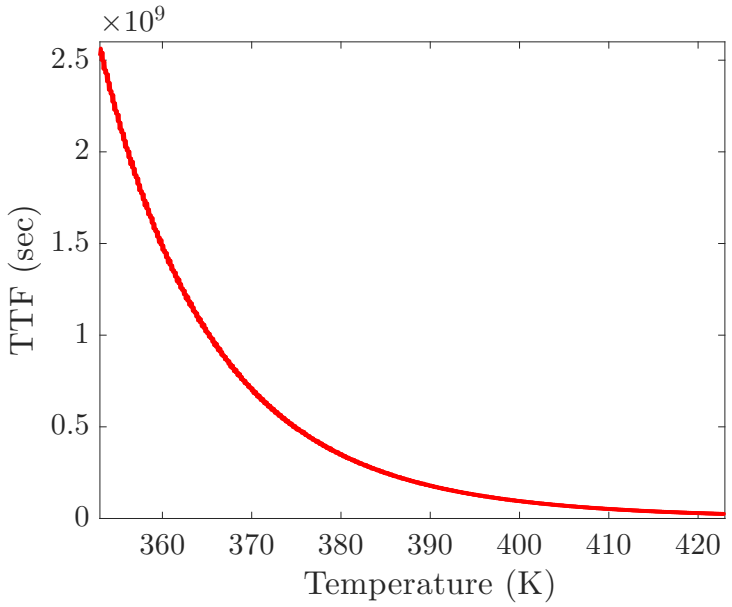


Figure 2.15: Impact of temperature on the structure shown in Fig. 2.10 operating under normal mode

However, as previously mentioned, temperature also accelerates other reliability effects, which is not desirable when the goal is to study EM in isolation. Hence, we will leverage temperature along with the aforementioned reservoir and sink based methods to achieve significant TTF acceleration while staying below the 150°C temperature limit. Before we

discuss the impact that increasing temperature has on the proposed structures operating in acceleration mode, we will first show the results for normal use mode. Specifically, we use the multi-segment structure shown in Fig. 2.10 where $L_R = 20um$, and $W_R = 1um$. This configuration is mortal in its default state and therefore is a good candidate to illustrate the effects of increasing temperature. The simulation results in Fig. 2.15 shows the impact on TTF as temperature is gradually increased. Few of the data points from this figure is shown in Table 2.3.

Table 2.3: Results from temperature based accelerated technique on the structure shown in Fig. 2.10 operating under normal mode

Temp.(°C)	$TTF_{EF}(hr)$	$TTF_{LF}(hr)$
79.85	7×10^5	7.17×10^5
89.85	3.58×10^5	3.64×10^5
99.85	1.7×10^5	1.73×10^5
109.85	8.08×10^4	8.25×10^4
119.85	4.39×10^4	4.47×10^4
129.85	2.24×10^4	2.29×10^4
139.85	1.29×10^4	1.32×10^4
149.85	7.03×10^3	7.19×10^3

As the results show, temperature has an exponential impact on the EM lifetime of mortal interconnect wires. In general, a design rule-of-thumb is *10% increase in temperature will lead to 10X reduction in TTF.*

2.5 Numerical results and discussions

In this section, we present the results from subjecting the proposed structures to the temperature-based testing conditions under acceleration mode. We will show that the proposed methods lead to the targeted 10^5X acceleration in TTF while staying below the

Table 2.4: Total acceleration results: Combining the proposed structure-based EM acceleration methods and temperature-based stressing conditions

Temp.	Reservoir-based Structure		Sink-based Structure		Hybrid Structure	
	TTF_{EF}	TTF_{LF}	TTF_{EF}	TTF_{LF}	TTF_{EF}	TTF_{LF}
79.85	1.49×10^3	1.69×10^3	1.98×10^3	2.45×10^3	263.33	271.94
89.85	747.22	836.11	861.11	1.13×10^3	121.94	125.83
99.85	336.11	380.56	438.89	541.67	58.89	60.83
109.85	168.33	189.72	216.94	266.94	29.44	30.28
119.85	92.5	103.33	110.03	136.67	15.17	15.67
129.85	46.39	52.22	57.5	71.67	8.08	8.36
139.85	25.47	28.61	30.83	94.17	4.47	4.61
149.85	15.25	17.11	19.5	23.97	2.56	2.64

$150°C$ temperature limit

2.5.1 The configurable reservoir based structure subjected to temperature based stressing conditions

First, we use the reservoir-based structure shown in Fig. 2.10 where $L_{MB} = 100um$, $W_{MB} = 0.05um$, $J_{MB} = 2 \times 10^{10} A/m^2$, $L_R = 20um$, and $W_R = 2um$. Table 2.4 summarizes the acceleration results as temperature is gradually increased from ∼80°C to ∼150°C. In this configuration, at ∼150°C, lifetime is reduced from 10+ years down to about 15.26 hours for the early failure case and 17.12 hours for the late failure case. This constitutes to an acceleration in the order of 10^4X.

2.5.2 The configurable sink based structure subjected to temperature based stressing conditions

Second, we use the sink-based structure shown in Fig. 2.12 where $L_{MB} = 7um$, $L_S = 4um$, and $J_{MB} = J_S = 5.3 \times 10^{10} A/m^2$. The resulting TTF for both early-failure

and late-failure cases as the temperature is increased from ∼80°C to ∼150°C is shown in Table 2.4. This configuration yields a TTF of 19.5 hours for the early failure case and 23.97 hours for the late failure case. The acceleration is still in the order of 10^4X. Our results show that the reservoir based method typically achieves slightly better acceleration but at the cost of a higher area-overhead.

2.5.3 The hybrid structure (reservoir + sink) subjected to temperature based stressing conditions

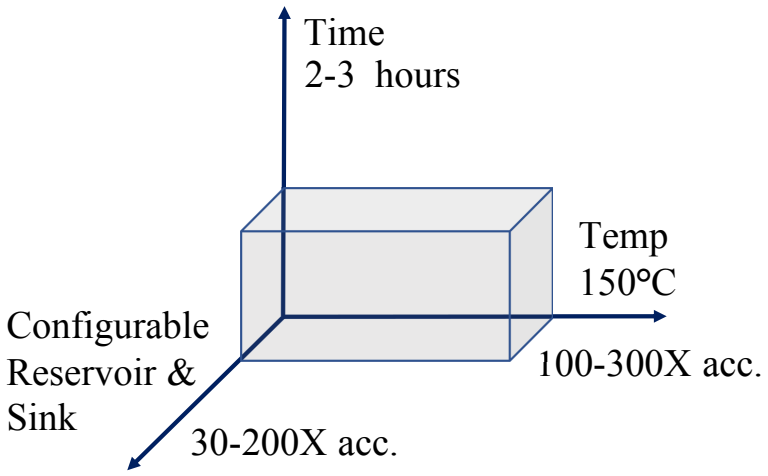


Figure 2.16: Leveraging both structure-based and temperature-based acceleration methods

Lastly, we repeat the same test on the hybrid structure shown in Fig. 2.14. By leveraging both the reservoir based and sink based acceleration methods together, we can achieve an even more impressive TTF reduction without needing to exceed the 150°C temperature limit as illustrated in Fig. 2.16. For the structure shown in Fig. 2.14 where $W_R = 2um$, $L_R = 20um$, $L_{MB} = 7um$, $L_S = 4um$, and $J_{MB} = J_R = J_S = 5.6 \times 10^{10} A/m^2$.

The TTF results for both early-failure and late-failure cases as the temperature is increased from ∼80°C to ∼150°C is shown in Table 2.4.

As the results show, we were able to achieve a TTF reduction from 10 years down to 2.56 hours for the early-failure case, and 2.65 hours for the late-failure case. This is a significant reduction in EM lifetime (10^5X) achieved at a much lower current density and temperature when compared to traditional burn-in testing.

2.6 Summary

In this chapter, we studied two structure-based EM acceleration techniques for fast failure testing. We showed that specially designed EM acceleration structures, based on the unique properties of atomic reservoir and sink segments, can be used to drastically alter the time-to-failure (TTF) of active interconnect wires. The proposed structures can be configured to achieve the desired TTF reduction during acceleration mode, while, at the same time, meet the 10+ year lifetime requirement during normal use. We demonstrated that these structures, when subjected to the traditional temperature based testing conditions, can achieve a lifetime reduction from 10+ years down to a few hours, while staying below a $150°C$ temperature limit. This satisfies the 10^5X reduction in TTF that is typically desired for accelerated testing. The proposed method, for the first time, allows EM testing and validation to be carried out in a controlled manner without the risk of accelerating other reliability effects in the process.

Chapter 3

Post-Silicon Heat-Source Identification and Temperature Estimation

3.1 Related Work and Motivation

Performance counter based power-consumption estimation methods for both high performance and mobile/embedded processors have been developed in the past [73–75]. These methods offer a software-based solution to runtime fine-grain power estimation rather than requiring component-wise power sensors which incur significant design overheads and are prone to sensing and process-based noise similar to embedded temperature sensors [76]. Additionally, performance counters along with the temperature readings from the embedded temperature sensors have also been used to predict the future readings from the embedded

sensors [76–78]. However, as previously mentioned, the number of embedded sensors on the chip is very limited due to their high area and power overheads and they may not always be placed in close proximity to the hot-spots on the chip.

To supplement the temperature readings from the embedded sensors, it is imperative to develop thermal models that can either estimate the temperature profile of the entire chip, or all the thermally vulnerable areas on the chip. To this end, interpolation based methods have been proposed to compute the full-chip thermal map from the sensor readings [79]. Since the number of sensors and their placement have a significant impact on the accuracy of the aforementioned interpolation, smart sensor placement algorithms have been proposed that can be used during design time to find the optimal placement for the given budget of embedded thermal sensors [4, 17, 18]. It has been shown that adapting the aforementioned sensor placement algorithms significantly improves the accuracy of soft-sensing or interpolation based methods that can be used to estimate the temperature of any arbitrary location on the die including the hot-spots. However, these methods are not suitable for chips that are not designed with the aforementioned smart sensor placement algorithm. There is still a lack of an exclusively post-silicon approach that requires no changes to the design of the chip.

Hence, in this work we propose a novel machine-learning based framework to post-silicon temperature estimation for commercial multi-core processors using high-level performance metrics. Here, the correlation between the utilization behavior of the processor shown by high-level performance monitors and the temperature response of the chip is automatically learned. With data being of utmost importance with any machine learning

based approach, in this work we present a thorough and systematic method to measure first-hand thermal and utilization data from commercial microprocessors. The overarching goal of this work is to propose an exclusively post-silicon method to identify all the thermally vulnerable areas of the die and build a thermal model that can be used to estimate the temperatures of these areas during runtime.

3.2 Proposed thermal modeling framework

3.2.1 The new thermal modeling and characterization overview

The proposed thermal modeling approach involves several critical steps. First and foremost, it requires an advanced IR thermography setup that is capable of recording lucid thermal maps of the processor under test while it is executing real workloads. This setup will be discussed in detail in the next subsection. The heatmaps acquired using this system will then be used to objectively determine the locations of prominent heat-sources (or power-sources) on the commercial processor. The located heat-sources will then be clustered together into dominant heat-source clusters, which are the thermally vulnerable spatial locations on the chip (hotspots). Our novel approach to locating these dominant heat-sources will be discussed in Sec. 3.3. Once the heat-sources are located, the IR setup will once again be used to record time-series temperature data of all the identified heat-sources while the processor is subjected to a variety of realistic workloads. At the same time, a suite of high-level performance metrics will be recorded in synchronous with the capture rate of the IR camera. After sufficient data is acquired, a variant of Recurrent-Neural-Networks (RNN) called Long-Short-Term-Memory (LSTM) network will be employed to train the

thermal model. Once trained, the thermal model will be able to use the performance metrics as inputs to estimate the temperatures of all the identified heat-sources in real-time.

3.2.2 Our IR thermography setup with rear-mounted cooling

With any machine-learning based approach, clean and stable datasets are of utmost importance. For the problem at hand, measured thermal data has been shown to be superior, compared to using simulators or other golden models [80, 81] for data generation. Hence, the acquisition of spatial and temporal heatmaps from the processor-under-test becomes and important aspect of the proposed approach.

Specifically, to develop the proposed thermal model for a given microprocessor, two critical pieces of data must be collected. Namely, a time-continuous sequence of spatial temperature data and high-level performance metrics of the microprocessor captured in synchronous with each-other. To this end, we have built an IR thermography setup that allows us to synchronously capture heatmaps ($T(x, y)_t$) and performance metrics ($M(j)_t$) at a constant frequency ($f = 1/\triangle t$). Here, x and y are spatial coordinates, t is time, $\triangle t$ is the time-span between two adjacent time-steps, and $1 < j < m$ where m is the total number of metrics supported by the performance monitoring software.

Our IR thermography setup, shown in Fig. 3.1, is based on the setup proposed in [29]. It features a thermo-electric (Peltier) device mounted on the PCB directly beneath the processor allowing it to be cooled from underneath. This leaves the front side of the processor fully exposed to the IR camera without any interference layer in-between. A programmable DC power supply is used to control the heat-flow through the thermo-electric

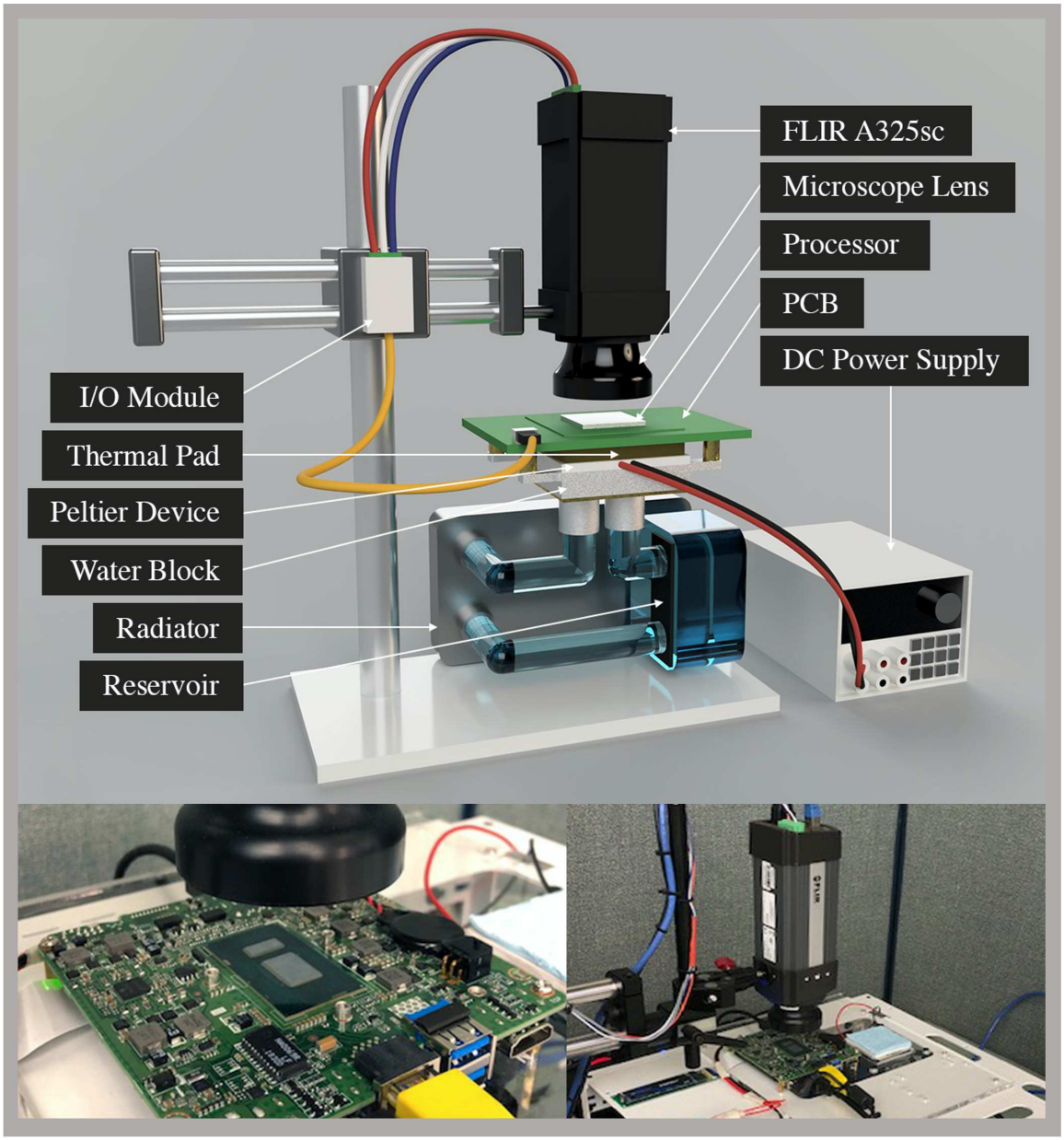


Figure 3.1: Our IR thermography setup

device so that the operating conditions can be matched to the baseline cooling unit (stock heat-sink) using the calibration method discussed in [29]. Unlike the traditional oil-based front-cooling methods, no de-embedding [39] is required in our setup. It should be noted

that this cooling system should only be used for processors that require heat-sinks. Many mobile and embedded processors are designed to be operated without heat-sinks. In such cases, the aforementioned cooling system should not be used during data acquisition.

Detailed description of the IR thermography setup is as follows. The IR camera used in this setup is a FLIR A325sc which supports a maximum imaging resolution of 320×240 pixels (px) with 16-bits of precision per px, and a maximum capturing frequency of 60Hz. The IR sensor is factory calibrated for accuracy across the temperature range of 0°C to 328°C, and resolves the IR spectral range of $7.5\mu m$ to $13\mu m$. A microscope lens is used to achieve the spatial resolution of $50\mu m$ per px. The FLIR A325sc has an internal waveform generator that outputs a square waveform in synchronous with the capture rate of the camera. An I/O module is used to interface the waveform generator to the processor-under-test so that the performance metrics (recorded in the processor) can be synchronized with the thermal data recorded by the IR camera. Mounted on the PCB directly underneath the processor is the thermo-electric-based cooling system which includes a Peltier device powered by a programmable DC power supply. The Peltier device is the primary cooling mechanism, keeping the chip operating at nominal working conditions. Thermal energy flows from the cool-side of the Peltier device to its hot-side, where it must be dissipated in order to ensure proper operation. To this end, an off-the-shelf liquid cooling system is used to cool the hot-side of the Peltier device.

It is crucial to note key caveats regarding the infrared thermography setup. First, the maximum inference frequency of the proposed model will be ultimately limited by the capturing frequency of the IR camera. In our case, the maximum inference frequency cannot

exceed 60Hz. If inference at a higher frequency is desired, then a more advanced IR camera will be required. Second, the peltier device used in our setup is the TEC1-12710 which has a maximum power rating of 110W [82]. It should be noted that the cooling potential of the peltier device must be significantly larger than the thermal design power (TDP) of the processors under test. This is due to the fact that the rear mounted cooler has to overcome the low thermal conductivity of the PCB which is now in-between the processor and the cooler. We found that the peltier device we used was able to easily match the cooling potential of the stock heat-sink of the two processors we tested. However, more powerful peltier devices may be required for higher-end processors. If the peltier device is not able to match the stock cooling system of the chip, but is able to maintain the chip's temperature under its thermal limits, then a method such as the one presented in [39] can be used to scale the captured thermal maps to the amplitude that would have been observed under the stock cooling system. Third, IR camera systems are factory calibrated on materials with high emissivity coefficient (ϵ). If the heat-spreader covering the processor's die has a low ϵ, then the temperature readings from the camera will not be accurate. One way to address this issue is to recalibrate the camera to the given heat-spreader using readings from the processor's integrated thermal sensors. However, this method is not recommended as the internal sensors have an accuracy of $\pm 5°$C [83]. Another method, which we prefer, is to improve ϵ of the heat-spreader by covering it with a thin layer of a better emitter. As suggested by the authors of [29] one simple option is masking tape ($\epsilon \approx 0.92$).

3.3 Heat-source identification

One important aspect of building a thermal model for a processor is identifying the dominant heat/power-source clusters. These are the critical areas of the chip for many online or dynamic thermal/power management schemes. Locating the heat-source clusters or hotspots during design time is trivial as it can be done through power/thermal simulation tools. However, post-silicon identification of these locations with no knowledge of the chip's proprietary design is not trivial. As a result, locating these heat-source clusters without the floorplan and layout information becomes an important problem. In this section, we will present our novel approach to locating these heat-sources on commercial processors exclusively using measured thermal data.

3.3.1 Laplacian operation for heat-source identification

We start with the general thermal diffusion equation shown below [84]

$$\rho C_p \frac{\partial T}{\partial t} - \nabla(\kappa \nabla T) = g, \tag{3.1}$$

where T is temperature (K), ρ is the mass density of the material ($\mathrm{kg} \cdot \mathrm{m}^{-3}$), C_p is the mass heat capacity ($\mathrm{J} \cdot \mathrm{kg}^{-1} \cdot \mathrm{K}^{-1}$), κ is the thermal conductivity and g is the spatial heat energy generation ($\mathrm{W} \cdot \mathrm{m}^{-3}$).

Since, in this step, we are only concerned with the spatial information that we can extract from a single time-step, we can ignore the transient terms in (5.7). We then get the 2D steady-state thermal equation (assuming homogeneous material with location independent κ):

$$-\kappa \nabla^2 T(x, y) = g_T(x, y) \tag{3.2}$$

where ∇^2 is the Laplace operator. From the simplified heat equation (3.2), we can see that *the negative spatial Laplacian of the temperature distribution across the die is equal to the spatial heat generation.* Therefore, we can perform the 2D spatial Laplacian on a given thermal map to locate the underlining heat-sources $g_T(x, y)$. The exact amplitude of $g_T(x, y)$ remains an unknown since we do not know the value of κ, however this is not important since we are only interested in the spatial locations of the heat-sources, hence relative amplitude will suffice. This method also works even if there is a thin heat-spreader layer with a conductive surface (for example a die with heat-spreader and package). This is due to the fact that, although the heat-spreader will distribute the heat across its surface and dissipate it, the spacial locations of the underlying heat-sources do not change.

To illustrate this idea, we simulate a simple structure in COMSOL Multi-Physics where three distinct heat-sources are placed underneath a thin heat spreader with a conductive boundary in-between. The simulation results (Fig. 3.2) show that, by applying 2D Laplacian transformation on the temperature distribution, T(x,y), observed on the heat spreader, the three distinct heat-sources located underneath the heat-spreader can be easily identified.

3.3.2 Comprehensive heat-source identification flow

In this subsection, based on the aforementioned principles, we present our approach to identifying the major heat-sources using measured thermal-maps captured from the commercial processor-under-test. First, the raw thermal-map is pre-processed to remove the inherent noise present in measured IR data. After this, 2D spatial Laplacian is applied

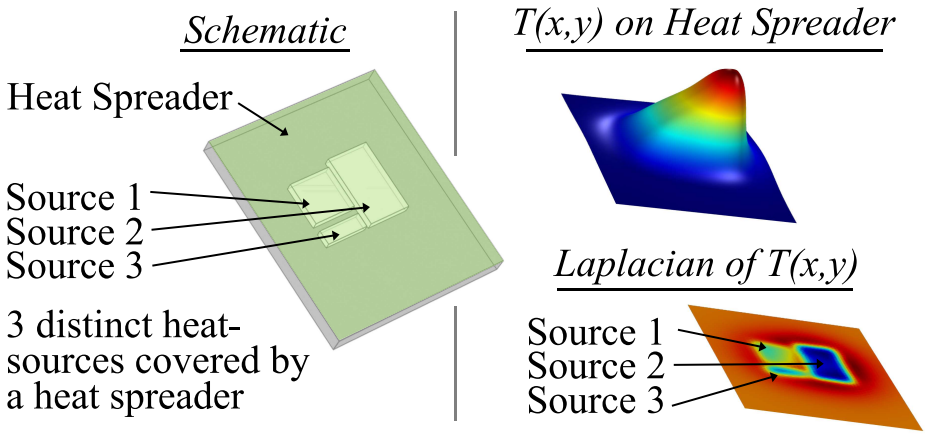


Figure 3.2: COMSOL validation of the heat-source identification method

to locate the active heat-sources in 2D space. This process is repeated on ten-of-thousands of heatmaps captured at different time instances under different workloads. Lastly, a K-means based clustering algorithm is invoked to find the dominant heat-source clusters with high densities of heat-sources. The proposed method is illustrated in Fig. 3.3. For clarity, we will demonstrate the algorithm using the following example.

Pre-processing for noise reduction via DCT

We start with a thermal-map (i.e. Fig. 3.4) captured from the dual core Intel i5-3337U processor under test.

The raw thermal-map may contain noise, which must be removed as a pre-processing step. This step is crucial because the 2D discrete Laplacian

$$\begin{aligned}\nabla^2 f(x,y) = f(x-1,y) + f(x+1,y) + f(x,y-1) \\ + f(x,y+1) - 4f(x,y),\end{aligned} \tag{3.3}$$

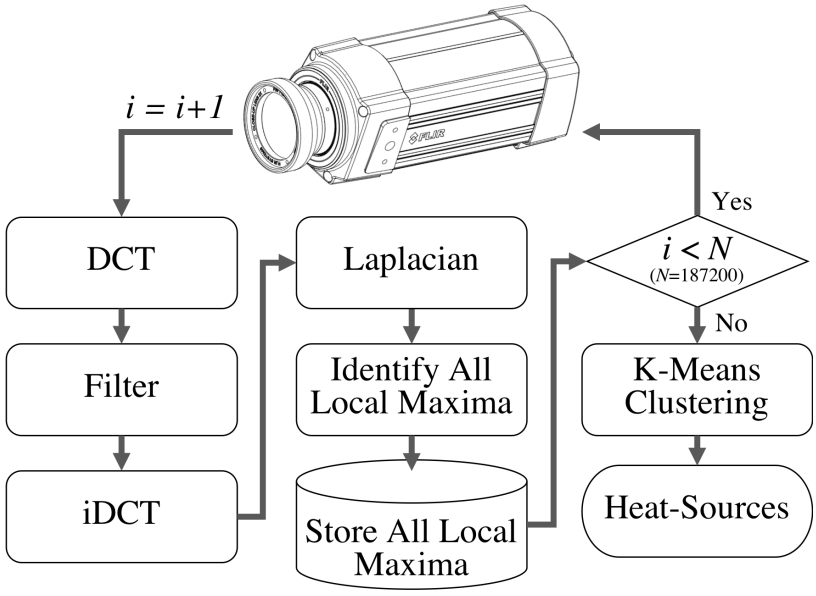


Figure 3.3: Illustration of our novel heat-source identification flow

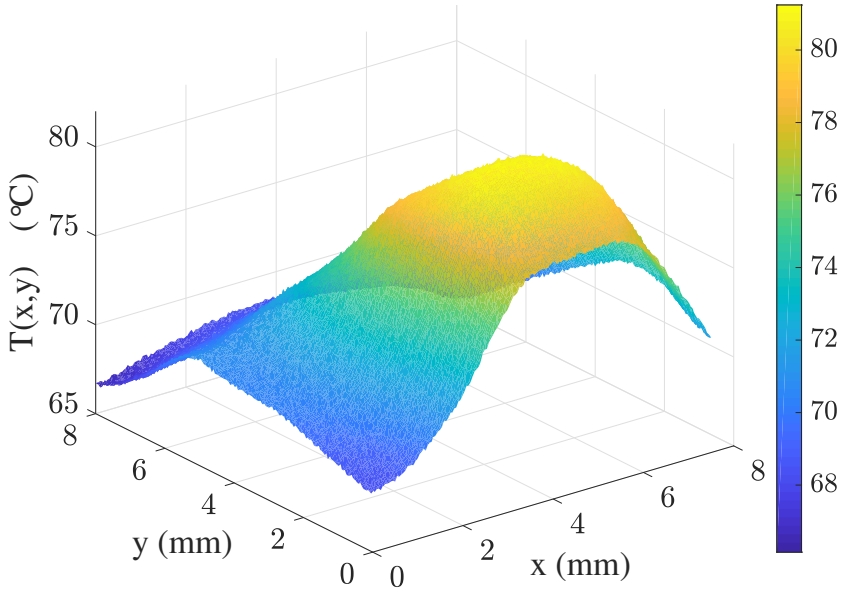


Figure 3.4: Heatmap of the Intel i5-3337U captured using our IR system

is very sensitive to local difference of adjacent pixels [1]. 2D discrete cosine transformation (DCT) filter is an effective method for eliminating high-frequency noise, by transforming the heatmap into spatial frequency domain, masking the high-frequency components, and then transforming back to the original space domain. A 2D DCT consists of two separate 1D DCT operations, which can be denoted as

$$f_k = \frac{a_0}{\sqrt{N}} + \sqrt{\frac{2}{N}} \sum_{i=1}^{N-1} a_i \cos \frac{(2i+1)k\pi}{2N}, \, 0 \leq k < N, \tag{3.4}$$

where vector $\{a_i\}$ is the original data, and $\{f_k\}$ is the result of 1D DCT. A 2D DCT is completed by applying 1D DCT on each column and then on each row of the matrix. With the heatmap $T(x, y)$ transformed to 2D frequency domain $F(x, y)$, a filtered frequency map $\mathcal{F}(x, y)$ can be obtained by applying a mask

$$\mathcal{F}(x, y) = F(x, y)m(x, y), \tag{3.5}$$

where $m(x, y)$ is the mask map valued 0 at high frequencies and 1 at low frequencies. The filtered heatmap $\mathcal{T}(x, y)$ is then obtained by taking the inverse 2D DCT on the filtered frequency map $\mathcal{F}(x, y)$. Similar to its forward counterpart, the inverse 2D DCT consists of two separate inverse 1D DCT steps on the rows and columns respectively. The inverse 1D transformation of (4.1) is

$$a_i = \frac{f_0}{\sqrt{N}} + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} f_k \cos \frac{(2i+1)k\pi}{2N}, 0 \leq i < N. \tag{3.6}$$

This operation performed on the noisy heatmap previously shown in Fig. 3.4 results in the filtered heatmap shown in Fig. 3.5.

[1] For a heatmap with 177×166 pixels, with temperature ranging from 65°C to 80 °C, the laplacian range is approximately at ± 0.025°C/pixel^2. While with noise introduced, the laplacian can easily go up to ± 1.0°C/pixel^2, which is much higher than the useful laplacian component.

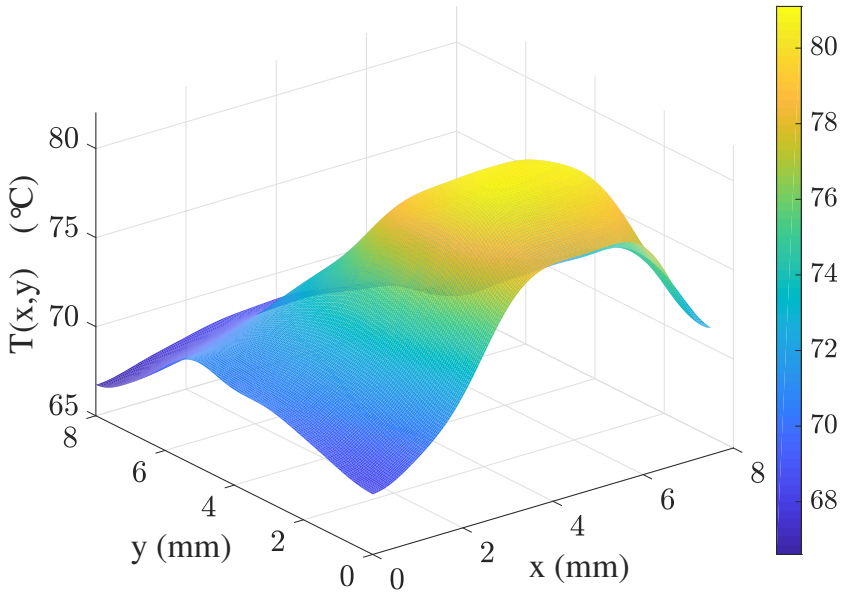


Figure 3.5: The noise-reduced heatmap of the Intel i5-3337U

Temperature Laplacian for heat-source identification

The Laplacian operation in (3.2) can now be applied to the noise-less heatmap, which reveals the locations of the internal heat-sources that were active during the time this particular heatmap was captured. These heat-sources can be identified by locating all the local maxima on the negative Laplacian of the temperature distribution as shown in Fig. 3.6.

K-means clustering for dominant heat-source localization

While the above step can be used to identify the heat-sources that were active during the time the heatmap shown in Fig. 3.4 was recorded, there is no guarantee that all the prominent heat-sources within the chip were active during that time. In fact, many of the

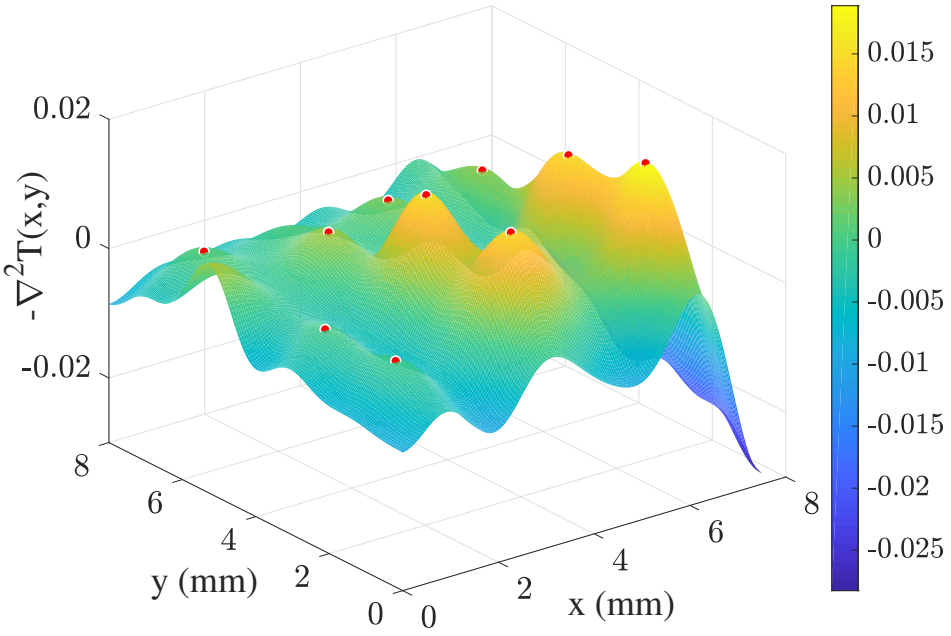


Figure 3.6: Negative Laplacian of the heatmap with all the heat-sources (red) identified

heat-sources are disabled at any given time due to the extreme power and clock gating used in modern processors. In order to ensure that most, if not all, of the prominent heat-sources on the chip are identified, we repeat the aforementioned heat-source identification process on many ($N \sim 2 \times 10^5$) heatmaps that were collected while the processor is subjected to a multitude of different workloads with varying execution patterns. This process increases the chances of activating all the prominent heat-sources on the chip, at least once, so that their thermal signature can be recorded. The aggregate of the local maxima identified using this method is shown as clusters of red dots in Fig. 3.7.

This method results in dense clusters of heat-sources. However, it is not possible to track the temperature of each point in the cluster. Instead, we use the K-means clustering algorithm (using the "elbow criterion" to determine the value of k) to identify the centroids

of the clusters (shown as yellow dots in Fig. 3.7). We will, from this point, refer to these centroids as our distinct heat-source clusters or simply as heat-sources. In total, we were able to identify 18 prominent heat-source clusters on the dual core Intel i5-3337U processor which has only 2 on-chip temperature sensors. Likewise, as we will show in Sec. 5.4, 20 heat-sources were identified on the quad core Intel i7-8650U which only has 4 on-chip temperature sensors.

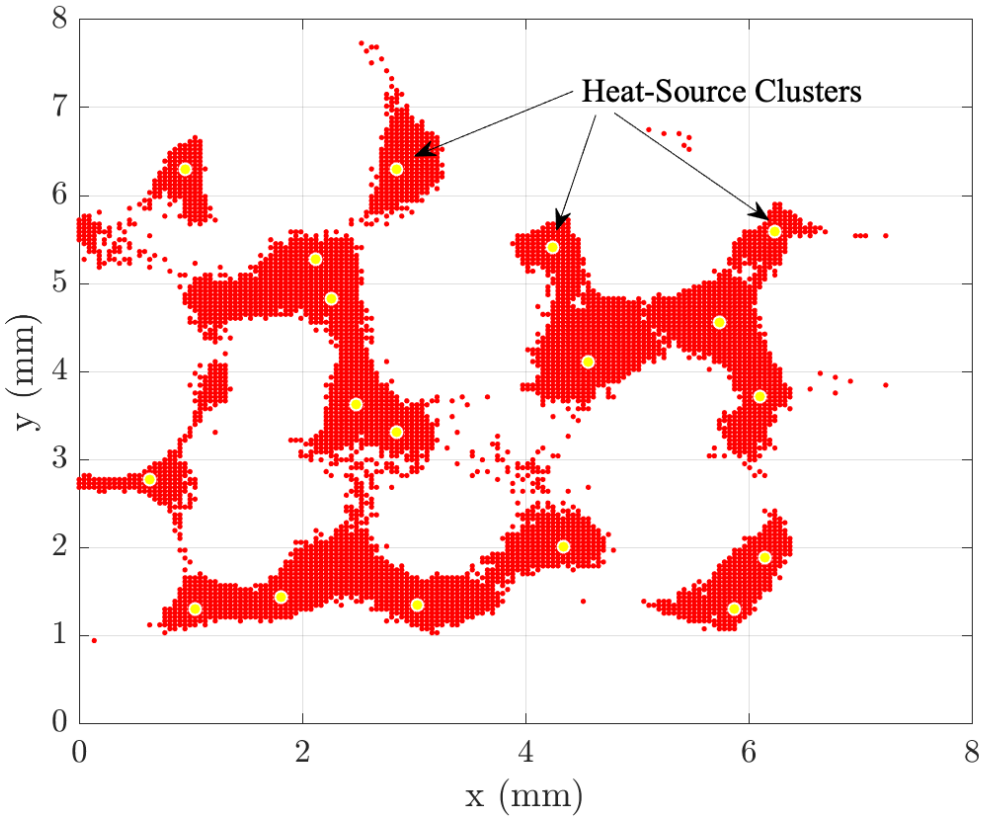


Figure 3.7: Distinct heat-sources (red) extracted from 187200 heatmaps of the Intel i5-3337U and dominant heat-source clusters (yellow) identified using k-means

With all the heat-sources on the chip identified, our next goal is to derive a model that can be used to estimate their temperatures in real-time. The derivation of this model

will be detailed in the subsequent sections. For clarity of the presentation, the discussions in Sec. 3.4 - 3.6 will only consider the Intel i5-3337U. These exact steps will also be implemented on the Intel i7-8650U. Sec. 5.4 will show the implementation of the proposed methodology and experimental results for both chips.

3.4 Machine learning based thermal modeling: Datasets

3.4.1 Runtime temperature measurement

The heat-source identification method discussed in Sec. 3.3 allowed us to locate 18 distinct heat-source clusters on the Intel i5-3337U dual core processor under test. With the thermal model, our goal is to accurately estimate the temperatures of these heat-sources during online operation. Hence, in order to train the regression model, we need time-series temperature data of these 18 heat-sources. With the IR thermography setup (Fig. 3.1) we can directly record the temperatures of the identified heat-sources while the processor is subjected to a variety of workloads. This temperature data measured directly from the processor gives us a significant advantage in developing an accurate thermal model, as opposed to relying on another previously established model or simulator to acquire the required data-sets. In this study, we strictly use first-hand measured data for training, and later for testing the accuracy of the trained model.

3.4.2 Runtime performance metrics

While the IR setup allows us to capture the temperature of the processor externally, the other major part of the data-set comes from monitoring the utilization of the

processor while the temperature data is being recorded. One way to monitor the processor's utilization is through online performance monitoring software, which are supported by most, if not all, major manufacturers of commercial microprocessors. In this work we use high-level performance metrics provided by tools such as Intel's Performance Counter Monitor (IPCM) [85]. These provide a comprehensive high-level view of the processor's utilization with system-level metrics such as the current frequency of the cores, instruction counts, cache hit/miss rates, sleep-state residency, temperature from the internal sensors, etc. In total, IPCM provides 80 performance metrics (I_1 to I_{80}) for the Intel i5-3337U. The complete list of these performance metrics is given in Table 4.1. Since these performance metrics are a good representation of the processor's utilization, we can train a model which can accurately estimate the temperature of the hot-spots using these metrics as inputs. Note, it is important to ensure that these performance metrics are captured in synchronous with the thermal data captured by the IR camera. Hence, as previously mentioned, the IR camera's internal waveform generator, along with an I/O device, is used to synchronize the capture rate of the camera and the performance metrics recorded on the test system. This setup ensures that, at the frequency of 60Hz, one set of IPCM data is recorded in tandem with each set of temperature data captured by the IR camera.

Thermal models based on runtime performance metrics have been demonstrated in the past [86–88]. However, the existing methods are not practical to implement in modern commercial processors for several reasons. First, for each functional unit (FU) on the chip, the low-level performance counters that have significant correlation with the power-draw of the given FU must be manually identified. However, this is under the assumption that

micro-benchmarks can be used to target a single FU in isolation so that this correlation can be determined, this is not feasible in modern processors. Second, even if these correlations can be found, the number of low-level performance counters that can be recorded in parallel is limited by the number of free programmable registers available in the processor. In the case of the Intel i5-3337U, only 11 registers were available. Since more than one metric is typically needed to model the temperature of a single FU, it is not possible to track the temperature of all the FUs on the chip in parallel. Alternatively, in this study we use high-level performance metrics offered by performance monitors such as Intel's Performance Counter Monitor (IPCM). The correlation between the transient behavior of the high-level performance metrics and the thermal response of the previously identified hot-spots are automatically learned through training. This makes the proposed method more practical for modern commercial processors with advanced micro-architectures. Moreover,the proposed approach does not require any information regarding the chip's architecture or floorplan, which is typically necessary for an FU-wise temperature estimation.

3.5 Machine learning based thermal modeling: IPCM input reduction

From the machine learning perspective, the larger the number of inputs and outputs, the more complex the model will need to be. Currently we have 80 inputs and 18 outputs. However, not all the inputs are important and relevant from the thermal/power perspective. To eliminate the irrelevant inputs, we need to identify the IPCM metrics that

Table 3.1: IPCM metrics for the Intel i5-3337U

Pkg.	Pkg.	Core1.1	Core1.2	Core2.1	Core2.2
exec	inst nom	exec	exec	exec	exec
IPC	inst nom%	IPC	IPC	IPC	IPC
freq	C2res%	freq	freq	freq	freq
afreq	C3res%	afreq	afreq	afreq	afreq
L3 miss	C6res%	L3 miss	L3 miss	L3 miss	L3 miss
L2 miss	C7res%	L2 miss	L2 miss	L2 miss	L2 miss
L3 hit	energy (J)	L3 hit	L3 hit	L3 hit	L3 hit
L2 hit	temp	L2 hit	L2 hit	L2 hit	L2 hit
L3 MPI		L3 MPI	L3 MPI	L3 MPI	L3 MPI
L2 MPI		L2 MPI	L2 MPI	L2 MPI	L2 MPI
read rate		C0res%	C0res%	C0res%	C0res%
write rate		C1res%	C1res%	C1res%	C1res%
inst count		C3res%		C3res%	
ACYC		C6res%		C6res%	
physIPC		C7res%		C7res%	
physIPC%		temp		temp	

have little correlation with the thermal response of the heat-sources. If these metrics are removed, we can reduce the number of inputs to our model while maintaining its accuracy. This step is especially crucial for real-time applications as it leads to a more compact model with lower inference latency.

To determine if a given IPCM metric is relevant to any of the heat-sources, we view the IPCM metrics as heat-source stimulants. Hence our goal will be to identify the IPCM metrics that are effective stimulators for the heat-sources on our chip. The IPCM metrics that fail to stimulate any of the heat-sources can be deemed thermally irrelevant and can therefore be removed. To this end, we first apply a DVFS heuristic algorithm to simulate spatial power ($W \cdot m^{-}3$) on a given heat-source induced by a targeted IPCM metric. We then set up the heat partial differential equation (PDE) for the given heat-source. The thermal coefficients for the PDE will be obtained by using the least-squares method. After this, we

can compute the estimated input power from the measured temperature at the given heat-source. If the targeted IPCM metric is relevant to the given heat-source, then the estimated power should agree with the IPCM metric activities. In the following subsections, we give detailed explanation of the proposed method.

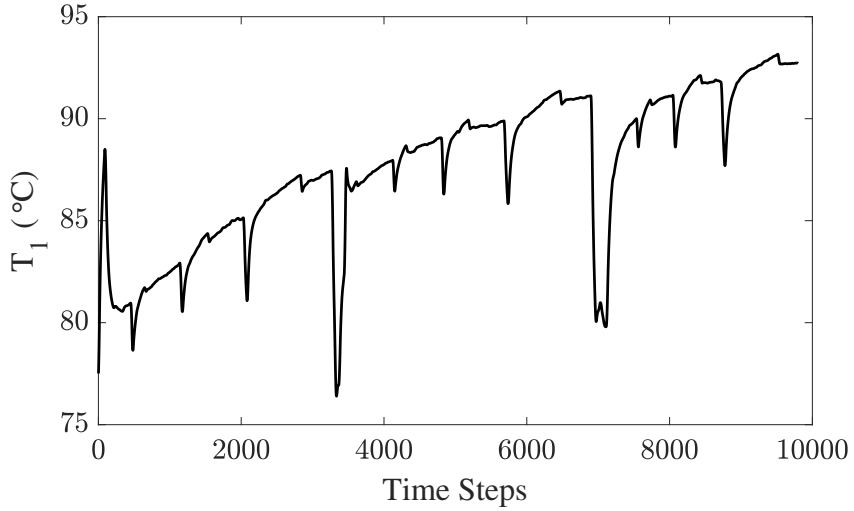


Figure 3.8: Transient temperature at heat-source #1 over time

3.5.1 Transient power estimation from measured thermal map

For our problem, we first need to estimate the transient power density at the heat-sources from transient heatmaps. Then we will calculate the correlation in the time domain. The total power of a CMOS digital chip is typically determined by clock frequency, supply voltage, and capacitance of transistors. Today's DVFS techniques couple frequency and voltage (called power states), which makes frequency the only variable needed to determine energy consumption. Official specifications released by Intel [89, 90] show that the active

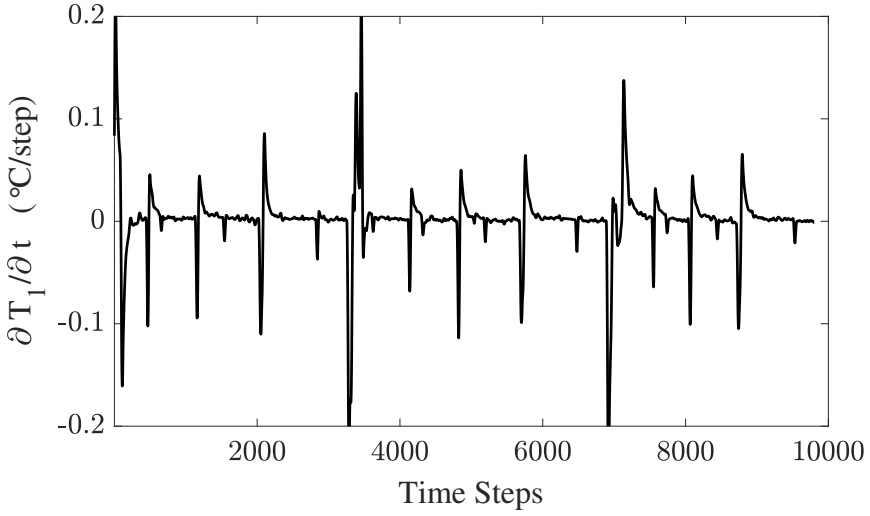


Figure 3.9: Time derivative of temperature at heat-source #1

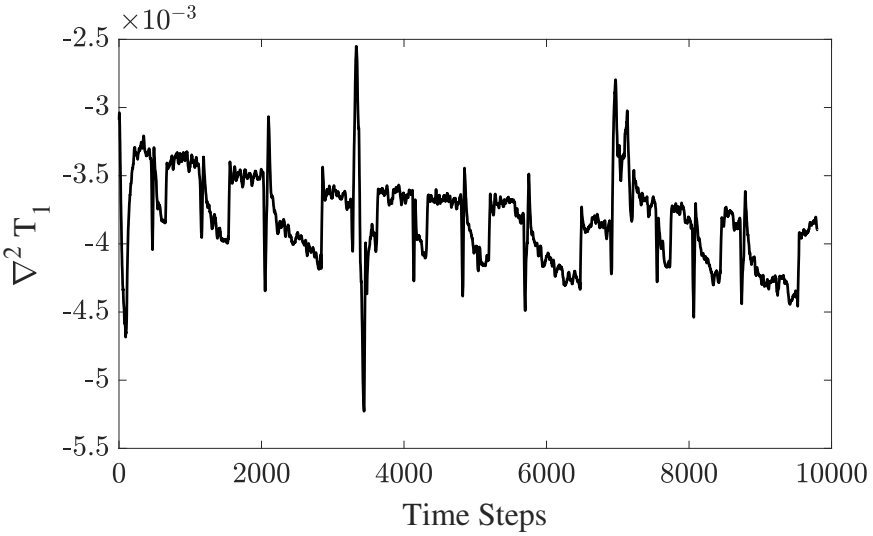


Figure 3.10: Laplacian of temperature at heat-source #1.

power of the CPU is linearly related to the operating frequency and squarely related to supply voltage, while the static power remains almost constant. In addition, [89] states that frequency and power are linearly related; the official data of power states in [90] illustrates

this linear relationship. Thus the total chip power is cubically related to frequency, which can be expressed by

$$P = CV^2 f + P^{(S)} = \eta f^{\alpha} + P^{(S)} \tag{3.7}$$

where P denotes the total power, C is capacitance, V is supply voltage, f is operating frequency, which is known, $\alpha = 3$ is a constant, η is the coefficient for frequency scaling and $P^{(S)}$ is the static power consumption.

As mentioned previously, we simulate the spatial power density by applying DVFS heuristic algorithm to IPCM metrics. The power density g_n at heat-source #n located at (x_n, y_n), can be computed by considering all the IPCM metrics together:

$$\begin{gathered} g_n(t) \propto \sum_{m=1}^{M} \eta_m (I_m(t) + \beta_m) f^{\alpha} + g_n^{(s)} \\ n \in [1, 2, 3, \cdots, N] \\ m \in [1, 2, 3, \cdots, M] \end{gathered} \tag{3.8}$$

where $g_n^{(s)}$ denotes static power density at heat-source #n. M and N denote the total number of IPCM metrics and heat-sources respectively. $I_m(t)$ is the value of the m^{th} IPCM metric at time step t, η_m is a coefficient and β_m serves as a constant bias for the m^{th} IPCM metric.

Let's consider the case where the power density at heat-source #n is very sensitive to one of the IPCM metrics. In this case, the power density at heat-source #n must have high correlation with this particular metric. Hence, power density can be estimated using just the targeted IPCM metric (3.9).

$$g_n(t) \approx \eta_k f^{\alpha} (I_k(t) + \beta_k) + g_n^{(s)} \tag{3.9}$$

where η_k is a constant coefficient for the targeted k^{th} IPCM metric. In (3.9), higher sensitivity will lead to higher equality.

In section 3.3, a simplified steady-state thermal diffusion equation was used to convert the heatmaps to scaled powermaps. But in this section we have to consider the transient effects so we start with the heat partial differential equation (PDE) in the time domain:

$$\rho C_P \frac{\partial T_n(t)}{\partial t} - \kappa \nabla^2 T_n(t) = \eta_k f^{\alpha}(I_k(t) + \beta_k) + g_n^{(s)} \tag{3.10}$$

In this equation, we assume that the heat-source temperature T_n is mainly stimulated by the k^{th} IPCM metric. However this may not always be true. When (3.10) approximately holds, then the k^{th} IPCM must be highly correlated with this heat-source, which is what we are looking for. If (3.10) does not hold, the weak correlation will manifest in the form of weak cross-correlation coefficient and spurious thermal constants which will be discussed in subsection 3.5.2. In subsection 3.5.2 we will refine the correlations and extract relevant IPCM metrics. Before this analysis, let us re-write (3.10) as:

$$\frac{\rho C_P}{\eta_k} \frac{\partial T_n(t)}{\partial t} - \frac{\kappa}{\eta_k} \nabla^2 T_n(t) = f^{\alpha}(I_k(t) + \beta_k) + \frac{g_n^{(s)}}{\eta_k} \tag{3.11}$$

where $T_n(t)$ is the temperature at heat-source #n at time step t acquired from measured thermal data. $\frac{\rho C_P}{\eta_k}$ and $\frac{\kappa}{\eta_k}$ are thermal constants scaled by η_k. Suppose we have sufficient amount of time steps, by stacking all the data along time vertically, (3.11) can be re-written as:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{3.12}$$

where $\boldsymbol{A}$ is a matrix with a total of 4 columns, $\boldsymbol{x}$ is an unknown vector with a total of four

elements, and $\boldsymbol{b}$ is a vector of known IPCM data such that

$$\begin{aligned}
\boldsymbol{A}_t &= \begin{bmatrix} \frac{\partial T_n(t)}{\partial t} & -\nabla^2 T_n(t) & -f(t)^\alpha & -1 \end{bmatrix} \\
\boldsymbol{x} &= \begin{bmatrix} \frac{\rho C_P}{\eta_k} & \frac{\kappa}{\eta_k} & \beta_k & \frac{g_n^{(s)}}{\eta_k} \end{bmatrix}^T \\
\boldsymbol{b}_t &= f(t)^\alpha I_k(t)
\end{aligned} \tag{3.13}$$

Here, the optimal solution for $\boldsymbol{x}$ can be obtained by applying the least-squares method,

$$\begin{aligned}
\boldsymbol{x}^* &= \begin{bmatrix} \left(\frac{\rho C_P}{\eta_k}\right)^* & \left(\frac{\kappa}{\eta_k}\right)^* & (\beta_k)^* & \left(\frac{g_n^{(s)}}{\eta_k}\right)^* \end{bmatrix}^T \\
&= (\boldsymbol{A}^T \cdot \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{b}
\end{aligned} \tag{3.14}$$

Then we obtain the two power densities for the given heat-source: The estimated power density, $g_{n,est}(t)$, by substituting x^* back to (3.11),

$$g_{n,est}(t) = \left(\frac{\rho C_P}{\eta_k}\right)^* \frac{\partial T_n(t)}{\partial t} - \left(\frac{\kappa}{\eta_k}\right)^* \nabla^2 T_n(t) \tag{3.15}$$

and the IPCM activity related power density, $g_{n,I_k}(t)$

$$g_{n,I_k}(t) = f(t)^\alpha \left(I_k(t) + (\beta_k)^*\right) + \left(\frac{g_n^{(s)}}{\eta_k}\right)^* \tag{3.16}$$

The transient power density estimation, $g_{n,est}(t)$, contains two contributions: the time derivative for the first term and the Laplacian of temperature for the second term. Fig. 3.9 and Fig. 3.10 show the two components for one heat-source location (heat-source #1). As we can see, both components are quite significant and should be considered. Now we are ready to compute the correlation by looking at those two power densities over time.

3.5.2 IPCM correlation analysis and refinement

We now compute the power density correlation between the heat-source #n and the targeted k^{th} IPCM metric. We use the cross-correlation (CC) definition [91] of two

deterministic and discretized digital signals, which measures the degree of similarity between the two time series signals:

$$\begin{aligned} X_{n,k}[t_i] &= \left(g_{n,est} \otimes g_{n,I_k}\right)[t_i] \\ &= \sum_{j=-\infty}^{\infty} g_{n,est}(j) \cdot g_{n,I_k}(t_i + j) \end{aligned} \tag{3.17}$$

where t_i is the discretized time point that we are interested in and $\otimes$ indicates the convolution operation. Then we use the normalized maximum absolute value of $X_{n,k}$, between $[0, 1]$, as the CC measurement of the two power signals.

In the following discussions, we take heat-source #1 (measured thermal data in Fig. 3.8) with the 11^{th} IPCM metric, i.e. *read rate* (Fig. 3.11), as an example to illustrate a strong correlation. We also analyze this same heat-source with the 31^{st} IPCM metric, i.e. *L3 hit* (Fig. 3.12), as a counter example to show weak correlation. The estimated power density and IPCM activity related power density for the two IPCM metrics are compared in Fig. 3.13 and 3.14 respectively. As we can see, a strong correlation (0.88, Fig. 3.15) is observed for the 11^{th} IPCM metric, while a weak correlation (0.28, Fig. 3.16) is observed for the 31^{st} IPCM metric. Although, heat-source #1 is weakly correlated with the 31^{st} IPCM metric, it is possible that this metric is better correlated with some other heat-source. It is also possible that more than one IPCM metrics are highly correlated with heat-source #1. In Fig. 3.17 we show the CC coefficient of heat-source #1 with respect to all 80 IPCM metrics. As we can see, it is highly correlated with more than one metric. Note that CC coefficient of $24^{th}, 40^{th}$, and 68^{th} IPCM metrics are set to zero since they are temperature

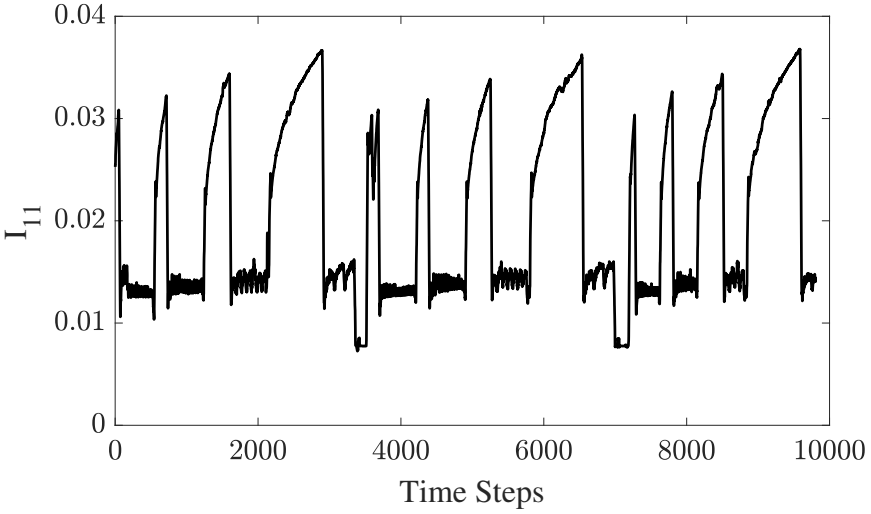


Figure 3.11: 11^{th} IPCM data I_{11}

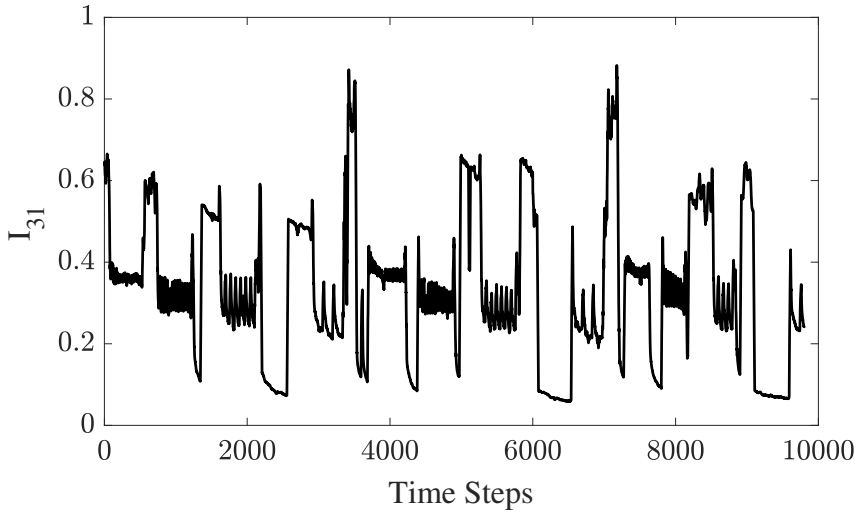


Figure 3.12: 31^{st} IPCM data I_{31}

sensor data which are not power stimulants.

Our detailed study shows that some correlations are still spurious. We observe that thermal constants ratio $(\rho C_P)/\kappa$ obtained from optimal solution x^* (3.14) are not

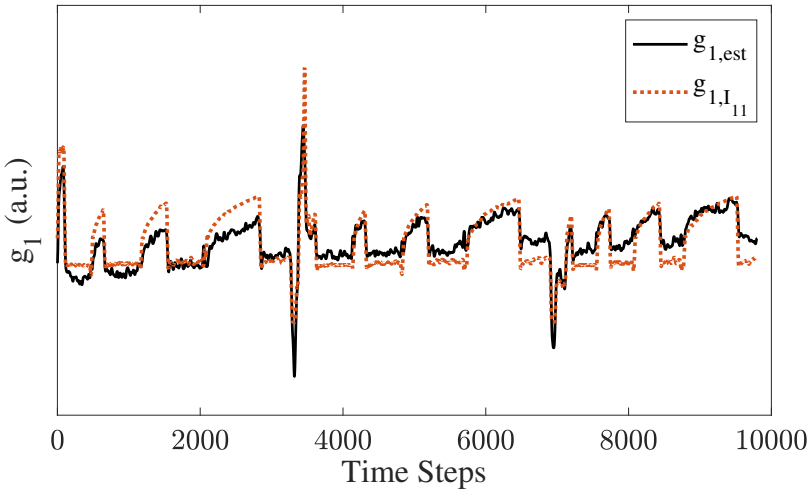


Figure 3.13: Estimated power density at heat-source #1 compared to 11^{th} IPCM-related power density

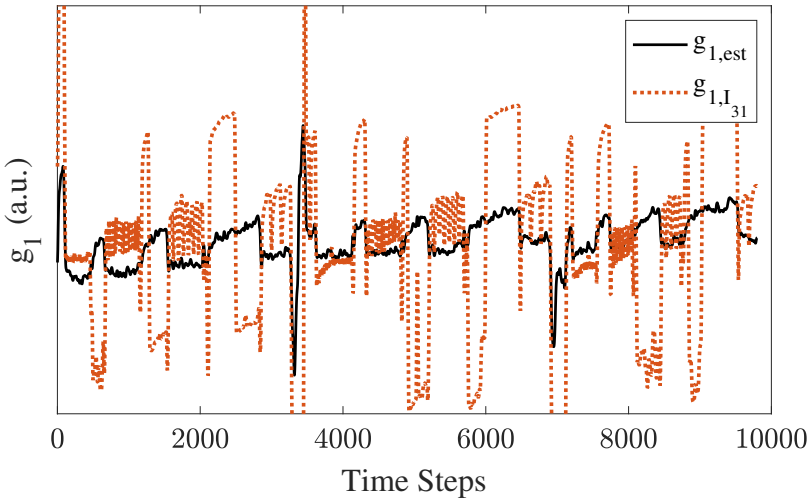


Figure 3.14: Estimated power density at heat-source #1 compared to 31^{st} IPCM-related power density

constant for different IPCM metrics as shown in Fig. 3.18. This indicates that the related correlations have some error and noise. In order to address this problem, we need to find the real ratio of thermal constants. A simple technique we apply is to extract the IPCM metrics

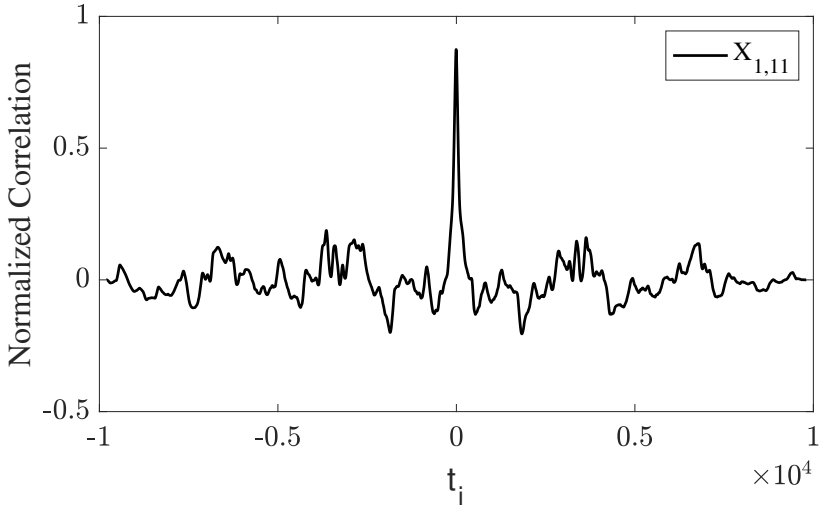


Figure 3.15: CC between heat-source #1 and 11^{th} IPCM with coefficient 0.88

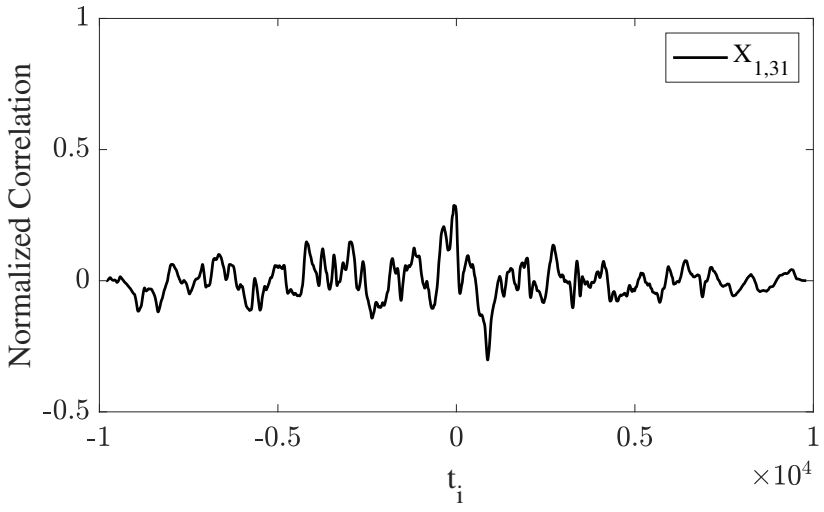


Figure 3.16: CC between heat-source #1 and 31^{st} IPCM with coefficient 0.28

that have high correlations (e.g. > 0.8), and then take an average of their ratios ($\rho C_P/\kappa$) as the real ratio. Then we update the correlations by fixing this ratio for (3.10). By using this technique, we identify 7 relevant IPCM metrics for heat-source #1; their associated

thermal constants ratios are shown in Fig. 3.19. After this, we proceed to compute the average ratio $(\rho C_P)/\kappa$, which is 0.0053. In this case study, the updated non-spurious CC coefficients are shown in red in Fig. 3.17.

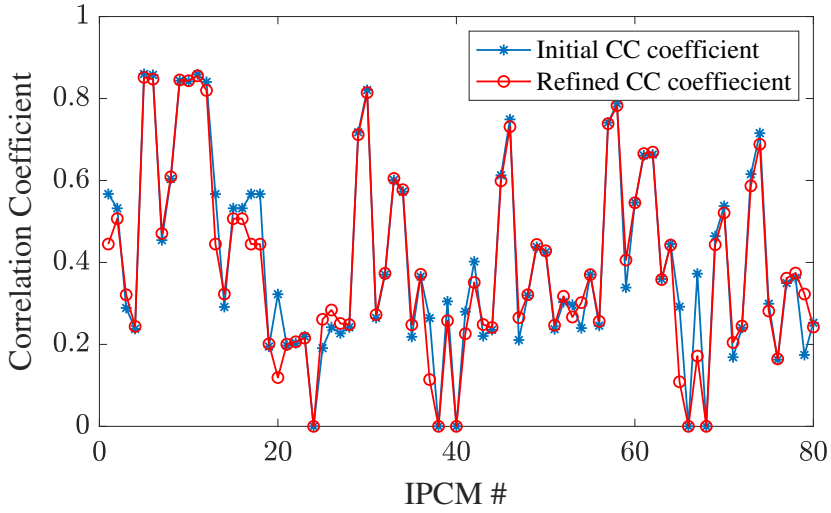


Figure 3.17: CC coefficient between heat-source #1 and 80 IPCM metrics. Blue dot-marked trace includes the spurious CC coefficient, while the red circle-marked trace is for after refinement

We performed this analysis for all 18 heat-sources which yielded an average ratio $(\rho C_P)/\kappa = 0.0048$. We then used this ratio for (3.10) to identify all the relevant IPCM metrics, which are listed in Table 3.2. In total we were able to identify 47 IPCM metrics that are highly correlated with the heat-sources on the Intel i5-3337U. This is a significant reduction from the original 80 IPCM metrics previously shown in Table 4.1.

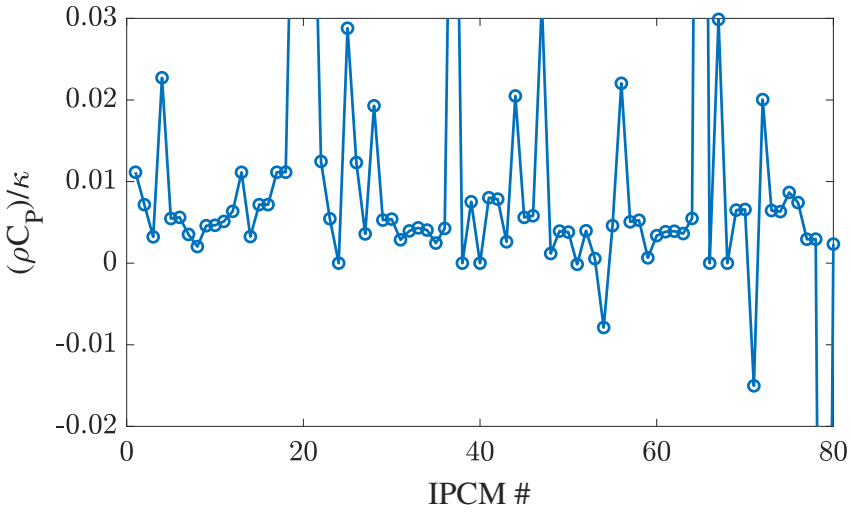


Figure 3.18: Ratio of thermal constants $(\rho C_P)/\kappa$ associated with 80 IPCM metrics

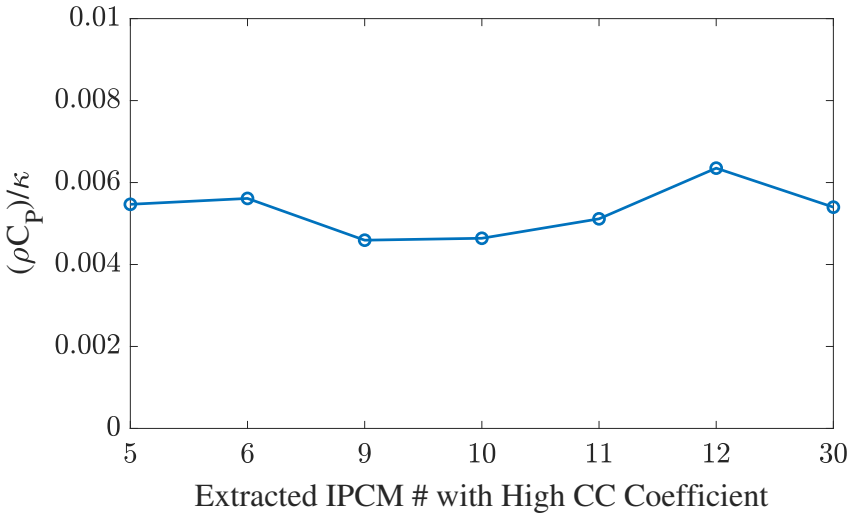


Figure 3.19: (Ratio of thermal constants for relevant IPCM metrics that have high CC coefficients.

3.6 Machine learning based thermal modeling: Network architecture

[2]In network configurations, LayerType$_n${**m**} is a condensed representation describing the structure of

Table 3.2: Reduced Performance metrics (Intel PCM)

Pkg.	Pkg.	Core1.1	Core1.2	Core2.1	Core2.2
exec	write rate	IPC	exec	exec	afreq
IPC	inst count	freq	IPC	IPC	L2 miss
freq	ACYC	L2 miss	freq	freq	
afreq	physIPC	L2 hit	afreq	L3 miss	
L3 miss	physIPC%	L2 MPI	L3 miss	L3 hit	
L2 miss	inst nom	C0res%	L2 miss	L3 MPI	
L3 hit	inst nom%	C7res%	L3 hit	L2 MPI	
L3 MPI	energy (J)	temp	L3 MPI	C0res%	
L2 MPI	temp		L2 MPI	temp	
read rate			C0res%		

Table 3.3: Performance comparisons between various NN configurations

Network Configuration[2]	All IPCM	
	RMS Error (°C)	Inference Time (sec.)
$LSTM_1\{10\}$ + $Dense_1\{18\}$	1.203	0.461×10^{-3}
$LSTM_1\{18\}$ + $Dense_1\{18\}$	0.821	0.482×10^{-3}
$LSTM_1\{30\}$ + $Dense_1\{18\}$	0.785	0.543×10^{-3}
$LSTM_1\{50\}$ + $Dense_1\{18\}$	0.791	0.668×10^{-3}
$LSTM_1\{100\}$ + $Dense_1\{18\}$	0.748	1.082×10^{-3}
$LSTM_2\{100,40\}$ + $Dense_1\{18\}$	0.827	1.748×10^{-3}
$LSTM_3\{100,70,40\}$ + $Dense_1\{18\}$	0.881	2.662×10^{-3}
Network Configuration	Reduced IPCM	
	RMS Error (°C)	Inference Time (sec.)
$LSTM_1\{10\}$ + $Dense_1\{18\}$	0.980	0.436×10^{-3}
$LSTM_1\{18\}$ + $Dense_1\{18\}$	0.927	0.459×10^{-3}
$LSTM_1\{30\}$ + $Dense_1\{18\}$	0.851	0.515×10^{-3}
$LSTM_1\{50\}$ + $Dense_1\{18\}$	0.679	0.581×10^{-3}
$LSTM_1\{100\}$ + $Dense_1\{18\}$	0.684	1.047×10^{-3}
$LSTM_2\{100,40\}$ + $Dense_1\{18\}$	0.717	1.687×10^{-3}
$LSTM_3\{100,70,40\}$ + $Dense_1\{18\}$	0.699	2.544×10^{-3}

the NN. Here LayerType refers to the type of hidden layer (i.e. LSTM or Dense), subscript n refers to the number of the aforementioned layers in the network, and the $1 \times n$ vector $\mathbf{m}$ refers to the number of nodes in each of the respective layers (i.e. $LSTM_3\{100,70,40\}$ + $Dense_1\{18\}$ refers to a network with 3 LSTM

Since online temperature estimation of a microprocessor is very much a time-series problem, we need a method that is naturally suited for modeling such a system. One option would be to use a statistical analysis based approach such as autoregressive-moving-average (ARMA) or even simple least-squares regression models. The proposed heat-source identification, data acquisition and preparation methods discussed previously can also be applied to fit these models as well. However, in this work we will instead be utilizing deep learning as the recent advancements in this area have shown promising results in time-series estimation and pattern recognition tasks [92, 93]. Recurrent-neural-networks (RNN) are the classical architecture designed for such tasks. In this work, we will utilize a specialized subset of RNNs, called Long-Short-Term-Memory (LSTM) network, which uses gated internal states making it ideal for problems that require substantial temporal resolution. For brevity, we refer the readers to [92] for detailed discussions and analysis of LSTM networks.

While the NN architecture of choice was obvious, there is no standard method of determining the size and depth of the network that is optimal for the problem at hand. In most cases, some experimentation is necessary to determine the smallest network that is robust enough to accurately model the given problem. The network size is especially crucial for online/real-time applications, like that one explored in this study, where the model should be light-weight enough for inference at moderate to high-frequencies with low computational overhead. In this work, our goal is to do nearly real time temperature estimation. Meaning, we want to estimate the temperature at time t by time $t + t_{inference}$, where $t_{inference}$ is the time taken for each inference. Hence, in order to be as close to real-time

layers, with 100, 70, and 40 nodes per layer respectively, and 1 dense layer with 18 nodes).

as possible, it is imperative to reduce $t_{inference}$ as much as possible without deteriorating accuracy. The IPCM reduction method from the previous section aids in reducing $t_{inference}$ by reducing the input dimensionality of the model, however, another factor that affects $t_{inference}$ is the network architecture itself. To this end, in this section we explore various network depths and layer sizes to determine the optimal configuration. Table 3.3 shows the estimation error and inference times for various networks. Each network was trained twice, first with all 80 IPCM metrics as inputs, and a second time with only the thermally-relevant 47 IPCM metrics as inputs. The results for both cases are shown in Table 3.3.

From the analysis in Table 3.3, it is clear that as the network size grows, so does the inference time and consequently the performance overhead of the model. The thermal time-constant (τ) for semi-conductor devices is typically in the order of 10^{-3}sec. (milliseconds), hence it is important to ensure that the inference time of the model is equal to or less than τ. To be conservative, we will aim for an inference time less than 1 millisecond (ms). In addition to minimizing inference time, the model must also yield usable accuracy. The analysis in Table 3.3 shows that the model with the fastest inference time also has the worst accuracy, hence there must be a trade-off between the two. However, we found that accuracy does not continue to improve as the network size grows; after a certain threshold, accuracy saturates or, in some cases, even declines. This is due to the fact that larger networks are generally more difficult to train, often requiring meticulous tuning. While such tuning can yield higher accuracy, a larger network performing worse than a smaller one is usually a sign that the network has grown larger than necessary for modeling the given problem.

Hence, based on the aforementioned observations, we chose to proceed with $\text{LSTM}_1\{50\}$

+ $\text{Dense}_1\{18\}$, which has 1 LSTM layer with 50 nodes and 1 dense layer with 18 nodes as shown in Fig. 4.5. This network offers a good trade-off between accuracy and inference time. When all 80 IPCM metrics are used as inputs, the network yields an overall RMS error of 0.79°C and an inference time of 0.67ms. Utilizing the input reduction method discussed in Sec. 3.5, the same network should maintain its accuracy while yielding a faster inference time. This is indeed the case, when trained with only the thermally-relevant 47 IPCM metrics as inputs, the network performs even better with an overall RMS error of 0.68°C and an inference time of 0.58ms. Here, as the input dimensionality is decreased, the inference time also decreases. Additionally, accuracy of the model improves as well since the IPCM metrics with little or no correlation with the thermal response of the chip are removed, resulting in less noise at the input. Further details of the experimental setup and performance analysis will be presented in the next section.

3.7 Experimental results and discussions

In this section, we present the experimental results from the proposed system-level thermal modeling approach. The proposed method has been implemented on two Intel processors: the Intel Core i5-3337U with 2-cores / 4-threads, which is the chip that was used in all of the previous discussions, and the Intel Core i7-8650U with 4-cores / 8-threads. It should be noted that, in this work, only one i5-3337U and one i7-8650U chip was used to collect all of the training data-set. However, it is generally recommended to use multiple sample chips of the same type to gather the data-sets in order to account for the statistical variations between the chips. Additionally, it should be noted that the proposed heat-

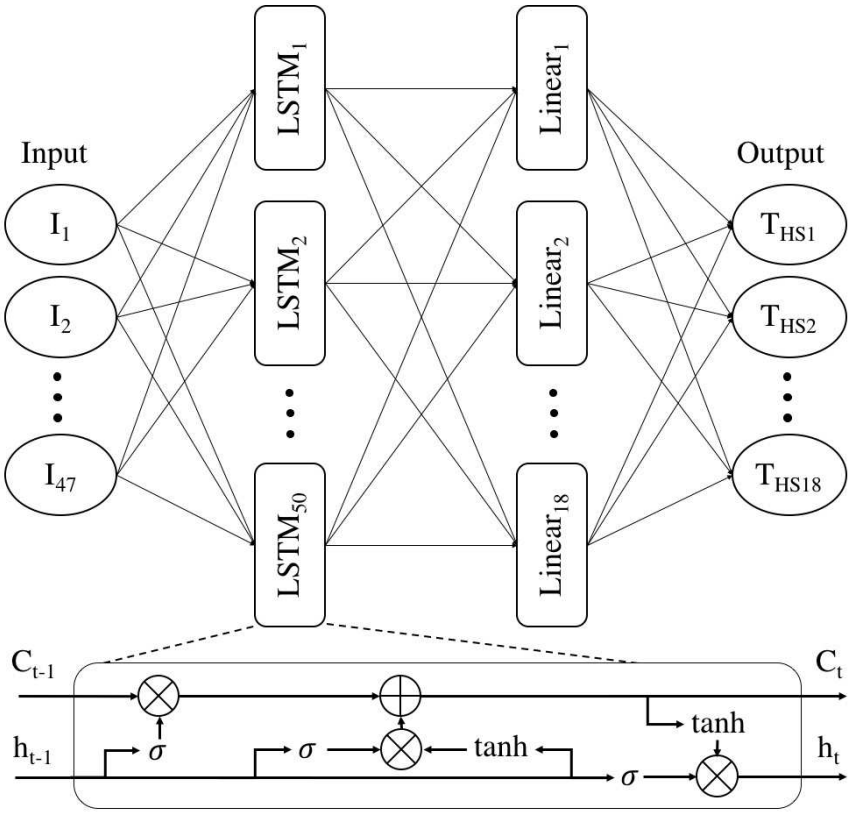


Figure 3.20: LSTM network architecture

source identification method and the proposed data-collection and preparation methods can be used to fit any regression-based model, not just train a neural-network based machine learning model. To this end, the same datasets used to train the LSTM network were used to fit a simple least-squares linear regression model so comparisons can be made between the estimation accuracy and inference times of the two approaches.

3.7.1 Results from Intel Core i5-3337U

From the Core i5-3337U, a total of 187200 data points were collected which, considering the capture rate of 60Hz, constitutes to 52 minutes of continuous runtime. Each data

point consists of 80 IPCM metrics captured internally on the test system, and the temperatures of the previously identified 18 heat-sources captured via the thermal imaging setup. During this time, the processor was subjected to a variety of realistic workloads. These range from lightweight workloads like idling, and word processing, to intensive workloads like data compression. Some workloads were primarily compute-intensive tasks while others were memory-intensive. The idea is to utilize all the different sub-systems in the processor during the course of data acquisition so that their thermal response can be recorded. The amount of data that is required for the proposed approach will depend on the processor and its application. As a rule of thumb, with any machine learning based approach, more data will generally always lead to a better model. Additionally, it is important to ensure that the workloads used during data acquisition are as diverse as possible. One option to ensure diversity is to use a benchmark suite, such a Phoronix test suite, that offers workloads that vary in hardware utilization and intensity [94].

Once all the data is acquired, the method discussed in Sec. 3.5 was used to select the IPCM metrics that are highly correlated with the previously identified heat-sources. Only using these metrics to train the model allowed us to reduce the input dimensionality of the model from 80 to 47. After this step, the network shown in Fig. 4.5, was trained for a total of 50 epochs with 60 time steps used for the LSTM layers. Out of the 187200 datapoints collected for this study, the first 60% were used for training, the next 15% were used for validation, and the remaining 25% were used for testing. Similarly, for the least-squares regression model, the first 75% of the data-set was used for fitting and the remaining 25% was used for testing.

Formal testing and validation carried out on the final LSTM model shows that it performs exceptionally well. The results presented in Fig. 3.21 shows the model estimating the runtime temperature of heat-source #1 for a duration of about 8 minutes. The measured temperature from the IR camera is overlaid on top of the estimation for comparison. For brevity we only show this plot for one heat-source, however the root-mean-square-error (RMSE) computed for all 18 heat-sources is presented in Table. 3.4. In summary, the highest RMSE was 0.76°C (HS #10) while the lowest was 0.55°C (HS #2). Considering the observed dynamic range of 58.73°C to 101.35°C, these constitute to a relative RMSE of 1.79% and 1.28% respectively. The same tests run on the least-squares linear regression based model shows that this approach produces lower estimation accuracy with the highest RMSE of 3.09°C (HS #6) and the lowest RMSE of 1.99°C (HS #4). These constitute to a relative RMSE of 7.25% and 4.67% respectively. RMSE computed for all 18 heat-sources is shown in Table. 3.4. While accuracy suffers with the simple regression model, it does have the advantage of extremely fast fitting time and inference time. While the LSTM model took more than a day to train, the least-squares model took a few seconds to fit after the data had been prepared. Similarly, while the inference time of the LSTM model is in the order of milliseconds ($\sim 0.58ms$), as discussed in Sec. 3.6, the inference time of the regression model is in the order of microseconds ($\sim 0.62\mu s$). Hence, there is a trade-off between the two black-box modeling approaches.

3.7.2 Results from Intel Core i7-8650U

To further validate the proposed approach, it was also implemented on the Intel Core i7-8650U. Using the heat-source identification method presented in Sec. 3.3 a total

of 20 heat-sources were detected on the Intel i7-8650U. Then 288000 time-steps of runtime temperature data of the 20 heat-sources were recorded along with synchronized runtime IPCM data. In total, IPCM provides 170 metrics for the Intel i7-8650U. With the method discussed in Sec. 3.5, 72 thermally-relevant metrics were selected to be used as inputs to the model. With this reduction, the final model will have an input dimensionality of 72 (for the 72 IPCM metrics) and an output dimensionality of 20 (for the temperatures of the 20 heat-sources). With this in mind, the analysis presented in Sec. 3.6 was then preformed to find a network that offers a reasonable trade-off between accuracy and inference time. The network that was selected is very similar to the one shown in Fig. 4.5 but with 75 LSTM nodes in the first layer and 20 dense nodes in the second layer. Once the network was selected, it was then trained for a total of 50 EPOCHS with 65% of the data used for training, 15% used for validation and the remaining 25% used for testing. The same data was also used to fit a simple least-squares linear regression model, where the first 75% of the data was used to fitting the model and the remaining 25% of the data was used for testing. Using the same training and testing data on two different black-box modeling approaches allows us to compare and contrast the advantages and disadvantages of the two. The results presented in Fig. 3.21 shows the LSTM model estimating the runtime temperature of heat-source #1 for a duration of about 8 minutes. The RMSE for each of the 20 heat-sources are given in Table 3.5 for both the LSTM model and the least-squares regression model.

The results for the Intel i7-8650U are very comparable to what was achieved with the Intel i5-3337U. Here the LSTM model yielded the highest RMSE of 0.93°C (HS#4) and the lowest of 0.62°C (HS#14). Considering the observed dynamic range of 28.9°C to

97.9°C, this constitutes to a relative RMSE of 1.35% and 0.89% respectively. While the least-squares linear regression model yielded the highest RMSE of 3.19°C (HS#2) and the lowest RMSE of 1.83°C (HS#14). These constitute to a relative RMSE of 4.62% and 2.65% respectively.

Table 3.4: Root-Mean-Square-Error for each heat-source (i5-3337U)

Idx	LSTM	LS-Reg	Idx	LSTM	LS-Reg	Idx	LSTM	LS-Reg
HS#1	0.694°C	2.528°C	HS#7	0.584°C	2.031°C	HS#13	0.710°C	2.809°C
HS#2	0.762°C	2.861°C	HS#8	0.615°C	2.383°C	HS#14	0.623°C	2.276°C
HS#3	0.696°C	2.921°C	HS#9	0.686°C	2.312°C	HS#15	0.686°C	2.609°C
HS#4	0.697°C	1.992°C	HS#10	0.731°C	3.013°C	HS#16	0.651°C	2.701°C
HS#5	0.736°C	2.796°C	HS#11	0.547°C	2.036°C	HS#17	0.734°C	3.084°C
HS#6	0.737°C	3.091°C	HS#12	0.678°C	2.244°C	HS#18	0.686°C	2.393°C

Table 3.5: Root-Mean-Square-Error for each heat-source (i7-8650U)

Idx	LSTM	LS-Reg	Idx	LSTM	LS-Reg	Idx	LSTM	LS-Reg
HS#1	0.737°C	1.830°C	HS#8	0.679°C	2.559°C	HS#15	0.710°C	2.002°C
HS#2	0.802°C	3.181°C	HS#9	0.734°C	2.146°C	HS#16	0.804°C	2.045°C
HS#3	0.898°C	2.799°C	HS#10	0.747°C	2.056°C	HS#17	0.694°C	2.480°C
HS#4	0.934°C	2.396°C	HS#11	0.852°C	2.775°C	HS#18	0.783°C	3.198°C
HS#5	0.869°C	2.741°C	HS#12	0.691°C	2.208°C	HS#19	0.717°C	2.365°C
HS#6	0.690°C	1.841°C	HS#13	0.859°C	2.837°C	HS#20	0.815°C	1.865°C
HS#7	0.654°C	2.089°C	HS#14	0.617°C	2.106°C			

3.8 Summary

In this chapter, we have presented a novel method of systematically identifying all prominent heat-sources on commercial processors and deriving a dynamic thermal model

to estimate the temperatures of the identified heat-sources during online use. Unlike many existing studies, this work exclusively utilizes measured data gathered directly from commercial off-the-shelf processors. Additionally, the proposed approach inherently avoids all the major obstacles faced by traditional methods that currently exist in literature, allowing it to be easily deployed by chip manufacturers and third-parties alike. Experimental results on two Intel multi-core CPUs showed that the proposed thermal model achieves very high accuracy (root-mean-square-error: 0.55°C to 0.76°C on the Intel i5-3337U and 0.62°C to 0.93°C on the Intel i7-8650U) in estimating the temperatures of all the identified heat-sources on the two chips. These results make the proposed approach very desirable for dynamic thermal management schemes which now rely heavily on the temperature data from just the on-chip temperature sensors alone. The high spatial resolution yielded by the proposed approach can help greatly in supplementing the temperature data from the on-chip sensors, allowing for the development of more robust and smarter online thermal/power control schemes.

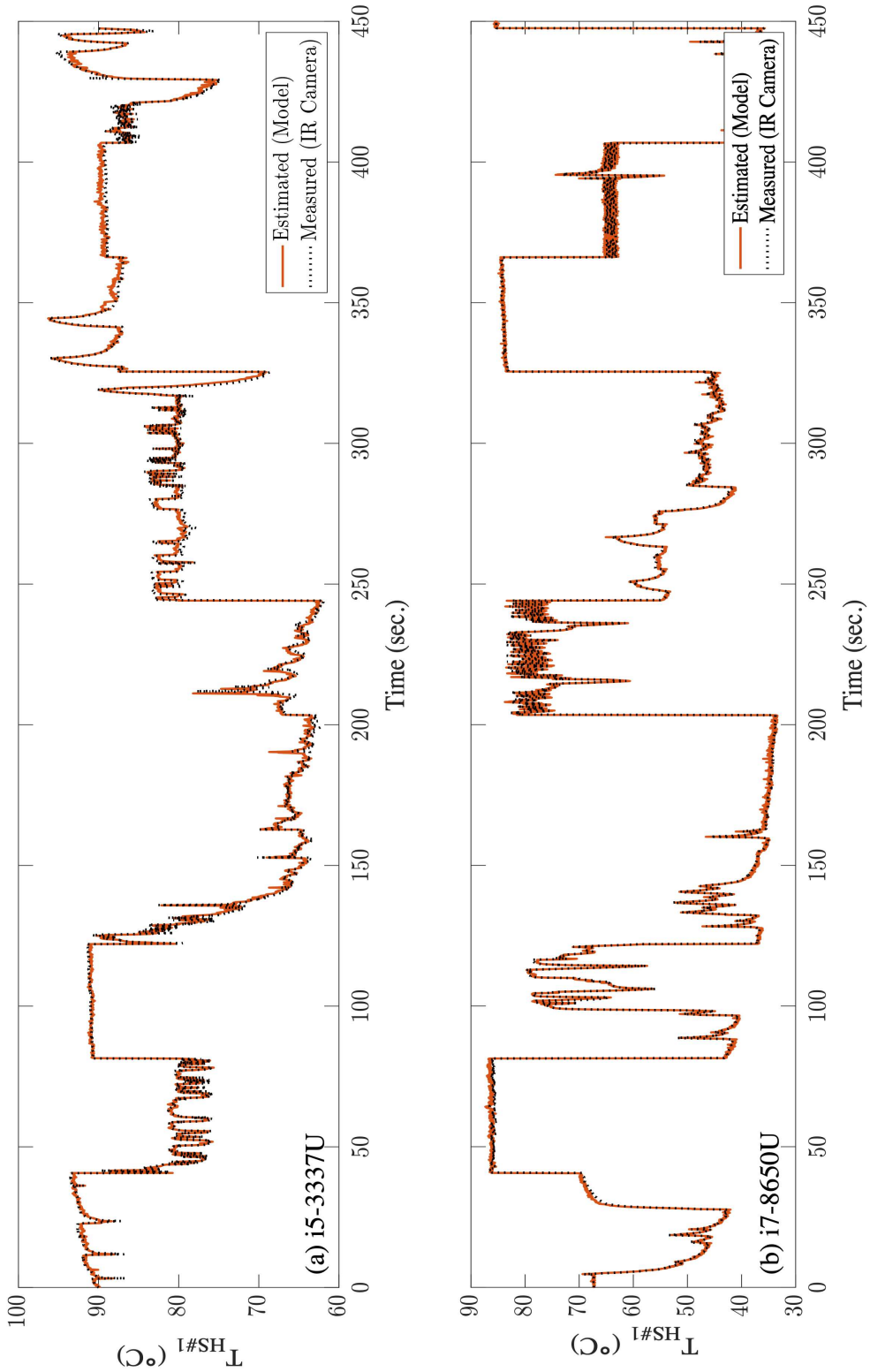


Figure 3.21: Estimated vs measured runtime temperature of heat-source #1 (a) i5-3337U (b)i7-8650U

Chapter 4

Real-Time Full-Chip Thermal Tracking

4.1 Related Work and Motivation

Hardware performance counters are a collection of special-purpose registers that are now present in most, if not all, commercial microprocessors. These registers can be configured to count low-level performance metrics such as the utilization rates of one or more functional units (FU). Due to the correlation between a FU's utilization rate and its power consumption, it has been shown that FU-wise thermal and power models can be built as functions of low-level performance metrics. Exploiting this correlation, low-level performance metrics and temperature readings from the embedded temperature sensors have been used in the past to predict the future readings from the embedded sensors (i.e. at time t predict $Tsens_{t+1}$) [77]. Predicting the future temperature aids in the development of

proactive thermal control schemes as opposed to the existing reactive ones [76,78]. However, in this study, we attempt to solve a different problem. As previously mentioned, the number of embedded sensors on the chip is very limited due to their high design overheads and they may not always be placed in close proximity to the hot-spots on the chip (Fig. 1.1). Hence it is imperative to develop methods to monitor the temperature and/or power distributions across the entire chip's surface-area in real-time.

To that end, software-based power-consumption and temperature estimation methods for both high performance and mobile/embedded processors have been developed [73–75, 86–88]. These methods offer a software-based solution to runtime FU-wise, or package-wise, power and temperature estimation which otherwise would require a vast number of embedded sensors that incur significant design overheads and are prone to sensing and process-based noise [76]. However, these methods typically rely on manually identifying the low-level performance metrics that are correlated with each FU on the chip. Additionally, for FU-wise estimation, at least one low-level performance metric must be recorded for each FU at all times. However, the number of configurable performance counting registers available in a typical microprocessor is limited, hence simultaneous monitoring of the entire chip is not feasible using this method.

To overcome these challenges, two general strategies have been explored. The first is to estimate the full-chip heatmaps from physics-based thermal models and power related information [31,32]. These thermal models can be built using the so-called "bottom-up" approaches such as HotSpot [30] based simplified finite difference methods, finite element methods [33], equivalent thermal RC networks [34], and recently proposed behavioral

thermal models based on the matrix pencil method [35] and the subspace identification method [36, 37]. However, such methods are typically not suited for real-time use and often require accurate component wise power-traces as inputs, which are not trivial to obtain [38, 39]. Second is to use an interpolation based approach to estimate the full-chip heatmaps from the embedded sensor readings [79]. Since the number of sensors and their placement have a significant impact on the accuracy of the aforementioned interpolation, smart sensor placement algorithms have also been proposed that can be used during design time to find the optimal placement for the given budget of embedded temperature sensors [4, 17, 18, 95, 96]. It has been shown that adapting smart sensor placement algorithms can improve the accuracy of soft-sensing or interpolation based methods that can be used to estimate the temperature of any arbitrary location on the chip [4, 18].

However, the aforementioned methods either require design-time hardware changes (inserting or relocating sensors) or at the very least require detailed knowledge of the chip's floorplan and constants specific to the technology-node which are not disclosed by the original chip manufacturer. An exclusively *post-silicon* approach to real-time estimation of the spatial temperature distribution across the entire chip area (i.e. at time t, estimate the full-chip spatial heatmap $T(x, y)_t$) remains a challenge for existing commercial microprocessors. Such an approach would aid the original chip manufacturer, as well as third-parties, such as system integrators and academic research labs, in developing more robust thermal, power, and reliability control schemes that can make use of both the real-time temperature data sensed by the existing embedded sensors, as well as the real-time estimation from the thermal model.

On the other hand, recently, machine-learning is gaining much attention due to the breakthrough performance in various cognitive applications such as visual object recognition, object detection, speech recognition, natural language understanding, etc., due to dramatic accuracy improvements in their time-series or sequential modeling capabilities [93]. Machine-learning for electronic design automation (EDA) is also gaining significant traction as it provides new computing and optimization paradigms for many of the challenging design automation problems that are complex in nature. For instance, machine learning methods have been applied to power modeling [97] and design space exploration [98]. Additionally, machine-learning based schemes have recently been explored to build a workload-dependent thermal prediction model [99], where the future steady-state temperature of the chip can be predicted by application characteristics and physical features.

Inspired by the recent breakthrough in deep-learning, in this work we present a machine-learning based framework to *post-silicon* full-chip heatmap estimation for commercial off-the-shelf microprocessors. The proposed method leverages the existing embedded temperature sensors and high-level performance monitors which provide system-level metrics such as core-wise frequency, instruction counts, cache hit/miss-rates, overall energy consumption, etc., providing a comprehensive view of the utilization of the entire microprocessor in real time. Deep learning is used to ascertain the relationship between the system-level utilization behavior of the microprocessor and its thermal behavior. The proposed data-driven modeling strategy is structured such that it can be applied to most, if not all, existing commercial multi-core microprocessors with no knowledge of the proprietary design/floorplan information.

4.2 Model Optimization

In *RealMaps*, the thermal model will be built via training a LSTM-based deep-neural-network (DNN). The DNN will be trained offline using the full-chip heatmaps and high-level performance metrics acquired using our IR thermography setup. In machine-learning terminology, the heatmaps will be our labels (output of the model) while the performance metrics are our features (input of the model). Once the model is trained, the goal is to deploy the model back into the processor in the form of a background application residing in the operating system (OS). This application will, in real-time, feed the performance metrics from the online performance monitor into the model; the model will in-turn periodically output the estimated heatmaps. Since the model is meant for real-time use, it is imperative for it to be as lightweight as possible with minimal overheads in processing time and memory usage.

From the DNN perspective, several techniques, such as weight pruning and quantization, exist for optimizing DNN models. However, in this work we will be utilizing Google's Tensorflow (TF) machine-learning library [100] to configure and train our model. The aforementioned weight pruning and quantization methods, along with a number of other optimizations are already performed by the aforementioned library (TF Lite); hence, from this perspective, there's limited margin to optimize our core model any further. We can however optimize the model from the input and output points of view. To this end, in this section we will discuss our approach to output and input dimensionality reduction via dominant spatial frequency extraction and simple cross-correlation analysis.

4.2.1 Heatmap compression

Heatmaps of the Intel i5-3337U and i7-8650U captured using our IR thermography setup have image resolutions of $177 \times 166px$ and $185 \times 154px$ respectively. This constitutes to a total pixel count of 29382 and 28490 respectively for the two chips. If we were to build DNN models capable of pixel-wise estimation of the full-chip heatmaps (image generation task), then the models for the two chips will need to have output sizes of 29382 and 28490 dimensions respectively. This would not only make the DNN models very large, thus unfit for online inference, but will likely be untrainable due to the large number of trainable parameters that the network will contain. One solution to this problem is to build a model to only estimate the dominant features of the heatmaps rather than the entire heatmap image. The estimated heatmaps can then be reconstructed using the dominant features outputted by the model.

Generally, feature extraction can be carried out using popular dimensionality reduction techniques such as principal component analysis (PCA). Here, compression or approximation is achieved by projecting a data-sample, heatmap $T(x, y)$, onto the subspace spanned by the dominant principal components (PCs) of the available dataset. Such a method would involve first calculating the PCs which form the columns of the change-of-basis matrix. The dominant features in this case would be the coefficients of expansion corresponding to the dominant PCs. However, using this non-standard basis for the application at hand is not recommended since this would require the dominant PCs to be stored in memory. Each PC has the dimensionality equivalent to the pixel-count of the heatmap image (29382 and 28490 single precision floating point values for the two chips respectively).

Hence, just storing a few PCs will incur a significant amount of memory overhead. An alternative would be to use a basis with established analytical transformation equations, so that the basis vectors do not need to be stored in memory.

In the case of spatial heatmaps, it has been shown that discrete cosine transformation (DCT) to the spatial frequency domain is an excellent option as the majority of the information from a heatmap can be expressed with just a few low-frequency coefficients of DCT [95]. To this end, we use 2D DCT to convert the measured heatmaps, $T(x, y)$, into spatial frequency-domain [101]. This allows us to extract the dominant low-frequency DCT coefficients of the heatmaps and train our DNN to only estimate these coefficients. Inverse DCT can then be performed at the model's output to recover the estimated heatmaps.

$$\begin{bmatrix} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & x_{max} \times y_{max} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 21 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 15 & 20 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 10 & 14 & 19 & \cdots & \cdots & \cdots & \cdots & \cdots \\ 6 & 9 & 13 & 18 & \cdots & \cdots & \cdots & \cdots \\ 3 & 5 & 8 & 12 & 17 & \cdots & \cdots & \cdots \\ 1 & 2 & 4 & 7 & 11 & 16 & \cdots & \cdots \end{bmatrix}$$

Figure 4.1: Order of frequency dominance in $F(x, y)$

2D DCT is a popular choice for signal and image processing with its "strong energy compaction property" [101]. In most applications, the bulk of the information can be represented by a few low-frequency components of the DCT. A 2D DCT consists of two separate 1D DCT operations, which can be denoted as

$$f_k = \frac{a_0}{\sqrt{N}} + \sqrt{\frac{2}{N}} \sum_{i=1}^{N-1} a_i \cos \frac{(2i+1)k\pi}{2N}, \; 0 \leq k < N, \tag{4.1}$$

where vector $\{a_i\}$ is the original $(1 \times N)$ data, and $\{f_k\}$ is the result of 1D DCT. A 2D DCT is completed by applying 1D DCT on each column and then on each row of the matrix.

For feature extraction, $1 \leq n \leq x_{max} \times y_{max}$ number of dominant DCT frequencies can be extracted from a spatial heatmap $(T(x, y))$ by transforming $T(x, y)$ into its spatial frequency domain representation $(F(x, y))$ using 2D DCT; then retaining only the first n dominant coefficients in $F(x, y)$. The order of dominance in $F(x, y)$ is shown in Fig. 4.1 where index labeled 1 and $x_{max} \times y_{max}$ represent the most and least dominant DCT frequencies respectively.

In an image compression scenario, a compressed frequency map $(\mathfrak{F}(x, y))$ is obtained by applying a mask to $F(x, y)$

$$\mathfrak{F}(x, y) = F(x, y)m(x, y), \tag{4.2}$$

where $m(x, y)$ is a mask map valued 1 at the n most dominant DCT frequency locations and 0 everywhere else. The compressed heatmap $(\mathcal{T}(x, y))$, can then be recovered by carrying out 2D inverse DCT (iDCT) on $\mathfrak{F}(x, y)$. Similar to its forward counterpart (4.1), 2D iDCT consists of two separate 1D iDCT steps (4.3) on the rows and columns respectively.

$$a_i = \frac{f_0}{\sqrt{N}} + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} f_k \cos \frac{(2i+1)k\pi}{2N}, 0 \leq i < N. \tag{4.3}$$

In this case, the higher the value of n, the more $\mathcal{T}(x, y)$ will resemble $T(x, y)$. For example, Fig. 4.2(a) shows a random heatmap of an Intel i7-8650U. The heatmap compressed using $n = 1$ spatial DCT frequencies is shown in Fig. 4.2(b). Since only 1 spatial frequency is used, the compressed heatmap only retains the approximate amplitude and no details of the spatial temperature distribution. As n in increased (Fig. 4.2(c) -

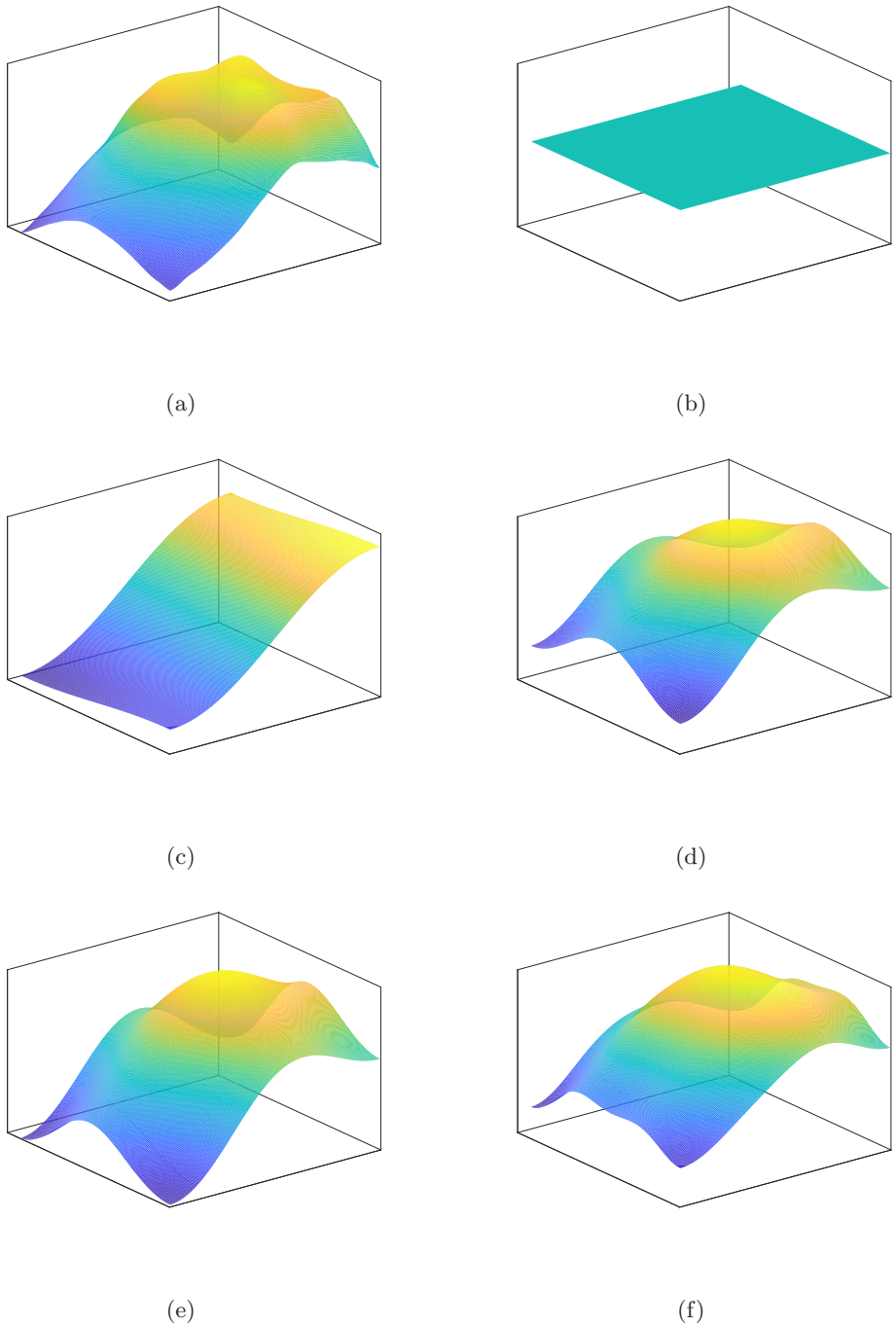


Figure 4.2: (a) A randomly selected uncompressed heatmap of an Intel i7-8650U. (b) Compressed ($n = 1$). (c) Compressed ($n = 3$). (d) Compressed ($n = 6$). (e) Compressed ($n = 10$). (f) Compressed ($n = 15$).

Fig. 4.2(f)), more and more nuanced spatial details are retained in the compression. For a heatmap of size $x_{max} \times y_{max}$, if $n = x_{max} \times y_{max}$, then $\mathcal{T}(x, y) = T(x, y)$.

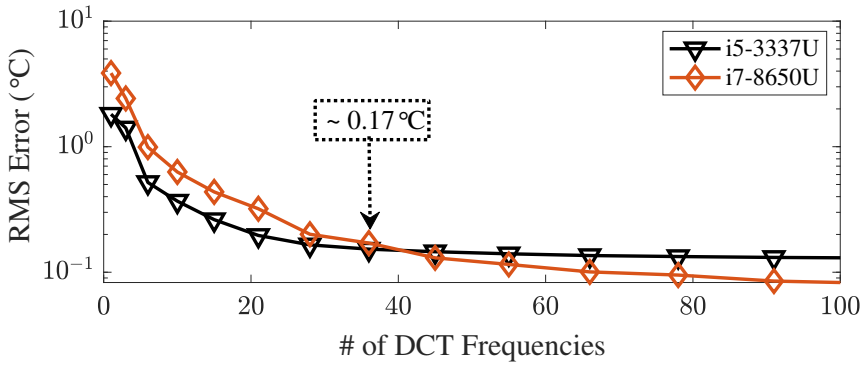


Figure 4.3: RMS error between actual heatmaps and heatmaps compressed using varying number of DCT coefficients

In this work, our goal is to identify the minimum value for n that results in minimal loss of spatial information from this compression. In order to determine the minimum number of DCT coefficients that we can use without introducing a significant amount of error, we compress 140,000 heatmaps from each of the two processors with varying number of DCT coefficients. Root-mean-square (RMS) error is then computed between the compressed heatmaps and their uncompressed counterparts. Fig. 4.3 shows the RMS error in °C as the number of DCT coefficients used in the compression is increased. Based on Fig. 4.3, we can see that it is sufficient to use only the first 36 most-dominant DCT coefficients, as increasing the number of features further produces marginal benefits. *With this compression, the output of the two models* ($\mathcal{F} = \textbf{vectorize}(\textbf{nonZero}(\mathfrak{F}))$ *only need the dimensionality of 36, instead of 29382 and 28490 respectively. This is a significant reduction to the size of the model, with a compression error of only* $0.17°C$ *RMSE as shown in Fig. 4.3.*

While we found that 36 DCT coefficients is sufficient for the two chips used in this

study, this number may not be true in general for all processor models. Some processors may require fewer, while other may require more. Hence, it is imperative to repeat the above analysis for each processor model to derive the minimum value for n that results in negligible loss due to the compression.

4.2.2 Performance metrics selection

As previously mentioned, two Intel chips are used in this study. Namely, the Intel i5-3337U (2 cores, 4 threads, released in 2012) representing an older generation and the Intel i7-8650U (4 cores, 8 threads, released in 2017) representing a relatively newer generation of chips from the Intel core family of microprocessors. The primary high-level performance monitoring software supported by Intel is Intel's Performance Counting Monitor (IPCM) [85]. IPCM provides the system-level utilization metrics that we will be utilizing in this work. For non-Intel chips, the equivalent performance monitors can be used (i.e. AMD uProf [102]).

IPCM provides package and core-wise performance metrics such as energy usage, package and core frequency, instruction counts, cache hit/miss-rates, etc., as well as the sensed temperature from the embedded sensors. Table 4.1 shows the complete list of IPCM performance metrics from both the package and core-wise domains. There are 30 metrics corresponding to the whole package domain, and 28 metrics for each core. In total, IPCM provides 86 metrics for the dual-core i5-3337U and 142 metrics for the quad-core i7-8650U.

If we were to use the entire IPCM suite as the input to our models, they would have input dimensionalities of 86 and 142 respectively for the two chips. However, although the entire suite of metrics may be useful from a performance monitoring perspective, not all

Table 4.1: High-level Performance Metrics (Intel PCM)

Package			Core		
Exec	Read	C1res%	$Exec_1$	$L2Miss_1$	$c0res\%_1$
IPC	Write	C2res%	$Exec_2$	$L2Miss_2$	$c0res\%_2$
Freq	INST	C3res%	IPC_1	$L3Hit_1$	$c1res\%_1$
AFreq	ACYC	C6res%	IPC_2	$L3Hit_2$	$c1res\%_2$
L3Miss	Time	C7res%	$Freq_1$	$L2Hit_1$	C3res%
L2Miss	PhysIPC	C8res%	$Freq_2$	$L2Hit_2$	C6res%
L3Hit	PhysIPC%	C9res%	$Afreq_1$	$L3MPI_1$	C7res%
L2Hit	INSTnom	C10res%	$Afreq_2$	$L3MPi_2$	T_{sens}
L3MPI	INSTnom%	Energy(J)	$L3Miss_1$	$L2MPI_1$	
L2MPI	C0res%	$_{sens}$	$L3Miss_2$	$L2MPI_2$	

The subscript 1 and 2 (i.e. $Exec_1$ and $Exec_2$) correspond to hardware threads 1 and 2 within a single core.

of the metrics may be relevant in modeling the temperature of the chip. Hence, if we can identify the metrics that are irrelevant to the application at hand, eliminating them from the input will aid in further reducing the size of the model. Removing the irrelevant IPCM metrics can be done simply by computing the Pearson's correlation coefficient between a given IPCM metric and each one of the 36 DCT frequencies discussed in Sec. 4.2.1. If a metric is not correlated with any of the dominant frequencies then it can be eliminated from the input (Algorithm 1). *This trivial approach allows us to reduce the input size from 86 to 58 for the i5-3337U and from 142 to 86 for the i7-8650U.* While this reduction is not as significant as the output reduction in Sec. 4.2.1, it will nonetheless contribute to the efficiency of the model. Additionally, only utilizing relevant features at the input will make the DNN easier and faster to train.

Algorithm 1 Relevant IPCM Metric Selection

Note: n = # of DCT Freq, m = # of IPCM metrics, $tmax$ = # of timesteps

Input: $\mathbb{F} = [\mathcal{F}_0; ..; \mathcal{F}_{tmax}]$; $\mathbb{M} = [allM_0; ..; allM_{tmax}]$

Output: M

1: $M = [\,]$

2: **for** $i \leftarrow 1$ to m **do**

3: $\quad maxC = 0$

4: $\quad$ **for** $j \leftarrow 1$ to n **do**

5: $\quad\quad currC =$ **PearCorr**$(\mathbb{M}[:, i], \mathbb{F}[:, j])$

6: $\quad\quad maxC =$ **max**$(currC, maxC)$

7: $\quad\quad$ **if** $maxC \geq 0.5$ **then**

8: $\quad\quad\quad M =$ **concatenate**$(M, \mathbb{M}[:, i])$

4.3 Framework and Implementation

In this section, we will present the framework and implementation specifics of *RealMaps*. We will detail the process of acquiring the necessary data, training and validating the DNN model, and deploying the end model for real-time inference.

4.3.1 Data acquisition and normalization

The data acquisition for the proposed approach involves simultaneously recording spatial heatmaps of the processor and performance metrics at a constant capture rate ($f = 1/\triangle t = 60Hz$). At time t, one complete heatmap matrix $T(x, y)_t$ of size $x_{max} \times y_{max}$ is captured, while at the same time IPCM vector $allM_t$ of size $1 \times m$ is recorded. Where

$x_{max} \times y_{max} = 177 \times 166$ and $m = 86$ for the i5-3337U and, $x_{max} \times y_{max} = 185 \times 154$ and $m = 142$ for the i7-8650U. The 36 most dominant DCT frequencies (vector $\mathcal{F}_t$), previously identified using the method presented in Sec. 4.2.1, are extracted from $T(x, y)_t$, while at the same time the relevant IPCM metrics (vector M_t), previously identified using the method presented in Sec. 4.2.2, are extracted from $allM_t$. Both $\mathcal{F}_t$ and M_t are then normalized to the range of -1 to 1 and saved. After a period of time ($\triangle t$), the next time-step of data is captured in the same manner. This process is repeated until the desired amount of data is acquired. To summarize, the proposed data acquisition flow is illustrated in Fig. 4.4.

For this study, a total of $t_{max} = 149760$ and $t_{max} = 230400$ time-steps of training data were collected for the i5-3337U and i7-8650U respectively at a capture rate of $60Hz$. This constitutes to a total runtime of 41.6 and 64 minutes respectively. During the initial data collection phase, it is difficult to judge exactly how much data will be needed. However, after the training process, if the model does not produce the desired accuracy then one course of action would be to collect additional data to train the model further. As a rule of thumb, with any machine learning based method, more data will generally lead to a better model. It is however important to denote that increasing the training dataset without increasing the validation dataset can heighten the risk of overfitting. This will result in the model performing well on the training dataset but poorly on the validation dataset and consequently in testing and deployment. In our study, we use 80% of the acquired data for training and 20% for validation. With sufficient validation data, it will be easier to detect and mitigate overfitting during the training process. As we will discuss in the next subsection, this is done by monitoring the learning curve during the training process.

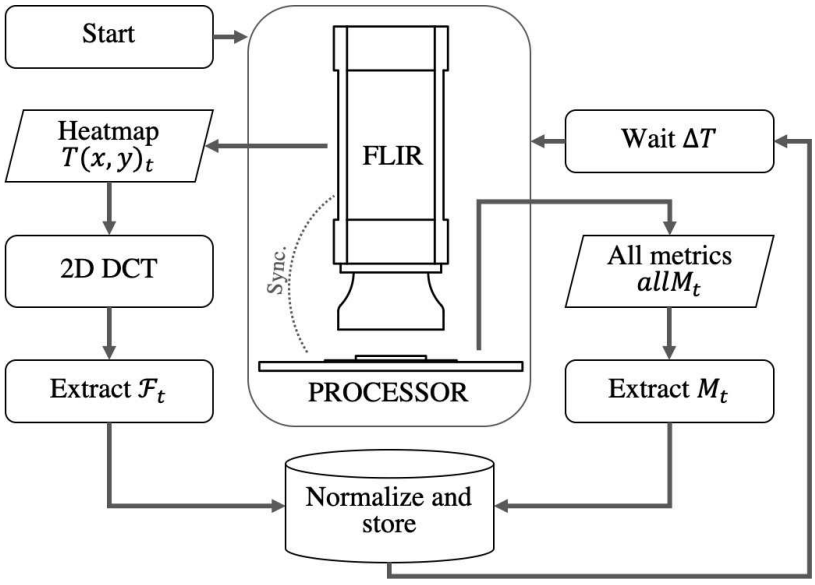


Figure 4.4: Data Acquisition Flow

During the course of data acquisition, the processors were subjected to a variety of workloads. These range from lightweight workloads like idling, to intensive workloads like data compression. Some workloads were primarily compute-intensive tasks while others were memory-intensive. It is important to note that the model itself will be workload independent as it only uses the performance metrics (Sec. 4.2.2) as the inputs, which contain no information of the workload that the processor is executing. The end goal of the proposed approach is to derive a model that performs reliability regardless of the workload that the processors will be subject to during deployment. However, when the training dataset is being collected, diversity in workloads is nonetheless important due to the extensive use of clock and power gating in modern processor architectures. If a FU is not used when the training data is being collected, it is likely that this FU will remain disabled during the entire time. Hence, it is crucial to utilize all the different sub-systems in the processor

Table 4.2: Phoronix Workloads Executed During Data Acquisition

Processor	System	Memory	Disk
aobench	cyclictest	stream	aio-stress
compress-7zip	phpbench	tinymembench	fio
encode-flac	gimp	t-test1	fs-mark
build-gcc	git	ramspeed	dbench
cachebench	blender	mbw	tiobench

during the course of data acquisition so that their thermal behavior can be recorded and consequently be "learned" by the model during the training process. One option to ensure diversity of workloads is to use a benchmark suite that offers a variety of workloads that range in hardware utilization and intensity [94, 103, 104].

For this study, we used the Phoronix test suite [94], which is an open-source benchmark software for Linux that offers an extensive range of workloads. In Phoronix, the workloads are categorized under four domains: processor, system, memory and disk. The processor and system workloads tend to be more compute intensive, while memory and disk workloads tend to be memory intensive. For this study, 5 workloads from each category were randomly selected as shown in Table 4.2. These workloads were randomly executed in the processor during the course of the data acquisition (Fig. 4.4).

4.3.2 Training and testing the LSTM model

As previously mentioned, in this study, we will be employing a LSTM-based DNN to train our online thermal model. A LSTM network is an improved variant of RNNs whose nodes (or neurons) have gated internal states, allowing the network to model problems that require substantial temporal dimensionality. Such a network is ideal for the problem

at hand, since the current temperature of a microprocessor (heatmap $T(x,y)_t$) is not just a function of its current utilization (vector M_t), but rather its recent utilization (matrix $[M_{t-s};..;M_t]$). In this work we will set $s = 59$, making our estimated heatmap ($\mathcal{T}(x,y)_t$) a function of 60 time-steps of utilization data ($[M_{t-59};..;M_t]$). With the acquisition rate of $60Hz$, 60 time-steps of M represents the processor's utilization for a time-span of 1 second. Given that the thermal time-constant for semiconductor chips is in the order of milliseconds, 1 second of temporal dimensionality at the input should be sufficient. Setting s to a substantially large number will make the model more difficult to train while yielding minimal improvements in accuracy. This is because, during the training process, the weights assigned to the inputs spanning significantly far back in time will be set very close to 0 as their contribution to the current output of the model (current temperature) is minimal.

The specific configuration of the LSTM network (# of nodes and # of layers) is not an exact science, especially for a regression problem. Generally, it is recommended to start with a smaller network and increase the size based on its performance. In this work, we will be utilizing the network illustrated in Fig. 4.5. This is a two layer network with k nodes and 60 time-steps of feedback in the LSTM layer, and 36 nodes in the linear output layer. Through experimentation, we determined that $k = 58$ and $k = 86$ (matching $\mathbf{size}(M)$) yielded a good trade-off between network size and inference time for the i5-3337U and i7-8650U respectively.

After the training data was acquired using the method outlined in Sec. 4.3.1, the aforementioned LSTM network was trained for a total of 150 EPOCHS with 80% of the data used for training and 20% used for validation. Training was carried out on a server with two

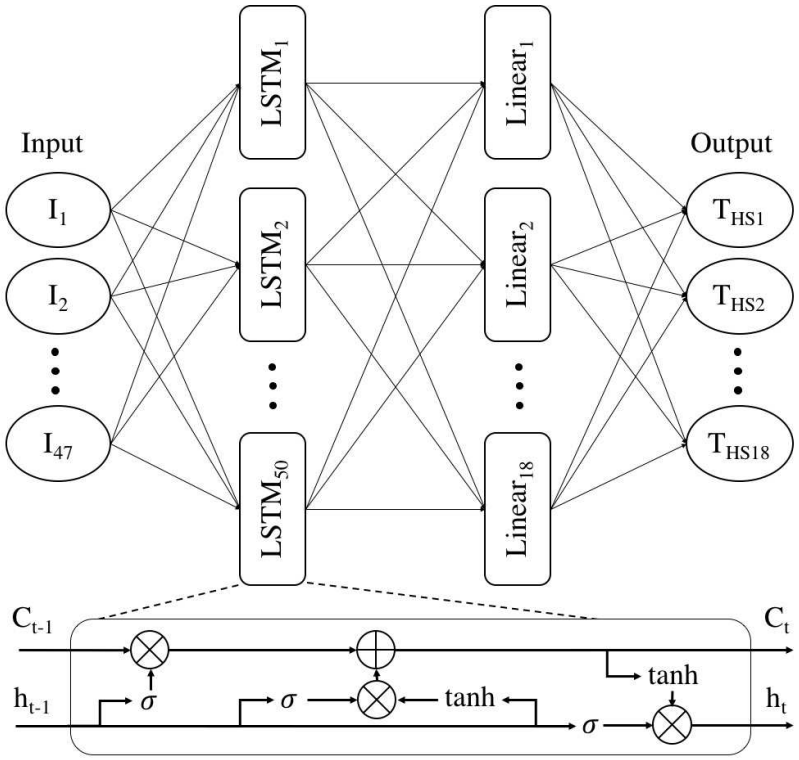


Figure 4.5: LSTM Network Configuration

Intel 22-core E5-2699 CPUs, and 320GB of memory. The total time elapsed for training was approximately 81 and 112 hours for the two models respectively. As previously mentioned, for each time-step t, 60 time-steps of IPCM metrics $[M_{t-s}; ..; M_t]$ were used as the features (input) while 1 time-step of the 36 most dominant DCT frequencies $\mathcal{F}_t$ extracted from $\mathcal{T}(x, y)_t$ was used as the label (output).

During the training process, it is essential to monitor the learning curve. If validation loss diverges significantly from training loss, this typically indicates overfitting. The learning curves for the two chips are shown in Fig. 4.7. In our case, overfitting was not found to be an issue. However, this will not always be true as overfitting is a common problem encountered during training. If overfitting is detected, then regularization techniques, such

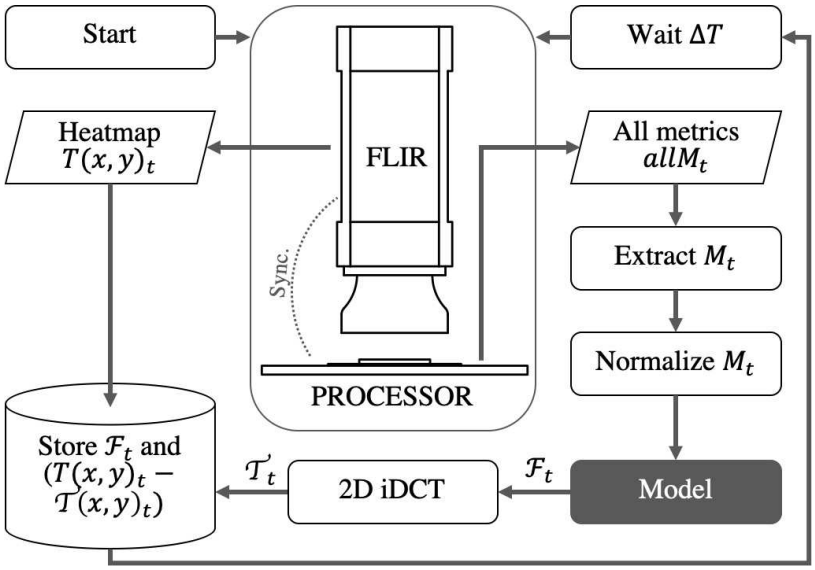
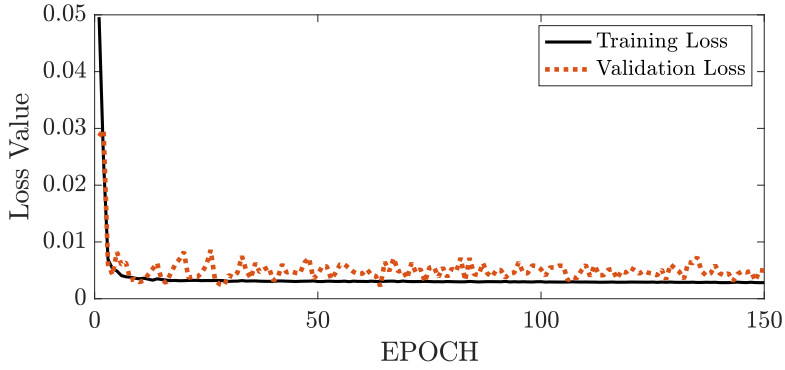


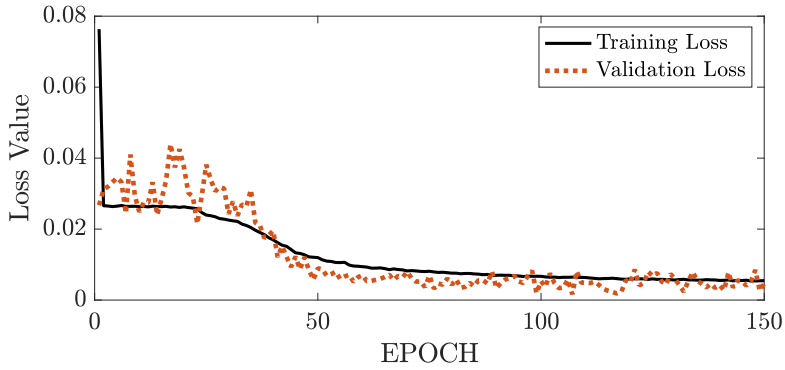
Figure 4.6: Testing Flow

as the popular Dropout [105] method, can be used to mitigate it. This will require further experimentation and tuning as with machine learning in general.

Once the model is trained, it is crucial to test it thoroughly with new data that was not used for training. As illustrated in Fig. 4.6, in this study, testing was done by capturing additional thermal data using the IR thermography setup and comparing the measured heatmaps ($T(x, y)$) with the estimated heatmaps ($\mathcal{T}(x, y)$) produced by the model. The error maps ($T - \mathcal{T}$) are saved in order to evaluate the accuracy of the model. For testing, it is recommended to execute a variety of workloads, ideally different than the ones used previously during the acquisition of the training dataset, in order to test both the accuracy and the generality of the model. In this study, the Phoronix test suite [94] was once again used for this purpose. Other benchmark suites such as PARSEC [103], and SPEC [104] or any random collection of workloads that vary in hardware utilization and



(a)



(b)

Figure 4.7: Learning curves: (a) i5-3337U (b) i7-8650U.

intensity will suffice as well. The randomly selected Phoronix workloads that were executed during the testing process are: cloverleaf, compilebench, cpp-perf-bench, himeno, pgbench, phpbench, bork, byte, node-octane, opt carrot, osbench, pyperformance, opencv-bench, pybench, sqlite-speedtest, ozone, postmark. This testing process allows us to calculate the error between the estimated heatmaps and the true measured heatmaps, while at the same time, measure the processing and memory overheads of the model. These results for both of the chips will be presented in the next section.

4.3.3 Deployment

Once the model has been trained and validated as shown in Sec. 4.3.2, it can be deployed in the processor as a OS-resident background application (software thermal monitor). Online inference can be achieved by directing the performance metrics to the thermal monitor, which in-turn outputs the estimated heatmap ($\mathcal{T}(x,y)_t$). This estimated heatmap can then be directed to a thermal/power controller or any other OS-resident application as desired. After a period of $\triangle t$, this process can be repeated again (Fig. 4.8). Note, it is important to set $\triangle t$ to be equivalent to the capture rate of the training datasets. In our case, it would be $\triangle t = 1/60sec$ since the training data was captured at the rate of $60Hz$. In the next section, we will present the experimental results from implementing the proposed framework on two test chips and analyze the model's performance both in terms of its estimation accuracy and its overheads compared to the current state-of-the-art approach.

4.4 Experimental Results and Comparisons

4.4.1 Experimental results

As outlined in Sec. 4.3.1 and Sec. 4.3.2, an extensive set of data were collected from both the i5-3337U and i7-8650U which were then used to train the respective RNN-based thermal models. Once the loss function saturates at a sufficiently low value, the two models were retrieved and put under the testing process illustrated in Fig. 4.6. As previously mentioned, the testing phase involves utilizing the model to estimate the processor's spatial heatmaps ($\mathcal{T}(x,y)$) while at the same time, the real heatmaps ($T(x,y)$) are captured using

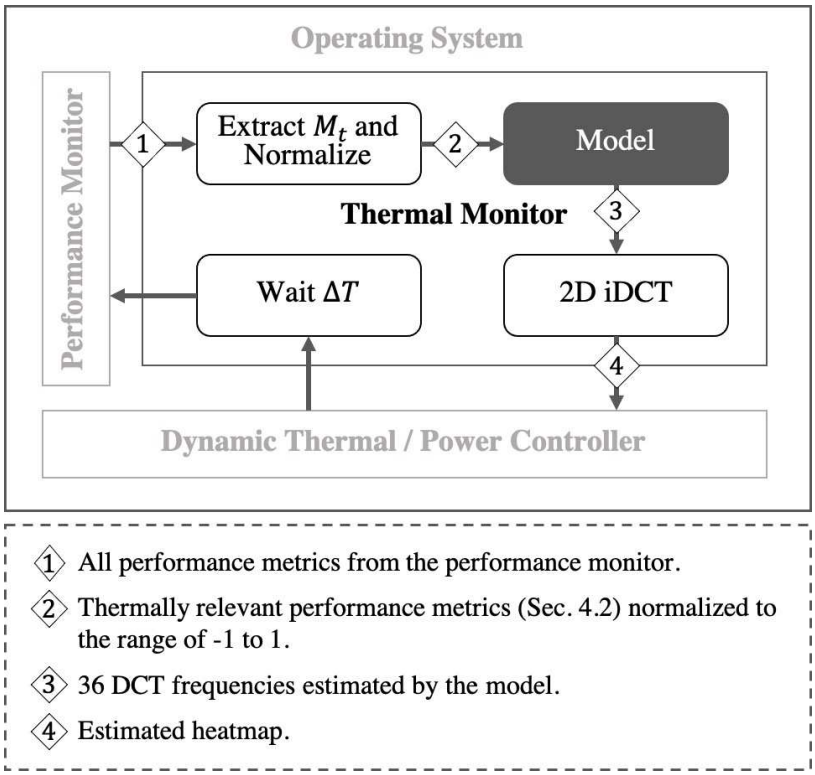
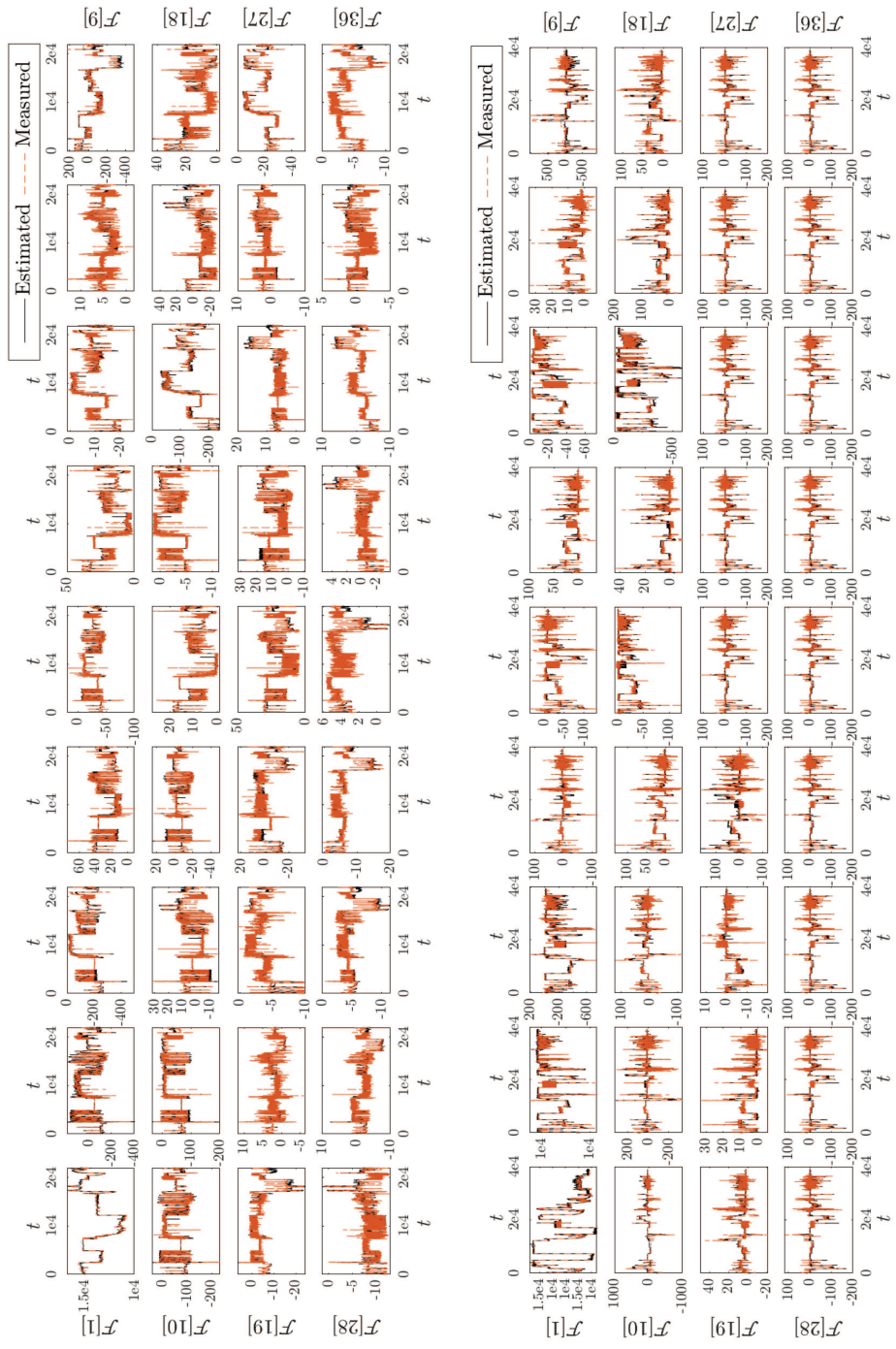


Figure 4.8: Model Deployment

the IR thermography setup. At each time-step, the error-maps ($T - \mathcal{T}$) are stored in order to compute the overall accuracy of the model.

Extensive testing conducted on the two chips show that the models perform exceptionally well. The results from the testing process are presented in Fig. 4.9, 4.10-4.15, and Table. 4.3. The estimated 36 DCT coefficients $\mathcal{F}$ follow the trends of the measured data with marginal error. The estimated DCT coefficients for both chips are shown in Fig. 4.9, plotted along with their measured counterparts from the testing process. For better visualization, we can randomly select a measured heatmap ($T(x, y)_t$) and compare it with the estimated heatmap ($\mathcal{T}(x, y)_t$) from the same testing time-step t. Fig. 4.10-4.15 show



(a) (b)

Figure 4.9: Estimated vs measured $\mathcal{F}[1]$ to $\mathcal{F}[36]$ (a) i5-3337U (b) i7-8650U.

a measured heatmap of the two chips alongside the estimated heatmaps generated by the models. The error-map $(T - \mathcal{T})$ is also shown. The heatmaps shown in Fig. 4.10, and 4.13 are from the randomly selected time-step $t = 15059$ and $t = 30073$ from the testing process for the two chips respectively.

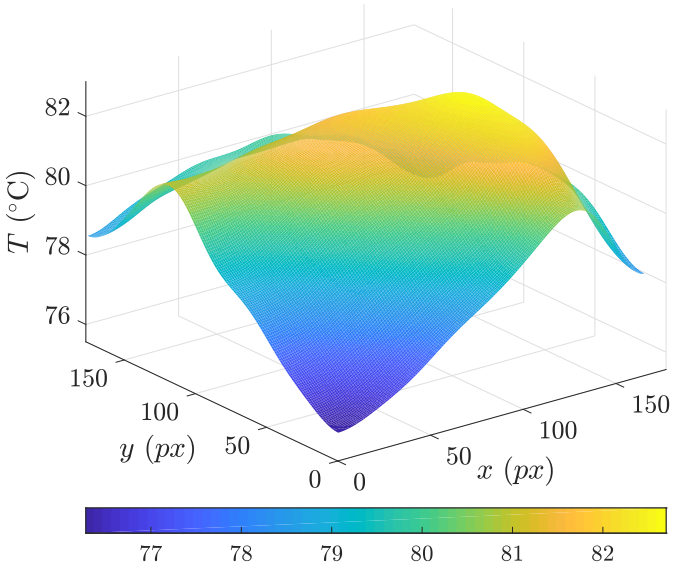


Figure 4.10: Measured $T(x, y)$ i5-3337U

In order to formally compute the overall accuracy of the model from the data acquired through the testing phase, we first assemble the error vector given in (4.4).

$$E = vectorize([T_1 - \mathcal{T}_1, .., T_{tmax} - \mathcal{T}_{tmax}]) \tag{4.4}$$

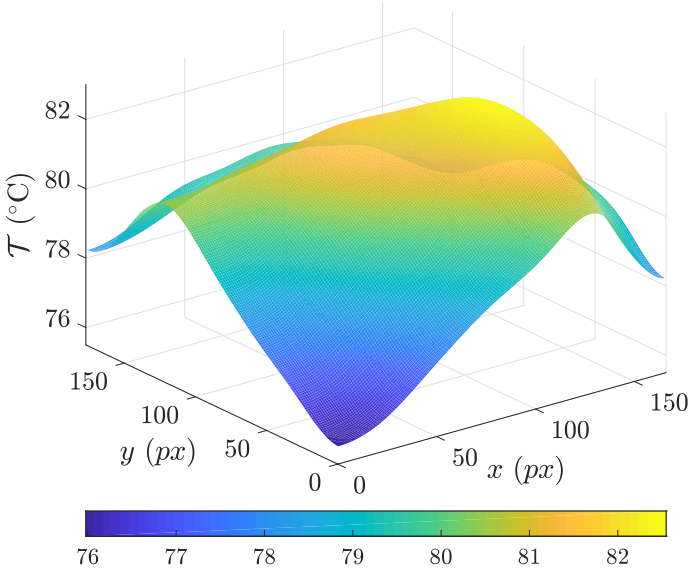


Figure 4.11: Estimated $\mathcal{T}(x, y)$ i5-3337U.

Table 4.3: Error stats - Realmaps

	RMS(E)	Mean(E)	Med(E)	Max(E)	Stdev(E)
i5-3337U	0.87°C	0.75°C	0.69°C	10.31°C	0.56°C
i7-8650U	1.24°C	0.86°C	0.70°C	9.74°C	0.75°C

Where matrices T and $\mathcal{T}$ are the measured and estimated heatmaps respectively, and $tmax$ is the final testing time-step. The total length of E is 8.8×10^8 and 1.3×10^9 elements for the i5-3337U and i7-8650U respectively. This error vector (E) captures all of the pixel to pixel errors between the measured heatmaps and the estimated heatmaps throughout the span of the entire testing process. Once E has been assembled, the error statistics shown in Table 4.3 can be calculated.

As shown in Table 4.3, the models yielded a root-mean-square error of 0.87°C and 1.24°C, max error of 10.31°C and 9.74°C and a mean error of 0.75°C and 0.86°C with a

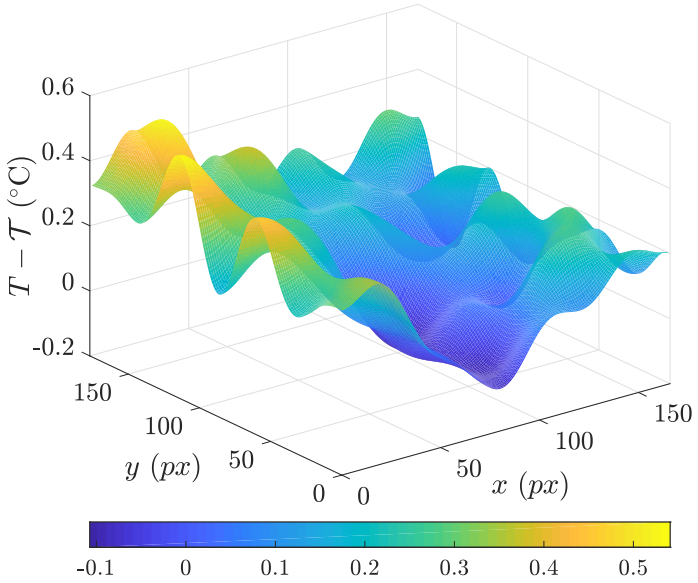


Figure 4.12: Error $T - \mathcal{T}$ i5-3337U

standard deviation of 0.56°C and 0.75°C for the i5-3337U and i7-8650U respectively. We believe that this is sufficient for full chip heatmap estimation, especially when considering the fact that the embedded temperature sensors are rated to have an error of ±5°C [83]. As previously mentioned, the heatmaps estimated by the model are to be used to supplement the temperature data sensed by the embedded temperature sensors, rather than being used as a substitute. For example, the readings from the embedded temperature sensors provide more accurate temperature data but only of a few pre-selected locations of the chip, whereas the proposed thermal model will offer full-chip temperature information albeit at lower accuracy. Hence, both the sensors and proposed model can be used together by the dynamic thermal and power controller in order to make a well informed regulation decision. In addition, the accuracy of the model is comparable to the existing pre-silicon techniques

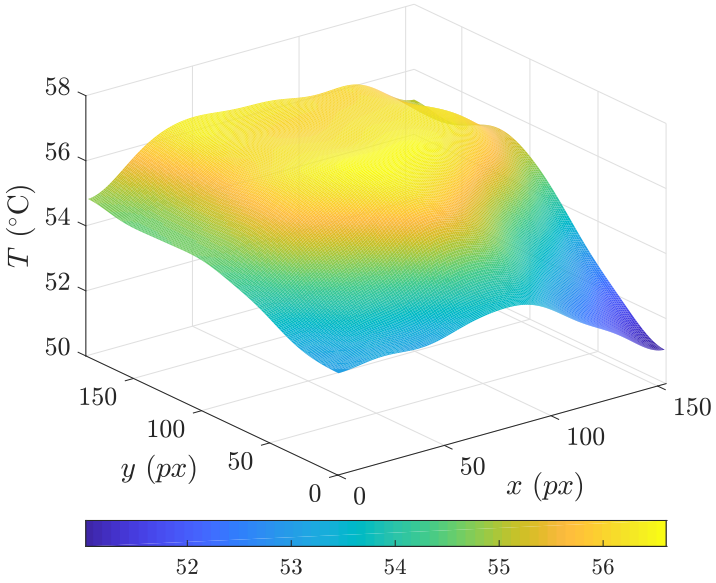


Figure 4.13: Measured $T(x, y)$ i7-8650U

which require specialized sensor placement algorithms to be adapted during design time and often require more sensors than the quota allocated for typical microprocessors [88, 95, 106, 107]. As we will show in the next subsection, this model is also more lightweight in terms of the computation and memory overheads when compared to the current state-of-the-art. The computation time of our model is, on average, 0.41 milliseconds per inference for both chips and its memory overhead, primarily incurred in storing the network weights, is 266Kb and 557Kb respectively for the i5-3337U and i7-3650U respectively. This makes the proposed full-chip heatmap estimation technique not only practical, but also highly desirable for online temperature estimation.

One caveat that should be noted, however, is the existence of variations between processor samples that stem from the manufacturing process. The model that is derived

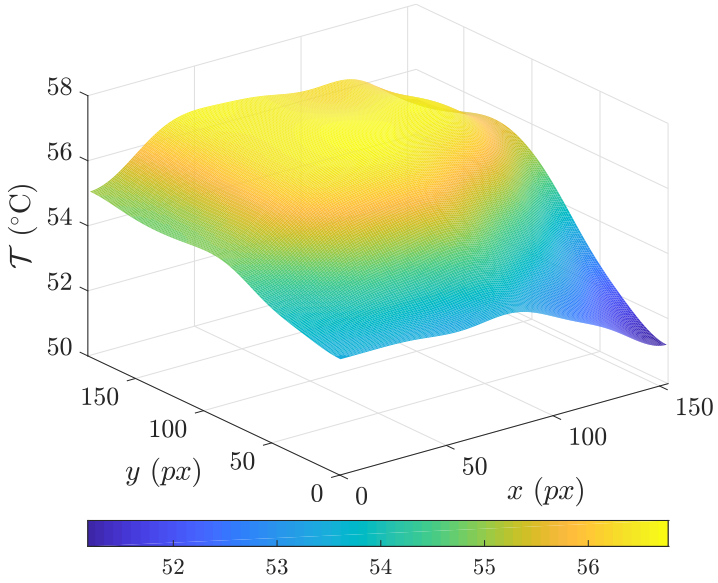


Figure 4.14: Estimated $\mathcal{T}(x, y)$ i7-8650U

using the proposed framework must be robust against such variations. In this study, only one sample of the i5-3337U and i7-8650U were used to collect the training dataset. However, in reality, it is recommended to use multiple samples of the given chip for both the acquisition of the training datasets, as well as for testing the end model. This aids in increasing the robustness of the model to such statistical variations.

4.4.2 Comparisons with the state-of-the-art pre-silicon approach

As previously mentioned, to our knowledge, the proposed framework is the first exclusively *post-silicon* approach to achieve real-time full-chip spatial heatmap estimation for commercial off-the-shelf microprocessors. However, as discussed in Sec. 4.1, pre-silicon methods based on smart embedded temperature sensor placement algorithms have been

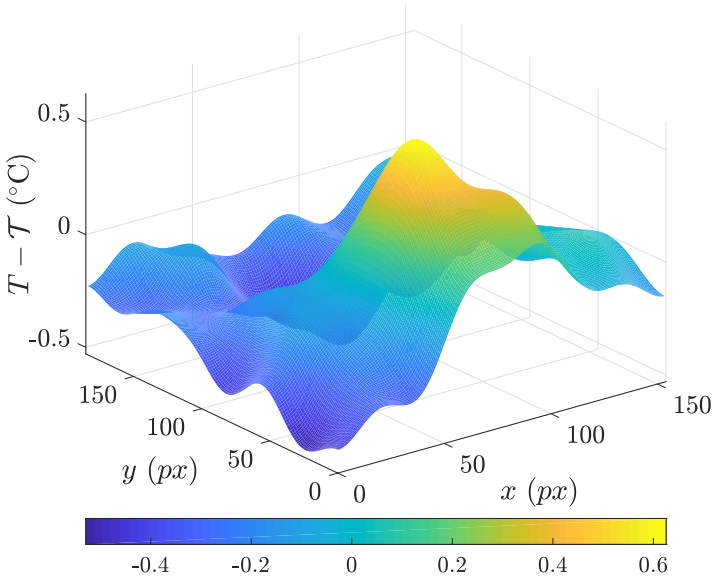


Figure 4.15: Error $T - \mathcal{T}$ i7-8650U

presented in the past [4, 95]. The current state-of-the-art pre-silicon method known as "*Eigenmaps*" was established by Ranieri *et. al.* [4]. In this subsection, we present the implementation of [4] for the two Intel chips used in this study (i5-3337U and i7-8650U) and compare the heatmap estimation accuracy as well as the overheads with that of the proposed post-silicon machine-learning based approach (*Realmaps*).

In summary, the approach in [4] is based on identifying the ideal basis (matrix Φ) for the vectorized spatial heatmaps of the given chip. Sensor locations are determined by calculating the correlation between all the rows of Φ and determining s least correlated spatial locations for sensor placement. Once the sensors are placed, the temperature readings from the sensors can be used to approximate the coefficients of expansion over Φ.

More specifically, let $v[i]$, where $0 < i < N = x_{max} \times y_{max}$, be the 1D vectorized

form of the 2D heatmap $T(x, y)$, where $0 < x < x_{max}$, $0 < y < y_{max}$, and x_{max} and y_{max} are the horizontal and vertical pixel counts of $T(x, y)$ respectively. This vectorization is done simply by vertically stacking the columns of $T(x, y)$ such that the 1D index i for v corresponds to the 2D index x, y for T according to (4.5)

$$v[i] = T\left[i \bmod y_{max},\ \left\lfloor \frac{i}{x_{max}} \right\rfloor\right] \tag{4.5}$$

In general, the vector v can be represented using a basis Φ as per (4.6).

$$v[i] = \sum_{j=1}^{N} \Phi[i, j]\alpha[j] \tag{4.6}$$

Here, vector $\alpha[j]$ contains the coefficients of expansion over basis Φ. With vectorized heatmap v expressed as (4.6), an approximate or compressed vectorized heatmap $\hat{v}$ can be described as a linear combination of the first K columns of Φ and the corresponding K elements of α as shown in (4.7). In other words, this process is a projection of the spatial heatmap onto the linear subspace spanned by the first K columns of basis Φ.

$$\hat{v} = \Phi[:, 0 : K]\alpha[0 : K] = \Phi_K \alpha_K \tag{4.7}$$

The optimal subspace Φ_K is the one that introduces the smallest error between v and $\hat{v}$. Finding this optimal subspace Φ_K is a classical problem that is better known as Principal Component Analysis (PCA). As per PCA, the ideal subspace Φ_K is the matrix whose columns are made up of the first K principal components (PCs) of matrix $V = [v_1, .., v_{tmax}]$. Ranieri *et. al.* aptly named these PCs as "*Eigenmaps*" as the analytical solution to computing the PCs involves calculating the Eigenvectors of the covariance matrix

of V. In [4], being a pre-silicon approach, V is derived by simulating the chip's layout using a thermal simulator. At each simulation time-step t, the simulated heatmap $T(x, y)_t$ is vectorized into v_t and stacked into column $V[:, t] = v_t$. This process is repeated until the final simulation time-step t_{max}. In this comparison, we will be using the measured heatmaps of our two chips (our entire training dataset from Sec. 4.3.1) as a substitute to the simulated heatmaps used in [4].

Similar to the DCT basis discussed in Sec. 4.2.1, here the higher the value of K, the better $\hat{v}$ will resemble v. However, since Φ_K is the ideal basis for the given problem, far fewer coefficients will be needed compared to the DCT basis. For example, Fig. 4.16 shows the RMSE computed between the measured heatmaps and the compressed counterparts using varying number (K) of columns in Φ. Comparing this with the same analysis done previously for the DCT basis (Fig. 4.3), it is clear that Φ_K is indeed a superior basis for the problem. However, the disadvantage of using Φ_K for real-time applications is that it has to be held in memory, which incurs a considerable memory overhead. Note, each columns of Φ_K contains $N = x_{max} \times y_{max}$ single-precision floating point values.

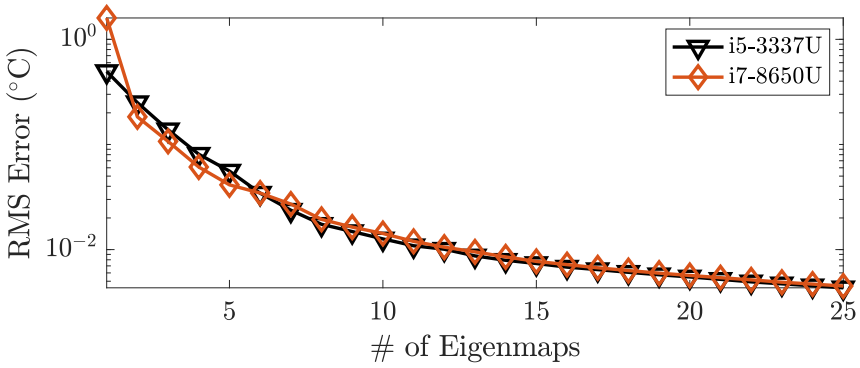


Figure 4.16: RMS error between actual heatmaps and heatmaps compressed using varying number of Eigenmaps.

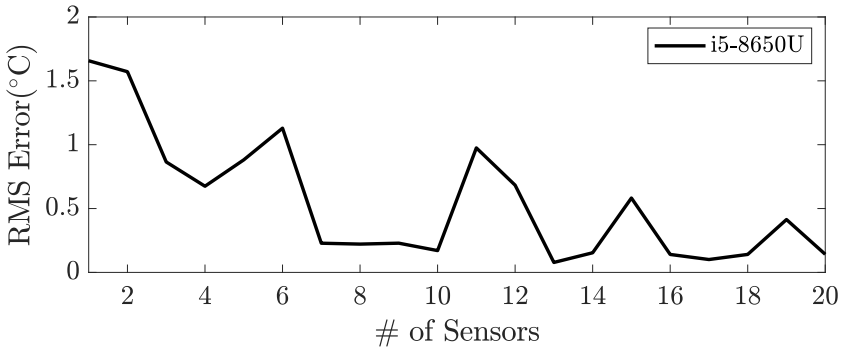
After Φ_K was calculated for the i5-3337U and the i7-8650U, the greedy sensor placement algorithm from [4] was implemented for the two chips to determine the optimal sensor locations given the allocation of s number of temperature sensors. Normally, after the sensors are placed in the design, and the chip is manufactured, the temperature readings from the embedded sensors (vector $v_s = [T_{sens\#1}, .., T_{sens\#s}]$) can be used to approximate the estimated vectorized spatial heatmap $\tilde{v}$ using (4.8).

$$\tilde{v} = \Phi_K(\tilde{\Phi}_K * \tilde{\Phi}_K)^{-1} * \tilde{\Phi}_K * v_s = C * v_s \tag{4.8}$$

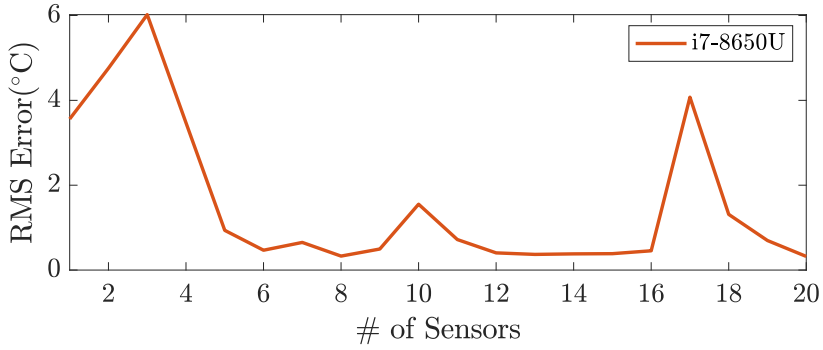
Here $\tilde{\Phi}_K$ is the matrix made up of the first s columns of Φ_K and the rows corresponding to the spatial coordinates of the selected sensor locations.

For our implementation of *EigenMaps*, we cannot physically embed the sensors as this would require the two chips re-manufactured with the sensors in place. Instead, sensor temperatures are **not sampled** from physical sensors but rather sampled from the measured heatmaps from our thermal imaging system. This is done by simply reading the temperature of the spatial locations where the sensor would have been located. For this comparison, RMSE between the measured heatmaps and the heatmaps estimated using the framework in [4] was calculated using various number of artificially embedded sensors, whose locations are determined by *EigenMaps'* sensor placement algorithm. The results for the two chips is shown in Fig. 4.17. The results show that the error between the estimated and measured heatmaps generally tend to decrease as the number of allocated sensors is increased. In reality, the number of sensors allocated for a processor typically depends on its core count. This is often equal to the number of cores + 1. For example, the dual-core

i5-3337U and the quad-core i7-8650U have 3 and 5 sensors respectively. Accordingly, we will consider $s = 3$ sensors, and $s = 5$ sensors for the two chips in this comparison. The accuracy results for Eigenmaps are presented in Table 4.4 for the two chips. As the results show, the estimation accuracy of Eigenmaps is comparable to what was achieved with Realmaps (Table 4.3), especially considering that Eigenmaps requires pre-silicon design considerations, where as Realmaps is an exclusively post silicon framework.



(a)



(b)

Figure 4.17: RMS error between measured heatmaps and heatmaps estimated using [4] as a function of the number of embedded temperature sensors. (a) i5-3337U (b) i7-8650U.

In terms of overheads, deploying the method in [4] for real-time inference would

Table 4.4: Error stats - Eigenmaps

	RMS(E)	Mean(E)	Med(E)	Max(E)	Stdev(E)
i5-3337U	0.86°C	0.57°C	0.34°C	9.02°C	0.65°C
i7-8650U	0.94°C	0.57°C	0.26°C	12.52°C	0.74°C

require the expression in (4.8) to be calculated for each inference. This is computationally very expensive and is therefore not suited for online use. Alternatively, the matrix C in (4.8) can be pre-calculated and stored in memory. This way, each inference is simply a matrix-vector-multiplication operation. While this is the more suitable option, it does require the entire matrix C to be stored in memory. Note, matrix C is an $N \times N$ matrix whose exact size is 863301924 and 811680100 single-precision floating point elements for the two chips respectively. This translates to a memory overhead of 3.45GB and 3.25GB respectively, which is quite expensive. This can however be remedied by considering lower resolution heatmaps as done in [4].

We remark that the comparison against *EigenMaps* [4] is not an apples-to-apples comparison. The reason being, *EigenMaps* is a *pre-silicon* approach that requires the placement of embedded sensors at specific locations during design time to achieve the reported accuracy; this is fundamentally different from the proposed approach which requires no design time considerations. However, the above comparison does show that the proposed data-driven *RealMaps* framework can yield very similar results in terms of accuracy, with a substantially lower computational overhead. Additionally, the *post-silicon* nature of *RealMaps* makes it feasible for existing commercial off-the-shelf processors. Moreover, it can be used by third parties who do not have control over and, in most cases, have no knowledge of the proprietary design details of the chip. This includes system integrators interested in developing ultra compact mobile devices that benefit from innovative thermal monitoring and

management software, and academic research labs that can use the full chip temperature estimation to develop advanced thermal control schemes.

4.5 Summary

In this chapter, we have proposed a machine learning based framework to real-time estimation of full-chip heatmaps for commercial microprocessors. The proposed approach, named *RealMaps*, only uses the existing embedded temperature sensors and system level utilization information, which are available in real-time. Moreover, it is structured to not require any knowledge of the proprietary design details or manufacturing process-specific information of the commercial processors. Consequently, the methods presented in this work can be implemented by either the original chip manufacturer or a third party alike. In this new approach, we start with accurate spatial and temporal heatmaps measured using an advanced infrared thermal imaging system. To build the transient thermal model, we utilize temporal-aware long-short-term-memory (LSTM) networks with system-level variables such as chip frequency, voltage, and instruction counts as inputs. Instead of a pixel-wise heatmap estimation, we use 2D spatial discrete cosine transformation (DCT) on the heatmaps so that they can be expressed with just a few dominant DCT coefficients. Our study shows that only 36 DCT coefficients are required to maintain sufficient accuracy. Experimental results show that *RealMaps* can estimate the transient heatmaps with 0.9°C and 1.2°C RMSE with minimal overheads for the two commercial chips tested in this study. Compared to the state-of-the-art *pre-silicon* method, the proposed approach shows similar accuracy, but with much less computational cost.

Chapter 5

Power-density Driven Thermoelectric Array Based Targeted Cooling

5.1 Related Work and Motivation

5.1.1 Thermoelectric effects in a nutshell

Thermoelectric cooling uses the Peltier effect to create a heat flux at the junction of two different types of materials. A Peltier cooler, heater, or thermoelectric heat pump is a solid-state active heat pump which transfers heat from one side of the device to the other, with consumption of electrical energy, depending on the direction of the current.

Thermoelectric coolers (TEC) are based on the principles of the thermoelectric (TE) effect where heat flux is generated at the intersection of two materials with different

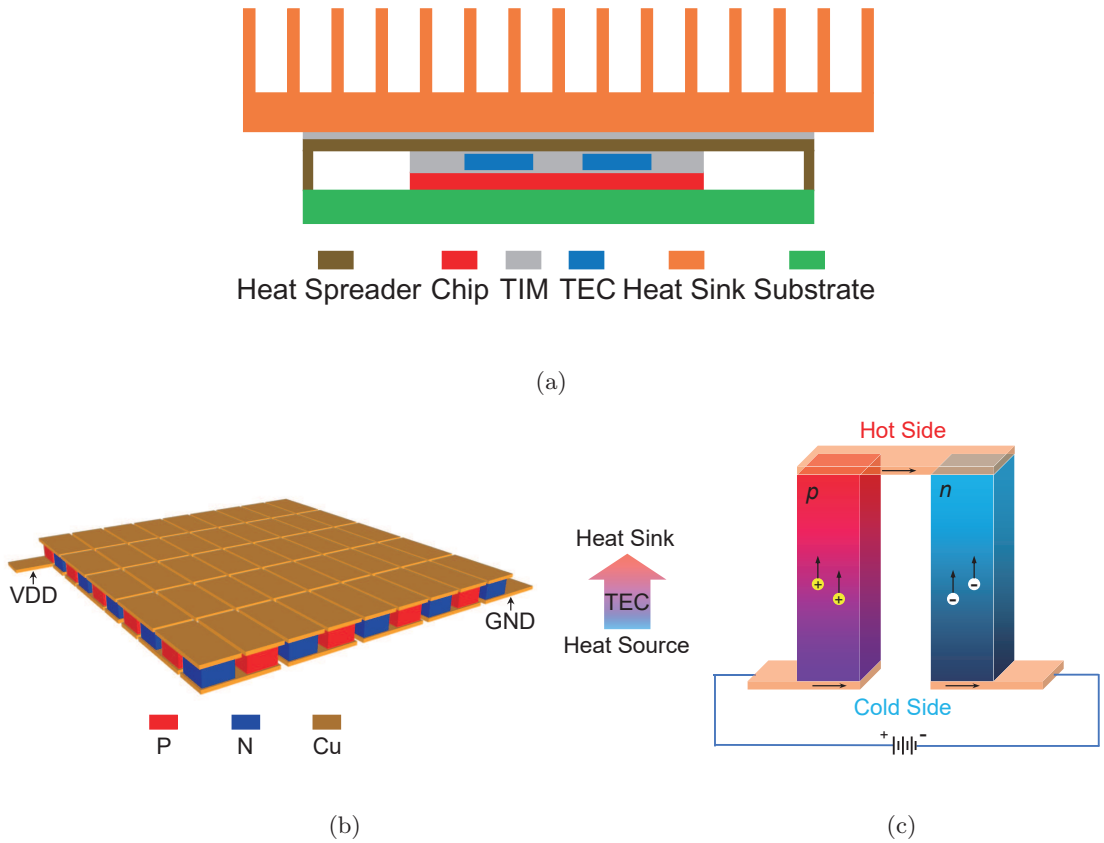


Figure 5.1: (a) The side view of the chip package. (b) 3D view of thin-film TEC devices. (c) Peltier effect for an N-P pair in the TEC devices.

electron densities, as shown in Fig. 5.1. TECs have two sides, namely the cold side and the hot side, where heat is transferred from the cold side to the hot side along the direction of DC current flow. In cooling applications, the hot side of the TEC is typically attached to a heatsink in order to keep it at ambient temperature allowing the cold side to go below ambient temperature. While the basic principles of the TE effect is intuitive, the physics governing this phenomenon is complex and therefore challenging to model and simulate.

The TE effect is an energy conversion phenomenon that consists of the Seebeck, Peltier, Thomson and Joule heating effects. We assume that the two ends of the TE material

are excited by an ideal voltage source, which is capable of supplying and maintaining the same voltage. Under these conditions, the Seebeck effect cannot impact the voltage and current of TE materials. Therefore, we do not consider the Seebeck effect. Thomson effect is typically ignored for constant Seeback coefficient. Hence, the dominant factors for modeling the TE effect are as follows.

First, under the Peltier effect, the current density $\mathbf{J}$ ($\mathrm{A/m^2}$) driven by the voltage across two ends produces heat flux $\mathbf{q}_\mathrm{P}$ moving from cold side to hot side, which is expressed as

$$\mathbf{q}_\mathrm{P} = P\mathbf{J} \tag{5.1}$$

where $P = S \times T$ is the Peltier coefficient, S is the Seebeck coefficient (V/K), T is the temperature, and heat flux $\mathbf{q}_\mathrm{P}$ is a flow of heat energy per unit area per unit time ($\mathrm{W/m^2}$).

Second, Fourier's law describes that the heat flux flows from high temperature to low temperature, which is expressed as

$$\mathbf{q}_\mathrm{Fourier} = -\kappa \nabla T \tag{5.2}$$

where κ is the thermal conductivity, and ∇ is the gradient vector operator.

Last, the power per unit volume ($\mathrm{W/m^3}$) due to Joule heating effect is calculated by

$$g_\mathrm{Jh} = \mathbf{J} \cdot \mathbf{E} = \frac{||\mathbf{J}||^2}{\sigma} \tag{5.3}$$

where σ is electrical conductivity, $\mathbf{E}$ is the electric field, "$\cdot$" is the dot product operator, $||\mathbf{J}||^2$ is the dot product of $\mathbf{J}$ and itself, and $\mathbf{J}$ is equal to the product of σ and $\mathbf{E}$. To model the TE effect, two models were primarily investigated in existing work [108–110]. One is

a simplified 1D energy equilibrium model and the other is a more complicated 3D coupled multiphysics model.

Simplified one-dimensional energy equilibrium model was first proposed to characterize the cooling heat flux of the TEC device based on energy balance on the cold side. The model accounts for the Peltier, Joule heating and heat transfer effects. Here cooling heat flux, which is a key parameter that represents the cooling capacity of TEC, is given by [108, 110, 111]

$$q_c = q_{\mathrm{P}} - q_{\mathrm{Jh}} - q_{\mathrm{Fourier}} = ST_cJ - \frac{1}{2}\frac{J^2L}{\sigma} - \frac{\kappa}{L}(T_h - T_c) \tag{5.4}$$

where T_c and T_h are the temperatures at the cold and hot side, respectively. L is thickness of TEC leg. q_{Fourier} is essentially the heat flux from high temperature to low temperature; which is Fourier's law expressed by

$$q_{\mathrm{Fourier}} = -\kappa\nabla T \approx -\kappa\frac{\Delta T}{\Delta x} = \kappa\frac{T_h - T_c}{L} \tag{5.5}$$

The effective heat flux caused by Joule heating at cold side is calculated by

$$q_{\mathrm{Jh}} \approx \frac{1}{2}\frac{Q_{\mathrm{Jh}}}{A} = \frac{1}{2}\frac{g_{\mathrm{Jh}}\mathcal{V}}{A} = \frac{1}{2}\frac{J^2L}{\sigma} \tag{5.6}$$

where A and $\mathcal{V}$ are cross-sectional area and volume of the TEC leg, respectively, and g_{Jh} is defined in (5.3); and Q_{Jh} is the heat produced by Joule heating.

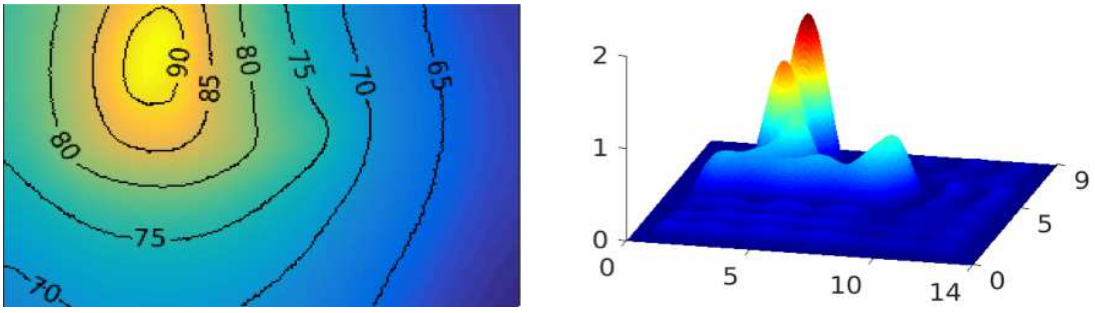
However, this model is overly simplified with approximated expressions to describe the Peltier, Joule heating and Fourier heat transfer effects. The results produced by the model are inaccurate when the thermal gradient is large because the formula is too simple to describe these complicated phenomena. An accurate TEC model needs to be developed to consider spatial temperature as we will discuss in Sec. 5.3.3.

5.1.2 Runtime power-map estimation

Real-time estimation of the spatial power-maps (or power-density maps) is essential for controlling the proposed TEC-Array. And real-time estimated thermal-maps are highly valuable since they can be used to estimate the spatial power-density maps using a thermal-to-power transformation method [5,112]. In [112], the authors developed a general blind power identification, called *BPI* method for power-maps from thermal measurements. Whereas [5] proposed a first-principle based efficient power-map estimation utilizing thermal measurements. In essence, the latter work simply calculates spatial power density map using the steady state thermal diffusion equation shown in (5.7).

$$-\kappa\nabla^2 T = g_T \tag{5.7}$$

Here T is the spatial thermal-map (in K), κ is the thermal conductivity, g is the volumetric power-density ($\mathrm{W}\cdot\mathrm{m}^{-3}$), which can be transformed to spatial power-density map by multiplying the silicon thickness factor, and ∇^2 is the Laplace operator. Hence, if T can be estimated using the method previously discussed in Ch. 4, then g_T can be calculated using (5.7). Note, κ is the effective thermal conductivity of the processor die. This can either be calculated by the manufacturer, with the knowledge of the exact material composition of the die, or experimentally estimated by a third-party using infrared measurements and using (5.7) as demonstrated in [5]. For example, Fig 5.2 shows the spatial thermal map (T) of an Intel i7-8650U (Fig 5.2(a)), in one thermal steady state, and the corresponding power-map (Fig 5.2(b)) calculated using Eq.(5.7). Here, the effective thermal conductivity was calculated to be $\kappa = 174\mathrm{W}\cdot\mathrm{m}^{-1}\cdot\mathrm{K}^{-1}$ in [5].



(a) (b)

Figure 5.2: Thermal-map to power-map conversion: (a) Experimentally measured thermal map (T), (b) Estimated power-density map (g_T) in 3D view. [5]

5.2 Powermap estimation overview

5.2.1 Powermap estimation summary

Accurately estimating the spatial powermaps of the chip is important for effectively controlling the proposed TEC Array. In this article we present simulation results from a commercial physics simulator, where the power density maps are known. However, in reality, when the processor is operational, determining the power density maps in real-time is not a trivial task. Alternatively, one can estimate the full-chip thermal maps and use the thermal-map to power-map conversion methodology to estimate the power-maps as discussed in Sec. 5.1.2.

We remark that from practical application perspective, the machine learning based full-chip power density map estimation approach discussed above can be trained using measured data from a thermal IR imaging system (shown below) or other numerical simulation methods, during the design-time, considering practical cooling, package and thermal conditions. Such modeling only needs to be done once during design time or the post-silicon

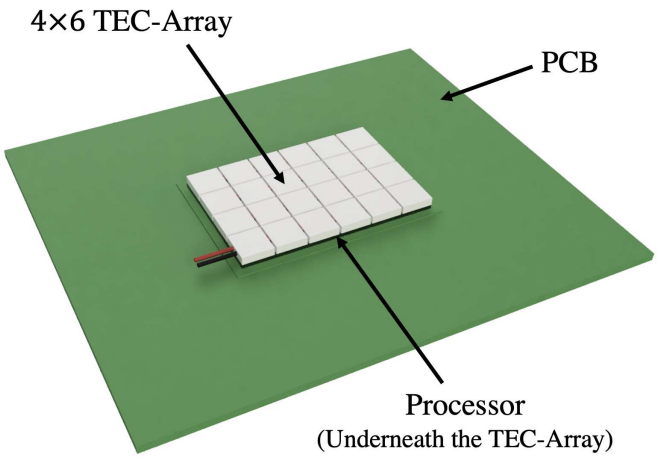
phase and the resulting models can be applied to different cooling conditions (with heat sink, without heat sink, etc.) as demonstrated recently in [5]. Different cooling conditions will change the temperature of the chip but they do not change the locations and shapes of hotspots, and thus (relative) power density values of those hotspots will remain the same. As a result, the power density map (relative distribution of power across the die are) will be **similar** under all cooling conditions as it fundamentally represents the spatial power consumption of silicon with respect to the chip's utilization. For TEC-based control, such relative valued power density map will be sufficient as it provides the **key differential values** for different TEC devices.

5.3 Proposed TEC array control framework

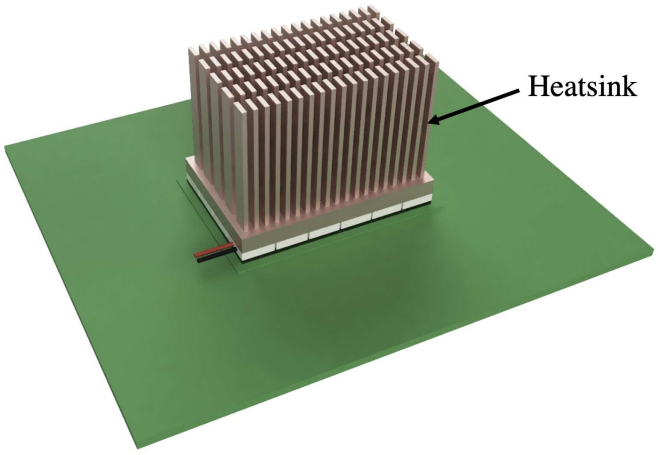
5.3.1 TEC-array architecture

The proposed TEC-Array consists of a number of small TEC modules arranged to form a 2D array. In our case, we will consider a 24 unit array (4×6 TEC modules) as shown in Fig. 5.3(a). The cold side of the TEC-Array is to be affixed over a microprocessor (Fig. 5.3(a)), with a traditional heatsink (or liquid cooling unit) affixed to the hot-side (as shown in Fig. 5.3(b)). The TEC-Array will "pump" the heat from the surface of the chip to its hot side, which is in turn extracted by the heatsink and ultimately transferred to ambient air through natural or forced convection.

The proposed TEC-Array is designed such that the rate of heat transfer through each TEC module in the array can be controlled by varying the amplitude of voltage applied



(a)



(b)

Figure 5.3: (a) TEC-Array affixed over a processor. (b) Heatsink affixed over the TEC-Array.

to each module. Such granularity in control can be possible because the proposed TEC-Array is affixed over the processor's heat-spreader (outside the packaging). For methods where the TEC modules are integrated into the packaging, this level of granular control is not possible due to design constraints [44]. This setup enables the ability to precisely control the cooling across the surface-area of the processor die in a non-homogeneous manner. Meaning, the areas of the chip consuming more power (hence generating more heat) can be cooled more intensely than the areas of the chip that are consuming less power (hence generating less heat). This level of control significantly aids in reducing the thermal gradients across the chips surface (difference in temperature in one area of the chip vs. another), consequently normalizing the rate of aging effects across the chips surface as previously mentioned. This is in contrast with the traditional TEC-based coolers where a single large TEC module is used, consequently limiting the spatial control.

5.3.2 TEC-array control flow

Controlling the TEC-Array can be done by generating a discrete voltage map (V) with the same dimensionality as the TEC-Array (4×6 in our case), where each index of V denotes the voltage that is to be applied to the corresponding TEC-module. For effective thermal control, at each timestep t, V_t should be proportional to the power density map (g_{T_t}).

To this end, we propose the TEC-Array control flow illustrated in Fig. 5.4. Here, at each time-step (t), the thermal-map of the chip (T_t) is estimated using a thermal model (Ch. 4), which is then used to calculate the power-density map g_{T_t}. Once g_{T_t} is determined,

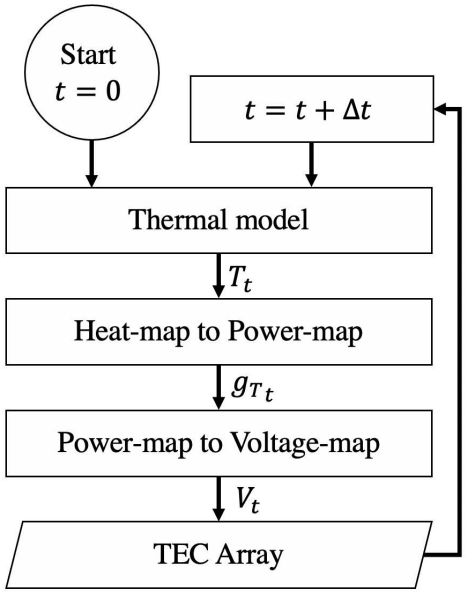


Figure 5.4: Proposed TEC-Array control flow

V_t can be calculated as per Eq. (5.8).

$$V = resize(\frac{g_T - g_{min}}{g_{max} - g_{min}} \times V_{max}) \tag{5.8}$$

Here, g_{min} and g_{max} denote the minimum and maximum power density observed across all time-steps, and V_{max} is the maximum usable voltage rating of the TEC modules. The resize function is used to reduce the dimensionality of V to match the TEC-Array (4×6 elements).

Eq. (5.8) generates V such that the TEC modules located in the areas of the chip with higher power density are assigned higher voltages than the TEC modules located in the areas of the chip with lower power density. For example, Fig. 5.5 and Fig. 5.6 show two power-density maps (g_T) of an Intel i7-8650U quad-core processor. Fig. 5.5 shows g_T

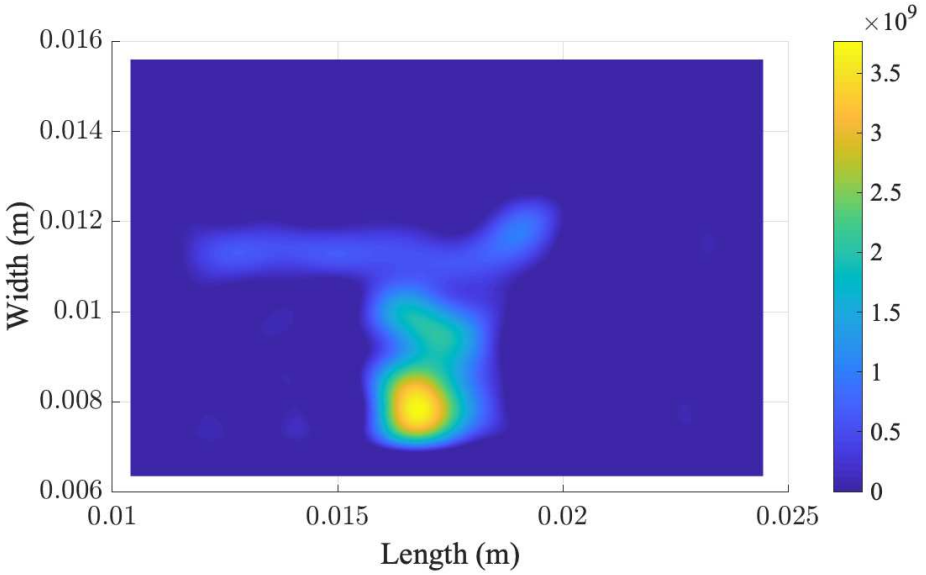


Figure 5.5: Powermap (W/m^2) of an Intel i7-8650U under a single-threaded workload

when the processor is executing a single-threaded workload (only one core active), where as Fig. 5.6 shows g_T when the processor is executing a multi-threaded workload (all four cores active). Here we see a single hotspot (location of high power density) in Fig. 5.5 and four distinct hotspots in Fig. 5.6, with significant portions of the chip under lower power-density in both cases. With the proposed TEC-Array and the aforementioned control scheme, the TEC modules located over the hotspots will be assigned higher voltages than the ones located over non-hotspots as shown in Fig. 5.7 and Fig. 5.8 (voltage-maps V). In contrast to cooling the entire chip at the same rate, this method allows targeted cooling of hotspots which in turn results in significant saving in the energy consumption of the cooling system

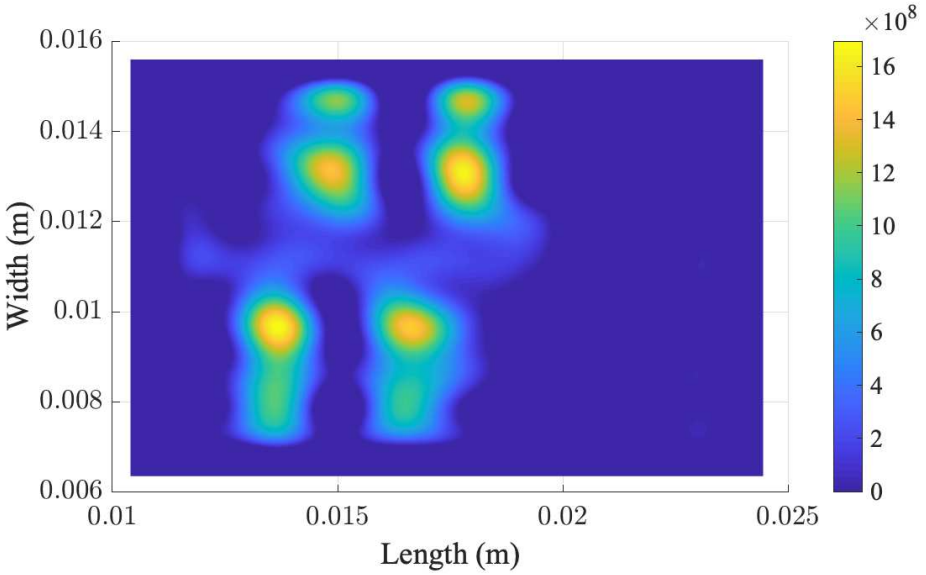


Figure 5.6: Powermap (W/m^2) of an Intel i7-8650U under a multi-threaded workload

as we will demonstrate in Sec. 5.4.

5.3.3 3D multiphysics model for TEC devices

COMSOL Multiphysics, a commercial finite-element-based physics simulation software, was used to simulate the proposed cooling system. The setup illustrated in Fig. 5.3 was configured in COMSOL using the built in 3D modeling graphical user interface. In terms of the material properties, COMSOL's default materials for the printed circuit board (PCB), silicon (for the processor) and copper (for the heatsink) were used. The default conductive boundary was assumed between the PCB, processor, TEC-Array and heatsink, with natural convection between the heatsink and static ambient air. The ambient temper-

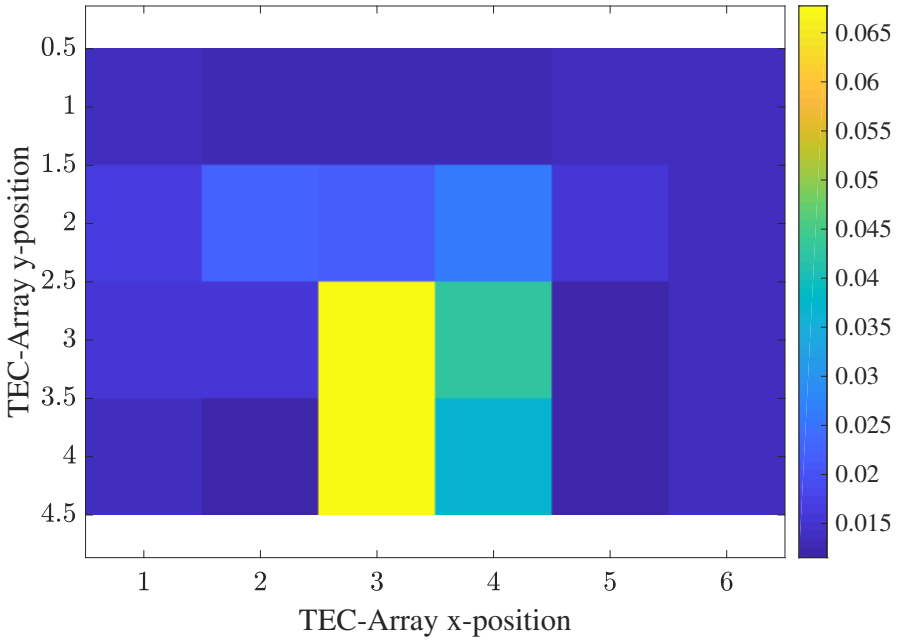


Figure 5.7: Voltage map ($V/position$) generated using Eq. (5.8) from g_T of Fig. 5.5

ature was set at 20°C. The processor is set as the heat-source with spatial power-density set as per g_{T_t} for each time-step t. The 3D geometry of the 4×6 TEC-Array was designed with the TEC leg height of 0.5mm. The physics of the TEC modules are configured as described below.

A 3D coupled multiphysics model is employed to characterize the behavior of TEC devices accurately [109, 110, 113–115]. Considering thermal and electrical fields, the thermoelectric analysis is a electro-thermal co-simulation, meaning the two fields are solved simultaneously. Therefore, this model is called a 3D coupled multiphysics model, where Peltier effects build the relationship between thermal and electrical fields.

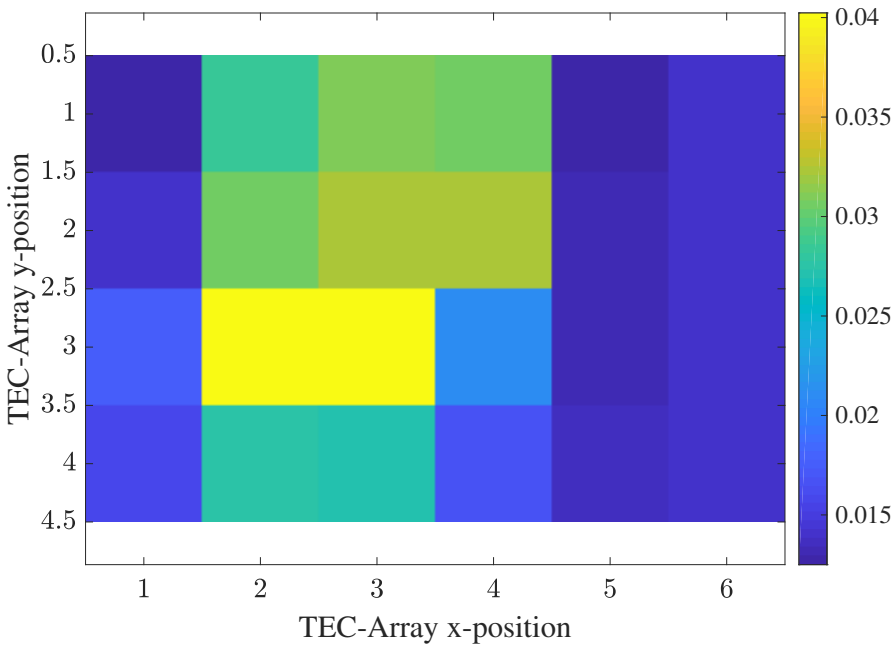


Figure 5.8: Voltage map ($V/position$) generated using Eq. (5.8) from g_T of Fig. 5.6

Specifically, for electric field, we have

$$\mathbf{E} = -\nabla\phi \tag{5.9}$$

where ϕ is the potential in the TEC. Then, the current continuity equation is formulated as

$$\nabla \cdot \mathbf{J} = \nabla \cdot (-\sigma\nabla\phi) = 0 \tag{5.10}$$

One side is set to ground boundary condition $\phi(0) = 0$ and the other side is excited by an ideal voltage source $\phi(L) = V$.

For thermal field, we integrate the Peltier effect into the heat flux

$$\mathbf{q} = \mathbf{q}_{\text{Fourier}} + \mathbf{q}_{\text{P}} = -\kappa\nabla T + P\mathbf{J} \tag{5.11}$$

Then, considering the Joule heating, the transient heat conduction equation is rewritten as

$$\nabla \cdot \mathbf{q} = \nabla \cdot (-\kappa \nabla T + ST\mathbf{J}) = \frac{||\mathbf{J}||^2}{\sigma} \tag{5.12}$$

The heat sink is placed at one side to dissipate the heat, and we set this side to a constant temperature boundary condition

$$T(0) = T_0 \tag{5.13}$$

where T_0 is the reference temperature (hot side for TEC device). Another side is used to remove the heat from chip (the cold side from TEC device), and boundary condition is set to the heat flux boundary condition

$$-\mathbf{n} \cdot \mathbf{q} = -\mathbf{n} \cdot (-\kappa \nabla T + ST\mathbf{J}) = q_c \tag{5.14}$$

where $\mathbf{n}$ is the unit outward normal vector of the boundary surface, q_c is the cooling heat flux.

Based on partial differential equation, 3D coupled multiphysics model considers thermal gradients and can simulate the heat conduction more accurately compared with simplified 1D energy equilibrium model. In this work, the Mathematics Module in COMSOL Multiphysics is used to numerically solve the 3D coupled multiphysics model (5.10) and (5.12).

5.4 Results and discussions

A total of 190 power-density maps calculated using thermal-maps of an an Intel i7-8650U test chip were used for this work [116]. For comparison purposes, three scenarios were

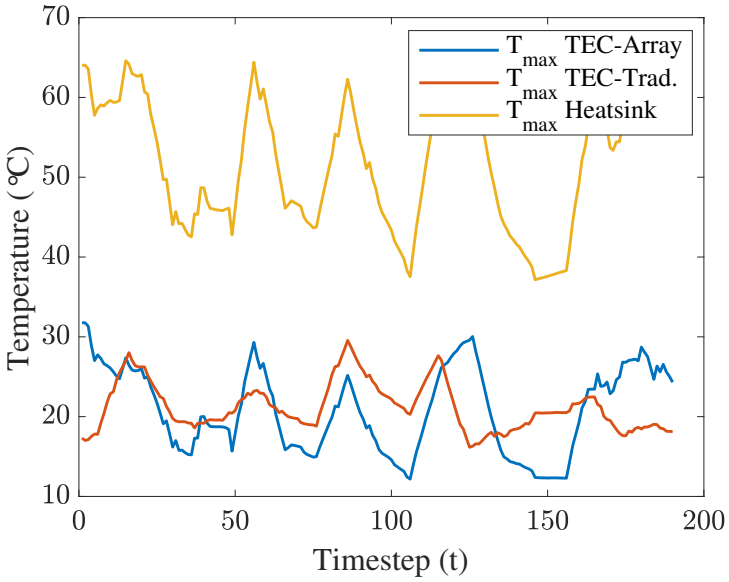


Figure 5.9: Max temperature

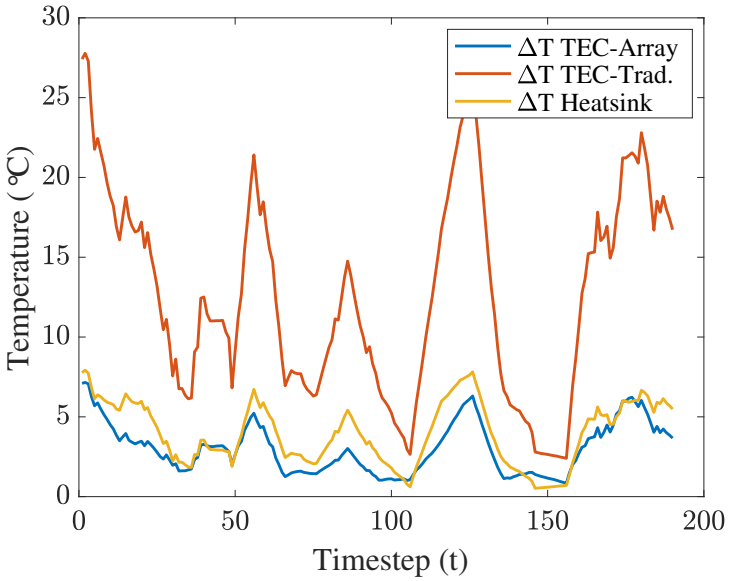


Figure 5.10: Spatial temperature range

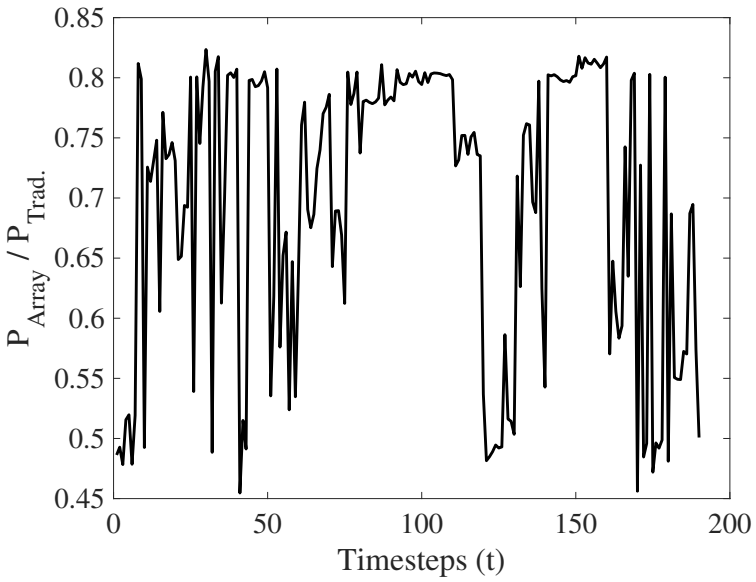


Figure 5.11: Relative power consumption

considered. First is the processor cooled using the proposed TEC-Array and the control scheme from Sec. 5.3. For the second scenario, we consider the traditional TEC cooling approach where a single large TEC module is used as opposed to the proposed array of TEC modules. In both cases, the same heatsink (Fig. 5.3(b)) was affixed over the TEC modules. For the third scenario, we remove the TEC layer and affix the heatsink directly over the processor to mimic standard passive heatsink cooling. All other parameters are kept exactly the same for all three scenarios.

All three scenarios were simulated for a total of 190 time-steps, using the aforementioned 190 power-density maps of the Intel i7-8650U. For the results, we present three metrics (T_{max}, ΔT, and $P_{Array}/P_{Trad.}$). Fig. 5.9 shows the max temperature (T_{max}), observed across the area of the chip. The max temperature is crucial because it directly impacts the performance of the processor. On-board dynamic voltage and frequency scal-

ing (DVFS) controllers dynamically throttle the processor's voltage and frequency according to max temperature. On Intel chips, the DVFS controller allows the frequency to exceed the chip's rated frequency (i.e. 4Ghz) when the max temperature is below a threshold (called Intel thermal velocity boost [83]); the DVFS controller will also scale the voltage and frequency down significantly if T_{max} is high in order to prevent the temperature from exceeding the maximum junction temperature of the given technology node [83]. Hence, T_{max} is an important factor that affects the processor's performance. From Fig. 5.9, we see that the processor cooled using the TEC-Array can maintain T_{max} similar to the traditional TEC system (T_{max} TEC-Trad.) where a single large TEC module is used to cool the entire surface of the chip at the same rate (also voltage modulated based on peak power). Note, both the TEC-Array and TEC-Trad outperform the heatsink-only cooling as expected.

The second metric presented in Fig. 5.10 shows ΔT; that is the difference between the maximum (T_{max}) and minimum temperature (T_{min}) across the die area. ΔT represents the spatial thermal gradients, which has huge impacts on the chip reliability due to effects such as thermal cycling [117], thermo-migration [118,119] and timing violation due to uneven stressing or aging of chip. For better visualization, we present Fig. 5.12 and 5.13 that shows the simulated thermal-map (spatial temperature distribution across the die area) at time-step $t = 53$. Fig. 5.12 shows the thermal-map under TEC-Array cooling where as Fig. 5.13 shows the thermal map under TEC-Trad cooling. Here we can see that, because the proposed TEC-Array enables spatially variable cooling, where the hotspots are cooled more intensely than the non-hotspots, ΔT is relatively low. This is in contrast with TEC-Trad which can have much larger temperature gradients. As a result, the proposed TEC-array

cooling actually can allow chip to run in higher performance modes via DVFS thanks to the increased thermal design power (TDP) as shown in Fig. 5.9 and at the same time maintaining or even increasing chip design lifetime due to the reduced thermal gradients across the chip. This can significantly reduce the thermal gradient induced stress such as thermal cycling for devices, thermo-migration for interconnects, and unbalanced thermal induced aging [24].

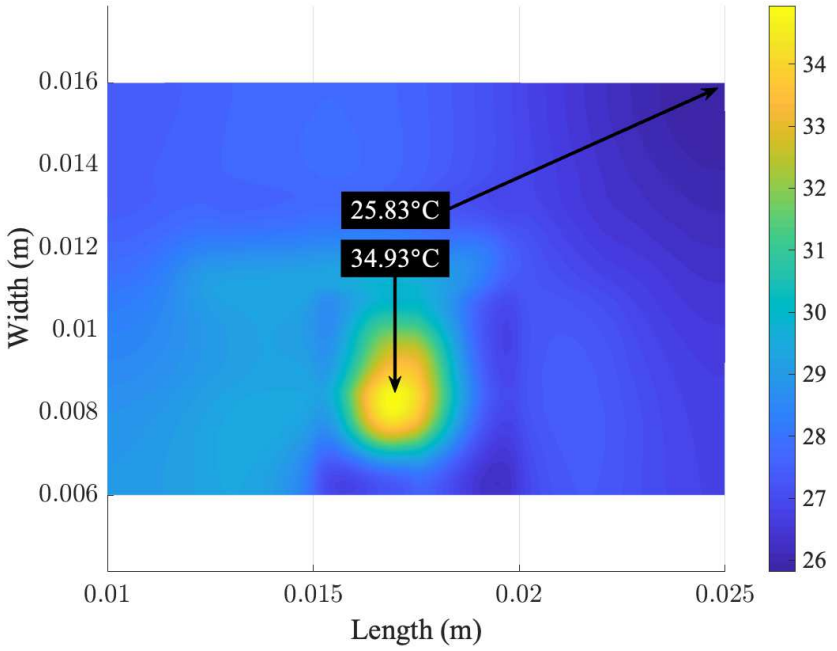


Figure 5.12: 2D Thermal-map (temperature in °C) at timestep 53 under TEC-Array

Furthermore, we observe that TEC-Trad leads to less efficient power consumption for TEC devices as the non-hotspots are cooled as intensely as the hotspots, resulting in a high ΔT and consequently incurring a significant wastage of power used by the cooling system. This can be seen in Fig. 5.11 where the power consumption of the TEC-Array is

shown in proportion to the power consumption of the TEC-Trad (P_{Array}/P_{Trad}). The data shows that the cumulative energy consumption of the TEC-Array is 66.23% of TEC-Trad across the 190 time-steps. Hence, over the long term, implementing the proposed TEC-Array will result in significant energy savings, compared to TEC-Trad, while yielding the same benefits (i.e. T_{max}).

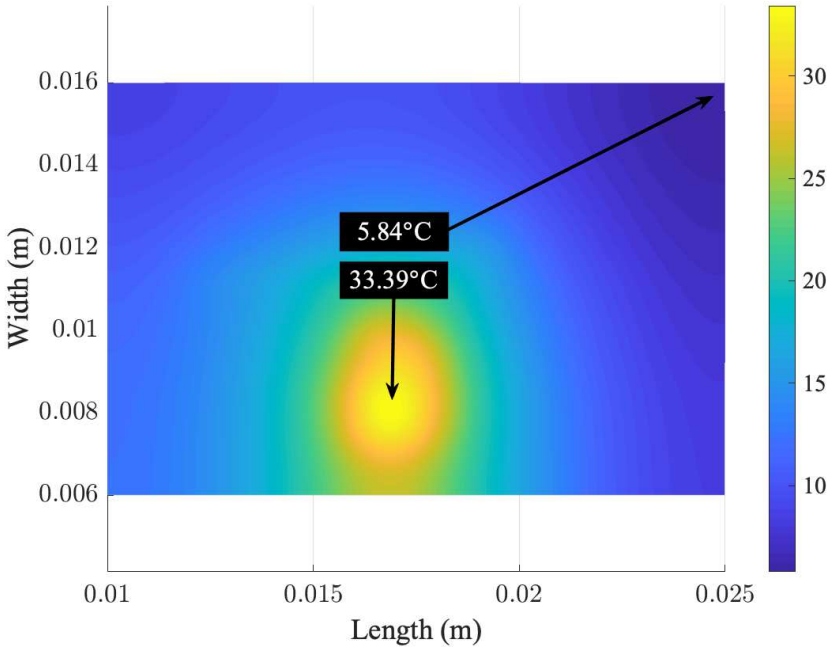


Figure 5.13: 2D Thermal-map (temperature in °C) at timestep 53 under TEC-Trad

5.5 Summary

In this chapter, we presented a general, power-density driven, thermoelectric-based active cooling solution for multi-core processors. The proposed cooling system involves a 2D array of TEC modules, called TEC-Array, that enables the ability to spatially vary

the rate of cooling across the surface of the processor die. The TEC-Array is controlled using a power-density based control scheme that allows the hot-spots to be cooled more intensely than non-hotspots. Moreover, the general nature of the proposed system allows it to be easily adopted for any commercially available processor. The numerical results on an Intel quad-core chip shows that the proposed TEC-array cooling can substantially reduce the peak temperatures compared to the traditional passive heat sink cooling method. Furthermore, compared to existing single TEC module based cooling method, the proposed method can reduce both TEC power and temperature gradients across chip under the same maximum temperature constraints. As a result, the new TEC-array cooling can enable more aggressive chip performance with increased thermal design power (TDP) while maintaining the chip design lifetime. The proposal TEC power management is also orthogonal to other power management techniques, such as DVFS, which can be used to reduce the total power of a chip.

Chapter 6

Conclusions

In this article, we reviewed the culmination of our work and shared our contributions to the areas of pre-silicon IC reliability analysis, post-silicon thermal estimation, and advanced microprocessor cooling. Specifically, in the first segment, we shared our novel structure-based approach to accelerating electromigration (EM) wear-out for the purposes of post-silicon qualification and burn-in testing (Ch .2). In the second segment of this manuscript, we presented our data-driven post-silicon approach that can be used to estimate the temperatures of all prominent hotspots on the chip (Ch .3), as well as estimate the full-chip spatial temperature distribution across the surface of the die in real time (Ch. 4). Lastly, in the third segment (Ch. 5), we showed that the estimated temperatures from the proposed model can be used to supplement the temperature information sensed from the embedded thermal sensors in order to make better informed thermal and reliability regulation decisions. Specifically, our contributions and results are summarized below.

6.1 Electromigration wear-out analysis

In Ch .2) of this manuscript, we showed that specially designed EM acceleration structures, based on the unique properties of atomic reservoir and sink segments, can be used to drastically alter the time-to-failure (TTF) of active interconnect wires. The proposed structures can be configured to achieve the desired TTF reduction during acceleration mode, while, at the same time, meet the 10+ year lifetime requirement during normal use. We demonstrated that these structures, when subjected to the traditional temperature based stressing conditions, can achieve a lifetime reduction from 10+ years down to a few hours, while staying below a $150°C$ temperature limit. This satisfies the 10^5X reduction in TTF that is typically desired for accelerated testing. The proposed method, for the first time, allows EM testing and validation to be carried out in a controlled manner without the risk of accelerating other reliability effects in the process.

6.2 Real-time heat-source temperature estimation

In Ch .3, we presented a novel method of systematically identifying all prominent heat-sources on commercial processors and deriving a dynamic thermal model to estimate the temperatures of the identified heat-sources during online use. Unlike many existing studies, this work exclusively utilizes measured data gathered directly from commercial off-the-shelf processors. Additionally, the proposed approach inherently avoids all the major obstacles faced by traditional methods that currently exist in literature, allowing it to be easily deployed by chip manufacturers and third-parties alike. Experimental results on two Intel multi-core CPUs showed that the proposed thermal model achieves very high accuracy

(root-mean-square-error: 0.55°C to 0.76°C on the Intel i5-3337U and 0.62°C to 0.93°C on the Intel i7-8650U) in estimating the temperatures of all the identified heat-sources on the two chips. These results make the proposed approach very desirable for dynamic thermal management schemes which now rely heavily on the temperature data from just the on-chip temperature sensors alone. The high spatial resolution yielded by the proposed approach can help greatly in supplementing the temperature data from the on-chip sensors, allowing for the development of more robust and smarter online thermal/power control schemes.

6.3 Real-time full-chip heatmap estimation

In Ch. 4, we proposed a machine learning based framework to real-time estimation of full-chip heatmaps for commercial microprocessors. This is an extension of the work proposed in Ch .3. In this new approach, we start with accurate spatial and temporal heatmaps measured using an advanced infrared thermal imaging system. To build the transient thermal model, we utilize temporal-aware long-short-term-memory (LSTM) networks with system-level variables such as chip frequency, voltage, and instruction counts as inputs. Instead of a pixel-wise heatmap estimation, we use 2D spatial discrete cosine transformation (DCT) on the heatmaps so that they can be expressed with just a few dominant DCT coefficients. Our study showed that only 36 DCT coefficients are required to maintain sufficient accuracy. Experimental results from this work showed that *RealMaps* can estimate the transient heatmaps with 0.9°C and 1.2°C RMSE with minimal overheads for the two commercial chips tested in this study. Compared to the state-of-the-art *pre-silicon* method, the proposed approach shows similar accuracy, but with much less computational cost.

6.4 Thermo-electric based targeted cooling

Lastly in Ch. 5, we presented a on-demand, spatially varying, thermoelectric-based active cooling solution for multi-core processors. The proposed cooling system consists of a 2D array of TEC modules, called TEC-Array, that has the ability to spatially vary the rate of cooling across the surface of the processor die. Using this approach, hotspots can be identified and cooled more intensely than non-hotspots. The numerical results on an Intel quad-core chip shows that the proposed TEC-array cooling can substantially reduce the peak temperatures compared to the traditional passive heat sink cooling method. Furthermore, compared to the existing single TEC module based cooling method, the proposed method can reduce both TEC power and temperature gradients across the chip under the same maximum temperature constraints. As a result, the new TEC-array cooling can enable more aggressive chip performance with increased thermal design power (TDP) while maintaining the chip design lifetime. The proposal TEC power management is also orthogonal to other power management techniques, such as DVFS, which can be used to reduce the total power of a chip.

Bibliography

[1] S. X.-D. Tan, H. Amrouch, T. Kim, Z. Sun, C. Cook, and J. Henkel, "Recent advances in EM and BTI induced reliability modeling, analysis and optimization," *Integration, the VLSI Journal*, vol. 60, pp. 132–152, Jan. 2018.

[2] L. Zhang, *Effects of Scaling and Grain Structure on Electromigration Reliability of Cu Interconnects*. PhD thesis, University of Texas at Austin, 2010.

[3] M. Lin and A. Oates, "An electromigration failure distribution model for short-length conductors incorporating passive sinks/reservoirs," *IEEE Transactions on Device and Materials Reliability*, vol. 13, pp. 322–326, March 2013.

[4] J. Ranieri, A. Vincenzi, A. Chebira, D. Atienza, and M. Vetterli, "Eigenmaps: Algorithms for optimal thermal maps extraction and sensor placement on multicore processors," in *Proceedings of the 49th Annual Design Automation Conference*, DAC '12, (New York, NY, USA), pp. 636–641, ACM, 2012.

[5] J. Zhang, S. Sadiqbatcha, M. O'Dea, H. Amrouch, and S. X.-D. Tan, "Full-chip power density and thermal map characterization for commercial microprocessors under heat sink cooling," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2021.

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

[7] S. Sadiqbatcha, Z. Sun, and S. X. . Tan, "Accelerating electromigration aging: Fast failure detection for nanometer ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 4, pp. 885–894, 2020.

[8] "International technology roadmap for semiconductors (ITRS)," 2015. http://www.itrs2.net/itrs-reports.html.

[9] H. Stork, "Electrified driving experience - expectations on automotive semiconductors," in *2017 IEEE International Integrated Reliability Workshop (IIRW)*, IEEE, 2017. Keynote speech.

[10] "Ansys totem." https://www.ansys.com/products/semiconductors/ansys-totem.

[11] "Ansys redhawk." https://www.ansys.com/products/semiconductors/ansys-redhawk.

[12] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," *Micro, IEEE*, vol. 32, pp. 122–134, May 2012.

[13] M. Taylor, "A landscape of the new dark silicon design regime," *IEEE/ACM International Symposium on Microarchitecture*, vol. 33, pp. 8–19, October 2013.

[14] K.Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *Proc. Intl. Symp. on Computer Architecture*, 2006.

[15] J. Kong, S. W. Chung, and K. Skadron, "Recent thermal management techniques for microprocessors," *ACM Comput. Surv.*, vol. 44, pp. 13:1–13:42, jun 2012.

[16] S. Sadiqbatcha, H. Zhao, H. Amrouch, J. Henkel, and S. X.-D. Tan, "Hot spot identification and system parameterized thermal modeling for multi-core processors through infrared thermal imaging," in *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2019.

[17] R. Cochran and S. Reda, "Spectral techniques for high-resolution thermal characterization with limited sensor data," in *Proc. Design Automation Conf. (DAC)*, pp. 478–483, 2009.

[18] S. Reda, R. Cochran, and A. N. Nowroz, "Improved thermal tracking for processors using hard and soft sensor allocation techniques," *IEEE Transactions on Computers*, vol. 60, pp. 841–851, June 2011.

[19] M. Ohring, *Reliability and Failure of Electronic Materials and Devices*. San Diego: Academic Press, 1998.

[20] D. Wolpert and P. Ampadu, *Temperature Effects in Semiconductors*, pp. 15–33. New York, NY: Springer New York, 2012.

[21] J. R. Black, "Electromigration-A Brief Survey and Some Recent Results," *IEEE Trans. on Electron Devices*, vol. 16, pp. 338–347, Apr. 1969.

[22] I. A. Blech, "Electromigration in thin aluminum films on titanium nitride," *Journal of Applied Physics*, vol. 47, no. 4, pp. 1203–1208, 1976.

[23] 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 *IEEE Int. Reliability Physics Symposium (IRPS)*, pp. 2C.1.1–2C.1.6, 2013.

[24] S. X.-D. Tan, M. Tahoori, T. Kim, S. Wang, Z. Sun, and S. Kiamehr, *VLSI Systems Long-Term Reliability – Modeling, Simulation and Optimization*. Springer Publishing, 2019.

[25] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," in *Proc. IEEE Int. Symp. on High-Performance Computer Architecture (HPCA)*, pp. 171–182, Jan. 2001.

[26] V. Hanumaiah and S. Vrudhula, "Energy-efficient operation of multicore processors by DVFS, task migration, and active cooling," *IEEE Trans. on Computers*, vol. 63, pp. 349–360, February 2014.

[27] Z. Liu, S. X.-D. Tan, X. Huang, and H. Wang, "Task migrations for distributed thermal management considering transient effects," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 2, pp. 397–401, 2015.

[28] H. Wang, J. Ma, S. X.-D. Tan, C. Zhang, H. Tang, K. Huang, and Z. Zhang, "Hierarchical dynamic thermal management method for high-performance many-core microprocessors," *ACM Trans. on Design Automation of Electronics Systems*, vol. 22, pp. 1:1–1:21, July 2016.

[29] H. Amrouch and J. Henkel, "Lucid infrared thermography of thermally-constrained processors," in *2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 347–352, July 2015.

[30] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 14, pp. 501–513, May 2006.

[31] Y. Yang, Z. P. Gu, C. Zhu, R. P. Dick, and L. Shang, "ISAC: Integrated space and time adaptive chip-package thermal analysis," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 1, pp. 86–99, 2007.

[32] H. Wang, S. X.-D. Tan, G. Liao, R. Quintanilla, and A. Gupta, "Full-chip runtime error-tolerant thermal estimation and prediction for practical thermal management," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, Nov. 2011.

[33] S. P. Gurrum, Y. K. Joshi, W. P. King, K. Ramakrishna, and M. Gall, "A compact approach to on-chip interconnect heat conduction modeling using the finite element method," *Journal of Electronic Packaging*, vol. 130, pp. 031001.1–031001.8, September 2008.

[34] Y. C. Gerstenmaier and G. Wachutka, "Rigorous model and network for transient thermal problems," *Microelectronics Journal*, vol. 33, pp. 719–725, September 2002.

[35] D. Li, S. X.-D. Tan, E. H. Pacheco, and M. Tirumala, "Parameterized architecture-level dynamic thermal models for multicore microprocessors," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 15, no. 2, pp. 1–22, 2010.

[36] 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.

[37] Z. Liu, S. X.-D. Tan, H. Wang, Y. Hua, and A. Gupta, "Compact thermal modeling for packaged microprocessor design with practical power maps," *Integration, the VLSI Journal*, vol. 47, January 2014. in press, online access: http://www.sciencedirect.com/science/article/pii/S0167926013000412.

[38] W. Wu, L. Jin, J. Yang, P. Liu, and S. X.-D. Tan, "Efficient power modeling and software thermal sensing for runtime temperature monitoring," *ACM Trans. on Design Automation of Electronics Systems*, vol. 12, no. 3, pp. 1–29, 2007.

[39] K. Dev, A. N. Nowroz, and S. Reda, "Power mapping and modeling of multi-core processors," in *International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 39–44, Sept 2013.

[40] J. Zhang, S. Sadiqbatcha, Y. Gao, M. O'Dea, N. Yu, and S. X.-D. Tan, "Hat-drl: Hotspot-aware task mapping for lifetime improvement of multicore system using deep reinforcement learning," in *Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD*, MLCAD '20, (New York, NY, USA), p. 77–82, Association for Computing Machinery, 2020.

[41] M. J. Dousti and M. Pedram, "Power-aware deployment and control of forced-convection and thermoelectric coolers," in *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2014.

[42] S. Jayakumar and S. Reda, "Making sense of thermoelectrics for processor thermal management and energy harvesting," in *2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 31–36, 2015.

[43] J. Long, S. Ogrenci Memik, and M. Grayson, "Optimization of an on-chip active cooling system based on thin-film thermoelectric coolers," in *2010 Design, Automation Test in Europe Conference Exhibition (DATE 2010)*, pp. 117–122, 2010.

[44] M. J. Dousti and M. Pedram, "Power-efficient control of thermoelectric coolers considering distributed hot spots," in *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 966–971, 2015.

[45] H. Amrouch, G. Zervakis, S. Salamin, H. Kattan, I. Anagnostopoulos, and J. Henkel, "Npu thermal management," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 11, pp. 3842–3855, 2020.

[46] C. Lundquist and V. Carey, "Microprocessor-based adaptive thermal control for an air-cooled computer cpu module," in *Seventeenth Annual IEEE Semiconductor Thermal Measurement and Management Symposium (Cat. No.01CH37189)*, pp. 168–173, 2001.

[47] X. Huang, T. Yu, V. Sukharev, and S. X.-D. Tan, "Physics-based Electromigration Assessment for Power Grid Networks," in *Proceedings of the 51st Design Automation Conference*, DAC '14, (New York, NY), pp. 1–6, ACM Press, Jun. 2014.

[48] V. Sukharev, X. Huang, H.-B. Chen, and S. X.-D. Tan, "IR-drop based electromigration assessment: parametric failure chip-scale analysis," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, pp. 428–433, IEEE, Nov. 2014.

[49] H. Chen, S. X.-D. Tan, X. Huang, T. Kim, and V. Sukharev, "Analytical modeling and characterization of electromigration effects for multibranch interconnect trees," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 11, pp. 1811–1824, 2016.

[50] X. Huang, A. Kteyan, S. X.-D. Tan, and V. Sukharev, "Physics-Based Electromigration Models and Full-Chip Assessment for Power Grid Networks," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, pp. 1848–1861, Nov. 2016.

[51] V. Mishra and S. S. Sapatnekar, "Predicting Electromigration Mortality Under Temperature and Product Lifetime Specifications," in *Proc. Design Automation Conf. (DAC)*, pp. 1–6, Jun. 2016.

[52] Z. Sun, E. Demircan, M. D. Shroff, T. Kim, X. Huang, and S. X.-D. Tan, "Voltage-Based Electromigration Immortality Check for General Multi-Branch Interconnects," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, pp. 1–7, Nov. 2016.

[53] X. Huang, V. Sukharev, T. Kim, and S. X.-D. Tan, "Electromigration recovery modeling and analysis under time-depdendent current and temperature stressing," in *Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, pp. 244–249, IEEE, Jan. 2016.

[54] X. Huang, V. Sukharev, and S. X.-D. Tan, "Dynamic electromigration modeling for transient stress evolution and recovery under time-dependent current and temperature stressing," *Integration, the VLSI Journal*, vol. 55, pp. 307–315, September 2016.

[55] S. Chatterjee, V. Sukharev, and F. N. Najm, "Power grid electromigration checking using physics-based models," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, pp. 1317–1330, July 2018.

[56] Z. Sun, E. Demircan, M. D. Shroff, C. Cook, and S. X.-D. Tan, "Fast Electromigration Immortality Analysis for Multisegment Copper Interconnect Wires," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, pp. 3137–3150, Dec. 2018.

[57] H. Zhao and S. X.-D. Tan, "Postvoiding fem analysis for electromigration failure characterization," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 26, pp. 2483–2493, Nov. 2018.

[58] C. Cook, Z. Sun, E. Demircan, M. D. Shroff, and S. X.-D. Tan, "Fast electromigration stress evolution analysis for interconnect trees using krylov subspace method," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 26, pp. 969–980, May 2018.

[59] R. De Orio, H. Ceric, and S. Selberherr, "Physically based models of electromigration: From black's equation to modern tcad models," *Microelectronics Reliability*, vol. 50, no. 6, pp. 775–789, 2010.

[60] W. K. Meyer, *Electromigration of Damascene copper for IC interconnect*. PhD thesis, Oregon Health and Science University, April 2004.

[61] V. Sukharev, A. Kteyan, and E. Zschech, "Physics-Based Models for EM and SM Simulation in Three-Dimensional IC Structures," *IEEE Trans. on Device and Materials Reliability*, vol. 12, no. 2, pp. 272–284, 2012.

[62] V. Sukharev, A. Kteyan, and X. Huang, "Postvoiding stress evolution in confined metal lines," *IEEE Transactions on Device and Materials Reliability*, vol. 16, no. 1, pp. 50–60, 2016.

[63] H.-B. Chen, S. X.-D. Tan, J. Peng, T. Kim, and J. Chen, "Analytical modeling of electromigration failure for vlsi interconnect tree considering temperature and segment length effects," *IEEE Transaction on Device and Materials Reliability (T-DMR)*, vol. 17, no. 4, pp. 653–666, 2017.

[64] S. Chatterjee, V. Sukharev, and F. N. Najm, "Power Grid Electromigration Checking Using Physics-Based Models," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, pp. 1317–1330, July 2018.

[65] M. A. Korhonen, P. Bo/rgesen, K. N. Tu, and C.-Y. Li, "Stress evolution due to electromigration in confined metal lines," *Journal of Applied Physics*, vol. 73, no. 8, pp. 3790–3799, 1993.

[66] S. Wang, Z. Sun, Y. Cheng, S. X.-D. Tan, and M. Tahoori, "Leveraging recovery effect to reduce electromigration degradation in power/ground TSV," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, pp. 811–818, IEEE, Nov. 2017.

[67] J. R. Lloyd, "New models for interconnect failure in advanced IC technology," *Physical and Failure Analysis of Integrated Circuits, 2008. IPFA 2008. 15th International Symposium on the*, pp. 1–7, 2008.

[68] C.-K. Hu, D. Canaperi, S. T. Chen, L. M. Gignac, B. Herbst, S. Kaldor, M. Krishnan, E. Liniger, D. L. Rath, D. Restaino, R. Rosenberg, J. Rubino, S.-C. Seo, A. Simon, S. Smith, and W.-T. Tseng, "Effects of overlayers on electromigration reliability improvement for cu/low k interconnects," in *Reliability Physics Symposium Proceedings, 2004. 42nd Annual. 2004 IEEE International*, pp. 222–228, IEEE, 2004.

[69] Z. Sun, S. Sadiqbatcha, H. Zhao, and S. X.-D. Tan, "Accelerating Electromigration Aging for Fast Failure Detection for Nanometer ICs," in *Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, pp. 623–630, Jan. 2018.

[70] M.-H. Lin and A. S. Oates, "Electromigration Failure Time Model of General Circuit-Like Interconnects," *IEEE Transactions on Device and Materials Reliability*, vol. 17, no. 2, pp. 381–398, 2017.

[71] Synopsys, "Synopsys University Program and Resources." `https://www.synopsys.com/community/university-program/teaching-resources.html`.

[72] S. Sadiqbatcha, C. Cook, Z. Sun, and S. X.-D. Tan, "Accelerating electromigration wear-out effects based on configurable sink-structured wires," in *Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, pp. 21–24, IEEE, July 2018.

[73] M. Witkowski, A. Oleksiak, T. Piontek, and J. Weglarz, "Practical power consumption estimation for real life hpc applications," *Future Generation Computer Systems*, vol. 29, pp. 208–217, 2013.

[74] M. J. Walker, S. Diestelhorst, A. Hansson, A. K. Das, S. Yang, B. M. Al-Hashimi, and G. V. Merrett, "Accurate and stable run-time power modeling for mobile and embedded cpus," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, pp. 106–119, Jan 2017.

[75] F. Pittino, F. Beneventi, A. Bartolini, and L. Benini, "A scalable framework for online power modelling of high-performance computing nodes in production," *2018 International Conference on High Performance Computing Simulation (HPCS)*, pp. 300–307, July 2018.

[76] R. Diversi, A. Tilli, A. Bartolini, F. Beneventi, and L. Benini, "Bias-compensated least squares identification of distributed thermal models for many-core systems-on-chip," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, pp. 2663–2676, Sep. 2014.

[77] R. Cochran and S. Reda, "Consistent runtime thermal prediction and control through workload phase detection," in *Proc. Design Automation Conf. (DAC)*, pp. 62–67, 2010.

[78] A. Bartolini, R. Diversi, D. Cesarini, and F. Beneventi, "Self-aware thermal management for high-performance computing processors," *IEEE Design Test*, vol. 35, pp. 28–35, Oct 2018.

[79] F. Beneventi, A. Bartolini, P. Vivet, and L. Benini, "Thermal analysis and interpolation techniques for a logic + wideio stacked dram test chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, pp. 623–636, April 2016.

[80] R. Cochran, A. Nowroz, and S. Reda, "Post-silicon power characterization using thermal infrared emissions," in *Proc. Int. Symp. on Low Power Electronics and Design (ISLPED)*, pp. 331–336, 2010.

[81] S. Reda, K. Dev, and A. Belouchrani, "Blind identification of thermal models and power sources from thermal measurements," *IEEE Sensors Journal*, vol. 18, pp. 680–691, Jan 2018.

[82] Thermonamic, "Thermoelectric module tec1-12710." `http://www.thermonamic.com/TEC1-12710-English.pdf`.

[83] Intel, "Technical resources: Intel core processors." `https://www.intel.com/content/www/us/en/products/docs/processors/core/core-technical-resources.html`.

[84] F. P. Incropera and D. P. DeWitt, *Fundamentals of Heat and Mass Transfer*. New York: John Wiley & Sons, 5th ed., 2002.

[85] Intel, "Intel Performance Counter Monitor (PCM)." `https://software.intel.com/en-us/articles/intel-performance-counter-monitor`.

[86] K. . Lee and K. Skadron, "Using performance counters for runtime temperature sensing in high-performance processors," in *19th IEEE International Parallel and Distributed Processing Symposium*, pp. 8 pp.–, April 2005.

[87] J. S. Lee, K. Skadron, and S. W. Chung, "Predictive temperature-aware dvfs," *IEEE Transactions on Computers*, vol. 59, pp. 127–133, Jan 2010.

[88] H. Wang, S. X.-D. Tan, S. Swarup, and X. Liu, "A power-driven thermal sensor placement algorithm for dynamic thermal management," in *Proc. Design, Automation and Test In Europe Conf. (DATE)*, pp. 1215–1220, March 2013.

[89] M. Dixon, P. Hammarlund, S. Jourdan, and R. Singhal, "The next-generation intel core microarchitecture," *Intel Technology Journal*, vol. 14, no. 3, 2010.

[90] E. Intel, "Speedstep® technology for the intel® pentium® m processor," *White Paper, Intel*, 2004.

[91] R. Bracewell, "Pentagram notation for cross correlation. the fourier transform and its applications," *New York: McGraw-Hill*, vol. 46, p. 243, 1965.

[92] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[93] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016. `http://www.deeplearningbook.org`.

[94] Phoronix, "Open-Source, Automated Benchmarking." `https://www.phoronix-test-suite.com/`.

[95] A. Nowroz, R. Cochran, and S. Reda, "Thermal monitoring of real processors: Techniques for sensor allocation and full characterization," in *Proc. Design Automation Conf. (DAC)*, 2010.

[96] X. Li, X. Li, W. Jiang, and W. Zhou, "Optimising thermal sensor placement and thermal maps reconstruction for microprocessors using simulated annealing algorithm based on pca," *IET Circuits, Devices Systems*, vol. 10, no. 6, pp. 463–472, 2016.

[97] J. L. Greathouse and G. H. Loh, "Machine learning for performance and power modeling of heterogeneous systems," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–6, IEEE, 2018.

[98] R. G. Kim, J. R. Doppa, and P. P. Pande, "Machine learning for design space exploration and optimization of manycore systems," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–6, IEEE, 2018.

[99] K. Zhang, A. Guliani, S. Ogrenci-Memik, G. Memik, K. Yoshii, R. Sankaran, and P. Beckman, "Machine learning-based temperature prediction for runtime thermal management across system components," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, pp. 405–419, Feb 2018.

[100] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[101] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, pp. 90–93, Jan 1974.

[102] AMD, "AMD uProf." `https://developer.amd.com/amd-uprof/`.

[103] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark suite: Characterization and architectural implications," in *International Conference on Parallel Architectures and Compilation Techniques (PACT)*, 2008.

[104] http://www.spec.org/cpu2000/CFP2000/.

[105] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[106] Y. Zhang, B. Shi, and A. Srivastava, "Statistical framework for designing on-chip thermal sensing infrastructure in nanoscale systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 2, pp. 270–279, 2014.

[107] H. Zhou, X. Li, C. Cher, E. Kursun, H. Qian, and S. Yao, "An information-theoretic framework for optimal temperature sensor allocation and full-chip thermal monitoring," in *DAC Design Automation Conference 2012*, pp. 642–647, 2012.

[108] D. M. Rowe, *Thermoelectrics Handbook: Macro to Nano.* Boca Raton, FL, USA: CRC Press, 2005.

[109] E. E. Antonova and D. C. Looman, "Finite elements for thermoelectric device analysis in ansys," in *Proc. 24th Int. Conf. Thermoelectrics (ICT)*, pp. 215–218, 2005.

[110] R. A. Kishore, A. Nozariasbmarz, B. Poudel, M. Sanghadasa, and S. Priya, "Ultra-high performance wearable thermoelectric coolers with less materials," *Nat. Commun.*, vol. 10, pp. 1–13, Apr. 2019.

[111] I. Chowdhury, R. Prasher, K. Lofgreen, G. Chrysler, S. Narasimhan, R. Mahajan, D. Koester, R. Alley, and R. Venkatasubramanian, "On-chip cooling by superlattice-based thin-film thermoelectrics," *Nat. Nanotechnol.*, vol. 4, pp. 235–238, Jan. 2009.

[112] S. Reda, K. Dev, and A. Belouchrani, "Blind identification of thermal models and power sources from thermal measurements," *IEEE Sensors Journal*, vol. 18, no. 2, pp. 680–691, 2018.

[113] M. Jaegle, "Multiphysics simulation of thermoelectric system—modeling of peltier cooling and thermoelectric generator," in *Proc. COMSOL Conf.*, pp. 4–6, 2008.

[114] C. Goupil, *Continuum theory and modeling of thermoelectric elements*. Weinheim, Germany: Wiley-VCH, 2015.

[115] Y. Shi, Y. Wang, D. Mei, and Z. Chen, "Numerical modeling of the performance of thermoelectric module with polydimethylsiloxane encapsulation," *Int. J. Energy Res.*, vol. 42, pp. 1287–1297, 2018.

[116] S. Sadiqbatcha, Y. Zhao, J. Zhang, H. Amrouch, J. Henkel, and S. X. D. Tan, "Machine learning based online full-chip heatmap estimation," in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 229–234, 2020.

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

[118] A. Abbasinasab and M. Marek-Sadowska, "RAIN: A tool for reliability assessment of interconnect networks—physics to software," in *Proc. Design Automation Conf. (DAC)*, (New York, NY, USA), pp. 133:1–133:6, ACM, 2018.

[119] M. Kavousi, L. Chen, and S. X.-D. Tan, "Electromigration Immortality Check considering Joule Heating Effect for Multisegment Wires," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, pp. 1–8, 2020.