BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Dissertation

IMPROVING PROCESSOR EFFICIENCY THROUGH THERMAL MODELING AND RUNTIME MANAGEMENT OF HYBRID COOLING STRATEGIES

by

FULYA KAPLAN

B.S., Middle East Technical University, 2011

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2017

Approved by

First Reader

Ayse K. Coskun, Ph.D.
Associate Professor of Electrical and Computer Engineering

Second Reader

Ajay Joshi, Ph.D.
Associate Professor of Electrical and Computer Engineering

Third Reader

Martin C. Herbordt, Ph.D.
Professor of Electrical and Computer Engineering

Fourth Reader

Sherief Reda, Ph.D.
Associate Professor of Engineering and Computer Science
Brown University

Fifth Reader

Wayne Burleson, Ph.D.
Professor of Electrical and Computer Engineering
University of Massachusetts, Amherst

If you change nothing, nothing will change.

Tony Robbins

Acknowledgments

First and foremost, I would like to thank my PhD advisor, Ayse K. Coskun, for her guidance and endless support. Her mentorship has been invaluable for me both as a PhD student and as a young woman starting her career in computer engineering. I appreciate her time, contributions, generosity, and her responsiveness to my needs. Her advice on research and many other aspects of success made my PhD experience productive and made me a more confident person. Throughout my PhD, I had the opportunity to interact and collaborate with many inspirational researchers and gained priceless work experience. All of these would not have been possible without her help and encouragement.

I would like to thank Prof. Martin Herbordt and Prof. Ajay Joshi for participating in my thesis committee. I appreciate their valuable time and feedback. I would also like to thank Prof. Wayne Burleson, for his insightful comments as a member of my thesis committee and my supervisor during my internship at Advanced Micro Devices, Inc. I extend special thanks to Prof. Sherief Reda for the stimulating discussions and his precious feedback as a research collaborator and a member of my thesis committee.

I thank Dr. Vitus J. Leung for his guidance during my internship at Sandia National Laboratories and during our continued collaboration afterwards. I would like to express my appreciation to Prof. David Atienza for granting me the internship opportunity in his research group at EPFL.

I am grateful to all my collaborators and co-authors for their productive collaboration: Manish Arora at Advanced Micro Devices, Inc., Prof. Dean Tullsen at the University of California, San Diego, Prof. Houman Homayoun at George Mason University, Scott K. Hemmert at Sandia National Laboratories, Prof. Gunar Shirner at Northeastern University, Dr. Marina Zapater and Artem Andreev at EPFL, Samuel Howes at Google, and Charlie De Vivero at Raytheon Integrated Defense Systems.

Former academic program managers at Boston University Electrical and Computer Engineering Department, Ms. Cali Stephens and Ms. Laura Cunningham, have supported me with their incredible managerial work and their sincere friendship. I extend my appreciation to them as well.

I am thankful to the members of the PEACLab research group. Their friendship and altruistic help were priceless in overcoming the struggles of the PhD life. I would like to especially acknowledge Dr. Jie Meng, Dr. Can Hankendi, Dr. Tiansheng Zhang, Ozan Tuncer and Onur Sahin for the collaborations and endless discussions.

I sincerely thank my best friends Su Sonia Herring and Efsun Selin Sezer for their lifelong support and companionship that exceed the limits of friendship and for helping me to become the person I am.

Finally, I would like to thank my family for making me feel unconditionally loved, valued, and supported everyday. I dedicate this dissertation to my mother, my father, and my sister.

The research that forms the basis of this dissertation has been partially funded by the NSF CAREER grant CNS-1149703 and by Advanced Micro Devices, Inc.

The contents of Chapter 3 are in part reprints of the material from the following papers:

- *Fulya Kaplan, Charlie De Vivero, Samuel Howes, Manish Arora, Houman Homayoun, Wayne Burleson, Dean Tullsen and Ayse K. Coskun. "Modeling and Analysis of Phase Change Materials for Efficient Thermal Management"*, in Proceedings of International Conference on Computer Design (ICCD), 2014.

- *Charlie De Vivero, Fulya Kaplan and Ayse K. Coskun. "Experimental Validation of a Detailed Phase Change Model on a Hardware Testbed"*, in Proceedings of ASME International Electronic Packaging Technical Conference and Exhibition (InterPack), 2015.

- *Fulya Kaplan, Sherief Reda and Ayse K. Coskun. "Fast Modeling of Liquid, Thermoelectric and Hybrid Cooling"*, in The Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITHERM), 2017.

The contents of Chapter 4 are in part reprints of the material from the paper, *Fulya Kaplan and Ayse K. Coskun. "Adaptive Sprinting: How to Get the Most Out of Phase Change Based Cooling"*, in Proceedings of International Symposium on Low Power Electronics and Design (ISLPED), 2015.

IMPROVING PROCESSOR EFFICIENCY THROUGH THERMAL MODELING AND RUNTIME MANAGEMENT OF HYBRID COOLING STRATEGIES

FULYA KAPLAN

Boston University, College of Engineering, 2017

Major Professor: Ayse K. Coskun, Ph.D.
Associate Professor of Electrical and Computer Engineering

ABSTRACT

One of the main challenges in building future high performance systems is the ability to maintain safe on-chip temperatures in presence of high power densities. Handling such high power densities necessitates novel cooling solutions that are significantly more efficient than their existing counterparts. A number of advanced cooling methods have been proposed to address the temperature problem in processors. However, tradeoffs exist between performance, cost, and efficiency of those cooling methods, and these tradeoffs depend on the target system properties. Hence, a single cooling solution satisfying optimum conditions for any arbitrary system does not exist.

This thesis claims that in order to reach exascale computing, a dramatic improvement in energy efficiency is needed, and achieving this improvement requires a temperature-centric co-design of the cooling and computing subsystems. Such co-design requires detailed system-level thermal modeling, design-time optimization, and runtime management techniques that are aware of the underlying processor architecture and application requirements. To this end, this thesis first proposes compact

thermal modeling methods to characterize the complex thermal behavior of cutting-edge cooling solutions, mainly Phase Change Material (PCM)-based cooling, liquid cooling, and thermoelectric cooling (TEC), as well as hybrid designs involving a combination of these. The proposed models are modular and they enable fast and accurate exploration of a large design space. Comparisons against multi-physics simulations and measurements on testbeds validate the accuracy of our models (resulting in less than $1°C$ error on average) and demonstrate significant reductions in simulation time (up to four orders of magnitude shorter simulation times).

This thesis then introduces temperature-aware optimization techniques to maximize energy efficiency of a given system as a whole (including computing and cooling energy). The proposed optimization techniques approach the temperature problem from various angles, tackling major sources of inefficiency. One important angle is to understand the application power and performance characteristics and to design management techniques to match them. For workloads that require short bursts of intense parallel computation, we propose using PCM-based cooling in cooperation with a novel Adaptive Sprinting technique. By tracking the PCM state and incorporating this information during runtime decisions, Adaptive Sprinting utilizes the PCM heat storage capability more efficiently, achieving 29% performance improvement compared to existing sprinting policies. In addition to the application characteristics, high heterogeneity in on-chip heat distribution is an important factor affecting efficiency. Hot spots occur on different locations of the chip with varying intensities; thus, designing a uniform cooling solution to handle worst-case hot spots significantly reduces the cooling efficiency. The hybrid cooling techniques proposed as part of this thesis address this issue by combining the strengths of different cooling methods and localizing the cooling effort over hot spots. Specifically, the thesis introduces LoCool, a cooling system optimizer that minimizes cooling power under

temperature constraints for hybrid-cooled systems using TECs and liquid cooling. Finally, the scope of this work is not limited to existing advanced cooling solutions, but it also extends to emerging technologies and their potential benefits and trade-offs. One such technology is integrated flow cell array, where fuel cells are pumped through microchannels, providing both cooling and on-chip power generation. This thesis explores a broad range of design parameters including maximum chip temperature, leakage power, and generated power for flow cell arrays in order to maximize the benefits of integrating this technology with computing systems. Through thermal modeling and runtime management techniques, and by exploring the design space of emerging cooling solutions, this thesis provides significant improvements in processor energy efficiency.

Contents

1 Introduction **1**

1.1 Problem Statement 2

1.2 Thesis Contributions 10

1.3 Organization 12

2 Background and Related Work **13**

2.1 Advanced Electronic Cooling Methods 13

2.2 Modeling Processor Temperature 17

2.2.1 Modeling Temperature of Systems with PCM 19

2.2.2 Modeling Microchannel Liquid Cooling 20

2.2.3 Modeling TEC 21

2.2.4 Modeling Hybrid Cooling 21

2.3 Thermal Management Techniques 22

2.3.1 Thermal Management on Systems with PCM 23

2.3.2 Liquid Cooling Management 25

2.3.3 TEC Device Optimization 26

2.3.4 Hybrid Cooling Management 27

2.4 Flow Cell Array Technology 28

2.5 Distinguishing Aspects from Prior Work 30

3 Modeling and Validation of Advanced Cooling Methods in Compact Thermal Simulators **34**

3.1 Modeling of Phase Change Materials 34

3.1.1 Proposed Modeling Methodology 35
3.1.2 Model Validation Using Multi-Physics Tools 38
3.1.3 Impact of PCM Properties on Temperature 42
3.1.4 Evaluation of the PCM Model on a Hardware Testbed 44
3.2 Modeling of Hybrid Cooling with TECs and Liquid Cooling 49
3.2.1 Proposed Modeling Methodology 49
3.2.2 Model Validation Using Multi-Physics Tools 54

4 Design-Time and Runtime Optimization Techniques 68
4.1 Adaptive Sprinting for Systems with PCM-based Cooling 69
4.1.1 Adaptive Sprinting Policy 70
4.1.2 Baseline Sprinting Policies 73
4.1.3 Performance Evaluation 75
4.2 Fighting Hot Spots Locally with Hybrid Cooling 80
4.2.1 Cooling Power Models 81
4.2.2 LoCool Optimization Technique 83
4.2.3 Experimental Methodology and Results 86

5 Analysis and Optimization of Systems with FCA 91
5.1 Design Space Exploration of FCA on MPSoC 94
5.1.1 Simulation Framework for FCA-based Cooling and Power Generation 94
5.1.2 Thermal Model of FCA 97
5.1.3 Experimental Methodology 98
5.1.4 Key Observations of the Analysis 102

6 Conclusions and Future Directions 110
6.1 Summary of Major Contributions 110
6.2 Future Research Directions and Open Problems 114

6.2.1 Thermal Modeling of Two-Phase Cooling and a System-Level Simulation Framework 114
6.2.2 Hybrid Cooling Optimizer for Heterogeneous Architectures . . 115
6.2.3 Optimization of Systems with Integrated FCAs 116

References **117**

Curriculum Vitae **126**

List of Tables

3.1 Maximum, mean, and standard deviation of error for the two melting models compared against COMSOL. 41

3.2 The parameters we used for the liquid microchannel and TEC models. 56

4.1 The parameters used for liquid microchannel and TEC models in hybrid cooling optimization. 83

4.2 Percent of the settings that meet the temperature constraint and out of that percent, the portion of them which provide lower cooling power than liquid cooling. N/A means liquid cooling did not meet the temperature constraint. 90

5.1 Examples of real systems for FCA integration and the corresponding power generation metrics. 109

List of Figures

3·1 Piecewise linear function for PCM specific heat capacity. Setting $c_{tr} \gg c_{ps}$ for the (T_1, T_2) interval models the phase change. 36

3·2 (a) Package layers; (b) Silicon and PCM grid cells. 36

3·3 Transient temperature comparison for two different power traces. Trace 1: square wave with 50% duty cycle; Trace 2: triangular wave with 1 sec period. 39

3·4 Scatter plot comparing the solution time for COMSOL and the proposed PCM model. 40

3·5 Percentages of time spent by the CPU within temperature ranges for 9 PCM configurations and for the case where no PCM is used. 43

3·6 IFC6410 SBC with copper box holding PCM, fitted on top of the Qualcomm Snapdragon SoC. 45

3·7 HotSpot model steady state error probability histogram. 46

3·8 Chip stack with hybrid cooling combining microchannel liquid cooling and TECs. TECs are placed on top of high heat flux areas to remove hot spots, while microchannels are used to remove the heat pumped by the TECs and the background heat. 49

3·9 (a) Solid grid cell, (b) Liquid grid cell, (c) TEC grid cell, d) Dimensions of the grid cells, (e) Connectivity of the grid cells building a chip stack. Current sources are shown only the rightmost for TEC and ceramic cells for clarity. 51

3·10 TEC device as we modeled in COMSOL. Example temperature distribution is shown for when TEC was biased at 4A current. 55
3·11 Comparison of processor layer temperature for the case without TECs and with varying heat transfer coefficient (htc) and heat flux (q) values. 57
3·12 Absolute temperature error for the case without TECs and with varying heat transfer coefficient (htc) and heat flux (q) values. 58
3·13 Comparison of processor temperature over TEC current for COMSOL and the proposed model. $htc = 10^6\ W/m^2K$ and $q = 20\ W/cm^2$. . . . 59
3·14 Comparison of the cold and hot side temperatures over TEC current for COMSOL and the proposed model. $htc = 10^6\ W/m^2K$ and $q = 20\ W/cm^2$. 60
3·15 Comparison of thermal maps corresponding to the processing layer and the TEC hot side, for COMSOL and the proposed model. $htc = 10^6\ W/m^2K$, $q = 20\ W/cm^2$ and $I = 4\ A$. 61
3·16 (a) Front view of the thin slice of chip stack we modeled for liquid cooling, (b) Side view of the chip stack as we modeled in COMSOL. . 62
3·17 Maximum processor temperature comparison for COMSOL, 3D-ICE and the proposed model for $q = 100\ W/cm^2$. 63
3·18 Comparison of the simulation speed across three simulators. As 3D-ICE does not have a TEC model, the bar is not shown. 63
3·19 Temperature difference introduced when the virtual node is placed at the bottom surface of a liquid cell. Placing the virtual node at the bottom surface results in underestimation of the temperature by up to $20°C$. 65

3·20 Comparison of the hot spot temperatures for pessimistic baseline #1 from prior work (Chowdhury et al., 2009), a more realistic baseline #2, and a system with TECs using different bias currents. 65

4·1 Proposed adaptive sprinting policy flowchart. 70

4·2 Performance, power and temperature simulation framework. 76

4·3 Running times of the benchmarks normalized to the no management (ideal) case for each application for (a) policies that are thermally-aware and (b) policies that cause significant thermal violation. 78

4·4 Percentage of thermal violation (i.e., time spent above $T_{max} = 80°C$) for each sprinting policy. 79

4·5 Average energy and EDP normalized to the no management case. . . 79

4·6 *LoCool* optimization flow. 84

4·7 Contour plot showing equal cooling power and temperature lines. . . 86

4·8 Front view of an example hybrid design combining microchannel liquid cooling and TECs. TECs are placed on top of high heat flux areas to remove hot spots, while microchannels are used to remove the heat pumped by the TECs and the background heat. 87

4·9 Total cooling power comparison of liquid and hybrid cooling for hot spot heat flux (HSHF) of 1000 W/cm^2. Results are normalized to liquid cooling at $T_{max} = 85°C$. 88

4·10 Total cooling power comparison of liquid and hybrid cooling for hot spot heat flux (HSHF) of 1300 W/cm^2. Results are normalized to liquid cooling at $T_{max} = 85°C$. Temperature constraint was not met for bars not shown. 89

5·1 Illustration of a system with integrated FCAs (Sridhar et al., 2014). . 92

5·2 Structure of the PowerCool compact electrochemical model for a discretized flow cell (Sridhar et al., 2014). 96

5·3 $FCA(\%)$ versus heat flux. Color bar shows the maximum chip temperature. 102

5·4 $NetFCA(\%)$ versus heat flux. Color bar shows the maximum chip temperature. 103

5·5 (a) P_{FCA} and (b) $NetFCA(\%)$ versus chip length at heat flux=20 W/cm^2, velocity=2.5 m/s, T_{inlet}=60°C, chip width=chip length, β=0.1, and κ=0.013. Color map shows the maximum temperature and it is the same for both plots. 104

5·6 $NetFCA(\%)$ versus maximum chip temperature for varying liquid inlet temperatures at heat flux of (a) 5 W/cm^2, (b) 20 W/cm^2, and (c) 50 W/cm^2. The other parameters are set as velocity=2.5 m/s, chip width=chip length=2.5 cm, β=0.1, and κ=0.013. 105

5·7 $NetFCA(\%)$ versus maximum chip temperature for (a) varying β when $\kappa = 0.013$ and (b) varying κ when $\beta = 0.1$ at heat flux=15 W/cm^2. For the other parameters, we use velocity=2.5 m/s, T_{inlet}=45°C, and show data corresponding to all chip lengths in the plots. 106

5·8 (a) P_{FCA} and (b) $NetFCA(\%)$ versus maximum chip temperature for varying flow velocity values at heat flux=5 W/cm^2. For the other parameters, we use chip width=chip length=2.5 cm, T_{inlet}=45°C, β=0.1, and κ=0.013. Color map is the same for both plots. 107

List of Abbreviations

CFD	.....	Computational Fluid Dynamics
CMOS	.....	Complementary Metal-Oxide-Semiconductor
CPU	.....	Central Processing Unit
CTM	.....	Compact Thermal Modeling
DRAM	.....	Dynamic Random Access Memory
DVFS	.....	Dynamic Voltage-Frequency Scaling
EDP	.....	Energy Delay Product
FCA	.....	Flow Cell Array
FEM	.....	Finite Element Method
FPGA	.....	Field-Programmable Gate Array
GMRES	.....	GPU-Accelerated Generalized Minimum Residual
GPU	.....	Graphics Processing Unit
HPC	.....	High Performance Computing
HTC	.....	Heat Transfer Coefficient
IC	.....	Integrated Circuit
L2	.....	Level Two
MPSoC	.....	Multiprocessor System on Chip
PCM	.....	Phase Change Material
RC	.....	Resistor Capacitor
TDP	.....	Thermal Design Power
TEC	.....	Thermoelectric Cooler
TSV	.....	Through-Silicon-Via
V/F	.....	Voltage/Frequency

Chapter 1

Introduction

One of the main challenges in building future high-performance processors is the requirement to alleviate high power densities. If not handled effectively, high power densities lead to elevated on-chip temperatures. High temperatures not only significantly limit energy efficiency by increasing leakage power (Lee and Groot, 2006; Kim et al., 2003), but they also degrade performance through built-in throttling mechanisms and shorten processor lifetime as a result of temperature dependent failure mechanisms (Srinivasan et al., 2003). Thus, achieving exascale computing strongly relies on designing novel electronic cooling solutions that are able to remove heat much more efficiently than the existing cooling methods.

A number of advanced electronic cooling solutions have been developed by thermo-mechanical engineers to overcome the temperature problem. Examples of advanced cooling methods include microchannel liquid cooling (Sharma et al., 2015; Gruener, 2008; Dang et al., 2010), two-phase cooling (Schultz et al., 2016; Thome, 2004), PCM-based cooling (Tan and Fok, 2007; Yoo and Joshi, 2004; Stupar et al., 2010; Raghavan et al., 2012) and TEC cooling (Chowdhury et al., 2009; Taylor and Solbrekken, 2008; Yazawa et al., 2012). However, each cooling solution comes with corresponding tradeoffs of performance, energy efficiency, area and cost. For example, while microchannel liquid cooling is a scalable promising solution for 3D-stacked systems, it may not be feasible for mobile platforms due to area and cost constraints. Another example is the TEC, which is able to handle high power densities in localized

areas, but becomes highly inefficient when used to cool down large areas. Moreover, these tradeoffs highly vary based on the target system properties and the applications running on the system. For example, the benefits of PCM-based cooling is more apparent for applications that have short periods of intense computation followed by long idle times. Hence, there is no cooling design that meets all requirements perfectly for an arbitrary computing platform.

This thesis claims that a *temperature-centric co-design* of the cooling and computing systems is a key enabler for building future high-performance computing systems. In this context, co-design refers to designing and optimizing the cooling system with the awareness of the underlying computing platform, the application characteristics, and the resulting on-chip heat distribution. Such a co-design approach requires the development of system-level thermal models that are able to characterize the temperature behavior in a fast and accurate manner. These models should be accompanied by design-time and runtime optimization algorithms that maximize system efficiency with consideration of the underlying processor architecture and application properties. This thesis enables the aforementioned co-design approach by providing compact thermal modeling of the advanced cooling methods, and by developing optimization techniques to maximize the energy efficiency of systems adopting these cooling solutions. In the next section, we describe the problems that this thesis aims to solve in more detail.

1.1 Problem Statement

A wide range of advanced cooling solutions have been and continue to being developed to tackle the thermal challenges in processors from different angles. These solutions include but are not limited to PCM-based passive cooling (Tan and Fok, 2007; Yoo and Joshi, 2004; Stupar et al., 2010; Raghavan et al., 2012), microchannel liquid

cooling (Sharma et al., 2015; Gruener, 2008; Dang et al., 2010), and TEC (Chowdhury et al., 2009; Taylor and Solbrekken, 2008; Yazawa et al., 2012), each having their own advantages and tradeoffs.

PCM is a passive cooler that can store large amounts of heat at near-constant temperature during phase transition (i.e., from solid to liquid), acting like a large thermal capacitor. Owing to its heat storage property, PCM has been commonly used in cooperation with performance boosting algorithms (Raghavan et al., 2012; Tilli et al., 2012; Raghavan et al., 2013; Shao et al., 2014). PCM is highly suitable for platforms where active cooling is not feasible due to area and power constraints. However, once fully melted, PCM requires idle duration to dissipate the stored heat and freeze back for reuse.

Liquid cooling is an active cooling solution, where a coolant fluid is pumped (using an electrical pump) through microchannels to remove heat. Liquid cooling is especially attractive for 3D-stacked processors, as interlayer cooling can be applied between the stacked layers, providing a scalable and effective solution (Sharma et al., 2015; Gruener, 2008; Dang et al., 2010; Coskun et al., 2011). However, liquid cooling introduces new challenges, such as the additional pumping power requirement and the large thermal gradients caused by the increase in fluid temperature as the liquid flows through the channels.

TECs operate based on the Peltier effect such that when electric current passes through the TEC device, heat is absorbed from one side and rejected on the other side, creating a temperature difference across the ends. The power consumption of the TEC device increases considerably when cooling large areas. At micro-scale, however, TECs are highly effective in handling high power densities, which makes them good candidates for localized cooling methods.

In addition to these cooling solutions, hybrid cooling designs, which combine the

strengths of the individual cooling solutions on the same platform, have been implemented to achieve higher cooling efficiency. Examples of such hybrid designs include TECs with air-cooled fans (Paterna and Reda, 2013; Jayakumar and Reda, 2015) or TECs incorporated on a liquid-cooled system (Yazawa et al., 2012; Sahu et al., 2012). In the former examples (Paterna and Reda, 2013; Jayakumar and Reda, 2015), TECs are used in cooperation with air cooling and dynamic voltage frequency scaling (DVFS) algorithms to achieve higher throughput for a given system power cap. In the latter case with combined liquid cooling (Yazawa et al., 2012; Sahu et al., 2012), TECs help remove high-density hot spots locally to reduce the total cooling power consumption.

Despite the variety of new advanced cooling techniques, early investigation and optimization of these cooling methods are significantly delayed due to the following reasons. It takes considerable amount of time until systems adopting such new cooling technologies become commercially available to researchers. By the time the products are available, the potential benefits from research are already left on the table. An alternative solution is to build prototypes, but prototyping new technologies also incurs monetary and engineering costs, and requires expertise in a wider range of areas, making this option unfeasible in most cases. In addition, having a few prototypes is often not sufficient to explore the design space to the desired extent.

It is possible to use commercial multi-physics simulators (e.g., COMSOL Multiphysics Software (COMSOL, 2017) or ANSYS (ANSYS, 2017)) to model a variety of cooling methods with high accuracy. However, such tools are prohibitively expensive in terms of the required simulation time and compute resources. It takes substantially long time to construct models in such tools, and at runtime, they incur long solution times as well as large memory requirements. For example, for a liquid-cooled chip, solving even a small steady-state problem corresponding to a slice

with two microchannels takes about 25 minutes in COMSOL and requires GBs of memory. For system-level analysis and optimization, however, one needs to model the whole processor stack and run longer real time simulations to understand the applications' runtime behavior. This kind of simulation takes from hours to multiple days or weeks in COMSOL. Such factors limit the use of multi-physics simulators for modeling advanced cooling.

All of these reasons indicate that it is critical to have detailed thermal models in order to estimate the processor temperature in a fast, accurate way and to enable the design and optimization of future computing systems.

A number of temperature simulators and thermal models exist, each focusing on different types of advanced cooling methods (Skadron et al., 2003b; Sridhar et al., 2010a; Ladenheim et al., 2016; Fourmigue et al., 2014; Yazawa et al., 2012; Sahu et al., 2012; Paterna and Reda, 2013; Raghavan et al., 2012; Tilli et al., 2012). These models represent the temperature behavior of the target cooling methods in a more compact way than the commercial tools in order to save time and resources. Nevertheless, the existing models are not sufficient due to three main reasons: ***(i)*** Majority of these models focus on a single cooling method and do not allow the user to explore other cooling solutions or hybrid cooling designs combining these cooling solutions on the same platform. For example, while HotSpot simulator (Skadron et al., 2003b) and Manchester Thermal Analyzer (Ladenheim et al., 2016) focus only on processors that are cooled using conventional heat spreader and heat sinks, 3D-ICE (Sridhar et al., 2010a) and ICTherm (Fourmigue et al., 2014) can model both microchannel liquid cooling and heat sinks. However, none of those simulators support modeling TECs or PCM-based cooling or hybrid solutions involving a combination of cooling methods. ***(ii)*** These models are often designed as stand-alone modules targeting a specific platform or applications and are not easily applicable to other systems. For

example, a TEC model from prior work (Paterna and Reda, 2013) is designed such that TECs units are the same size as the cores of a processor and that TECs cover the whole processor layer, not allowing the simulation of localized cooling with TECs. In another model for PCM cooling (Raghavan et al., 2012), PCM is assumed to have fixed properties and a certain location on the chip stack, and melting duration is estimated based on those specific properties. ***(iii)*** Some of the proposed thermal models (Raghavan et al., 2012; Tilli et al., 2012) rely on simplifying assumptions and cannot capture the complex temperature behavior of the corresponding cooling solution, resulting in a large modeling error. Hence, a thermal modeling infrastructure incorporating emerging cooling solutions in a single simulation environment with acceptable modularity and accuracy is currently not available.

Solely applying advanced cooling methods is not sufficient to achieve dramatic improvements in the energy efficiency. It is essential to develop temperature-aware design-time and runtime optimization techniques that consider the cooling and computing subsystems as a whole. To be more specific, when designing optimization techniques, the processor architecture, applications, physical layout, heat distribution on the chip, and efficiency requirements of the cooling technology should be considered all together. Power and performance characteristics of the workloads vary across applications and also dynamically change within an application over time. This dynamism indicates that the cooling demand of the processor also varies at runtime and fixing the operating point for the cooling system will lead to under-/over- cooling depending on the workload. Thus, runtime management techniques that can adapt to these changes are necessary for optimum operation. Moreover, for some applications, being highly responsive to short computational demand is crucial for performance, while for others, achieving sustained performance for longer durations is more important. The definition of optimum operation as well as the approach to achieve it

will significantly differ for those two cases. Hence, the runtime management policy should be aware of such application requirements. Furthermore, there are the efficiency requirements of each cooling solution that need to be considered. For example, the cooling efficiency of the TEC starts to decrease as the area and bias current are increased beyond a certain level. When designing a system with TECs for localized cooling of hot spots, it is necessary to optimize these design parameters to maximize TEC benefits. Such optimization is highly dependent on the properties of the target hot spots, i.e., their heat fluxes, locations and sizes. Therefore, the optimization algorithm should be aware of the efficiency requirements of the underlying cooling mechanism and the heat distribution on the chip.

A body of temperature-aware performance boosting techniques exist in the literature, each targeting a different source of processor inefficiency. One such technique is *computational sprinting* (Raghavan et al., 2012), which refers to exceeding the thermal design power (TDP) of a processor to respond to short bursts of intense computational demand. *Computational sprinting* activates all CPU cores of a system to take advantage of the thread-level parallelism for performance speedup. PCM has been used in cooperation with *computational sprinting* to extend sprinting duration for higher performance gains. While existing sprinting policies provide performance benefits (Raghavan et al., 2012; Raghavan et al., 2013; Tilli et al., 2012; Shao et al., 2014), there is potential for further improvement that has not been explored in prior work. Existing sprinting work treats the PCM as a single large *heat storage* unit, and assumes that this storage is consumed by the individual on-chip computing elements equally. However, we observe that PCM melts non-uniformly across the layer due to the heterogeneity in on-chip heat distribution. Thus, even if the *heat storage* capability is fully consumed on parts of the chip, there is still opportunity to continue sprinting using the rest of the chip to achieve further performance gain.

Heterogeneous heat distribution is not only important in the context of sprinting, but in general it is a considerable source of inefficiency in current processors. Even if two processors consume the same total power, maximum chip temperature of these two processors can differ significantly depending on where and with what density this power is consumed across the chip (Shafique et al., 2014). Most cooling systems are designed to remove a target amount of heat per unit area and keep the maximum chip temperature under a given constraint. However, they do not provide more or less cooling to certain locations of the chip based on the spatial variation of cooling demand. Hot spots, on the other hand, occur on different locations of the processor with varying areas and heat fluxes reaching 1-2kW/cm^2 (Schultz et al., 2016; Lu et al., 2016). Distributing the cooling effort equally across the chip leads to significant losses in cooling efficiency and causes over-/under- cooling of on-chip elements. Hybrid designs aim to address this issue by localizing the cooling effort over the hot spots and selecting the most suitable cooling methods for a given system. One such hybrid design combines TECs with liquid cooling, where TECs are used to remove high density hot spots and liquid cooling removes the lower intensity background heat (Yazawa et al., 2012; Sahu et al., 2012). Such a hybrid cooling system achieves lower hot spot temperature for a given cooling power cap, compared to a homogeneous design with liquid cooling only. Prior techniques on hybrid cooling mostly focus on optimizing the TEC device geometry (Yazawa et al., 2012; Sahu et al., 2012) and assume a fixed operation point for the liquid-cooled system. However, a generally applicable system-level optimization approach is essential to maximize the benefits of hybrid cooling systems and is not provided in prior work.

The ultimate goal of a cooling system is to efficiently remove the dissipated by the computing elements so that the processor can run at its maximum performance while maintaining reliable operation. As we mentioned earlier, the requirements for

achieving this goal is tightly coupled to the target processor architecture. In the last decade, architectural designs shifted from single-core to multicore processors in order to maintain performance scaling while preserving energy efficiency. This shift was followed by the introduction of 3D-stacked architectures, which enable stacking multiple processor and memory dies connected using through-silicon-vias (TSVs), providing lower on-chip communication latency and higher bandwidth. The benefits of 3D stacking are hindered by the heat removal problems and power delivery issues. Temperature problems escalate in 3D-stacked systems due to the additional thermal resistance introduced by vertical stacking. This brings the necessity for *scalable cooling* solutions in order to achieve the maximum potential in 3D designs. Another important challenge affecting the performance of processors, especially in 3D-stacked systems, is related to power delivery. The amount of power that can be delivered to the vertically stacked dies depends on the number of power TSVs. TSV area is limited and is shared between signal and power TSVs, constraining the computational density of the stacked layers.

In order to address the scalable cooling and power delivery challenges in computing systems, a new design concept has been recently introduced. In this concept, the cooling subsystem provides cooling of the processor and on-chip power generation simultaneously. The emerging integrated *flow cell array* (FCA) technology is a realization of this concept, where fuel cells are pumped through microchannels to provide both cooling and on-chip power generation through electrochemical reactions. FCA technology is a promising solution to the aforementioned cooling and power delivery problems faced in 3D-stacked processors and can also be applied to 2D designs to provide significant reduction in the wall-power consumption, leading to self-sustaining systems.

Recent work provides a preliminary analysis on a IBM POWER7+ platform (Srid-

har et al., 2014; Sabry et al., 2014) and demonstrates potential benefits of integrating FCAs in processors. However, the extent of the benefits and tradeoffs of FCA technology is yet to be explored. Such exploration requires consideration of the broad range of design parameters including the channel geometry, fluid flow rate, fluid inlet temperature, processor dimensions and heat flux levels, as well as the tradeoffs between the maximum chip temperature, generated power, leakage power and pumping power. This kind of thorough analysis is necessary to gain insight on how to maximize the benefits of FCA technology in future processors.

In summary, even though there exist a number of thermal models, optimization techniques and emerging technologies targeting energy efficiency, there is substantial headroom for improvement on each of these domains because (i) the existing models are not sufficient in enabling the exploration of a variety of cooling solutions together in a modular environment, (ii) there is potential for further improvement regarding the cooling optimization techniques especially through hybrid design and runtime optimization of cutting edge cooling, and (iii) the benefits and tradeoffs of the emerging technologies are yet to be explored.

1.2 Thesis Contributions

This thesis improves energy efficiency in processors through (1) devising novel thermal models for advanced hybrid cooling solutions, (2) developing design-time and runtime optimization techniques that are aware of the underlying computing and cooling subsystems, and (3) exploring the design space to maximize the benefits of emerging technologies. **Specific contributions of this thesis are as follows:**

- We propose compact thermal models for the design and evaluation of advanced cooling solutions, mainly, PCM-based cooling (Kaplan et al., 2014), TECs and microchannel liquid cooling, as well as the hybrid designs involving a combina-

tion of these. Our proposed models provide a fast and accurate way of exploring the large design space. We validate the accuracy of our models by comparing their results against multi-physics simulations (Kaplan et al., 2014) and measurements on testbeds (Vivero et al., 2015) and demonstrate less than $1°C$ error on average with up to four orders of magnitude shorter simulation times.

- In order to demonstrate the feasibility of the PCM-based cooling as well as our proposed PCM thermal model, we build a hardware testbed with a PCM unit placed on top of the processor package and experimentally validate our PCM model through measurements on the testbed (Vivero et al., 2015). Moreover, we propose a soft PCM capacity sensor to be used in cooperation with sprinting algorithms. Proposed PCM sensor estimates the remaining unmelted PCM at runtime through measurements. We show potential benefits of such PCM sensor by comparing PCM-aware policies against the ones that are oblivious to PCM state and demonstrate up to 4.5% performance improvement.

- We propose an *Adaptive Sprinting* (Kaplan and Coskun, 2015) algorithm in order to boost performance of multithreaded applications in systems with PCM-based cooling. *Adaptive sprinting* monitors the PCM state at runtime and uses this information to decide on the (i) number, (ii) location, and (iii) voltage/frequency (V/F) setting of the sprinting cores. The PCM-aware nature of the *adaptive sprinting* policy helps utilize the PCM storage capability more efficiently, leading to extended sprinting duration and 29% higher performance compared to the existing sprinting strategies.

- In order to mitigate high density hot spots more efficiently, we propose a cooling optimization algorithm, which focuses on hybrid cooling designs combining TECs and microchannel liquid cooling. The proposed cooling optimization algo-

rithm, *LoCool*, jointly minimizes the TEC and liquid pumping power for a given temperature constraint. By localizing the cooling effort over the hot spots and determining the best operating point for each of the cooling solutions, *LoCool* saves cooling energy by up to 28% compared to using liquid cooling only.

- We provide a comprehensive exploration of the architectural design space to maximize the power generation in FCA-integrated computing systems and point out target platforms that could benefit from FCAs the most. We analyze a wide range of parameters including channel geometry, fluid flow rate, fluid inlet temperature, processor dimensions, power density levels, and leakage characteristics of the processor. Our analysis provides insight on the interplay between the maximum chip temperature, leakage power, pumping power and the generated power and suggests that, for small low-power chips, up to 76% of the total system power can be generated on-chip using the FCAs. For larger processors with higher power densities, FCA can generate power (up to 60W) that is equivalent to the leakage power plus the pumping power consumption of the processor.

1.3 Organization

The rest of this thesis starts with providing the background and related work on advanced processor cooling techniques, thermal modeling and runtime management methods, and the FCA technology. Chapter 3 presents the details of the proposed thermal modeling approaches and provides validation results. In Chapter 4, we introduce our proposed optimization algorithms, i.e., *adaptive sprinting* and *LoCool*, and evaluate their performance by comparing against existing techniques. We then provide a design space exploration of FCAs in Chapter 5. Chapter 6 concludes this thesis and discusses future research directions.

Chapter 2

Background and Related Work

This thesis proposes thermal modeling of advanced cooling techniques, develops management techniques to improve the processor efficiency in hybrid cooling designs, and explores the potential benefits of the emerging FCA technology. In this chapter, we first briefly discuss selected advanced solutions in electronic cooling. We continue with a background on processor temperature modeling techniques and discuss the prior modeling approaches. We then present a detailed overview of the recent processor thermal management techniques that target systems utilizing different cooling solutions. Finally, we describe the operation principles and the existing work on FCA technology.

2.1 Advanced Electronic Cooling Methods

A number of advanced cooling methods have been proposed to address the cooling efficiency problems in modern processors. In this thesis, we focus on a subset of these cooling methods, namely TEC cooling, single-phase microchannel-based liquid cooling and PCM cooling. We select these cooling methods as their operation principles significantly differ from each other and they introduce very different tradeoffs. For example, TECs and liquid cooling are active cooling methods, while PCM is a passive cooling solution. Liquid cooling and PCM have a slower response time ranging from hundreds of milliseconds to seconds, while TECs respond within microseconds. Moreover, the selected cooling methods target distinctively diverse platforms and have

been widely studied in the research community. This section provides an overview of the existing advanced cooling solutions together with their strengths and tradeoffs.

TEC cooling is one of the emerging methods in mitigating hot spots (Chowdhury et al., 2009; Paterna and Reda, 2013; Taylor and Solbrekken, 2008; Yazawa et al., 2012). TEC is a device that works according to the Peltier principle; that is, when a bias current passes through the thermocouples, heat is absorbed on one side and rejected to the other side, creating a temperature difference. The amount of heat pumped by the TEC depends on the bias current, intrinsic material properties, as well as the temperatures of the cold and hot sides. Recently, the use of superlattice-based thin-film thermoelectrics has been proposed owing to their high heat pumping capabilities reaching $1300 W/cm^2$ (Chowdhury et al., 2009). Superlattice-based TEC devices are composed of ultrathin (5-10um) Bi_2Te_3-based p-n thermocouples sandwiched between thin ceramic plates. TEC devices are compatible with silicon manufacturing process (Chowdhury et al., 2009; Sahu et al., 2015), which makes them promising solutions to target hot spots at micro-scale. TECs can mitigate localized high density hot spots efficiently; however, they consume considerable amount of cooling power when cooling down large areas.

Recently, the use of PCM has been explored as a passive cooling solution (Raghavan et al., 2012; Tilli et al., 2012; Alawadhi and Amon, 2003; Tan and Fok, 2007; Yoo and Joshi, 2004; Stupar et al., 2010). PCMs store large amounts of heat during phase change (e.g., from solid to liquid) at near-constant temperature; thus, they act like large thermal buffers and delay the rise of temperature. PCM-based cooling is attractive for systems where active cooling methods may not be feasible due to area and power constraints, such as mobile platforms.

Another type of advanced cooling solution is liquid cooling, which can be performed by attaching a cold plate with built-in microchannels on the back of the

processor (i.e., back-side liquid cooling) or by fabricating microchannels between the layers of the chip (i.e., embedded liquid cooling). A coolant fluid is then pumped through the channels to remove the heat. Embedded microchannel liquid cooling has become an attractive solution to overcome temperature problems in 3D-stacked architectures due to the higher heat removal capability of liquids in comparison to air (Sharma et al., 2015; Gruener, 2008; Dang et al., 2010; Coskun et al., 2011). In addition, the heat removal ability of this interlayer cooling approach scales with the number of stacked layers. Current technology allows fabricating the infrastructure to enable interlayer liquid cooling. IBM Zurich Research Laboratory has built a 3D chip that uses microchannel liquid cooling (Gruener, 2008). Their cooling system can remove heat at a rate of $180W/cm^2$ per layer through $50\mu m$ wide channels from a stack with $4cm^2$ footprint.

Embedded microchannel liquid cooling introduces additional complexity during the microchannel etching and the bonding phases, which translates to around 20% additional manufacturing cost compared to a design without microchannels (Coskun et al., 2011). On the other hand, it has been shown that embedded liquid cooling provides a much higher cooling efficiency in comparison to back-side liquid cooling in both 2D and 3D systems (Brunschwiler et al., 2015; Yueh et al., 2015; Sahu et al., 2015). Brunschwiler *et al.* compares the cooling performance of three liquid cooling designs: (i) back-side cooling with a lid attached between the cold plate and the chip, (ii) back-side cooling with integrated direct-attached cold plate, and (iii) embedded liquid cooling. They compare the thermal gradient from fluid inlet to the maximum junction temperature and show that the direct-attached cold plate decreases the gradient from $120°C$ to $80°C$, and embedded liquid cooling further reduces it to $50°C$. Embedded cooling achieves better cooling as it provides lower thermal resistance by eliminating additional contact materials, increases the surface

area for heat transfer and brings the liquid closer to the heat source. Another benefit of embedded cooling is that it reduces the footprint of the cooling system and provides a scalable solution for 3D-stacked architectures. Recent work compares in-package and external microfluidic cooling experimentally on a mobile platform and shows that in-package cooling can increase the cooling performance per volume by almost two orders of magnitude (Yueh et al., 2015). Sahu *et al.* demonstrate the benefits of on-chip microchannel cooling over the off-chip configuration experimentally on a hybrid cooling system that combines liquid cooling and TECs (Sahu et al., 2015). They show that the on-chip configuration provides more than twice the cooling compared to the off-chip design as it reduces the parasitic heat transfer to the TEC device.

Liquid cooling brings new challenges with it, such as large on-chip thermal gradients created by the fluid temperature increase and the additional power required by the pump. As the fluid flows through the microchannel, it absorbs heat from the processor and gets hotter, resulting in higher temperatures at locations closer to the outlet. Increasing the liquid flow rate can reduce thermal gradients; however, required pumping power quadratically increases with flow rate and also, the maximum flow rate is limited by the maximum pressure drop for safe operation of the system.

Two-phase cooling aims to address some of these limitations of the single-phase liquid cooling. Examples of two-phase cooling methods include two-phase microchannel cooling (Schultz et al., 2016; Thome, 2004), thin film evaporation (Zhu et al., 2016), and nanoporous evaporation (Lu et al., 2015). In two-phase microchannel cooling, the coolant fluid evaporates as it flows through the channel, absorbing large amounts of heat. In nanoporous evaporators (Lu et al., 2015), the working fluid is delivered across microchannels and is drawn in through the manifolds towards the heated surface via capillary forces using a thin nanoporous membrane. Subsequently, the vapor generated by evaporation exits through the membrane and is guided to an external

condenser where the liquid is recirculated back to the pump.

A few other cooling methods that are worth mentioning for the sake of completeness are as follows. Under the family of liquid cooling, traditional microchannel heat sinks (Lee et al., 2005) and manifold microchannel (MMC) heat sinks (Sharma et al., 2013; Escher et al., 2010) are included. MMC heat sinks consist of embedded microchannels and a manifold layer above that involves multiple inlet and outlet ports, providing lower overall pressure drop and higher thermal efficiency. Another category of cooling is using impingement jets (Kandlikar and Bapat, 2007), which can be either air-powered or use some kind of fluid, such as water. High speed jet impingement on a component surface creates a thin boundary layer, and thus, provides a high heat transfer. Heat pipe is another cooling solution (Xie et al., 1998), where the working fluid inside the pipe absorbs heat from a thermally conductive surface and turns into vapor. The vapor then travels to the cold interface and condenses back. Heat pipes in modern computer systems are used to move heat away from separate components on a larger medium, such as inside a laptop case. In the next section, we will discuss the existing methods that have been used to model these advanced cooling solutions.

2.2 Modeling Processor Temperature

Thermal modeling is essential for the design and evaluation of the current and future cooling systems. There are two commonly used approaches for chip-level thermal modeling. The first approach is finite element method (FEM), which is used in computational fluid dynamics (CFD) software such as COMSOL (COMSOL, 2017). FEM divides a chip into many small subdomains and uses variational methods to model the thermal conditions of the whole chip (Reddy, 1993). This method provides high accuracy, however, it is time-consuming (requiring many hours to days of simulation time for large systems) and computationally-intensive and thus, it is not suitable for

system-level simulations.

The second approach is compact thermal modeling (i.e., also adopted by simulators such as HotSpot (Skadron et al., 2003b)), which models the chip as a thermal Resistor Capacitor (RC) network. In the RC network, R stands for thermal resistance and C represents thermal capacitance. Current flowing through R represents heat flow, while C models the transient behavior of temperature. Solving the thermal RC network for a given processor power distribution gives the temperature of each node (Skadron et al., 2003b; Coskun et al., 2009b; Sridhar et al., 2010a; Sridhar et al., 2010b; Fourmigue et al., 2014; Liu et al., 2012). Compact thermal modeling approach trades off some accuracy for considerable reduction in simulation time and is suitable for design-time thermal analysis. Within compact thermal models, various solution methods have been proposed for improving the simulation efficiency even further (Sridhar et al., 2010b; Fourmigue et al., 2014; Liu et al., 2012; Ladenheim et al., 2016). For example, Zanini *et al.* (Zanini et al., 2009) propose a novel matrix state-space compatible representation of the processor thermal behavior. Using this representation together with adaptable ordinary differential equation solvers, this work examines the tradeoffs between accuracy and simulation time under various runtime conditions. ICTherm (Fourmigue et al., 2014) simulator provides an alternate solver that is second-order accurate in time, unconditionally stable, and has linear-time complexity. It also provides a parallel and scalable implementation. Manchester Thermal Analyzer (MTA) (Ladenheim et al., 2016) provides fully adaptive spatio-temporal mesh refinement features for improved accuracy and computational efficiency. It also solves the linear systems using a multigrid iterative method, which gives superior performance for 3D transient analysis.

Commercial tools based on FEM such as COMSOL and ANSYS are commonly used to verify the accuracy of compact modeling techniques (Skadron et al., 2003b;

Sridhar et al., 2010a). While these tools provide a well representation of the temperature behavior in absence of real testbeds, it should be noted that the accuracy of a COMSOL or ANSYS model depends on how it is setup, the assumed parameters, as well as the solver settings. Thus, researchers usually validate the feasibility of the FEM model setup based on some measurement data when available, and then they verify the accuracy of their compact model by comparing against the FEM model.

2.2.1 Modeling Temperature of Systems with PCM

Various methods have been used to model the phase change behavior in prior work. Sridhar *et al.* propose simulation of two-phase energy and mass balance (STEAM), a compact simulator that models two-phase liquid cooling, focusing on the liquid-vapor phase change (Sridhar et al., 2013). They model phase change from liquid to vapor, while our work focuses on phase change from solid to liquid. Tan *et al.* carry out CFD simulations, which are computationally-intensive, to analyze the PCM behavior, but do not consider real-life workloads (Tan and Fok, 2007).

Raghavan *et al.* define an RC network for the silicon and PCM layers (Raghavan et al., 2012). In order to compute the phase change duration, this model uses McPAT (Li et al., 2009) to estimate the energy consumed by the cores and use these estimations to drive the RC model. Tilli *et al.* consider a more detailed PCM model, where they use a thermal RC network and carry on latent heat energy calculations (Tilli et al., 2012). Their model assumes homogeneous heat distribution across the PCM layer and assigns a *single RC* value for the PCM layer. In this model, during phase change from solid to liquid, temperature of the PCM layer stays constant until PCM absorbs energy that is equal to its latent heat of fusion.

The aforementioned PCM models (Raghavan et al., 2012; Tilli et al., 2012) are not sufficient for accurate modeling of PCM-based cooling as they rely on simplifying assumptions regarding the phase change duration and the PCM thermal properties.

Raghavan's model (Raghavan et al., 2012) is based on *a priori* characterization of the energy consumption of the CPU cores. However, this approach is not highly accurate, because the latent energy stored in the PCM at runtime depends not only on the energy consumed by the cores, but also on the temperatures of the cores and PCM, as well as the thermal properties of the PCM and the chip package. On the other hand, assigning a single temperature value for the whole PCM layer and assuming constant temperature during phase change (Tilli et al., 2012) results in considerable loss of accuracy. This is because on-chip heat distribution is not homogeneous, thus, some parts of the PCM melt faster while other parts might still be in solid phase. PCM models that cannot capture these effects result in significantly high temperature error.

2.2.2 Modeling Microchannel Liquid Cooling

Liquid cooling has been a topic of interest in recent years because it provides a scalable and effective solution especially for emerging architectures such as 3D-stacked processors (Coskun et al., 2009b; Sridhar et al., 2010a; Sridhar et al., 2010b; Fourmigue et al., 2014; Liu et al., 2012). Coskun *et al.* propose a liquid cooling model, where a grid-level thermal RC network is constructed and thermal properties of different interlayer materials (i.e., through silicon vias (TSVs), microchannels) are specified (Coskun et al., 2009b). This model is able to incorporate the difference between the thermal resistances of solids and liquids. However, it cannot account for the convective heat flow along the channel. Sridhar *et al.* propose 3D-ICE (Sridhar et al., 2010a), which is a simulator that also includes the convective heat in the direction of the liquid flow, and thus, it can model the thermal gradient between the inlet and outlet ports of the liquid microchannels. The accuracy of 3D-ICE has been validated against ANSYS CFX computational fluid dynamics (CFD) tool (ANSYS, 2017). The follow-up of this work (Sridhar et al., 2010b) adds the support for modeling enhanced

heat transfer geometries such as pin-fin structures. This updated model also simplifies the computation in the microchannel layers by homogenizing the channels into porous medium.

Another body of work focuses on speeding up the long solution time required when simulating liquid-cooled ICs. ICTherm is a recently introduced simulator that implements an efficient algorithm to compute the transient temperature in linear-time complexity in liquid-cooled ICs (Fourmigue et al., 2014). Other researchers tackle the long simulation time problem by using a GPU-accelerated generalized minimum residual (GMRES) method and provide up to two orders of magnitude speedup compared to single-threaded CPU-GMRES method (Liu et al., 2012).

2.2.3 Modeling TEC

Modeling of TEC thermal behavior is widely studied in the research community (Paterna and Reda, 2013; Taylor and Solbrekken, 2008; Yazawa et al., 2012; Chowdhury et al., 2009). Compact thermal models represent the heat absorbed and rejected on either side of the TEC elements by using current sources entering and leaving the thermal nodes (Taylor and Solbrekken, 2008; Yazawa et al., 2012; Chowdhury et al., 2009). Chowdhury *et al.* compare their numerical compact model against measurements on a test device and show the impact of non-idealities on the cooling potential (Chowdhury et al., 2009). Others perform comparison of their 1D analytic TEC model (i.e., modeling only the vertical dimension of heat flow) against 3D numerical simulations using ANSYS tool (Yazawa et al., 2012).

2.2.4 Modeling Hybrid Cooling

Hybrid cooling techniques combine one or more cooling methods on the same platform to achieve higher cooling efficiency. Hybrid cooling with TECs and liquid microchannels has been proposed as an energy-efficient solution for mitigating high density hot

spots (Yazawa et al., 2012; Sahu et al., 2012; Sahu et al., 2015). Sahu *et al.* show the thermal benefits and characterize the behavior of such a hybrid cooling scheme on an experimental setup incorporating on-chip TEC units and a microchannel heat sink (Sahu et al., 2015). Other works rely on compact models to demonstrate the cooling energy savings of a hybrid solid-state and microfluidic cooling system over solely using microfluidic cooling (Yazawa et al., 2012; Sahu et al., 2012). They use the aforementioned compact models for TEC modeling, and represent the effect of microchannel-based liquid cooling using a high effective heat transfer coefficient at the boundary. This is a simplified way of modeling hybrid cooling and it does not consider important aspects of liquid cooling, such as the rise of coolant temperature as it flows from the inlet to the outlet. Such aspects become critical when, for example, exploring the impact of hot spot locations on resulting cooling power. Hot spots that are located closer to outlet of the microchannels get hotter than the ones that are closer to the inlets, and failing to model this effect results in optimistic evaluation of systems.

2.3 Thermal Management Techniques

Extensive research has been done on thermally-aware optimization techniques due to the crucial role of optimization in the overall energy efficiency of computing systems. Dynamic Thermal Management (DTM) techniques aim to keep the temperature below a certain threshold by adjusting various control knobs at runtime. Temperature-aware job scheduling (Coskun et al., 2008b; Coskun et al., 2008a; Coskun et al., 2007; Coskun et al., 2009a) and dynamic task migration (Zhao et al., 2013) techniques control temperature by intelligently determining when and on which cores to run the applications. Dynamic Voltage Frequency Scaling (DVFS) is a technique that adjusts the hardware control knobs to control temperature (Jayaseelan and Mi-

tra, 2009; Meng et al., 2012). DVFS can reduce the temperature by reducing the core power consumption at the cost of performance. In this section, we focus on temperature-aware optimization techniques that have been designed to address the specific challenges of the target cooling solutions.

2.3.1 Thermal Management on Systems with PCM

The existing work on PCM based thermal management can be divided into two main groups: (1) using PCM as a heat spreader/heat sink enhancer, (2) exploiting PCM as part of performance boosting strategies. The first group of work focuses on designing more efficient heat spreader or heat sink units by incorporating PCM in the cooling package (Alawadhi and Amon, 2003; Tan and Fok, 2007; Yoo and Joshi, 2004; Stupar et al., 2010; Lingamneni et al., 2014). Tan *et al.* show the thermal benefits of PCM by performing CFD simulations on a large mobile phone with a PCM filled heat storage unit (Tan and Fok, 2007). They investigate eight different cases including different PCM and polymer casing materials, where they compare the temperature traces of the heat source. They show that a heat storage unit (HSU) filled with PCM can reduce the temperature compared to the HSU filled with aluminum material. Alawadhi *et al.* study the effectiveness of a thermal control unit composed of PCM and a thermal conductivity enhancer on a portable electronic device using experimental and numerical analysis (Alawadhi and Amon, 2003). The design of hybrid heat sinks that incorporate air-cooling and PCM together has also been explored (Yoo and Joshi, 2004; Stupar et al., 2010). Yoo *et al.* investigate the energy savings of using PCM with a heat sink as an alternative to a fan-cooled heat sink (Yoo and Joshi, 2004). The results of that investigation shows that PCM can provide both energy savings changing between 5.4%-12.4% in fan-cooled systems and a size reduction of heat sinks. Stupar *et al.* propose a hybrid air-cooled heat sink containing PCM for high peak load, low duty cycle applications (Stupar et al., 2010).

In their work, the authors introduce different PCM heat sink configurations, describe an optimization approach to maximize peak temperature reduction for a given load, and demonstrate that 10-20°C of peak temperature reduction is achievable.

Most PCM materials have low thermal conductivity, which significantly limits their potential benefits. Recent work addresses this problem by proposing the use of metal-PCM composites as heat spreaders in mobile devices (Lingamneni et al., 2014). In their work, the authors show the tradeoff between thermal conductivity and latent heat capacity by performing a parametric analysis on the metal fraction of the composite.

PCM has also been used in large-scale computing environments. Recent work by Skach *et al.* studies the impact of placing PCM in servers to reduce cooling costs of a data center (Skach et al., 2015).

The second group of work centers around designing performance boosting policies that exploit PCM properties (Raghavan et al., 2012; Raghavan et al., 2013; Tilli et al., 2012; Shao et al., 2014). *Computational sprinting* allows temporarily exceeding the TDP of a chip to improve the responsiveness during short bursts of computation (Raghavan et al., 2012). In the context of *computational sprinting*, the authors also explore the benefits of using PCM in extending the sprinting duration. In the proposed sprinting technique (Raghavan et al., 2012), all of the cores are activated at the highest V/F setting until the cores hit a temperature threshold, after which the execution continues with a single core. The authors' later work verifies the feasibility of *computational sprinting* on a hardware/software testbed (Raghavan et al., 2013). The concept of *sprint pacing* is introduced in their follow-up work as well, where the cores sprint at a lower frequency when half of the PCM has melted. Other techniques aim to sprint periodically for longer durations (Tilli et al., 2012; Shao et al., 2014). *Safe computational re-sprinting* policy targets periodic hard deadline tasks and ad-

justs the V/F settings of the cores to reserve the minimum amount of PCM latent heat capacity to guarantee re-sprinting at full power (Tilli et al., 2012). The authors evaluate the benefits of their policy using a simple PCM model and simulations. Shao *et al.* consider repeated sprints with a fixed duty cycle, which is the ratio of the sustained power over sprint power (Shao et al., 2014). They implement their technique on a thermal test chip with an on-chip phase change heat sink as a proxy for a smart phone processor. They experimentally show that on-chip PCM heat sink with duty cycle sprinting can reduce peak temperature from $85°C$ to $69°C$ in comparison to having no PCM.

2.3.2 Liquid Cooling Management

Liquid cooling provides much higher heat removal efficiency compared to air cooling, but also brings new management challenges such as large on-chip thermal gradients and additional pumping power to push the liquid through the channels. Higher liquid flow rate provides lower peak temperature, however, operating the liquid-cooled system always at the highest flow rate will consume pumping power unnecessarily. The reason is that the cooling demand of the processor will dynamically change depending on the utilization of the system and the workload characteristics. Thus, under low utilization for example, the system can satisfy the same temperature constraint at a lower flow rate. Driven by that observation, Coskun *et al.* adjust the liquid flow rate at runtime to save pump power (Coskun et al., 2010). Their algorithm predicts the maximum temperature and adjusts the flow rate to the minimum value that satisfies the thermal limits. Sabry *et al.* propose a fuzzy controller to decide on the most efficient core voltage-frequency setting and flow rate at runtime (Sabry et al., 2011). They also show that combining the fuzzy controller with flow-aware load balancing in 3D-stacked systems provides significant reduction in thermal gradients.

Large thermal gradients, which significantly deteriorate reliability (JEDEC, 2009),

is another management challenge in liquid-cooled systems. The main source of large thermal gradients is that the temperature of the coolant fluid rises as it flows along the channel and absorbs heat from other blocks. Having narrower microchannels provides a slower rise of fluid temperature along the channel. GreenCool (Sabry et al., 2013) is a design-time method that exploits this observation to reduce thermal gradients by modulating the channel width. GreenCool computes the optimal channel width profile that minimizes the pumping energy under thermal gradient constraints.

Another body of work customizes the cooling effort based on the demands of the computing elements to save cooling power. Qian *et al.* propose an efficient channel clustering and flow rate allocation algorithm, in which different flow rates are assigned to groups of microchannels based on the on-chip heat distribution (Qian et al., 2013). Saving pump power by non-uniformly distributing the microchannels across the cooling layer is also possible (Shi and Srivastava, 2014; Sharma et al., 2015). One such technique co-optimizes the number, locations, dimension, and flow rate of the microchannels to minimize pumping power for a given chip power profile (Shi and Srivastava, 2014). Another similar approach is to design a non-uniform liquid cooling layer such that microchannels are denser (i.e., narrower and higher in number) above hot spots (Sharma et al., 2015). Their approach also utilizes a manifold microchannel sink, with a manifold layer above the microchannels with multiple inlets/outlets, to reduce the pressure drop across the channel (Sharma et al., 2015).

2.3.3 TEC Device Optimization

TECs have been widely studied in efficient hot spot mitigation (Chowdhury et al., 2009; Yazawa et al., 2012; Sahu et al., 2015; Taylor and Solbrekken, 2008; Paterna and Reda, 2013; Hu et al., 2013; Jayakumar and Reda, 2015). Superlattice-based thin film TECs made of Bi_2Te_3 as the bulk material are the state-of-the-art, owing to their high intrinsic figure-of-merit (Chowdhury et al., 2009). Thin film TECs are

silicon micro-fabrication compatible and can be directly integrated or fabricated on the back of a silicon chip (Chowdhury et al., 2009; Sahu et al., 2015). A group of work focuses on optimizing TEC device geometry and supply current to maximize coefficient of performance (COP) (Sahu et al., 2012; Yazawa et al., 2012; Taylor and Solbrekken, 2008). Another body of work shows the integration of TECs on the back of a silicon test chip to cool hot spots with heat fluxes up to $1250 W/cm^2$ (Chowdhury et al., 2009; Sahu et al., 2015). Chowdhury *et al.* show that for a hot spot with $1250 W/cm^2$ heat flux, up to $9.6°C$ reduction in hot spot temperature is achievable using a Bi_2Te_3-based, 3.5mm × 3.5mm TEC unit (Chowdhury et al., 2009).

2.3.4 Hybrid Cooling Management

Hybrid designs incorporate two or more cooling solutions on the same platform. The first group of hybrid designs focus on TECs working together with liquid cooling. This hybrid combination is promising owing to the ability of TECs to remove localized hot spots and the ability of liquid cooling to remove background heat efficiently. Sahu *et al.* experimentally explore the impact of design parameters on the cooling ability of a test vehicle, which combines a microchannel heat sink with SiGe-based TECs (Sahu et al., 2015). In their work, the authors vary the TEC sizes (70, 100, 120 μm side length), the location of the microchannel heat sink (on-chip/off-chip), ambient temperature, and the type of fluid as design parameters, and show a maximum temperature drop of $3°C$ at $200 W/cm^2$ heat flux and $85°C$ ambient temperature. Yazawa *et al.* show 10× cooling power reduction for a microchannel and TEC-based hybrid cooling system compared to using microchannel cooling only (Yazawa et al., 2012). The benefits of a similar hybrid cooling scheme have been also demonstrated on a 3D-stacked system through simulations (Hu et al., 2013).

The second group of work combines TECs and fan cooling to maximize throughput under thermal limits. Paterna and Reda find the optimum {TEC current, voltage-

frequency} pair to distribute a given power budget between TECs and cores to maximize throughput for a fixed fan speed (Paterna and Reda, 2013). The follow-up of this work demonstrates the tradeoffs between TEC power, leakage power, and fan power on an experimental setup and add fan speed as a parameter in the optimization scheme (Jayakumar and Reda, 2015). This work targets low heat flux rates ($\sim$20-28W/cm^2) and does not focus on localized use of the TECs. The authors demonstrate that for a given total power cap, using TECs in cooperation with fans and DVFS techniques can provide 19% higher performance compared to using only fans and DVFS.

2.4 Flow Cell Array Technology

As the demand for computational capacity in microprocessors is steadily increasing, maintaining energy-efficient operation becomes more challenging. In order to continue performance scaling while maintaining energy-efficient operation, architectural designs evolved from single-core to multicore systems. Multicore processors bring new challenges related to on-chip communication latency, power delivery and effective heat dissipation. 3D stacking technology aims to address some of these challenges by enabling the stacking of multiple logic and memory layers and connecting them via TSVs, hence, providing lower communication delay and higher communication bandwidth compared to 2D designs. However, high temperatures and power delivery are remaining challenges, which constraint the stacking of multiple layers and limit the potential of 3D architectures. In order to address these challenges, scalable cooling solutions and novel power delivery approaches are needed.

A new design concept has been recently introduced to overcome the aforementioned challenges in computing systems. In this concept, also called as *electronic blood*, the cooling subsystem provides cooling and on-chip power generation together,

similar to how blood running through the veins provide cooling and energy in biological systems (Ruch et al., 2011; Ruch et al., 2013). Flow Cell Array (FCA) is a realization of this concept in microprocessors, where fuel cells (also called redox cells) are pumped through the microchannels to remove heat while engaging in electrochemical reactions with each other to generate electrical power. FCA design constitutes microchannels etched on the silicon and connected in an electrically parallel manner. FCA technology is compatible with silicon manufacturing process and the FCA channels can be produced in a similar way as liquid microchannels are etched in a liquid-cooled system. Integrating FCAs in microprocessors has promising benefits for both 2D and 3D designs. It can boost efficiency in 3D-stacked architectures by relaxing the constraint on power delivery and allowing more layers to be stacked, while it can lead the way to self-sustaining 2D systems (i.e., reducing need for external power by generating a large percentage of the system power on-chip).

Existing work on FCAs focuses on modeling the behavior of temperature and power generation on FCAs integrated in Multiprocessor System-on-Chips (MPSoCs) (Sabry et al., 2014; Sridhar et al., 2014). Initial work (Sabry et al., 2014) builds a numerical model in COMSOL and validates the COMSOL model against experimental data from prior work (Kjeang et al., 2007). The follow-up of this work introduces PowerCool, a simulation infrastructure based on compact modeling approach, that can simulate the microfluidic cooling and power generation on 3D-stacked MPSoCs (Sridhar et al., 2014). 3D-ICE simulator (Sridhar et al., 2010a) is used for the temperature part of the simulation and it is coupled with the electrochemical simulation module. In PowerCool, the authors also provide a small analysis of the temperature and power generation levels that is based on IBM Power7+ processor (IBM, 2010) and demonstrate on-chip power generation of about $6W$.

The potential benefits of FCA, however, are not limited to this specific architec-

ture. There are many architectural design aspects contributing to the microfluidic cooling efficiency and power generation, which have not been explored yet, such as the die size, heat flux, and the technology dependent leakage parameters. Each of these design parameters introduce tradeoffs between generated power, leakage power, pumping power, and maximum chip temperature. Thus, a detailed exploration of the design space is needed in order to determine the system properties that maximize the advantages of FCAs.

2.5 Distinguishing Aspects from Prior Work

This thesis advances the state-of-the-art processor cooling research in the following specific directions.

Thermal Modeling:

We propose fast compact thermal models to enable the exploration, evaluation, and optimization of advanced cooling methods. We validate our models by comparing against CFD simulations and demonstrate significant speedup in simulation time while providing sufficient accuracy.

Our proposed PCM thermal modeling technique (Kaplan et al., 2014) differs from previous work in the following aspects. We propose a detailed thermal model that accurately captures complex phase change behavior, such as local melting around hotter parts of the chip, which cannot be observed using prior compact thermal models. We compare our model against COMSOL CFD simulations and demonstrate $0.22°C$ error on average. Contrary to some prior models, our model does not rely on *a priori* characterization of energy consumption. The proposed PCM model is up to 37.5x faster than carrying out CFD simulations, and is easily applicable to a variety of systems with different power, performance, temperature characteristics. Thus, our work in PCM modeling advances the latest PCM research by enabling the exploration

of the design space and runtime behavior at a finer spatial and temporal granularity in a fast and accurate manner.

Our work on hybrid thermal modeling is the first to devise a compact steady-state model for the design and evaluation of systems using TECs and liquid microchannels with sufficient detail and modularity. We integrate our models into HotSpot (Skadron et al., 2003b), an open source thermal simulator commonly used in the research community, and plan to make them available for others to use. Our work advances the research on processor cooling by contributing the following improvements over the existing models: (1) Our model captures complex thermal behavior of microchannel liquid cooling and TEC cooling with sufficient detail. (2) Our model is sufficiently general to be applied to a wide range of platforms. (3) It is modular in the sense that users can plug-in the cooling elements (either TECs or microchannels) with desired size, properties, and granularity. In this way, it enables researches to explore a wider design space in a fast and accurate manner. (4) Compared to using computationally expensive commercial multi-physics tools (i.e., COMSOL), our compact model provides high accuracy while saving considerable amount of time (up to four orders of magnitude shorter simulation time) and processor resources.

Hardware Testbed with PCM:

We build a hardware testbed with PCM unit placed on top of the package and run real life applications on it. Our implementation of hardware testbed with PCM cooling (Vivero et al., 2015) is novel in the following aspects. Our work is the first to experimentally demonstrate the accuracy of a PCM thermal model on a hardware testbed. Prior models have been compared against CFD simulations only. We also implement and evaluate for the first time a soft PCM capacity sensor that monitors the remaining PCM capacity at runtime based temperature measurements and equivalent thermal resistances of the package on our testbed.

Design-Time and Runtime Management:

We develop thermally-aware management techniques to maximize the energy efficiency in PCM-enhanced systems and well as hybrid cooling systems.

We propose a PCM runtime management technique, ***adaptive sprinting*** (Kaplan and Coskun, 2015), which brings the following innovations over the state-of-the-art: (1) We observe that PCM melts at different rates around different locations of the chip heterogeneous on-chip heat distribution, which was not captured in prior work. Our work is the first to consider the non-uniform melting of the PCM and exploit this observation to extend sprinting duration and provide further performance gains. (2) We claim that power consumption during sprinting is highly application dependent and assuming a fixed sprinting power (as in prior work) leads to lower thermal efficiency. Our technique does not rely on a priori assumptions about application's power consumption. (3) Existing sprinting policies either merely apply DVFS or alternate between sprinting and resting modes to control temperature. The proposed *adaptive sprinting* adds another control knob, the number of sprinting cores, and applies it in conjunction with DVFS technique. (4) We propose to monitor the remaining unmelted PCM at runtime to decide on the number, location, and the V/f levels of the sprinting cores. By utilizing the PCM-related information when making runtime decisions, *adaptive sprinting* can utilize the PCM storage capability more efficiently, providing 29% higher performance and 22% energy savings compared to the best performing sprinting policy.

In order to maximize the energy-efficiency of hybrid cooling systems with TECs and liquid cooling, we develop *LoCool* optimizer. *LoCool* jointly determines the liquid flow rate and TEC current to minimize cooling power for a given temperature constraint. We demonstrate that, a hybrid cooling design optimized with *LoCool* can remove high intensity hot spots effectively and saves cooling power by up to 28%

compared to a design that uses liquid cooling only. We also show that if the same hybrid design is not optimized, it can lead to higher cooling energy consumption compared to the liquid-cooled design in at least 80% of the cases.

FCA Technology:

We provide a thorough analysis of the architectural design parameters that would maximize the power generation on systems with integrated FCAs. Design parameters such as the chip size, channel geometry and heat density directly impact the amount of power generation. On the other hand, runtime control parameters such as the fluid inlet temperature and flow rate also significantly affect the overall performance of the FCA system. Constraining the exploration of FCA on a single system and not considering all of those parameters may result in underestimation of FCA's potential benefits or tradeoffs. Our work addresses this issue by conducting a detailed analysis of the broad design space, while taking into account the tradeoffs between generated power, maximum chip temperature, leakage power, and pumping power. The insights we provide as part of this thesis helps determine candidate platforms with desired properties that would benefit from FCA technology the most. Our analysis shows that for smaller low power chips, up to 76% of the total processor power can be generated on-chip by the FCAs. For higher power processors, FCA can generate the amount of power that is equivalent to the temperature dependent leakage power plus the liquid pumping power.

Chapter 3

Modeling and Validation of Advanced Cooling Methods in Compact Thermal Simulators

In this chapter, we provide details of our proposed modeling methods that are able to characterize the temperature behavior of cutting-edge cooling mechanisms. We focus on three main advanced cooling methods, PCM, TECs, and microchannel liquid cooling, as well as a hybrid combination of them. We integrate our models in a compact thermal simulator, HotSpot (Skadron et al., 2003b), and demonstrate the accuracy by providing comparison against multi-physics simulations (i.e., COMSOL) and testbed measurements. We also provide insights on the factors influencing the modeling accuracy and the impact of the accuracy on the design of runtime management policies.

3.1 Modeling of Phase Change Materials

Having a detailed phase change model is essential for the design and true evaluation of systems with PCM cooling. Such a phase change model is needed in order to explore a variety of PCM material and volume choices as well as for the development of runtime management policies to maximize PCM benefits. To address this need, we propose a PCM thermal model (Kaplan et al., 2014) that is able to provide highly accurate temperature estimations within a short simulation time. This section

explains our proposed PCM modeling method in detail. It continues with a discussion of how we implement the models used in prior work. We then demonstrate the accuracy of our model by comparing it against COMSOL. We show the significance of modeling accuracy by evaluating a runtime management policy using our proposed and prior work's model. Finally, using our PCM model, we analyze the impact of PCM properties on the temperature profile of a processor.

3.1.1 Proposed Modeling Methodology

We leverage the compact modeling strategy for temperature modeling. In compact thermal simulators such as HotSpot (Skadron et al., 2003a), temperature is modeled based on an equivalent RC network. The temperatures of nodes are computed by solving the differential equations corresponding to that RC network. HotSpot models both lateral and vertical heat flow, as well as the chip package, including the heat spreader and the heat sink. HotSpot also allows the user to model basic 3D-stacking by defining multiple layers of silicon, thermal interface material, or any other desired layer. Fine-grained simulation is carried out using the grid model, in which the floorplan is divided into smaller grid cells and temperatures are computed for each grid cell.

During phase change, PCM stores a large amount of energy at close-to-constant temperature, acting like a large thermal capacitor. The heat stored by PCM is called the *latent heat of fusion* and melting continues until PCM absorbs an amount of energy equal to its *latent heat of fusion*. Our goal is to construct a model that can estimate the impact of phase change on temperature, as such a feature is not currently available in compact thermal simulators.

In this work, we focus on phase change from solid to liquid state and vice versa. We propose modeling phase change behavior using the *apparent heat capacity* method (Alawadhi and Amon, 2003). In this method, a nonlinear temperature-dependent

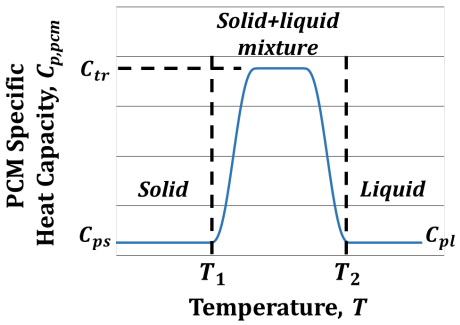


Figure 3·1: Piecewise linear function for PCM specific heat capacity. Setting $c_{tr} \gg c_{ps}$ for the (T_1, T_2) interval models the phase change.

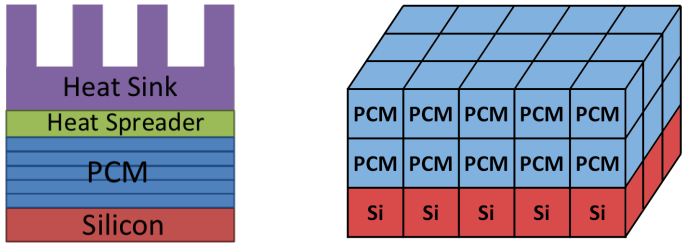


Figure 3·2: (a) Package layers; (b) Silicon and PCM grid cells.

specific heat capacity is assigned to the PCM layer as shown in Figure 3·1. The transition of the PCM from solid to liquid occurs over a temperature interval, where the specific heat capacity is very high compared to the material's heat capacity in the solid and liquid phases. Due to the high specific heat capacity, the rate of change of temperature is very low during phase transition.

We implement our model in HotSpot as follows: We first define a layer of PCM material. The PCM layer is placed on top of the silicon layer and has the same layout as the silicon layer. Using the layer configuration files in HotSpot, we set the thermal conductivity and thickness of the selected PCM. We also modify HotSpot to define the melting point and latent heat of fusion of the PCM. Figure 3·2(a) shows the package layers for a chip with PCM. For thick PCMs, we divide the PCM layer into thinner layers in order to improve accuracy. Figure 3·2(b) illustrates the grid

cell structure for the silicon and thin PCM layers. The bottom layer is the silicon layer, which has the processing units where the heat is generated. On top of that is the PCM layer, which does not dissipate any power. Next, we assign each individual PCM grid cell a temperature-dependent specific heat capacity as in Equation 3.1:

$$C_{p,pcm}(T) = \begin{cases} c_{ps} & T < T_1 \\ c_{tr} & T_1 \leq T \leq T_2 \\ c_{pl} & T > T_2 \end{cases} \tag{3.1}$$

where c_{ps}, c_{pl}, and c_{tr} are the specific heat capacities of solid, liquid, and phase transition states, respectively. We use $c_{ps} = c_{pl}$ similar to prior work (Ogoh and Groulx, 2012). T_1 is the onset temperature and T_2 is the end temperature of the phase transition. In our experiments, we use a transition temperature interval of 3^oC (Srinivas and Ananthasuresh, 2006), $c_{ps} = 1.57 \cdot 10^6 J/m^3K$, and $c_{tr} = 305 \cdot 10^6 J/m^3K$ (corresponding to Cerrobend PCM). We refer to the melting temperature as the center point of (T_1, T_2) interval. We implement the temperature-dependent heat capacity of Equation (3.1) using a smoothed piecewise linear function. At each time step, we update the specific heat capacity of each PCM grid cell depending on its temperature.

An important feature of our model is that it accounts for non-uniform melting of the PCM layer. In the case where there are idle and active cores, some portions of the PCM layer may melt earlier while other parts remain in the solid phase. Our model captures this behavior as we carry out phase change computations at a grid cell level.

Implementation of Phase Change Models in Prior Work:

As briefly discussed in Section 2.2.4, coarse-grained PCM models have been proposed and used in prior work (Raghavan et al., 2012; Tilli et al., 2012). For comparison purposes, we focus on a model proposed by Tilli *et al.*, where they use an RC network and a *latent heat energy* model (Tilli et al., 2012). In this model, a lumped RC network is defined for the silicon layer. On the other hand, the PCM layer is

treated as a single large cell where single R and C values are assigned to the whole layer assuming uniform heat distribution. During phase change, the PCM temperature is kept fixed at the melting temperature. To account for the melting duration, they compute the latent heat energy absorbed by the PCM using the heat transfer equation as follows:

$$\dot{U} = \sum_{k=1}^{N} \frac{T_k - T_{PCM}}{R_v} - \frac{T_{PCM} - T_{AMB}}{R_{PCM}} \tag{3.2}$$

where $\dot{U}$ is the rate of change of internal energy of the PCM, N is the number of silicon cells, T_k is the temperature of silicon cell k, T_{PCM} and T_{AMB} are the temperatures of the PCM layer and the ambient, respectively. R_v represents the contact resistance between the silicon and PCM layers in the vertical direction and R_{PCM} represents the thermal resistance of the PCM layer. We implement a model to mimic this *latent heat energy – single RC* model (Tilli et al., 2012) in HotSpot for comparison purposes.

The key differences of our model compared to *latent heat energy – single RC* model of (Tilli et al., 2012) are as follows: (1) we use the *apparent heat capacity* method to account for both temperature calculation and phase change duration, and (2) we carry out phase change computation at a grid cell level for the PCM layers.

3.1.2 Model Validation Using Multi-Physics Tools

This section provides a validation approach and demonstrates the accuracy of our phase change model by comparing its reported temperatures against the ones obtained from COMSOL (COMSOL, 2017). COMSOL models the chip package geometry as a set of 3D blocks stacked on each other, forming the layers of the package: silicon, PCM, heat spreader, and heat sink. The geometry is turned into a mesh composed of finely-sized tetrahedrals, comparable in size to the grid elements used in HotSpot. To model phase change behavior in the PCM layer, COMSOL uses the *apparent heat*

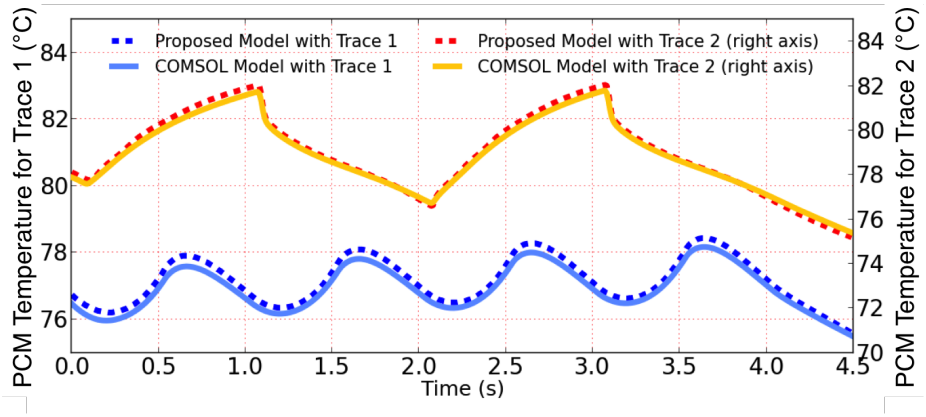


Figure 3·3: Transient temperature comparison for two different power traces. Trace 1: square wave with 50% duty cycle; Trace 2: triangular wave with 1 sec period.

capacity method (Alawadhi and Amon, 2003). COMSOL implements the two steps in the piecewise function of Equation (3.1) using a smoothed Heaviside function with a continuous 2^{nd} derivative. This modeling method has been used in similar COMSOL simulations involving phase change behavior (Ogoh and Groulx, 2012).

We carry out the validation experiments by simulating an AMD Opteron 6172 processor, using $8W$ and $2.63W$ for high and low power levels, respectively. Figure 3·3 compares the PCM layer transient temperature obtained by using our model against the COMSOL model. There are two example traces shown in the figure. Trace 1 uses a triangular wave with 1 second period for the power consumption signal and a 0.3 mm thick PCM with 77^oC melting point. Trace 2 uses a square wave with 50% duty cycle for the power signal and a 0.5 mm thick PCM with 80^oC melting point. Figure 3·3 shows that the temperature trace of our proposed model closely follows that of COMSOL. It should be noted that while the sophisticated COMSOL model is useful for validation, it runs far too slowly to evaluate the rapidly changing power traces we analyze in typical architectural simulations. Moreover, COMSOL requires several GB of storage even for a few seconds worth of real-life simulation; thus, it is not easily scalable to solve for longer traces. The scatter plot in Figure 3·4 compares

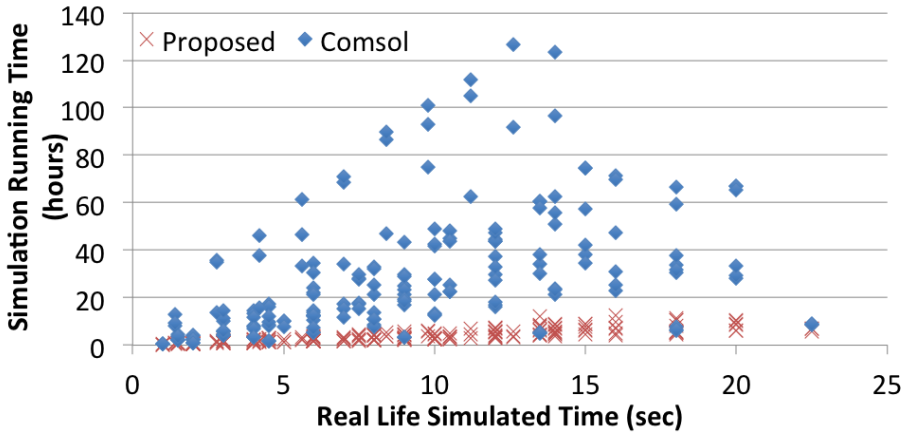


Figure 3·4: Scatter plot comparing the solution time for COMSOL and the proposed PCM model.

the simulation running times of COMSOL and the proposed model in HotSpot for various benchmarks and simulation lengths. Our proposed model implementation provides $37.5x$ maximum and $6.9x$ average simulation time savings in comparison to COMSOL. The running time difference between COMSOL and proposed model is higher for benchmarks with rapid power variations.

We investigate the accuracy of our phase change model by running a large set of experiments using various power traces, PCM thicknesses, and melting points. In Table 3.1, we report the maximum, mean, and standard deviation of error for a selected subset of these experiments. We present the temperature error across all units on both the silicon and PCM layers for our proposed model, as well as for the *latent heat energy – single RC* model (Tilli et al., 2012), both compared against COMSOL. As highlighted in the table, the maximum temperature error is significantly larger for the *latent heat energy – single RC* model, reaching up to $9.18°C$. On the other hand, our proposed model gives a maximum error of only $2.73°C$. The higher error occurs for benchmarks with abrupt power variations. Our model also reduces RMS error from $1.6°C$ to $0.27°C$ compared to the *latent heat energy – single RC* model.

ERROR (°C)	PHASE CHANGE MODEL	Square Wave 25% Duty Cycle	Square Wave 50% Duty Cycle	Square Wave 75% Duty Cycle	Triangular Wave 1 sec Period	Triangular Wave 2 sec Period	bzip2	calculix	GemsFDTD	hmmer	lbm	leslie3d	gcc
MAX	Single RC	7.59	7.78	9.18	7.52	6.96	3.45	7.33	5.95	6.33	4.21	4.71	5.26
	Proposed	1.23	1.11	1.08	0.9	0.9	1.93	1.99	2.22	2.73	0.76	1.33	2.59
MEAN	Single RC	1.22	1.61	2.49	1.22	1.7	0.91	1.34	1.38	1.51	1.23	1.3	1.24
	Proposed	0.17	0.21	0.23	0.18	0.19	0.22	0.22	0.24	0.31	0.16	0.17	0.43
STD DEV	Single RC	1.16	1.53	2.21	0.97	1.22	0.67	1.18	1.1	1.25	0.79	0.86	1.17
	Proposed	0.16	0.18	0.2	0.13	0.13	0.17	0.22	0.21	0.22	0.08	0.12	0.4

Table 3.1: Maximum, mean, and standard deviation of error for the two melting models compared against COMSOL.

Impact of Modeling Accuracy on the Evaluation of Runtime Management Policies

Next, we evaluate the impact of modeling accuracy on the runtime management policy decisions and show that a better modeling approach changes the design time evaluation of those strategies. For this purpose, we implement a temperature-aware throttling policy and evaluate the behavior of the policy using the two phase change models. In this policy, if a core's temperature exceeds a predefined upper threshold, it is put to idle for a fixed amount of throttling time. We use $80°C$ as the temperature threshold and 10 ms of throttling time, which is used in current systems. At the end of the throttling time, the policy checks if the temperature has fallen below the threshold value and if not, triggers the mechanism again. We simulate various utilization levels by activating different number of cores. We record the percentage of time spent in throttled mode for each core.

For lower utilization levels, the *single RC* model over-estimates the core temperatures, leading to higher percentage of throttling being reported. For example, for 60% utilization, *single RC* model does not detect any melting as the average PCM layer temperature does not reach melting temperature. As there is no melting, temperatures of active cores keep rising, leading to 12% more reported throttling time compared to the actual (i.e., the proposed model). Our model, on the other hand, captures the localized melting behavior and reflects the benefit of phase change on the core temperature.

As the utilization increases and the melting occurs, the *single RC* model starts to under-estimate the core temperatures and the percentage of throttling time. For example, for 100% utilization case, both models show that PCM goes through melting within the simulated time. However, the *single RC* model assumes constant temperature across the whole PCM layer during melting, leading to underestimation of core temperatures as well. In reality, some portions of PCM complete melting and the PCM temperature (and eventually the core temperature) starts rising at those locations. As the *single RC* model misses this observation, it reports 5.6% less throttling time than actual.

3.1.3 Impact of PCM Properties on Temperature

In this section, we analyze the impact of PCM thermal properties on the temperature profile through design space exploration. PCM materials vary in their latent heat of fusion values and melting temperatures. The larger the latent heat of fusion, the more heat we can store during melting. Typical values for melting temperature are between 30^oC to 70^oC (Hale et al., 1971); however, we explore a wider range of temperatures. In many applications, it is desirable to have the melting temperature close to, but below the maximum allowed chip temperature.

The thermal conductivity of the PCM is also an important parameter. It is desirable to have high conductivity to help homogeneous melting/freezing as well as to avoid overheating. Higher conductivity results in lower maximum temperature as it provides a smaller equivalent thermal resistance between the silicon layer and the heat sink. Cerrobend (19 W/mK) and gallium (33.7 W/mK) are examples of higher conductivity PCM (Hale et al., 1971). Most other PCMs, such as paraffin, have very low conductivity. Conductivity enhancement techniques have been proposed to overcome this challenge by embedding the PCM into a metal matrix (Mills et al., 2006; Lingamneni et al., 2014).

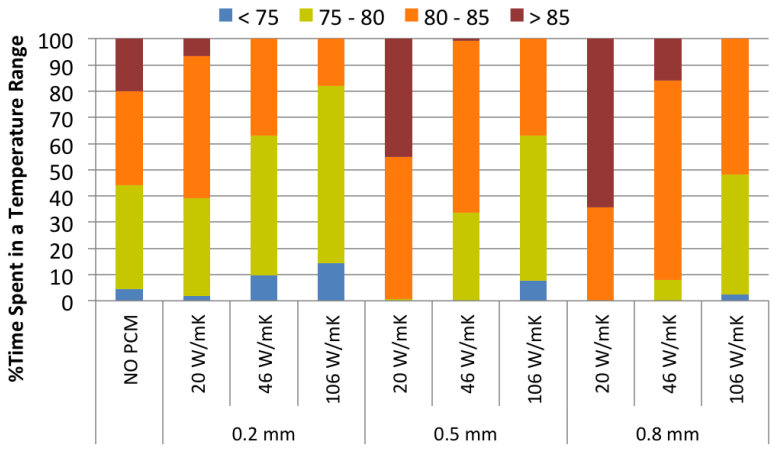


Figure 3·5: Percentages of time spent by the CPU within temperature ranges for 9 PCM configurations and for the case where no PCM is used.

The amount of PCM also impacts temperature profiles strongly. While a thick PCM can maximize the amount of heat absorbed and delay entry into fully melted (liquid) state, it can also interfere with the effectiveness of the heat sink after melting is complete. This is because the relatively lower conductivity of the PCM can reduce the efficiency of heat transfer to the high-conductivity heat sink.

In our design space exploration, we mainly focus on the conductivity and the thickness of the PCM as design parameters. We assume the use of highly thermally-conductive copper-PCM matrix (Lingamneni et al., 2014) with various PCM fractions, and explore the impact of PCM for 0.2-0.8 mm thickness and 20-106 W/mK conductivity (corresponding to PCM fractions 100%-70%). We set the melting temperature as 80^oC and the total simulation time as 10 seconds. Figure 3·5 shows the percentages of time spent by the CPU within different temperature ranges for 9 different PCM configurations as well as for no-PCM case. We see that the PCM properties have a significant effect on the temperature profile of the processor. With higher PCM conductivity, cores spend less time in the higher temperature ranges. This impact of conductivity becomes even more apparent for the 0.8 mm PCM. While the amount of time spent in the highest temperature range is 65% for 20 W/mK PCM, it decreases

to 15% and 0% for 46 W/mK and 106 W/mK PCMs, respectively. In terms of the maximum and average temperatures, we observe up to 20.1°C and 11.7°C difference, respectively, among the 9 PCM configurations. Another interesting result is that choosing the wrong PCM may result in higher temperatures than having no PCM at all. For example, for the system with no PCM, the temperature exceeds 85°C 20% of the time; while for the 0.5mm, 20 W/mK PCM, it rises to 45% as the poor conductivity of the PCM interferes with the effectiveness of the heat sink. In general, having the highest conductivity PCM available is preferred for all cases. However, the cost and availability of a high conductivity material becomes a tradeoff.

3.1.4 Evaluation of the PCM Model on a Hardware Testbed

In this section, we present a hardware testbed with a PCM unit installed on top of the chip package (Vivero et al., 2015). We start with a description of the testbed setup. Next, we create a model of our testbed using HotSpot which includes our proposed PCM model. We compare the temperature traces obtained from HotSpot simulations against the real-life measurements obtained from the testbed. On our testbed, we implement for the first time a soft PCM capacity sensor that monitors the remaining unmelted PCM at runtime. Finally, we evaluate runtime management policies using our testbed and the PCM sensor.

We use an Inforce Computing IFC6410 single-board computer (SBC) as our computing platform. The platform is powered by a Qualcomm Snapdragon 600 System-on-Chip (SoC), which includes a quad-core 1.2 GHz mobile processor (with a 2 MB shared L2 cache) commonly found in modern mobile devices. The IFC6410 provides 2.0 GB of RAM, and runs Android 4.1. The Snapdragon processor does not have a heat sink, thus, the processor is normally exposed to ambient air. We build a copper box enclosure that holds the PCM and place it on top of the Snapdragon processor as shown in Figure 3·6. We use a single thermocouple to measure the PCM temper-

ature and it is placed at the bottom surface of the copper PCM container. Thermal Interface Material lies in between the processor die and the PCM enclosure. We use $0.175g$ of paraffin wax as our PCM.

We sample the total power consumption of the SBC using a multimeter and a current probe (70Hz). We measure PCM layer temperature via the thermocouple and record CPU core temperatures using the internal temperature sensors (1.0Hz). We run a selection of computational kernels from the SciMark 2.0 Java benchmark (Pozo and Miller, 2017) (*Jacobi Successive Over-Relaxation (sor)*, *Sparse Matrix Multiply (smult)*, and *Dense LU Matrix Factorization (lu)*), with small problem sizes to focus on exercising CPU-intensive loads on the testbed.

Experimental Evaluation of a PCM Model

This section explains the details of how we experimentally evaluate the proposed PCM thermal model on our hardware testbed. We first create a model of the Snapdragon processor in HotSpot. For this purpose, we estimate the floorplan using McPAT (Li et al., 2009) modeling tool and the location of the VCC pads. We then extract the core

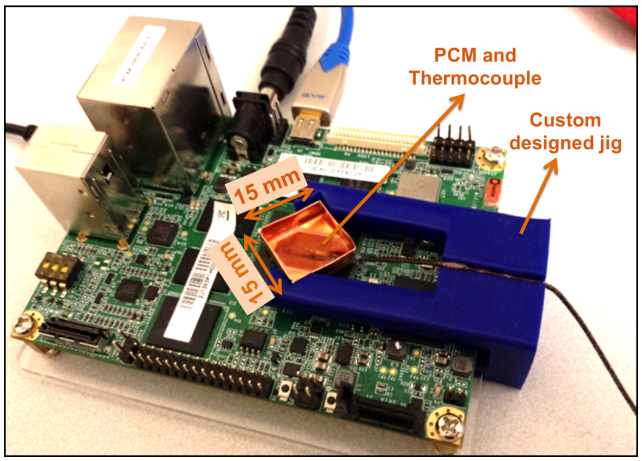


Figure 3·6: IFC6410 SBC with copper box holding PCM, fitted on top of the Qualcomm Snapdragon SoC.

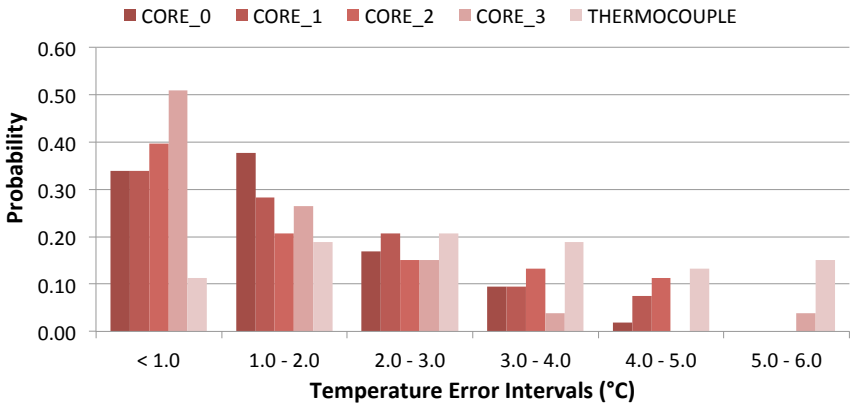


Figure 3·7: HotSpot model steady state error probability histogram.

and un-core powers from the total power using the data sheets and measurements. We finally create a chip stack in HotSpot using the estimated and measured geometry of Snapdragon. As the Snapdragon SoC does not have a heat sink while HotSpot software does not allow the removal of the heat sink, we model the heat sink as our PCM by integrating our phase change model into the package.

We carry out two main sets of comparisons: steady state temperature and transient temperature. For each case, we run a set of experiments (e.g., 4 different CPU frequency settings, different number of active CPU cores, 4 different power traces) and report the difference between the simulated and measured temperatures.

Figure 3·7 is a histogram plot that shows the steady state error probability. The x-axis represents the temperature error intervals in ^oC and the y-axis shows the probability of having a temperature error within the corresponding interval. We generate this plot using the following approach: (1) For each steady state experiment, we compare the measured and simulated temperatures of CPU cores and the PCM unit, and find the absolute steady state temperature error for each unit. (2) To find the probability of having a temperature error less than 1^oC, for example, we count the number of times we encounter an error that is less than 1^oC and divide it by

the total number of experiments. Error probability represents a comparison of the temperature trend between our simulations using the phase change model and the real-life measurements. According to Figure 3·7, the steady state temperature error is less than 4^oC with 0.89 probability and less than 2^oC with 0.6 probability. We carry out similar experiments for the transient case, where we simulate a 60-second time frame. The transient temperature error is found to be less than 4^oC with 0.63 probability. The temperature range in our experiments is 68^oC, where 4^oC of error corresponds to only 5.8%.

Implementation of a PCM Sensor

Monitoring PCM capacity enables estimation of the remaining sprinting capability. For this purpose, we implement a soft PCM capacity sensor that monitors how much of the PCM remains unmelted at runtime. Our soft sensor targets real life use as part of our proposed thermal management strategies.

The PCM capacity sensor is a counter that accumulates the amount of latent heat energy stored in the PCM at a given time, during phase change. At the beginning of the phase transition, the amount of latent energy stored in the PCM is zero. In order to fully melt, the PCM needs to store energy that is equal to its latent heat of fusion. PCM sensor estimates this stored energy by using the temperature sensor measurements and thermal resistances of the package as in the following formula:

$$P_{NET} = \frac{T_{CPU} - T_{PCM}}{R_{Si\_to\_PCM}} - \frac{T_{PCM} - T_{AMB}}{R_{PCM\_to\_AIR}} \tag{3.3}$$

$$E_{STORED,t} = E_{STORED,(t-1)} + P_{NET} \times t_{sampling} \tag{3.4}$$

where T_{CPU}, T_{PCM} and T_{AMB} are the temperatures of the CPU cores (we use the average of the four CPU cores), the PCM, and ambient air, respectively. $R_{Si\_to\_PCM}$ and $R_{PCM\_to\_AIR}$ are the equivalent thermal resistances seen from the silicon to PCM

and from PCM to air, respectively. In the right hand side of Equation (3.3), the first term represents the heat entering the PCM from the silicon layer and the second term represents the heat leaving the PCM to the ambient air per unit time. $E_{STORED,t}$ is the latent heat energy stored in the PCM at time t and $t_{sampling}$ is the sampling interval. We use $t_{sampling} = 1$ sec, as the CPU temperatures are recorded at a rate of 1.0 Hz. Equation (3.4) is an accumulation operation, which approximates taking the integral of the net input power over time. We measure the overhead of the PCM monitor (including the temperature sensing and the calculations) in terms of CPU utilization on our testbed, which is less than 0.4% .

Evaluation of Management Policies on the Testbed

We next evaluate runtime management policies on our testbed. We compare two policies: *temperature triggered DVFS (tt-dvfs)* and *PCM-aware* policy from prior work (Raghavan et al., 2013). *tt-dvfs* policy decreases the V/F level in steps if any of the cores reach the critical temperature (i.e., 80^oC), and increases back in steps when the temperature falls to a predefined value (i.e., 70^oC). We add a feature to *tt-dvfs* policy such that it takes proactive action before hitting the temperature threshold. Thus, when any of the cores reach 75^oC *tt-dvfs* decreases the V/F level by 1 step.

PCM-aware policy switches to a lower V/F level when the remaining PCM latent heat capacity falls to 50%. The aim is to use the PCM capacity at a lower rate, thus, extend the sprinting duration. In that sense, this policy also has a proactive nature. We experiment with different thresholds for the remaining PCM capacity (i.e., 25%, 50%, and 75%).

The purpose of this comparison is to show the benefit of using the PCM state information while taking runtime actions. Evaluations on our testbed using the PCM sensor shows that *PCM-aware* with 75% PCM threshold gives 4.5% higher performance compared to *tt-dvfs*. This is because *PCM-aware* takes action earlier than the

tt-dvfs based on the remaining PCM capacity.

3.2 Modeling of Hybrid Cooling with TECs and Liquid Cooling

In this section, we present implementation details of the hybrid cooling model including TECs and liquid microchannels. Figure 4·8 illustrates an example hybrid cooling design, where a TEC unit is placed above the processing layer on top of the hot spot location, and a microchannel liquid cooling layer is placed on top. We first explain how we model TECs and liquid microchannels using compact modeling. We then provide an approach for validation using COMSOL and 3D-ICE simulations. We conclude this section by discussing the important aspects of modeling hybrid cooling systems and how they influence the modeling accuracy.

3.2.1 Proposed Modeling Methodology

TEC Model

A TEC operates based on the Peltier effect such that when current passes through the device, heat is absorbed from one side (cold side) and rejected to the other side (hot side), creating a thermal gradient across the two sides. The amount of heat removed by the TEC depends on the Seebeck coefficient (S), applied current (I),

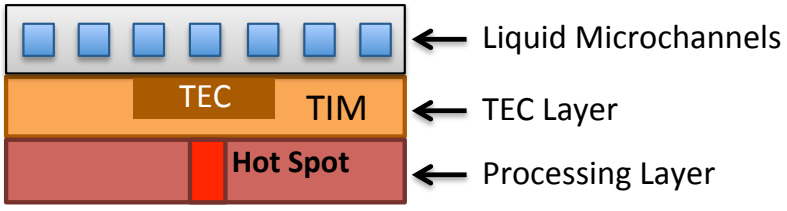


Figure 3·8: Chip stack with hybrid cooling combining microchannel liquid cooling and TECs. TECs are placed on top of high heat flux areas to remove hot spots, while microchannels are used to remove the heat pumped by the TECs and the background heat.

electrical resistivity (ρ_{tec}), thermal conductivity (k_{tec}) of the TEC device, and the temperatures of the hot (T_h) and cold (T_c) sides. Superlattice-based thin film TECs made of Bi_2Te_3 have high figure-of-merit (ZT). They are silicon micro-fabrication compatible and can be directly integrated or fabricated on the back of a silicon chip (Chowdhury et al., 2009; Sahu et al., 2015). On-chip TEC devices are composed of ultrathin (5-10um) Bi_2Te_3-based p-n thermocouples sandwiched between copper mini-headers and are covered with ceramic insulator plates at the outmost surfaces (Chowdhury et al., 2009).

There are three main contributors to heat flow within a TEC unit: (i) the Peltier term which accounts for the heat absorbed/rejected on the cold/hot sides, (ii) the conductive heat flow term, and (iii) Joule heating term that represents the resistive heat generated by passing current through the TEC. Mathematical representation of these terms are:

$$Q_c = N(SIT_c - \frac{T_h - T_c}{R_t} - \frac{1}{2}I^2R_e) \tag{3.5}$$

$$Q_h = N(SIT_h - \frac{T_h - T_c}{R_t} + \frac{1}{2}I^2R_e) \tag{3.6}$$

where Q_c and Q_h stand for the heat absorbed and rejected on the cold and hot sides, respectively. T_c and T_h are the cold and hot side temperatures. N is the number of p-n couples placed in the TEC unit. $R_t = h_{tec}/k_{tec}A$ is the thermal resistance and $R_e = \rho_{tec}h_{tec}/A$ is the electrical resistance of a TEC unit of thickness h_{tec} and area A.

We implement this model in HotSpot in the following way. We use the grid model in HotSpot, in which, each layer on the processor stack is divided into smaller grid cells representing a thermal node in the thermal R network. We add functionality to define a block on the floorplan as a TEC unit. We then assign TEC thermal properties only to the grid cells corresponding to these TEC units. For this purpose,

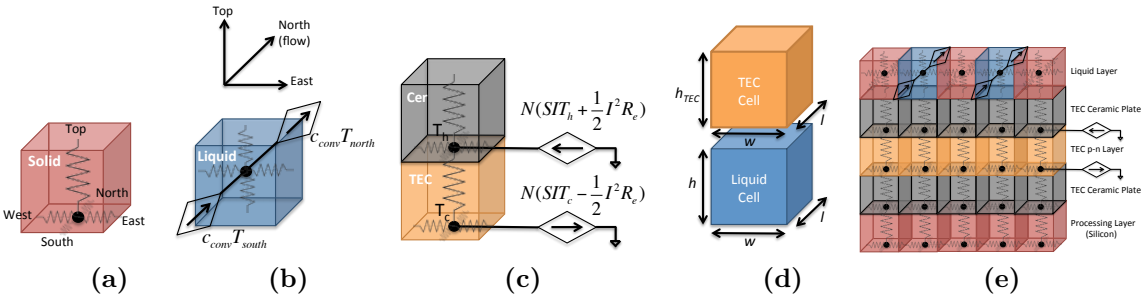


Figure 3·9: (a) Solid grid cell, (b) Liquid grid cell, (c) TEC grid cell, d) Dimensions of the grid cells, (e) Connectivity of the grid cells building a chip stack. Current sources are shown only the rightmost for TEC and ceramic cells for clarity.

we use the heterogeneous 3D modeling feature of HotSpot as mentioned earlier. By default, HotSpot accounts for the conductive heat flow (term (ii)) for solid cells as shown in Figure 3·9(a). In order to represent the Peltier term and Joule heating term on the cold and hot side of the TEC units described in Equations (3.5) and (3.6), we define current sources entering and leaving the TEC cells as illustrated in Figure 3·9(c). In the Figure, bottom surface of the TEC cell corresponds to the cold side temperature, while the bottom surface of the cell in the upper adjacent layer (i.e., the ceramic plate) corresponds to the hot side temperature.

Liquid Cooling Model

We adopt the 4 resistor model-based (4RM) liquid cooling model presented in 3D-ICE (Sridhar et al., 2010a). In the 4RM-based model, the discretization of the thermal grids is done such that the entire cross section of a microchannel forms a liquid grid cell. There are two main contributors to heat flow regarding a liquid grid cell: (i) convective heat transfer from the walls of the channel to the liquid and (ii) convective heat transfer in the direction of the liquid flow into and out of the current liquid cell. Figure 3·9(b) illustrates a liquid grid, where the term (i) is represented by resistive elements in four directions and the term (ii) is represented by using current sources in the direction of the flow (from South to North). The numerical values of the

resistances are given as follows (Sridhar et al., 2010a):

$$R_{top,bottom} = \frac{1}{h_{f,vertical} \cdot w \cdot l} \tag{3.7}$$

$$R_{east,west} = \frac{1}{h_{f,side} \cdot h \cdot l} \tag{3.8}$$

where $h_{f,vertical}$ and $h_{f,side}$ are the heat transfer coefficients for microchannel forced convection; w, l, and h are the width, length, and height of the microchannel cell, respectively (See Figure 3·9(d) for the cell dimensions.). As also stated in 3D-ICE work (Sridhar et al., 2010a), $h_{f,vertical}$ and $h_{f,side}$ (i.e., the vertical and side heat transfer coefficients) can be obtained from empirical correlations or numerical presimulation for a given system. For computing the heat transfer coefficients, prior work provides the following formulas assuming imposed axial heat flux and radial isothermal conditions:

$$h_{f,vertical} = h_{f,side} = \frac{k_{coolant} \cdot Nu}{d_h} \tag{3.9}$$

$$\begin{aligned} Nu = 8.235 \cdot (1 - 2.0421AR + 3.0853AR^2 \\ -2.4765AR^3 + 1.0578AR^4 - 0.1861AR^5) \end{aligned} \tag{3.10}$$

In these formulas, $k_{coolant}$ is the thermal conductivity of the coolant and $d_h = \frac{2h \cdot w}{h+w}$ is the hydraulic diameter of the channel. Nusselt number (Nu) was derived in prior work (Shah and London, 1978) as a function of channel aspect ratio ($AR = min\{h/w, w/h\}$). As Equations (3.9) and (3.10) may differ under different system assumptions, the original 3D-ICE simulator defines $h_{f,vertical}$ and $h_{f,side}$ as input parameters specified by the user.

Next, the values of the convective terms in the flow direction (i.e., the current sources) are computed as follows:

$$I_{in} = c_{conv} \cdot T_{south} \tag{3.11}$$

$$I_{out} = c_{conv} \cdot T_{north} \tag{3.12}$$

$$c_{conv} = C_v \cdot u_{avg,y} \cdot \Delta A_y \tag{3.13}$$

where I_{in} and I_{out} represent the convective heat flow into and out of the cell, respectively. T_{south} and T_{north} are the interface temperatures at the south and north surfaces of the cell. C_v is the specific heat capacity of the coolant, $u_{avg,y}$ is the average coolant velocity, and $\Delta A_y = w \cdot h$. The surface temperatures are approximated as the average of the cell temperatures which share that interface. We assume that for the southmost cell, $T_{south} = T_{inlet}$ (i.e., temperature of the coolant at the microchannel inlet) and for the northmost cell $T_{north} = T_{cell}$.

Note that by default, HotSpot places the virtual temperature node at the bottom surface of the grid cell in the vertical direction as illustrated in Figure 3·9(a). This convention is useful for modeling the TEC cells as the thermal effect is observed at the bottom and top surface of the TEC device. However, for liquid cells, we need to place the virtual node in the middle of the cell to be able to include the heat flow from the top/bottom walls in an accurate manner. Doing otherwise results in underestimation of the chip temperature by up to $20°C$ for liquid-cooled systems, according to our analysis (Refer to Section 3.2.2 for more detail). Thus, we construct the thermal resistance network in HotSpot such that for liquid cells, the node is placed in the middle; while for all other cells including TECs, the node is placed at the bottom surface. This way of constructing the thermal resistance network is our novel contribution. In Figure 3·9(e), we demonstrate how the grid cells of each type are connected in the chip stack building a thermal R network, for a single row of cells.

3.2.2 Model Validation Using Multi-Physics Tools

TEC Model Validation

In order to validate our TEC model, we compare temperatures reported by our model against the ones obtained from COMSOL simulations. For this purpose, we first select a prototype TEC device that has been fabricated on the back of a silicon chip and has been characterized in prior work (Chowdhury et al., 2009). We then create a model of this TEC device in COMSOL using the *heat transfer module*. Figure 3·10 illustrates the TEC device and the chip layers as we modeled in COMSOL. It is a superlattice-based thin film TEC made of Bi_2Te_3 as the bulk material and has high intrinsic figure-of-merit (ZT) (Chowdhury et al., 2009). TEC is composed of an array of 7×7 p-n thermocouples and has a total size of 3.5mm×3.5mm. Thermocouples are sandwiched between copper mini-headers and the top and bottom surface of the device is covered by ceramic plates to provide electrical insulation. Legs of the p-n thermocouples are ultra-thin ($8\mu m$) and the total thickness of the TEC device including the ceramic plates is $100\mu m$. Since the length and width of the thermocouple legs were not specified in prior work, we estimated them such that the 7×7 array fits nicely in the 3.5mm×3.5mm area. Based on this estimation, the leg width and leg length are $400\mu m$ and $150\mu m$, respectively. This corresponds to 0.833 p-n thermocouples per mm^2 area and it is used when calculating the parameter N (i.e., the number of p-n thermocouples) per grid cell in the proposed model. Detailed parameters of the TEC are given in Table 3.2. Note that for the temperature dependent parameters such as S, ρ_{tec} and k_{tec}, we assume constant values at steady-state temperature as reported in prior work (Chowdhury et al., 2009). The reported thermal conductivity k_{tec} is used for calculating the vertical thermal resistance. Since there is air between the p-n thermocouples, lateral heat transfer within the TEC unit is minimal. Thus, we assign a very large number to the thermal resistance in the horizontal direction for

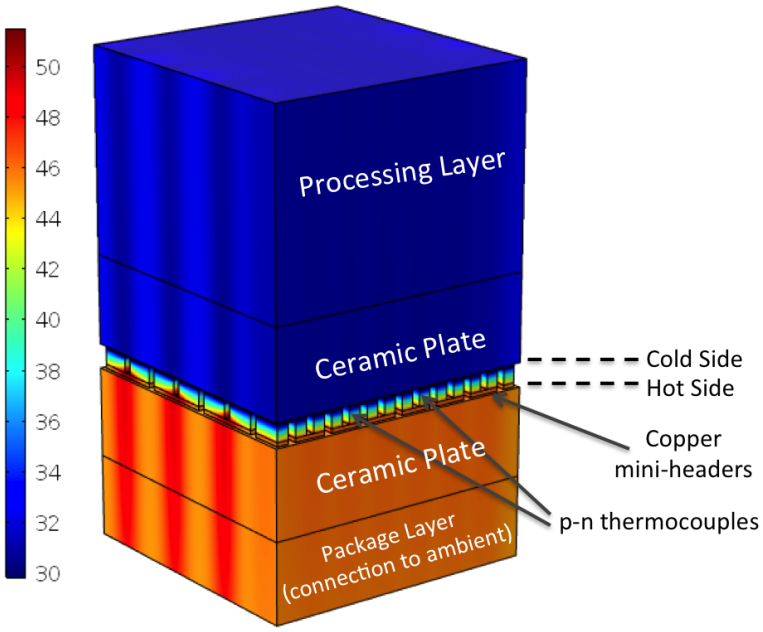


Figure 3·10: TEC device as we modeled in COMSOL. Example temperature distribution is shown for when TEC was biased at 4A current.

the TEC device. For all other layers, we include the lateral heat flow based on the corresponding material properties.

Next, we model the processing layer, where the heat is generated and is represented as a heat flux value (i.e., power dissipated per unit area), using a $100\mu m$-thick silicon layer at the cold side of the TEC. As TECs pump heat from the cold side to the hot side, an additional cooling mechanism is usually needed on the hot side of the TEC to avoid overheating and provide proper operation. Thus, at the hot side of the TEC, we define another layer, which represents the chip package and additional cooling mechanism (e.g., heat sink with fans, cold plates) that removes the heat pumped by the TEC. We assume silicon properties for this layer, set its thickness as $40\mu m$, and assign a heat transfer coefficient (htc) at the surface to the ambient to represent the additional cooling mechanism. *Htc* corresponds to the cooling capability of the additional cooling method with a higher number representing more effective cooling. We modify HotSpot's package model to be able to define a similar layer with

Table 3.2: The parameters we used for the liquid microchannel and TEC models.

Microchannel height	h	$100\mu m$
Microchannel width	w	$50\mu m$
Grid cell width & length	$w = l$	$50\mu m$
Microchannel length	L	$10mm$
Coolant thermal conductivity	$k_{coolant}$	$0.6069W/mK$
Coolant specific heat	C_v	$4181J/kgK$
Coolant inlet temperature	T_{inlet}	$27°C$
Coolant density	$\rho_{coolant}$	$998kg/m^3$
Coolant viscosity	μ	$8.89 \times 10^{-4} Pa.s$
Average coolant velocity	u_{avg}	$\leq 3m/s$
TEC width & length	$w_{tec} = l_{tec}$	$3.5mm$
Seebeck coefficient	S	$301\mu V/K$
Thermocouple thickness	h_{tec}	$8\mu m$
Copper mini-header thickness	h_{Cu}	$2\mu m$
Ceramic plate thickness	h_{Cer}	$44\mu m$
TEC electrical resistivity	ρ_{tec}	$1.08 \times 10^{-5} Ohm.m$
TEC thermal conductivity	k_{tec}	$1.2W/mK$
Copper thermal conductivity	k_{Cu}	$400W/mK$
Ceramic thermal conductivity	k_{Cer}	$175W/mK$
Silicon thermal conductivity	k_{Si}	$130W/mK$

connection to ambient using the *htc* parameter.

Note that in COMSOL, we model the TECs in detail by defining the individual p-n legs, the copper mini-headers connecting the thermocouples in series, the VDD and ground nodes one by one. In the proposed model, we define the TEC device as a block, where the details of individual p-n legs and the empty space between them are omitted for the sake of simplicity and speed. In order to account for the differences introduced by these simplifications, we calibrate our proposed model empirically based on COMSOL. Based on our experiments, we observe that such effects demonstrate themselves as a scaling factor on the equivalent electrical resistivity of the TECs, experimentally determined as 14.

We run two sets of experiments in COMSOL: (i) without the TEC device for varying *htc* and heat flux (q) levels, and (ii) with the TEC device using a bias current changing from 0 to 7A with varying q levels. For the case with TECs, we define a multi-physics problem, which combines *heat transfer in solids* with *thermoelectric*

effect, electromagnetic heat source and thermal coupling elements. We use the segregated solver in COMSOL to solve the multi-physics problem iteratively using GMRES method for both sub-parts of the problem. The resulting mesh consists of 164088 domain elements, 125204 boundary elements and 16422 edge elements. Number of degrees of freedom solved for is 1810396.

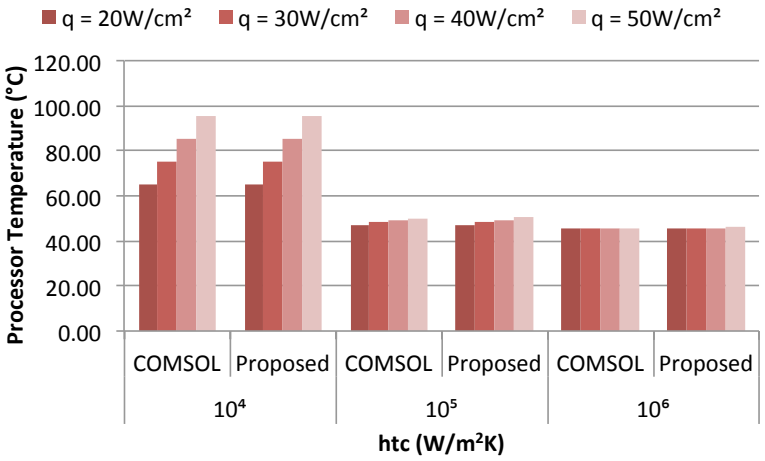


Figure 3·11: Comparison of processor layer temperature for the case without TECs and with varying heat transfer coefficient (htc) and heat flux (q) values.

For the rest of this section, we will refer to the results corresponding to our proposed hybrid model as *proposed*. Figure 3·11 compares the temperature of the processor layer for the case without TECs, and Figure 3·12 reports the absolute temperature difference between the proposed model and COMSOL. As seen from the figure, there is a good match between the two simulators with an absolute error of less than $0.5°C$ across all htc and q combinations.

Next, we present the comparison results for the case with TECs. Figure 3·13 compares the average temperature of the processor layer over a range of TEC bias currents. For this experiment, $htc = 10^6\ W/m^2K$ and $q = 20\ W/cm^2$. Our proposed TEC model closely follows the temperature results obtained from COMSOL with an

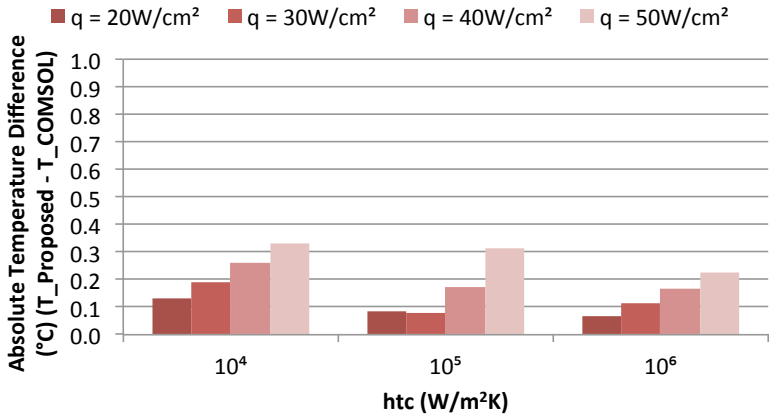


Figure 3·12: Absolute temperature error for the case without TECs and with varying heat transfer coefficient (htc) and heat flux (q) values.

error less than $1.5°C$. As expected, the processor temperature starts to reduce as the TEC bias current increases. At some point (i.e., around 6A), impact of Joule heating becomes dominant, resulting in a slight increase in the processor temperature. In Figure 3·14, we report the cold and hot side temperatures of the TEC for the same simulation. At 0A bias current, $T_{cold} > T_{hot}$ due to the additional resistance presented by the TEC device. At around 0.5A of bias current, amount of heat that is pumped by the TEC overcomes its own resistance and $\Delta T = (T_{hot} - T_{cold})$ becomes positive and starts to increase.

In Figure 3·15, we compare the thermal maps obtained from the two simulations for $q = 20\ W/cm^2$ and TEC current of 4A. The plots on the left correspond to the temperatures of the processing layer, while the plots on the right show the temperatures on the hot side. We carry out similar analysis for other q values ranging from 20 to 50 W/cm^2 and observe that the absolute maximum error is $3.57°C$. We also report $2.07°C$ of average and $2.25°C$ of RMS error.

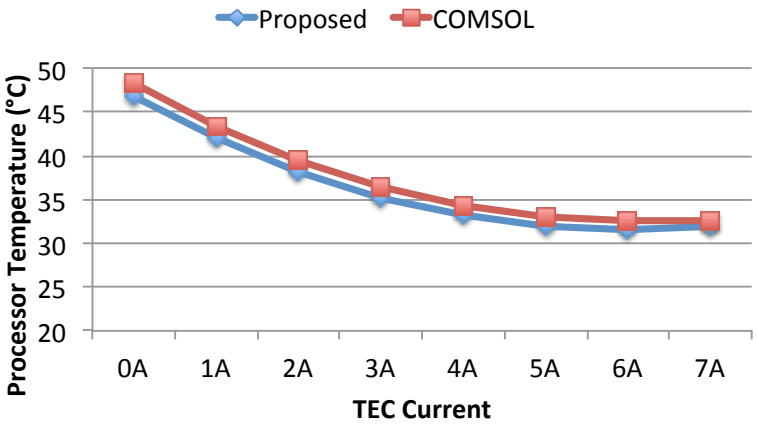


Figure 3·13: Comparison of processor temperature over TEC current for COMSOL and the proposed model. $htc = 10^6\ W/m^2K$ and $q = 20\ W/cm^2$.

Liquid Cooling Model Validation

We validate our microchannel liquid cooling model by comparing it against two different simulators: (i) COMSOL Multiphysics tool, and (ii) 3D-ICE (Sridhar et al., 2010a) simulator, which has been well validated against ANSYS CFX tool. During validation of the 3D-ICE simulator, two different chip stacks were modeled: (i) two active dies and one microchannel layer in between them and (ii) three active dies and four microchannel layers adjacent to them. Experiments with various flow rates and heat flux profiles have been carried out and a maximum temperature error of $1.5°C$ was reported.

For validation of our proposed model in COMSOL, we first create a chip stack with liquid microchannels. Figure 3·16(a) illustrates the cross-section of the chip stack, where the liquid microchannel layer is placed on top of the processor layer, and an additional bulk silicon layer (with $40\mu m$ thickness) is placed on top to provide closure to the microchannels. We simulate a thin slice of this chip stack as in prior work (Sridhar et al., 2010a). The width and length of the slice are $250\mu m$ and 5mm,

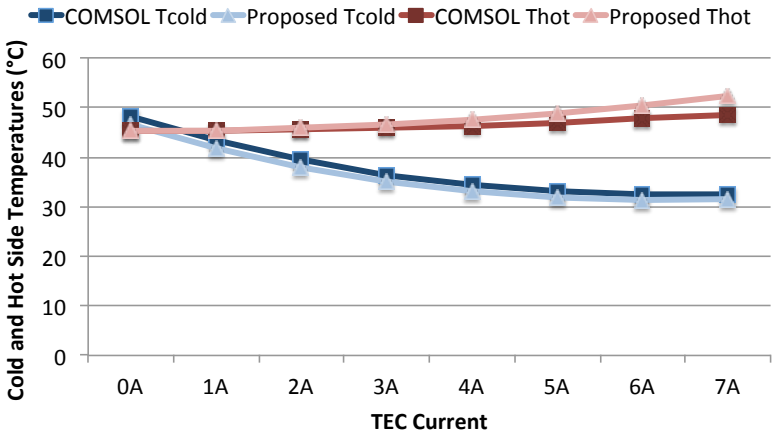


Figure 3·14: Comparison of the cold and hot side temperatures over TEC current for COMSOL and the proposed model. $htc = 10^6\ W/m^2K$ and $q = 20\ W/cm^2$.

respectively. We set the microchannel width as w=50μm (also equal to the wall width) and channel thickness as h=100μm. With these microchannel parameters, the simulated slice includes two microchannels interleaved between three channel walls. At the top surface of the bulk silicon layer, we assign a very small heat transfer coefficient (i.e., $htc = 0.01\ W/m^2K$) to represent minimal convection to air. We assume water as the coolant and use the coolant properties given in Table 3.2.

Similar to the case with TECs, the problem we define in COMSOL is a multi-physics problem, which combines *heat transfer in solids*, *heat transfer in liquids*, and *laminar flow* elements. We use the segregated solver in COMSOL to solve the multi-physics problem, where the segregated step 1 (corresponding to the laminar flow) is an iterative solver using GMRES method, and the segregated step 2 (corresponding to heat flow) is a direct solver using PARDISO method. We construct a fine mesh, which consists of 628237 domain elements, 66162 boundary elements and 4332 edge elements. Number of degrees of freedom solved for is 514554.

We model the same chip stack in 3D-ICE simulator for the second set of comparisons. As the computation of $h_{f,vertical}$ and $h_{f,side}$ coefficients significantly differ in

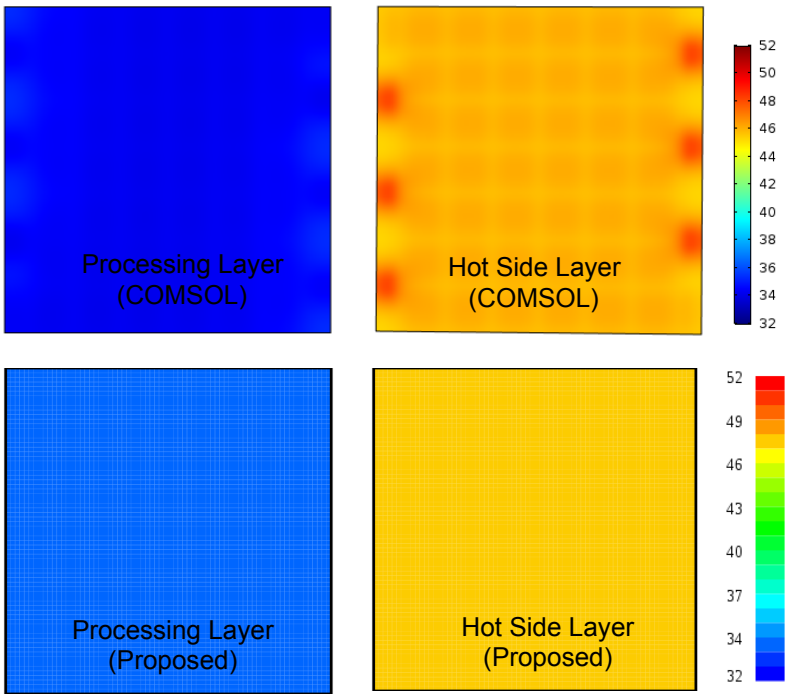
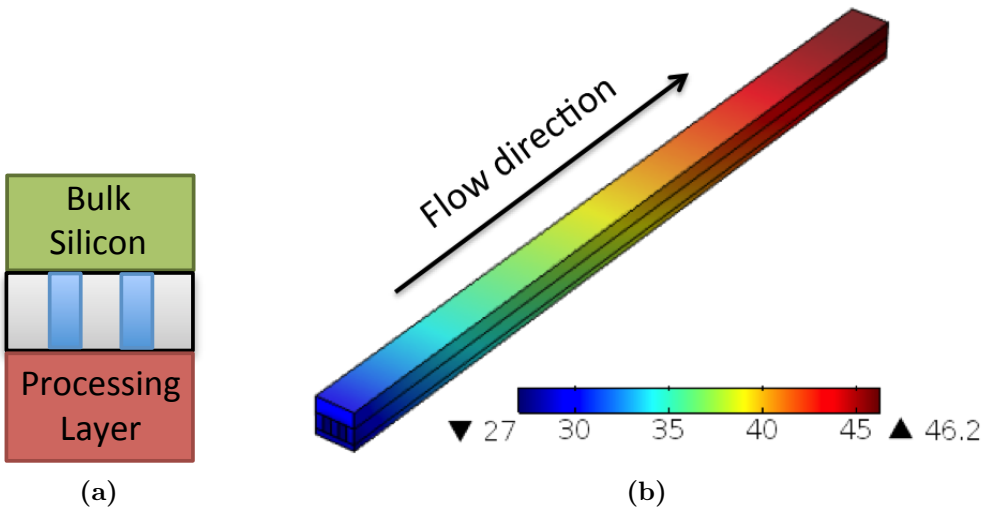


Figure 3·15: Comparison of thermal maps corresponding to the processing layer and the TEC hot side, for COMSOL and the proposed model. $htc = 10^6\ W/m^2K$, $q = 20\ W/cm^2$ and $I = 4\ A$.

COMSOL and 3D-ICE, we first experimentally estimate the coefficients from COMSOL simulations and then use them as inputs to the proposed model and 3D-ICE simulator. This way, we can carry out a consistent comparison of the three models. We extract the coefficients from COMSOL as follows: to find $h_{f,side}$, we select the surface of a side wall facing a microchannel and record the surface average of the total normal heat flux value ($ht.ntflux$ in COMSOL). We then record the surface average of the side wall temperature (T_{wall}), and the volume average of the liquid temperature (T_{liquid}). Finally, we compute $h_{f,side}$ using the equation below:

$$h_{f,side} = \frac{ht.ntflux}{(T_{wall} - T_{liquid})} \tag{3.14}$$

We carry out similar computation for $h_{f,vertical}$ using the top and bottom walls. We



(a) (b)

Figure 3·16: (a) Front view of the thin slice of chip stack we modeled for liquid cooling, (b) Side view of the chip stack as we modeled in COMSOL.

repeat the same steps for the flow velocities that we experiment with and assign the average computed value to the heat transfer coefficients. For our system, we determine that $h_{f,side} \approx h_{f,vertical} = 1.05 \times 10^5\ W/m^2K$. We use these values as inputs to the proposed model and 3D-ICE simulator.

We run steady-state simulations for a range of q values of $12.5, 25, 50$, and $100\ W/cm^2$ as well as for different flow velocities, $u_{avg} = 0.5, 1.0, 1.5, 2.0\ m/s$, and record the maximum temperature of the processing layer for the proposed model, COMSOL, and 3D-ICE. Figure 3·17 shows the maximum processor temperatures obtained from COMSOL, 3D-ICE, and our proposed model for all u_{avg} combinations where $q = 100\ W/cm^2$. Among all experiments, compared to COMSOL simulations, our proposed model provides maximum, average and RMS error of $2.46°C$ (corresponds to 2.8%), $0.36°C$, and $0.72°C$, respectively. In comparison to 3D-ICE simulator, the error of the proposed model is less than $0.04°C$.

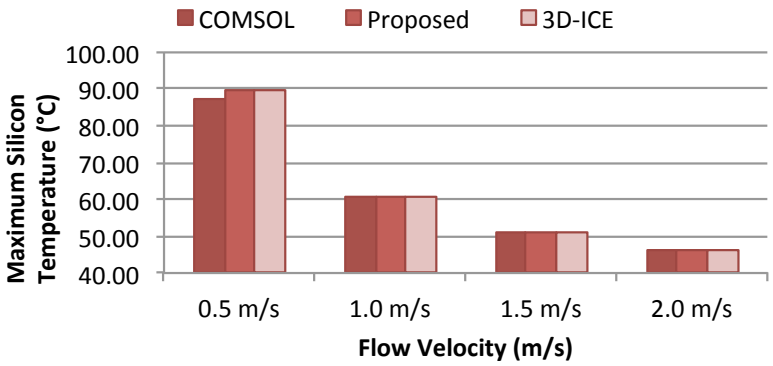


Figure 3·17: Maximum processor temperature comparison for COMSOL, 3D-ICE and the proposed model for $q = 100\ W/cm^2$.

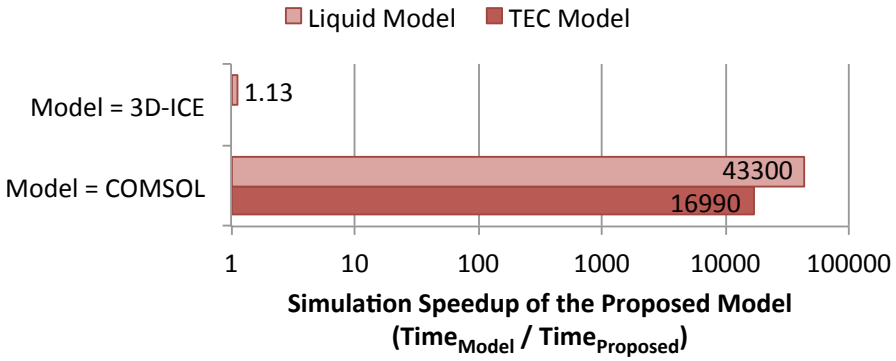


Figure 3·18: Comparison of the simulation speed across three simulators. As 3D-ICE does not have a TEC model, the bar is not shown.

Placement of the Virtual Thermal Node

We observe a number of important aspects of implementing hybrid cooling in compact thermal simulators that could lead to significant inaccuracy if overlooked. One aspect is related to where to place the virtual thermal nodes on the grid cells while constructing the thermal resistance network. As we briefly discussed in Section 3.2.1, HotSpot simulator by default places the virtual node at the bottom surface of a grid cell as shown in Figure 3·9(a). This is very convenient for TEC modeling, where we focus on either side of the TEC cell (i.e., cold and hot sides) when applying Kirchhoff's current law at the nodes and inserting the current terms into the equation (see

Figure 3·9(c)). However, for the liquid cooling model, we have found out that this approach results in significant underestimation of processor temperatures. This is because when modeling the temperature of the liquid cells, one should account for both the heat transferred from the solid cell (conduction) to the walls and from the walls to the liquid (convection). When the virtual node is placed at the bottom surface, the vertical heat transfer from the cell above is fully attributed to convection, while the heat transfer from the bottom cell is fully attributed to conduction (instead of a combination of them from each direction). This asymmetric representation of the resistances creates an affect as if liquid absorbs more heat from the processing layer than it actually does. This assumption also affects the convection in the direction of the flow as the values of the convective terms depend on the temperatures of the south and north faces (i.e., T_{south} and T_{north}) of the liquid cells, eventually resulting in underestimation of the processor temperatures.

We demonstrate this effect in Figure 3·19 for the following system. We assume the same chip stack illustrated in Figure 3·16(a), but simulate a 10mm×10mm die composed of four blocks with equal area, representing a conventional chip. We experiment with $q = 12.5, 25, 50, 100, 200\ W/cm^2$.

As shown, placing the virtual grid at the bottom surface of a grid may result in up to $20°C$ lower processor temperature in comparison to placing it in the middle (which is the adopted approach in the proposed model and gives matching results compared to 3D-ICE). This is an important factor as it would significantly change the outcome when evaluating different cooling designs.

Effect of TIM Assumptions on Fair Comparison of Cooling Designs

A second important aspect is modeling TIM. We will first describe an example case from prior work and show how the assumptions on the TIM thickness and properties may lead to overestimation of TEC benefits.

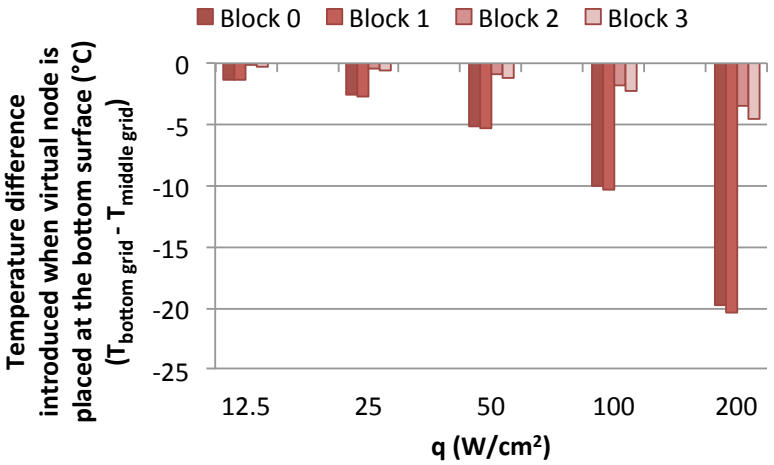


Figure 3·19: Temperature difference introduced when the virtual node is placed at the bottom surface of a liquid cell. Placing the virtual node at the bottom surface results in underestimation of the temperature by up to $20°C$.

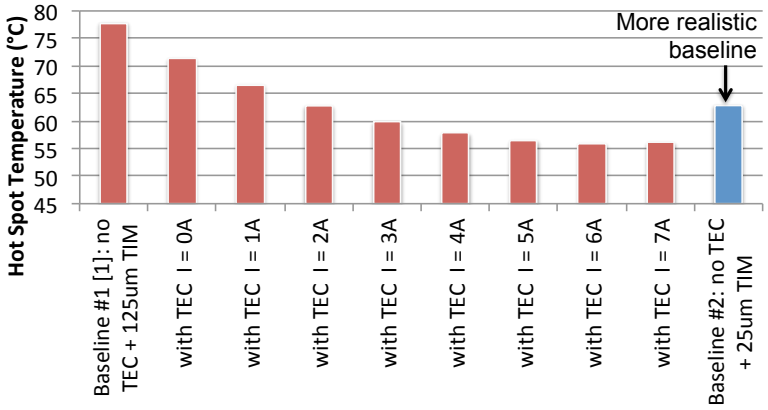


Figure 3·20: Comparison of the hot spot temperatures for pessimistic baseline #1 from prior work (Chowdhury et al., 2009), a more realistic baseline #2, and a system with TECs using different bias currents.

Prior work demonstrates the benefits of TECs regarding the removal of high density hot spots (Chowdhury et al., 2009). There is a very small hot spot area placed at the center of the processing layer. The size of the hot spot is $400\mu m \times 400\mu m$ and hot spot heat flux is $q = 1.25\ kW/cm^2$, while the background heat flux is $q_{bgnd} = 42.7\ W/cm^2$. On top of the processing layer, there is TIM followed by a top packaging

layer representing the heat sink. In order to demonstrate the benefits of TECs, two cases are compared: (i) chip stack with processor, $125\mu m$ TIM, and the package layers, and (ii) chip stack with processor, $25\mu m$ TIM, $100\mu m$ TEC layer with a TEC unit placed above the hot spot, and the package layers. The TIM conductivity was assumed 1.75 W/mK.

Prior work claims that by simply adding the TEC layer, even at 0A bias current, there would be a passive cooling effect introduced by higher thermal conductivity of the TEC material. This claim is true, if we assume a $125\mu m$-thick TIM as the baseline without TECs (let us call this baseline #1). However, in a system without TECs, a much thinner TIM, i.e., one with $25\mu m$ thickness, can be utilized (let us call this baseline #2). Moreover, there are TIM materials with much higher reported thermal conductivities (Jayakumar and Reda, 2015). Using our simulation framework, we evaluate the results of each assumption. As the heat sink properties were not specified in prior work, we assign $htc = 10^6$ W/m^2K without loss of generality to represent the heat sink. We assume a higher quality TIM material from recent work (Jayakumar and Reda, 2015) with 8.5 W/mK conductivity. In Figure 3·20, we compare baselines #1 and #2 against the system with different TEC bias currents and report the maximum temperatures. As seen from the figure, baseline #1 results in about $15°C$ higher temperature compared to a more realistic baseline #2. In fact, when we add TECs and do not apply any bias current, we are introducing additional thermal resistance which increases the temperature by $9°C$ compared to the more realistic baseline #2. However, if one assumes the very thick TIM from baseline #1, it seems like TEC is providing cooling even without being activated, which leads to overestimation of its benefits. For the hot spot heat flux we have in this experiment, TEC starts to provide benefit over the baseline #2 only after 2A of bias current.

We think that such assumptions on the TIM thickness can affect conclusions when

comparing two different cooling designs, thus, are highly important.

Chapter 4

Design-Time and Runtime Optimization Techniques

Design-time and runtime optimization of the system is essential for maximizing the efficiency of systems with advanced cooling. In this chapter, we identify sources of inefficiency that are specific to the target cooling systems and propose solutions through thermal management techniques. When applications are comprised of short bursts of intense parallel computation, high responsiveness becomes important. PCM-based cooling combined with *computational sprinting* algorithms addresses this issue by allowing more cores to be activated during phase transition while keeping the temperature stable. We propose a new ***adaptive sprinting*** (Kaplan and Coskun, 2015) policy that extends this sprinting duration by tracking the PCM state on different locations of the chip and taking runtime actions based on this information.

In addition to the performance characteristics of the applications, the heat dissipation profile of the processor is a significant parameter affecting the overall energy efficiency. Localized hot spots occur at different locations of the chip, considerably limiting the cooling efficiency. Hybrid designs combine multiple cooling solutions, such as TECs and liquid cooling, on the same platform to mitigate hot spots effectively. In order to maximize the benefits of hybrid-cooled systems, we propose a cooling optimization method, *LoCool*, which jointly optimizes the TEC power and liquid pumping power under temperature constraints.

4.1 Adaptive Sprinting for Systems with PCM-based Cooling

This section proposes a novel runtime management technique to improve the performance of multithreaded workloads on systems with PCM. Our proposed ***adaptive sprinting*** policy monitors the remaining PCM energy corresponding to each core at runtime, and using this information, it decides on the number, the location and the V/F setting of the sprinting cores. We first introduce the motivation behind our technique and the details of *adaptive sprinting* policy. We then evaluate our policy by comparing its performance against the state-of-the-art sprinting policies using a full system simulation framework.

The current research focuses on using PCM in the context of *computational sprinting* to extend sprinting duration. Prior techniques on *computational sprinting* alternate between sprinting with all cores and not sprinting by switching to idle mode or single core operation (Raghavan et al., 2012; Tilli et al., 2012; Raghavan et al., 2013; Shao et al., 2014). However, existing techniques ignore the following observation: due to the inherent heterogeneous heat distribution across a chip, different parts of PCM melt at different rates depending on their location. For example, the center cores typically get hotter and force the center part of the PCM to melt faster. When center cores exhaust their PCM capacity and hit a temperature threshold, the side cores still have thermal headroom to continue sprinting. Thus, sprinting with an *all or nothing* approach as in prior work wastes the yet unused PCM capacity, leading to substantially suboptimal performance. On the other hand, if we monitor the remaining PCM energy at various locations and take actions based on that, we can utilize the PCM heat storage capability much more efficiently.

Similarly, existing techniques do not consider the application's power consumption and assume a fixed sprinting power. These policies operate under the worst case power

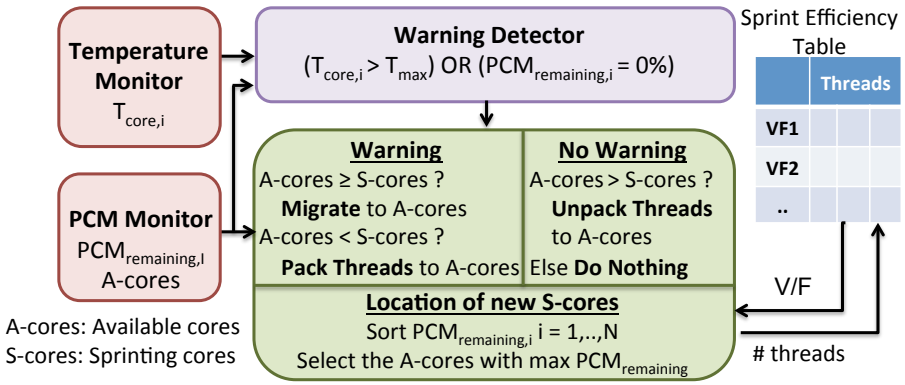


Figure 4·1: Proposed adaptive sprinting policy flowchart.

consumption scenario, and thus, potentially incur performance losses for applications that consume lower power. In fact, factors such as application's power consumption or the number of cores to sprint with are significant factors in determining the sprinting duration. Sprinting policies that do not consider these factors cannot exploit full benefits of PCM. The proposed *adaptive sprinting* policy addresses these limitations of the existing sprinting techniques and provides further performance boost.

4.1.1 Adaptive Sprinting Policy

The goal of our *adaptive sprinting* policy is to operate in sprinting mode as long as possible by leveraging the observations described previously and exploiting the PCM capacity to near exhaustion. By monitoring the PCM state, our policy determines how much sprinting capability is left for each core. We also determine the most sprint-efficient V/F setting using a lookup table. Based on this information, the policy decides on (a) the number, (b) the locations, and (c) the V/F setting of the sprinting cores. The policy changes the number of sprinting cores at runtime by applying thread packing (or unpacking), which refers to pinning the threads to a lower (or a higher) number of cores (Raghavan et al., 2013; Hankendi et al., 2013; Reda et al., 2012).

Figure 4·1 gives an overview of our *adaptive sprinting* policy. We first introduce

the terminology we use to describe our policy. The active cores are the *sprinting cores* (*S-cores*). *Available cores* (*A-cores*) are the cores which have more than 0% remaining unmelted PCM above them, and have lower temperature than the critical temperature (i.e., $T_{max} = 80^oC$). *A-cores* can be active or idle at a given time. A *warning* is raised if for any of the cores, the PCM portion that lies above that core is fully melted (i.e., remaining PCM storage capacity falls to 0%) **or** if the core temperature exceeds T_{max}.

Our policy checks for a *warning* every 50 ms (a temperature sampling rate that incurs negligible overhead in real systems) and if there is a *warning*, it determines the number of *A-cores* by checking their PCM capacity. If the number of *A-cores* is higher than or equal to the number of *S-cores*, we merely migrate the threads to the *A-cores*. If not, we continue sprinting with fewer cores by packing the threads to the *A-cores* (i.e., binding the threads to a lower number of cores). While thread packing, in order to determine the location of the new *S-cores*, we sort the remaining PCM capacities of the cores and select the ones that have maximum amount of remaining unmelted PCM. If we are left with no *A-cores*, we put all cores to idle state until some portion of the PCM capacity is recovered (i.e., 10%).

In order to determine the V/F setting at a given time, we follow an offline analysis approach. For this purpose, we run all benchmarks for each of the {thread count, V/F setting} pair when no management policy is applied. We record the original application running times (T_{run}) and the number of instructions executed ($Inst_{spr}$) until the first thermal violation. T_{run} is a measure of performance for the given pair. $Inst_{spr}$ represents how much work can be done when sprinting at a given setting until thermal violation. Based on these recordings, we define a new metric, *sprint efficiency* = $Inst_{spr}$ / T_{run}. Choosing the pair with higher *sprint efficiency* corresponds to choosing a configuration with higher performance while considering

the tradeoff between power and allowed sprinting capability. We create a lookup table of *sprint efficiency* values and our policy selects the V/F setting with the maximum *sprint efficiency* for a given thread count.

Our policy also addresses the fact that while the *S-cores* are using up the PCM capacity in parts of the chip, PCM capacity is being recovered around the idle cores. PCM recovery may also occur when a benchmark enters a low power phase. Thus, in case of no *warning*, the policy checks if there are more *A-cores* than the current number of *S-cores*. If there are, sprinting continues with the number of *A-cores* by *unpacking* the threads.

Monitoring the PCM capacity: An important aspect of our policy is that it takes actions based on the current PCM state. We monitor the percentage of melted PCM for each core individually (i.e., for each core, we track the latent heat stored in the PCM portion that lies on top of that core and has the same area as that core). In this way, at a given time, we know the sprinting capability for each core. In our HotSpot simulations, we monitor the percentage of melted PCM corresponding to each core using Equation (3.1) (i.e., % of melted PCM for a grid cell increases linearly within the (T_1, T_2) interval). In real systems, a similar estimation can be done using a soft PCM sensor that accumulates the amount of energy stored in the PCM during phase change. Details regarding the real-life implementation of such soft PCM monitor can be found in Section 3.1.4.

Performance Impact of Thread Packing: Our policy decreases the number of sprinting cores by binding the threads to a subset of available cores. In that case, the active threads get multiplexed on the active cores, which may incur performance penalty due to synchronization and context switches. For example, binding all 16 threads to a single core may give worse performance than running with a single thread. For PARSEC benchmarks (Bienia, 2011), prior work reports that the performance

degradation due to thread packing is less than 3.6% (for an 8-core system (Reda et al., 2012)) and is 7.3% on average (when packing 12 threads to 4 cores (Hankendi et al., 2013)). For benchmarks whose performance is severely affected by thread packing, a task-queue based worker thread execution framework can mitigate this effect (Raghavan et al., 2013). In our simulations based on a 16-core system, when using the *adaptive sprinting* policy, we observe that the number of sprinting cores does not fall below 10. Thus, we assume that the performance of packing the threads to N cores is equal to running with N threads with negligible additional penalty.

4.1.2 Baseline Sprinting Policies

We implement a large subset of the state-of-the-art sprinting policies to compare with our proposed policy. This section describes them in detail.

Truncated Sprints: This policy (Raghavan et al., 2012) activates all cores of a system at the highest V/F level (i.e., full intensity) during sprinting. Sprinting is truncated if the PCM capacity is fully exhausted or if any core temperature exceeds T_{max}. Upon sprint truncation, execution continues on a single core and all other cores are put to idle mode until the application finishes (i.e., *sustained mode*). In this policy, as the application running time gets longer, more time is spent in the *sustained mode*, which overshadows the benefits of sprinting. We implement an improved version of this policy and use it for comparison. After a truncated sprint, we allow re-sprinting if some portion of the PCM capacity is recovered (i.e., 10% per core), as opposed to running in *sustained mode* until the benchmark execution ends.

Fixed Duty Cycle Sprinting: This policy has been proposed for extended computations (Raghavan et al., 2013; Shao et al., 2014). It alternates between the sprint and rest modes (i.e., all cores are put to idle state) based on a fixed duty cycle. Duty cycle (D) is determined by the ratio between the sustained power and the sprinting power

to allow enough time to cool down after sprinting. For example, for 1 W of sustained power and 10 W of sprinting power, $D = 1 : 10$. Assuming a sprinting duration (i.e., the time it takes to reach T_{max} while sprinting) of 1 second, this corresponds to 9 seconds of rest time. Some limitations of this policy are as follows: (1) It assumes a fixed sprinting power and D for all benchmarks. Having a fixed duty cycle requires considering the worst case scenario (i.e., the highest possible sprinting power) while setting D in order to avoid thermal violations. Thus, for benchmarks that consume lower power, rest time is longer than needed, which incurs performance penalties. (2) It poses significant performance loss in some cases. For example, an application that originally completes in a little over 1 second would wait for 9 seconds in rest mode to complete the remaining small portion of the work.

Sprint Pacing: This policy (Raghavan et al., 2013) sprints with full intensity until half of the PCM thermal capacity is consumed. After that, it switches to a lower intensity sprint by keeping all cores active, but changing to the lowest V/F setting. However, prior work does not address how this policy behaves in case of a thermal violation at the lowest V/F setting. Thus, we implement two different versions of this policy: *sprint pacing*, which does not take any action after switching to the lower intensity sprinting, and *modified sprint pacing*, which puts the cores to idle mode (until 10% of the PCM capacity is recovered) if a thermal violation occurs during the lower intensity sprint.

Reactive DVFS: This policy represents the DVFS policies in current processors and it is oblivious to the PCM state. *Reactive DVFS* decreases the V/F setting by one step upon temperature violation in any of the cores. If the violation occurs at the lowest V/F setting, all active cores are put to idle mode. After cooling down and recovering a certain thermal headroom (i.e., 2^oC) to T_{max}, cores continue executing and V/F setting is increased in steps.

4.1.3 Performance Evaluation

Full System Simulation Infrastructure

We simulate a 16-core processor with private L2 caches, in which the core architecture is based on the AMD Opteron 6172 processor manufactured using a 45 nm silicon on insulator process. The architectural parameters for the cores and caches are taken from recent work (Conway et al., 2009).

Our simulation framework consists of microarchitectural performance simulation (Gem5 (Binkert et al., 2006)), power simulation (McPAT (Li et al., 2009) and CACTI (Thoziyoor et al., 2008)), temperature simulation (HotSpot), and a database that decouples time-consuming performance and power simulation from thermal simulation as shown in Figure 4·2.

We run each benchmark for 1 billion instructions in detailed mode in their parallel phase, and collect performance statistics every 2 million instructions with a total of 500 samples. We calibrate the McPAT dynamic core power values based on real measurements collected on the Opteron processor. We scale the CACTI values based on cache access rates. Figure 4·2 shows the available V/F levels and the corresponding average core powers for our processor. We use the HotSpot default package properties, except that we use a $1mm$ thick heat sink with 0.2 K/W convection resistance to represent a system without an advanced heat sink.

As there is a large time scale gap between performance-power simulations and temperature simulation, we decouple these two parts using an approach from prior work (Coskun et al., 2009c). We first generate a database of performance and power traces for each benchmark. The database maintains power and time information for each 2 million instruction frame for each application at all possible thread counts and V/F settings. At every sampling interval, HotSpot polls the database for acquiring the power data of the corresponding benchmark at a specific instruction count. As each

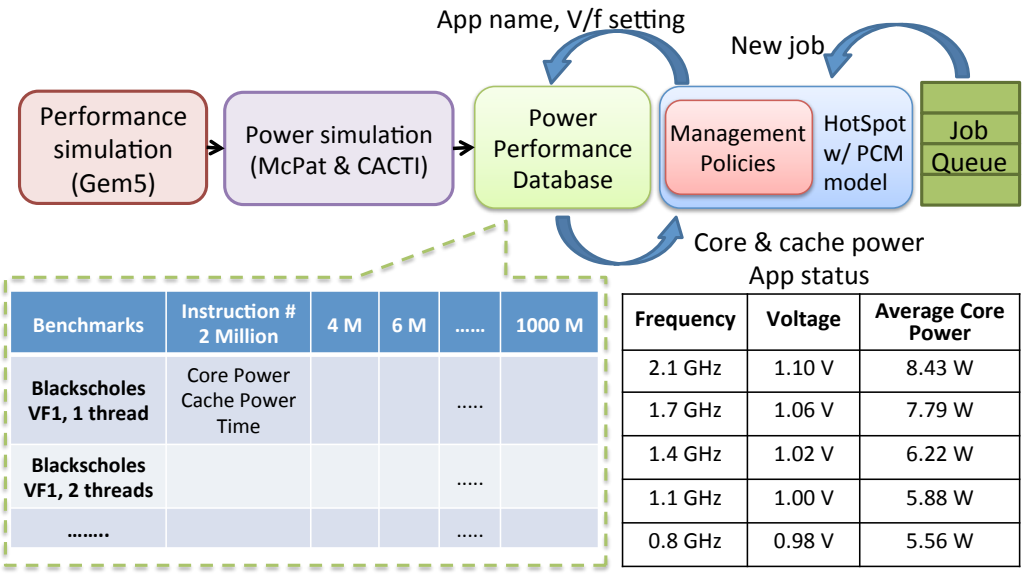


Figure 4·2: Performance, power and temperature simulation framework.

cell in the database represents an instruction frame and not time, we can switch from one V/F setting to another by reading from the cell of the desired V/F setting in the next instruction frame. Thus, we can apply DVFS policies or change the thread count (i.e., thread packing) at runtime. This framework has acceptable accuracy because (i) each core has private caches, (ii) V/F scaling is applied to all cores at the same time (which is reasonable for the type of multithreaded benchmarks we run), and (iii) thread packing overhead is small (see Section 4.1.1). For evaluating any policy that uses DVFS or thread migration, we include the DVFS and migration overheads, which are reported as less than 200 μs (Park et al., 2013) and 1 ms (Coskun et al., 2009c), respectively.

We run benchmarks from the PARSEC Suite (Bienia, 2011) as our workload. For each benchmark, we generate performance and power traces at various thread counts (i.e., 1, 2, 4, 6, 8, 10, 12, 14, and 16) using the sim-large input set. We assume equal power consumption for individual threads of an application as inter-thread differences are minimal for PARSEC running on the Opteron CPU.

As our PCM, we assume *cerrobend* material, which has a high thermal conductivity (19 W/mK) and a high latent heat of fusion (305×10^6 J/m^3). We also assume a metal mesh structure that contains the PCM, prevents it from mixing around and provides a higher effective thermal conductivity.

Experimental Results

We evaluate our *adaptive sprinting* policy by comparing against the baseline policies and the case where no management policy is applied. In the no management case, benchmarks run using 16 threads at the highest V/F setting, which gives the ideal performance.

Figure 4·3 shows the running times of the individual benchmarks normalized to the no management case. Figure 4·3(a) corresponds to the thermally-aware policies with negligible to no thermal violation, while Figure 4·3(b) shows results for the policies that cause significant thermal violation. As indicated, *truncated sprints* and *fixed duty cycle* policies result in the worst performance. Performance of benchmarks such as *blackscholes* and *swaptions* are severely degraded by *truncated sprints* (running times reaching up to 4.2x of their ideal value). This is because the performance of these applications scale well with the number of threads, thus, truncating a sprint leads to losing the benefit of performance scalability. *Fixed duty cycle* sprinting results in similar performance for all benchmarks, however, the penalty is slightly higher for the benchmarks that consume lower power (e.g., *x264*).

Sprint pacing provides similar performance as *adaptive sprinting*, but *sprint pacing* results in a maximum temperature of $87°C$ and causes temperature violation for up to 60% of the time as illustrated in Figure 4·4. On the other hand, *modified sprint pacing*, which is a thermally-aware version of *sprint pacing*, does not perform as well. Proposed *adaptive sprinting* policy also provides higher performance than *reactive DVFS*. Keeping all cores active and merely applying *reactive DVFS* cannot mitigate

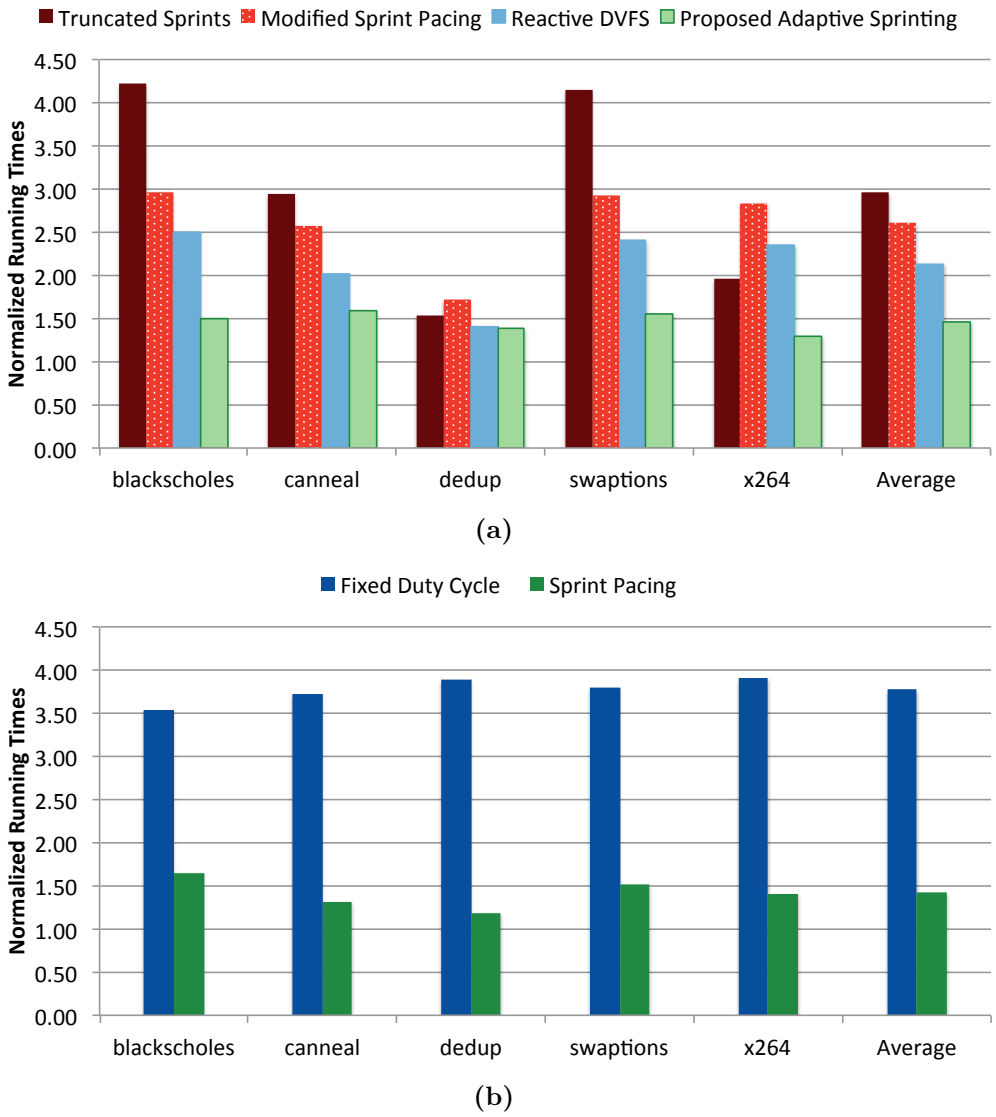


Figure 4·3: Running times of the benchmarks normalized to the no management (ideal) case for each application for (a) policies that are thermally-aware and (b) policies that cause significant thermal violation.

the temperature problem. Thus, once the thermal headroom is exhausted, *reactive DVFS* has to put the cores to idle. On the other hand, *adaptive sprinting* allows extended sprints with fewer cores and avoids idling. Our policy provides 29% and 42% higher the performance on average compared to the *reactive DVFS* and *modified sprint pacing*, respectively, without exceeding the thermal limits.

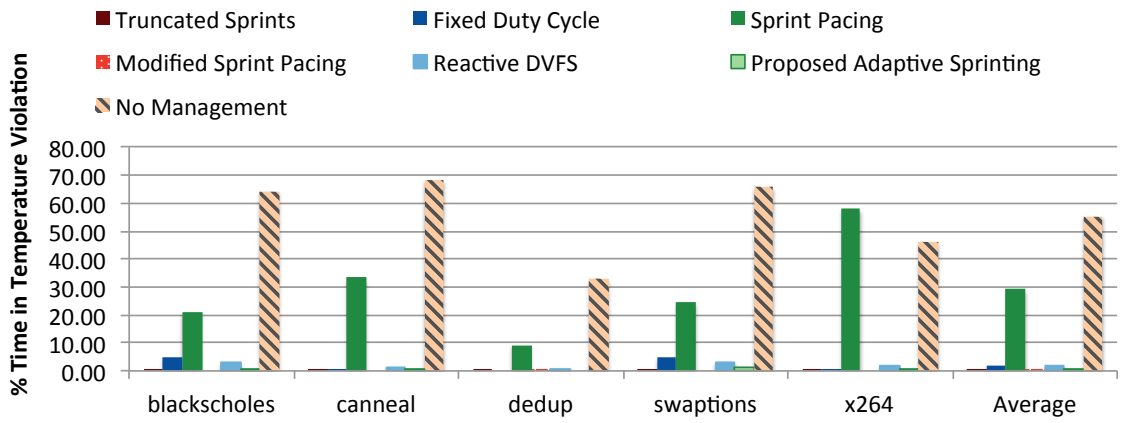


Figure 4·4: Percentage of thermal violation (i.e., time spent above $T_{max} = 80°C$) for each sprinting policy.

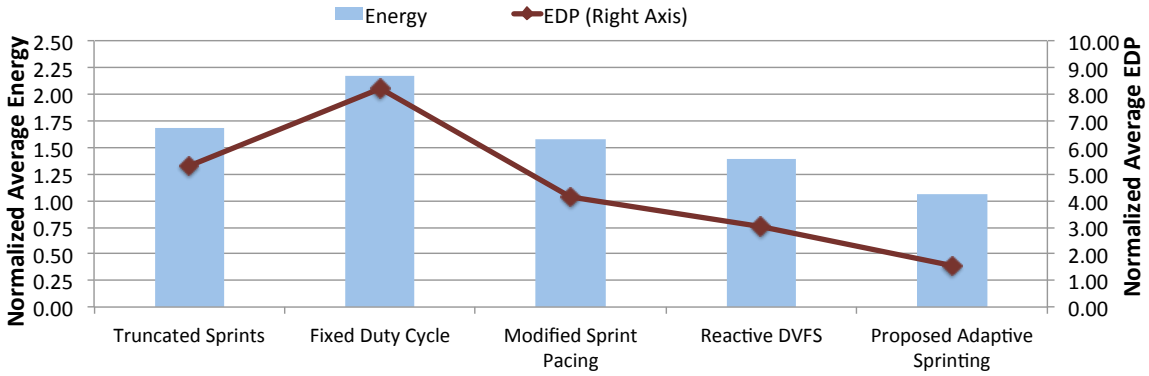


Figure 4·5: Average energy and EDP normalized to the no management case.

Figure 4·5 shows the average energy and EDP values normalized to the no management case for the thermally-aware policies. *Adaptive sprinting* saves energy by 22% and 32% on average in comparison to the *reactive DVFS* and *modified sprint pacing* policies, respectively. It also provides 43% and 59% lower EDP on average compared to the *reactive DVFS* and *modified sprint pacing* policies, respectively.

4.2 Fighting Hot Spots Locally with Hybrid Cooling

The current trend in processor cooling is to design the system to remove the worst-case TDP per unit area. However, large spatial variations in cooling demand exist across the chip. Localized hot spots occur at different locations with varying area and intensities. Hot spots with areas as small as 0.04 mm^2 and with heat fluxes reaching 1 kW/cm^2 are anticipated in next generation processors (Schultz et al., 2016; Lu et al., 2016). This heterogeneity in on-chip heat distribution is likely to increase with the integration of heterogeneous architectures on a single die, such as CPUs, GPUs, and FPGAs. Designing a homogeneous cooling system to remove such high (but local) power densities leads to undercooling of the hot spots or overcooling of the rest of the chip; thus, significantly lowers the efficiency. In order to achieve high cooling efficiency and reduce cooling power, it is essential to customize the cooling subsystem based on the demand across the chip.

Hybrid cooling strategies address this problem by combining the strengths of different cooling methods and localizing the cooling effort over the hot spots. One such hybrid design includes TECs and liquid cooling on the same platform (Sahu et al., 2015; Yazawa et al., 2012; Hu et al., 2013), where TECs are placed between the processing layer and liquid microchannel layer around the hot spot areas. Microchannel liquid cooling is well-suited to remove the background heat on large chips and 3D-stacked architectures. However, the fluid gets hotter as it flows along the channels; thus, the heat removal capability decreases on the locations that are far away from the liquid inlet (Sabry et al., 2013). TECs are successful in handling high power densities in small areas, but they consume a significant power when used for cooling large areas. When used in cooperation in a hybrid cooling framework, they provide considerably higher cooling efficiency compared to the homogeneous designs.

Existing work on hybrid TEC and liquid cooling mostly focuses on optimizing

the dimensions and bias current of TECs, assuming a fixed operating point for liquid cooling. However, we observe that liquid flow rate has an impact on both the liquid pumping power and the cooling performance of the TECs; thus, a co-optimization approach is necessary to achieve high efficiency. To this end, this thesis propose *LoCool*, a hybrid cooling optimization algorithm to maximize cooling efficiency in systems with high heat flux hot spots. Given a chip power map and thermal constraints, *LoCool* jointly tunes both cooling methods, namely the liquid flow rate and the bias current for the TEC units, to meet the given temperature constraint using minimum cooling power.

The rest of this section starts with a discussion of the factors contributing to the cooling power of a hybrid-cooled system with TEC and liquid cooling. We then describe the details of the *LoCool* optimization method. We finally evaluate the cooling efficiency of a hybrid design optimized with *LoCool* against the unoptimized hybrid designs as well as a design that uses liquid cooling only.

4.2.1 Cooling Power Models

The cooling power of a system with liquid cooling and TECs depend on two main parameters: (i) the power consumed by the pump to push liquid across the microchannels[1], and (ii) electrical power consumed by the TECs when a bias current is applied. Pumping power depends on the channel geometry and the liquid flow rate. TEC power on the other hand, depends on the bias current, electrical resistivity and the temperatures of the cold and hot sides.

Cooling power for an individual liquid-cooled system is calculated as follows (Sabry et al., 2013):

[1]In a data center setting, there is also external chiller power that is impacted by cumulative characteristics of a number of systems. We focus on a single liquid-cooled system in this work.

$$P_{pump} = \frac{\Delta P \cdot V}{\eta} \tag{4.1}$$

$$\Delta P = \frac{2 \cdot f_r \cdot \rho_{coolant} \cdot u^2 \cdot L}{d_h} \tag{4.2}$$

$$d_h = \frac{2h \cdot w}{h + w} \tag{4.3}$$

where ΔP (Pa) is the pressure drop across the channel and V is the total volumetric flow rate (m^3/s), and η is the pump efficiency (generally between 10-40%). f_r, $\rho_{coolant}$, and u are the friction factor, coolant density, and average coolant velocity, respectively. L is the length and d_h is the hydraulic diameter of the channel. h and w are channel height and channel width. Friction factor was derived in prior work for fully developed conditions as follows:

$$\begin{aligned} f_r \cdot Re = 24 \cdot (1 - 1.3553AR + 1.9467AR^2 \\ -1.7012AR^3 + 0.9564AR^4 - 0.2537^5) \end{aligned} \tag{4.4}$$

$$AR = min\{h/w, w/h\} \tag{4.5}$$

$$Re = \frac{u \cdot d_h \cdot \rho_{coolant}}{\mu} \tag{4.6}$$

$$\tag{4.7}$$

where Re is Reynold's number given for laminar flow conditions (i.e., $Re \leq 2300$) and AR is the channel aspect ratio. μ is the dynamic viscosity of the coolant.

The power consumed by the TECs is given by the following formula:

$$P_{tec} = Q_h - Q_c = N(SI(T_h - T_c) + I^2 R_e) \tag{4.8}$$

where I and R_e are the bias current and electrical resistivity of the TECs. The values we use for all constant parameters are listed in Table 4.1. We account for both

h	$100\mu m$	μ	$8.89 \times 10^{-4} Pa.s$
w	$50\mu m$	η	25%
$k_{coolant}$	$0.6069 W/mK$	S	$301\mu V/K$
L	$20mm$	h_{tec}	$8\mu m$
C_v	$4181 J/kgK$	h_{AlN}	$46\mu m$
T_{inlet}	$27°C$	ρ_{tec}	$1.08 \times 10^{-5} Ohm.m$
$\rho_{coolant}$	$998 kg/m^3$	k_{tec}	$1.2 W/mK$
u_{avg}	$\leq 2.6 m/s$	k_{AlN}	$285 W/mK$

Table 4.1: The parameters used for liquid microchannel and TEC models in hybrid cooling optimization.

the pump and TEC power in our experiments, and integrate the described power computation model in our simulation framework.

4.2.2 LoCool Optimization Technique

The **goal** of our algorithm is to find the {*coolant flow velocity, TEC current*} pair that minimizes the total cooling power while meeting temperature and cooling technology constraints. A formal definition of the optimization problem is as follows:

$$\begin{aligned} \text{minimize} \quad & P_{pump}(u) + P_{tec}(I) = \alpha \cdot u^2 + \beta \cdot I^2 \\ \text{subject to} \quad & T(u, I) < T_{max} \\ & u \leq u_{max} \\ & 0 \leq I \leq I_{max} \end{aligned} \tag{4.9}$$

where α and β are constants determined by the channel geometry and TEC properties. We compute $\alpha = 0.4954$ and $\beta = 0.057$ according to our system. T_{max} is the maximum temperature constraint, while u and u_{max} are average and maximum allowed coolant velocities, respectively. Maximum applied pressure drop recommended by manufacturers determines u_{max}. We use 3.3 bar for maximum pressure drop, which corresponds to $u_{max} = 2.6\ m/s$ for our geometry. We use $I_{max} = 7\ A$ as the maximum TEC current constraint (Chowdhury et al., 2009). We use a simplified version of the TEC power in our goal function as the quadratic portion dominates the TEC power.

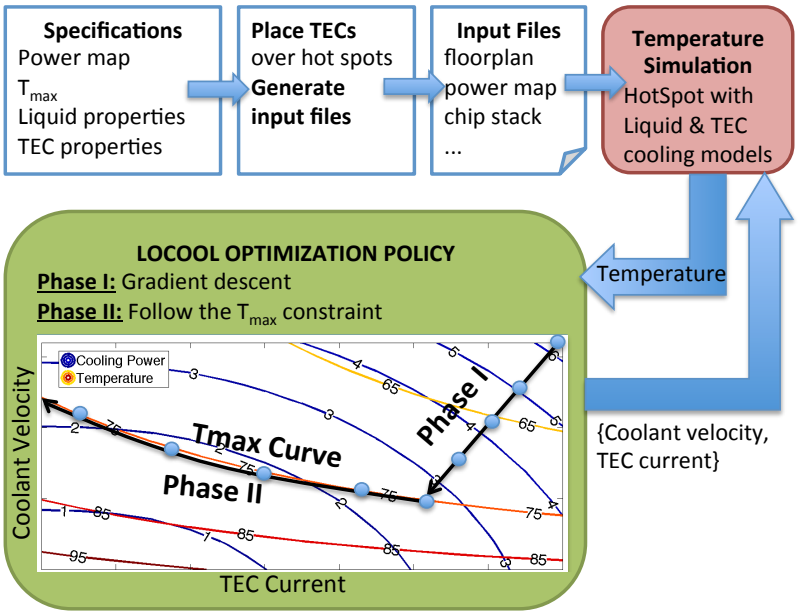


Figure 4·6: *LoCool* optimization flow.

LoCool optimization flow is illustrated in Figure 4·6. Given a power density map with several hot spot areas, we place TEC units above each hot spot. We assume power density maps where the hot spot heat flux is much higher than the background heat flux to model future high-performance systems as suggested in prior work (Schultz et al., 2016) (up to 40× difference has been reported). Any block with over 10x heat flux compared to the background is recognized as a hot spot. We use an iterative approach, where we call the HotSpot simulator at every iteration to check whether the temperature constraint is met. *LoCool* is composed of two main phases, where Phase I is the *descending* phase and Phase II is *following the temperature constraint*. Phase I starts from the highest cooling power setting and descends to lower cooling power settings using a *gradient descent* algorithm. *Gradient descent* is a first order iterative optimization algorithm, where one takes steps proportional to the negative gradient of the function to be minimized. At each iteration, the algorithm decreases each variable (i.e., flow velocity and TEC current) by a fraction of the gradient with respect to that variable. This way, during the descent, cooling

Inputs: T_{max}, u_{max}, I_{min}, I_{max}, α, β
Initialize: $u \leftarrow u_{max}$, $I \leftarrow I_{max}$, $i \leftarrow 0$
1: $f(u, I) = \alpha \cdot u^2 + \beta \cdot I^2$
2: **while** True **do**
3: $u(i+1) = u(i) - \gamma_u \frac{\partial f(u(i), I(i))}{\partial u}$
4: $I(i+1) = I(i) - \gamma_I \frac{\partial f(u(i), I(i))}{\partial I}$
5: $u(i) \leftarrow u(i+1)$, $I(i) \leftarrow I(i+1)$
6: $T \leftarrow HotSpot(u(i), I(i))$
7: $i \leftarrow i + 1$
8: **if** $|T - T_{max}| < 1°C$ **then**
9: **break**
10: **end if**
11: **end while**

Algorithm 1: Gradient Descent

power decreases, while temperature increases. Phase I ends when T is in the close vicinity of T_{max}, i.e., $|T - T_{max}| < 1°C$. Using the *gradient descent* algorithm, we can approach the maximum temperature constraint curve fast by following a steep path as shown in Figure 4·6.

In Phase II of *LoCool*, we follow the temperature constraint curve in the direction of decreasing cooling power. For this purpose, we leverage our observations on how the temperature and cooling power curves change based on the $\{u, I\}$ pairs. Figure 4·7 is a contour plot showing equal cooling power and temperature curves for a range of $\{u, I\}$ pairs. Phase II starts on a point that is close to the T_{max} curve. Due to the shape of the temperature curves, in order to minimize power, one needs to either (i) go up and left or (ii) down and right depending on where we are located on the curve. For example, if Phase I ended on the bottom right point of the curve (e.g., $\{u, I\} = \{1.5\ m/s,\ 6.0\ A\}$) and $T_{max} = 75°C$, then we need to go up and left (decrease current and increase velocity) to minimize power consumption. Similarly, if we are on the top left part (e.g., $\{u, I\} = \{2.2\ m/s,\ 0.5\ A\}$), we need to go down and right. To decide on which direction we should go, we compute $D = \nabla_{\vec{d}} f(u, I)$, which is the derivative of $f(u, I)$ in the direction of $\vec{d} = 0.1\vec{i} - 0.5\vec{j}$, where $\vec{i}$ and $\vec{j}$ are

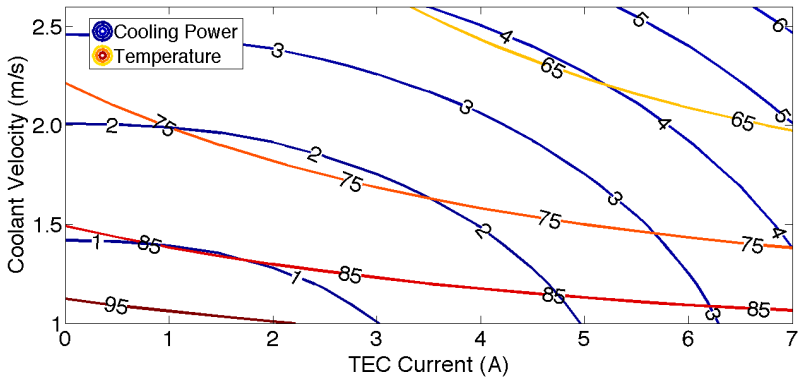


Figure 4·7: Contour plot showing equal cooling power and temperature lines.

the unit vectors in the cartesian coordinates. $\vec{d}$ represents an up and left motion and D changes from a negative value to an increasing positive value along a temperature curve. Once we decide on one of the two directions, we follow the direction by alternating between vertical and horizontal moves. We keep updating the minimum cooling power that meets the thermal constraints along the path. Phase II ends when we reach a boundary of valid $\{u, I\}$ pairs.

We evaluate the optimality of our algorithm by comparing its results against exhaustive search of all possible $\{u, I\}$ pairs. We tested 12 small examples and *LoCool* was able to find the optimum setting for all cases in less than 23 iterations.

4.2.3 Experimental Methodology and Results

In this section, we evaluate the benefits of hybrid cooling designs that are optimized using *LoCool* in comparison to using liquid cooling only. We use the proposed hybrid cooling thermal model described in Section 3.2.1 for evaluation. We experiment with a set of heat flux values and report the resulting total cooling power for the two designs.

Our target hybrid system is illustrated in Figure 4·8, where TECs are placed above of the hot spot areas of the processing layer and a microchannel liquid cooling

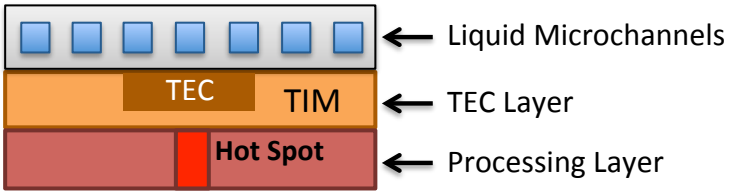


Figure 4·8: Front view of an example hybrid design combining microchannel liquid cooling and TECs. TECs are placed on top of high heat flux areas to remove hot spots, while microchannels are used to remove the heat pumped by the TECs and the background heat.

layer is placed on top. The geometry and the properties of the TECs and the liquid microchannels are given in Table 4.1. We assume a large chip with 20mmx20mm size. The background heat flux (BGHF) is set to 50 W/cm^2. We define hot spot blocks with $500\mu m$x$500\mu m$ size and we vary their location and hot spot heat flux (HSHF). We experiment with HSHF values of $1000, 1300, 1500$, and 2000 W/cm^2, following examples from prior work (Schultz et al., 2016; Chowdhury et al., 2009). We adopt the TEC size of 3.5mmx3.5mm from prior work (Chowdhury et al., 2009). We compare the minimum cooling power for the liquid cooled system against the hybrid cooling system for varying temperature constraints (i.e., $T_{max} = 85, 80, 75°C$). For the hybrid cooling case, we report the results we obtain from our *LoCool* algorithm.

Figures 4·9 and 4·10 show a subset of the results for a single hot spot case with HSHF of 1000 and 1300 W/cm^2, respectively. Hot spot is located close to the outlet of the channels. Figures 4·9 and 4·10 indicate that an optimized hybrid cooling system saves significant cooling energy by focusing the cooling effort on the hot spot. For HSHF=1000, hybrid cooling with *LoCool* saves cooling power by 9%, 16%, and 22% at $T_{max} = 85, 80, 75°C$, respectively. Intuitively, power saving increases for higher HSHF values. At HSHF=1300, *LoCool* provides up to 28% cooling power savings. The simple explanation is that as the temperature constraint gets tighter and hot spots get denser, liquid cooling starts to pump coolant at a much higher rate just to cool the hot spots. On the other hand, hybrid cooling focuses the cooling effort on

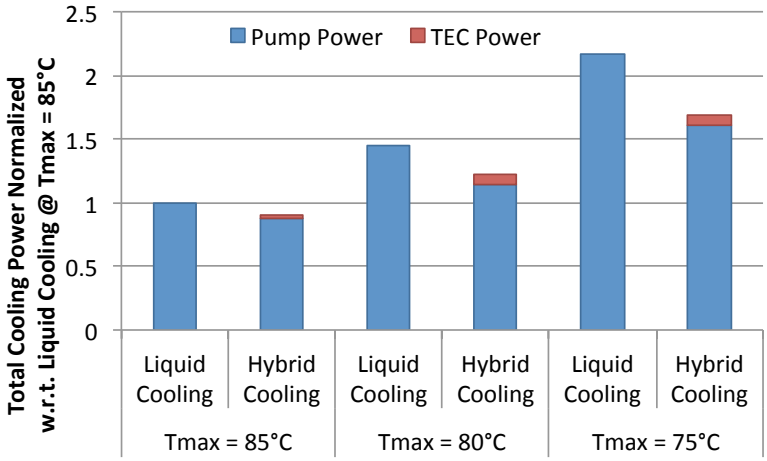


Figure 4·9: Total cooling power comparison of liquid and hybrid cooling for hot spot heat flux (HSHF) of 1000 W/cm^2. Results are normalized to liquid cooling at $T_{max} = 85°C$.

the hot spot and meet the same temperature constraint at a lower flow rate, thus, providing a more gradual cooling power curve.

An interesting observation is that liquid cooling cannot satisfy the $T_{max} = 75°C$ constraint at HSHF=1300 without exceeding the maximum pressure drop limit. Hybrid cooling, however, is able to meet that constraint using $\{u, I\} = \{2.2\ m/s,\ 3.0\ A\}$ settings, which is a significant achievement considering that 2.2 m/s corresponds to only 85% of the maximum pressure drop limit. Similarly, for the highest heat flux case (i.e., HSHF=2000), *LoCool* can cool the hot spot down to $80°C$ by biasing the TECs with maximum current, while liquid cooling fails to meet any of the temperature constraints.

In comparison to hybrid designs where TECs are combined with fans, using TECs with liquid cooling provides higher cooling efficiency. The reason is that TECs require some form of cooling mechanism to remove the heat pumped to the hot side in order to avoid self heating. Liquid cooling acts as a very efficient heat sink improving the TEC performance as it achieves much lower thermal resistance compared to conventional

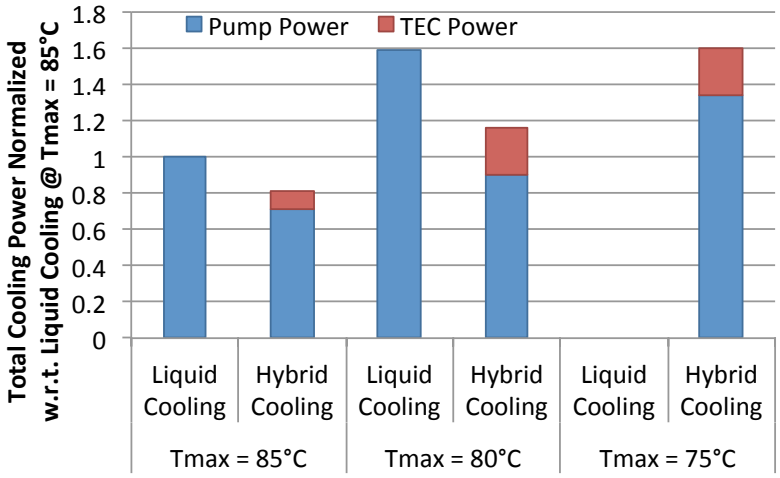


Figure 4·10: Total cooling power comparison of liquid and hybrid cooling for hot spot heat flux (HSHF) of 1300 W/cm^2. Results are normalized to liquid cooling at $T_{max} = 85°C$. Temperature constraint was not met for bars not shown.

heat sinks with fans.

The importance of having an optimized hybrid cooling system as opposed to a non-optimized system becomes more clear when we examine the design space for the resulting temperatures and cooling powers. We summarize such analysis in Table 4.2. The left half of the table shows the percentage of settings that meet the thermal constraints for various cases. We observe that for a hybrid cooled system, only a fraction of the available settings will meet the thermal limits, and this fraction decreases sharply down to %29 at tighter constraints. Out of that fraction, the right half shows the percentage of settings that save cooling power compared to liquid cooling system. For rather loose constraints, the cooling power consumption of the liquid and hybrid systems are close to each other. Thus, finding an optimal solution is crucial to provide benefits over liquid cooling, as only a small portion (e.g., %1.3) of the settings will achieve that. As the constraints become tighter (e.g., HSHF=1300 and $T_{max} = 80°C$), the inherent benefit of hybrid cooling becomes more significant. Thus,

	A = % of the settings where $T < T_{max}$			% of the settings out of A where $P_{hybrid} < P_{liquid}$		
HSHF (W/cm^2)	@85°C	@80°C	@75°C	@85°C	@80°C	@75°C
1000	84%	72%	56%	1.3%	3.8%	13.1%
1300	69%	51%	29%	6.9%	21.7%	N/A

Table 4.2: Percent of the settings that meet the temperature constraint and out of that percent, the portion of them which provide lower cooling power than liquid cooling. N/A means liquid cooling did not meet the temperature constraint.

even for suboptimal $\{u, I\}$ settings, the setting we converge to provides substantial savings compared to liquid cooling.

Chapter 5

Analysis and Optimization of Systems with FCA

The growing demand for computing power imposes many challenges in the design and energy-efficient operation of current and future processors. These challenges include but are not limited to handling high heat fluxes, reducing on-chip communication latency, achieving higher I/O bandwidth, and supplying sufficient power to the processor. In order to keep the performance scaling while preserving energy efficiency, architectural designs have initially shifted from single to multicore processors. This shift was followed by the introduction of 3D-stacked architectures, which enable stacking multiple processor and memory dies connected using through-silicon-vias (TSVs), providing lower on-chip communication latency and higher bandwidth.

The benefits of 3D stacking is hindered by the heat removal problems. Temperatures in 3D-stacked systems are usually much higher than the 2D systems dues to the additional thermal resistance introduced by vertical stacking. This brings the necessity for *scalable cooling* solutions in order to achieve the maximum potential in 3D designs. *Scalability in cooling* refers to the ability of a cooling method to maintain a similar size and cost per additional computing layer.

Another important challenge affecting the performance of processors, especially in 3D-stacked systems, is related to power delivery. The amount of power that can be delivered to the vertically stacked dies depends on the number of power TSVs. TSV area is limited and is shared between signal and power TSVs, constraining the

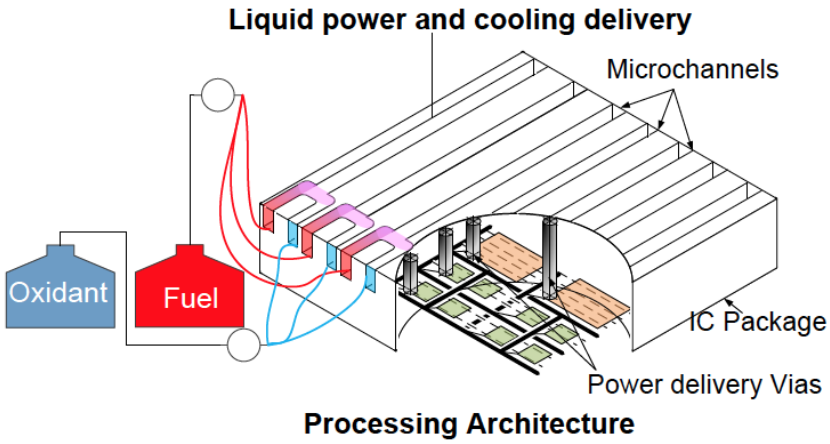


Figure 5·1: Illustration of a system with integrated FCAs (Sridhar et al., 2014).

computational density of the stacked layers.

Flow cell array (FCA) technology has recently been proposed to overcome the scalable cooling and power delivery challenges in computing systems (Sridhar et al., 2014; Sabry et al., 2014). FCA technology provides the ability to simultaneously remove heat and generate power on-chip. In a system with integrated FCAs, microchannels are etched between the stacked layers and an electrolytic solution is pumped through the channels as illustrated in Figure 5·1. While flowing through the microchannels, fuel and oxidant solutions engage in electrochemical reactions, producing electrical power while acting as a microfluidic heat sink, removing heat from the processor. FCA geometry can be manufactured in the same way as microchannel-based liquid cooling systems. FCA technology is a promising solution to the aforementioned cooling and power delivery problems faced in 3D-stacked processors and can also be applied to 2D designs to provide significant reduction in the wall-power consumption, leading to self-sustaining systems.

Recent work on FCA introduces a compact simulator named PowerCool (Sridhar

et al., 2014) for estimating the temperature and electrochemical power generation on systems with integrated FCA. PowerCool (Sridhar et al., 2014) and its follow-up work (Sabry et al., 2014) demonstrate that up to 6 W of electrical power can be generated on-chip while maintaining a peak temperature of $41°C$ using this technology. These studies were performed assuming an 8-core IBM POWER7+ processor (IBM, 2010), which is a high performance processor with peak power density of 26.7 W/cm^2. These works also provide initial analysis on the positive relationship between runtime control knobs (i.e., the fluid inlet temperature and flow rate) and the generated power.

The potential benefits of FCA, however, is not limited to this specific architecture. There are many architectural design aspects contributing to the microfluidic cooling efficiency and power generation, which have not been explored yet, such as the die size, heat flux and technology dependent leakage parameters. Each of these design parameters introduce tradeoffs between generated power, leakage power, and maximum chip temperature. Moreover, the interplay between these architectural design parameters and the runtime control knobs is another angle yet to be explored. Thus, a thorough investigation of the architectural design space is essential to be able to identify target platforms that could benefit from FCA integration the most.

Another strong motivation for such investigation is the emergence of new architectures that are targeting energy efficiency in various different environments. Integration of heterogeneous architectures such as CPUs, GPUs, special purpose accelerators and FPGAs (Burt, 2016) on the same platform is becoming a popular approach to boost efficiency, and is applicable to both 2D or 3D designs. Recent examples include utilizing FPGAs in cooperation with conventional CPUs to accelerate communication in Cloud (Caulfield et al., 2016) and supercomputing environments (George et al., 2016). Besides that, service providers are considering deployment of servers composed of multiple low-power cores (Bass and King, 2017), indicating a shift from conven-

tional data center design. We believe that FCAs can provide valuable advantages in such settings.

To this end, the work included in this chapter provides exploration of the architectural design space parameters including the chip size, heat flux levels, and technology dependent leakage parameters, and identify target systems whose energy efficiency will improve the most from FCA integration. Our analysis involves the exploration of the interplay between the architectural and runtime parameters, such as the flow rate and coolant inlet temperature, while we consider the tradeoffs between generated power, leakage power, maximum processor temperature, and pumping power. We show that in small low-power chips, FCA can provide up to 76% of the total system power. For higher power processors, FCA-generated power amount corresponds to the sum of the temperature dependent leakage power and the pumping power.

5.1 Design Space Exploration of FCA on MPSoC

In this section, we briefly describe the simulation infrastructure that we adopted from prior work (Sridhar et al., 2014). We then present our analysis involving a broad range of architectural design and runtime parameters. We continue with a description of our experimental methodology, our system assumptions, and the details of the design parameters that we study. We finally provide key observations regarding the impact of these parameters on the FCA power generation.

5.1.1 Simulation Framework for FCA-based Cooling and Power Generation

To carry out the design space exploration we use an updated version of the Power-Cool simulator, which incorporates a 3D MPSoC compact thermal model (Sridhar et al., 2010a), flow cell array compact electrochemical model (Sridhar et al., 2014) as well as CMOS temperature-dependent leakage model (Narendra and Chandrakasan,

2010) and pumping power estimation. It allows to investigate mutual dependencies of various parameters of the system while comparing with each other important metrics such as overall power consumption of the system, amount of power generated by the FCA, leakage power, pumping power and the maximum chip temperature. In this section we provide a detailed overview of our simulation framework.

Electrochemical Model of FCA Cells

The original PowerCool electrochemical model was presented in (Sridhar et al., 2014). It models operation of an array of identical individual flow cells, connected in parallel. The flow cells are represented as straight microchannels with rectangular cross-section, the electrodes are placed on the side walls of the channel, also overlapping some parts of the top and the bottom walls. Due to microscopic dimensions of the channels, the flow of the electrolyte streams is laminar, and there is no turbulent mixing between them, which removes the necessity of a separating membrane between the two streams. however, the cross-contamination of the electrolytes occurs due to the diffusion of one electrolyte species in the bulk of the other. We estimate the maximum possible size of the cross-contamination region in our simulated scenarios and choose the size of the electrodes accordingly, so the electrodes will not get in the contact with the electrolyte species of the opposite half-cell, because it will lead to a significant performance degradation of the flow cell.

While the solutions flow along the channels, they are gradually getting depleted, so the electrochemical parameters vary along the channels. We split every channel in a number of smaller elementary cells, so inside each cell the parameters vary insignificantly and can be considered constant. In Fig. 5·2 we show an electric circuit, which is used to model the behavior of a single elementary flow cell. The voltages generated by the reactions are represented by the voltage sources E^{+}_{OCP} and E^{-}_{OCP}. The losses described by the current-overpotential equation are represented by the

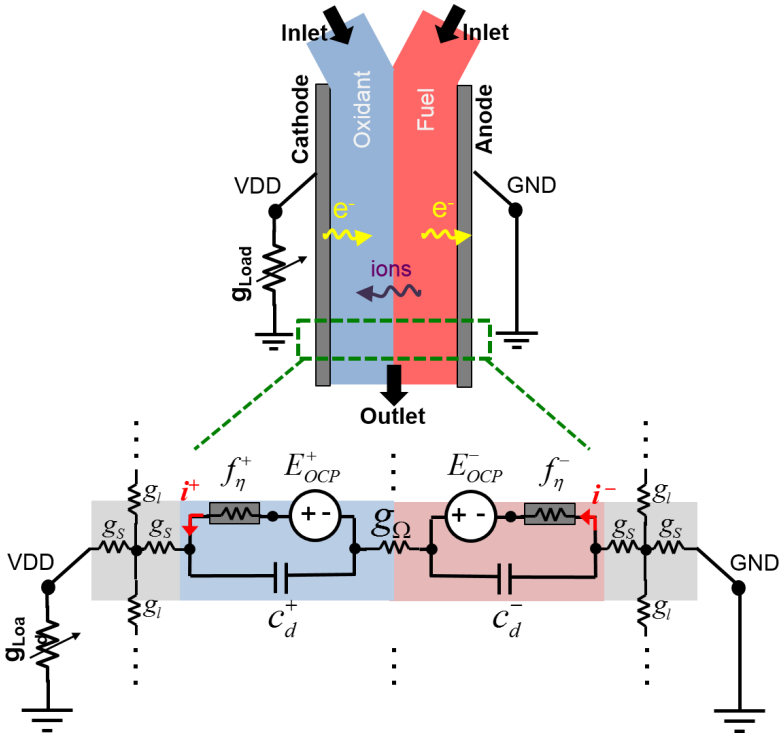


Figure 5·2: Structure of the PowerCool compact electrochemical model for a discretized flow cell (Sridhar et al., 2014).

non-linear resistors f_η^+ and f_η^-. The conductance of the solution (g_Ω) and the electrodes (g_l represents longitudinal conductance and g_s – lateral conductance) should be also considered to derive the output voltage and current, supplied by the FCA. $c_d^{+/-}$ represents double-layer capacitance, which is caused by accumulation of ions near the surfaces of the electrodes. This capacitance affects dynamic operation of the flow cell, and therefore is important for the transient analysis but could be left beyond the scope for the steady state design space exploration.

To be able to solve the system of the circuit equations written for every elementary cell it is necessary to compute the concentrations $C_{ox}^{+/-}$ and $C_{red}^{+/-}$, which define the voltages E_{OCP}^+ and E_{OCP}^- as shown in Equation (5.1):

$$E_{OCP}^{+/-} = E_0^{+/-} + \frac{RT}{nF} \log \frac{C_{ox}^{+/-}}{C_{red}^{+/-}}, \tag{5.1}$$

where R is the universal gas constant, F is the Faraday constant, n is the number of electrons exchanged in redox reaction, T is the temperature of solution in K and C_{ox} and C_{red} are the concentrations of the oxidized and reduced species at the surfaces of the electrodes respectively. Computing the concentrations requires consideration of mass transfer of reactants and products inside the channels together with the rates of chemical transformations of reactants into products on the surfaces of the electrodes. The rate of these reactions defines the electrical current which flows through the cell, and it results in a strong non-linearity of the system (current causes concentration changes, which causes voltage changes, which in turn causes current changes). We use the Newton-Raphson method to solve the system and find the current and the voltage produced by FCA for a given time.

5.1.2 Thermal Model of FCA

Integrated FCA not only acts as microfluidic heat sink for an MPSoC, but also depends on the temperature of the solutions, since many of its parameters (such as ions diffusion coefficients, exchange current densities, conductances, Nernst voltages and current-overpotential equations) vary with temperature. This is the reason why the aforementioned electrochemical model was integrated with the 3D-ICE compact thermal model (Sridhar et al., 2010a).

The thermal model uses an analogy between heat and electrical conduction to construct a linear system of circuit equation representing the thermal behavior of an MPSoC with integrated microfluidic cooling (electrical capacitance and conductivity represent heat storage and thermal conductivity respectively, current — heat flux and voltage — temperature). The system is solved using LU decomposition, and the calculated temperature distribution is used to define the coefficients and parameters of electrochemical model.

We modify the original 3D-ICE model in order to account for the temperature-

dependent subthreshold leakage, since it can represent a significant portion of the total power consumption and therefore, can affect the temperature distribution (Narendra and Chandrakasan, 2010). We describe leakage power as a sum of the subthreshold leakage ($P_{leak,subth}$), which has an exponential dependence on the temperature, and the static gate leakage ($P_{leak,stat}$), which is temperature-independent (Hall and Kopcsay, 2014):

$$P_{leak}(T) = P_{leak,stat} + P_{leak,subth}(T) = \alpha \cdot P_{ref} + \beta \cdot P_{ref} \cdot \mathrm{e}^{\kappa(T-T_{ref})} \tag{5.2}$$

$$P_{ref} = HeatFlux \cdot Area_{chip} \tag{5.3}$$

where α and β are constants representing the percentage of static and subthreshold leakage power with respect to a reference power P_{ref} at the reference chip temperature T_{ref}. κ is an exponential factor related to the technology node of the processor. P_{ref} is the total power of the processor including the dynamic (P_{dyn}) and leakage powers, and we compute P_{ref} based on the heat flux (W/cm^2) and the area of the chip.

After calculating the temperature distribution for a given system with MPSoC and FCA, we use Equation 5.2 to calculate the subthreshold leakage power corresponding to this temperature. We use this subthreshold leakage power value to iteratively update the power consumption and temperature until convergence. We extend PowerCool to include the above leakage model.

For the pumping power model, we use the model described previously in Section 4.2 and use the electrolyte properties as the coolant properties.

5.1.3 Experimental Methodology

We conduct design space exploration by running steady-state thermal and electrochemical simulations using our framework. The input design parameters we explore

are listed as follows:

Chip Length: Chip length, which also corresponds to the channel length, affects both the generated power and liquid cooling efficiency. As the electrolyte solution flows through the channel, it absorbs heat and its temperature increases. Hence, for fixed coolant velocity, longer chips lead to lower cooling efficiency and higher peak temperature. On the other hand, longer channels enable more electrochemical reaction surface, leading to higher power generation. Thus, chip length introduces a tradeoff that is worth exploration. In our analysis, we consider chip lengths ranging from 1 cm to 4.5 cm in 0.5 cm steps. We consider the lengths between 3.5 cm and 4.5 cm as they might correspond to the large interposer size in 2.5D designs.

Heat Flux: In this exploration, we initially assume a uniform on-chip heat distribution to simplify the large design space involving many dimensions. We consider heat fluxes representing a wide range of systems including low-power mobile platforms, FPGAs, as well as high performance servers. The heat fluxes we consider are 5, 10, 15, 20, 30, 50, and 100 W/cm^2.

Leakage Parameters: The amount of leakage power significantly impacts the net power gain of FCAs. As described in Section 5.1.2, leakage power is composed of a static and a temperature dependent part, which depend on the technology node, manufacturing process, material properties, as well as the temperature. In our analysis, we set the percentage of the static leakage, α, as 0.1 and experiment with different combinations of β and κ to analyze the impact of temperature dependent leakage power on the resulting efficiency of the FCA system. We use β values of 0.1, 0.2, 0.3, and κ values of 0.011 and 0.013, based on technology node trends. The leakage parameters considered in this work spans a wide range of systems with 20-40% leakage power to total power ratio.

Coolant Flow Velocity: Prior work shows how the voltage-current density (i.e., V-I) curve of a fuel cell changes for different flow velocities (Sridhar et al., 2014; Sabry et al., 2014). At higher flow velocities, one can achieve a higher current density for the same voltage. In other words, the limiting current density increases at higher flow velocity levels, leading to more power generation. Higher flow velocity also provides better cooling, but it comes at the cost of larger pumping power. To explore these tradeoffs, we experiment with flow velocities 0.2, 0.5, 1.0, 1.5, 2.0, and 2.5 m/s.

Coolant Inlet Temperature: Higher coolant inlet temperature directly increases the maximum chip temperature and the leakage power. On the other hand, higher inlet temperature leads to more power generation due to higher diffusion rates. We examine inlet temperatures of 27, 36, 45, 54, and 60$°C$.

Chip Width: We assume adiabatic boundary conditions and uniform heat distribution across the chip. Under these assumptions, the temperature distribution across the chip will be a horizontally symmetrical repeating pattern. Thus, we simulate a 0.45 cm-wide slice of a chip in order to speed up simulations without loss of accuracy. This slice corresponds to 45 microchannels for the geometry with 50 μm-wide channels. We can extrapolate the reported output power values to any given chip that is wider than 0.45 cm.

In our analysis, we consider all combinations of these input parameters and present the cases that result in feasible power consumption levels and maximum chip temperatures (i.e., $T_{max} < 110°C$, $P_{max} < 500\ W$).

Comparison Metrics: The output parameters we focus on are (i) the maximum chip temperature (T_{max}), (ii) generated FCA power (P_{FCA}), (iii) temperature dependent portion of the leakage power ($P_{leak,subth}$), and (iv) pumping power (P_{pump}). As the simulations are based on a 0.45 cm-wide slice of a chip, we scale the reported values for generated power and pumping power based on the ratio between the number of

channels when considering wide chips. Similarly, we scale the reported leakage power based on the ratio of the chip widths. For each combination of the input parameters, we simulate the generated power for a set of different load factors (representing different load resistances). In the results, we report the maximum generated power achieved across all load factors.

We quantify three main values using the following metrics:

1. P_{FCA} is the amount of power generated by the FCAs in Watts. We report P_{FCA} in order to give a measure of the range of absolute power generated on the system.

2. $FCA(\%)$ is the ratio of the generated FCA power over the total system power consumption. It answers the question of how much of the total system power can be generated on-chip using FCAs and is computed as:

$$FCA(\%) = \frac{P_{FCA}}{P_{dyn} + P_{leak,stat} + P_{leak,subth} + P_{pump}} \times 100 \tag{5.4}$$

3. $NetFCA(\%)$ is the ratio of the net FCA power over the total system power consumption. We compute net FCA power by subtracting the subthreshold leakage power and pumping power from the generated power. We define this metric in order to observe the impact of temperature dependent leakage and pumping power affects more clearly. If a design achieves $NetFCA(\%) > 0$, it means that, after the contribution of each design parameter is included, the generated power corresponds to an amount that is equal to the sum of the leakage and pumping powers, representing a strong design point. We compute $NetFCA(\%)$ as follows:

$$NetFCA(\%) = \frac{P_{FCA} - P_{leak,subth} - P_{pump}}{P_{dyn} + P_{leak,stat} + P_{leak,subth} + P_{pump}} \times 100 \tag{5.5}$$

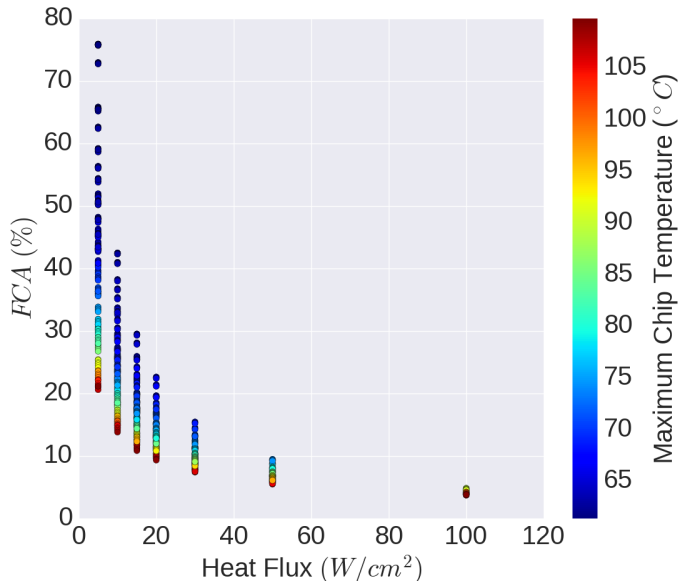


Figure 5·3: $FCA(\%)$ versus heat flux. Color bar shows the maximum chip temperature.

5.1.4 Key Observations of the Analysis

We next summarize our key observations by showing the impact of each design parameter on the output metrics. In the scatter plots included in this section, x-axis corresponds to the range of the design parameter we sweep in the experiment and y-axis represents the output metric we are investigating. In each plot, there is a color bar showing the maximum chip temperature.

Impact of Heat Flux: Figures 5·3 and 5·4 plot $FCA(\%)$ and $NetFCA(\%)$, respectively, for changing heat flux values. Figures indicate that as the heat flux level increases, both $FCA(\%)$ and $NetFCA(\%)$ decrease. For 5 W/cm^2 case, $FCA(\%)$ can reach up to 76%. As the heat flux increases, the range of $FCA(\%)$ values as well as the peak achievable value reduces. By comparing the two figures, one could also get an idea of the difference between the two metrics. $FCA(\%)$ is always a positive value. When we switch from $FCA(\%)$ to $NetFCA(\%)$, we observe the significance of

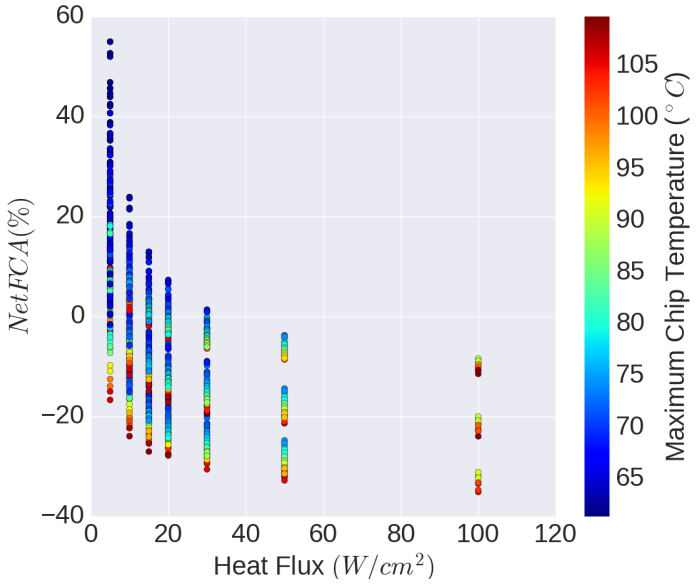


Figure 5·4: $NetFCA(\%)$ versus heat flux. Color bar shows the maximum chip temperature.

the leakage and pumping power in an amplified manner. For example, in Figure 5·4, we see that for low heat flux values such as 5-10 W/cm^2, generated power corresponds to the amount of power lost to leakage and pump, plus an additional useful power of up to 55%, which is a significant improvement. As the heat flux increases, the values in Figure 5·4 become negative, meaning that leakage affects start to dominate and even though we still generate power, the generated power compensates for a portion of the leakage and pumping powers.

Impact of Chip Length: Figures 5·5(a) and (b) plot P_{FCA} and $NetFCA(\%)$ metrics for varying chip lengths when the other parameters are fixed at the following values: heat flux=20 W/cm^2, velocity=2.5 m/s, T_{inlet}=60°C, chip width = chip length, β=0.1, and κ=0.013. As demonstrated in the plots, P_{FCA}, i.e., the absolute generated power, increases with chip length, reaching up to ∼60 W. This is also intuitive because longer channels provide a larger surface for the electrodes, while a wider chip (as chip width = chip length) contains a larger number of microchannels,

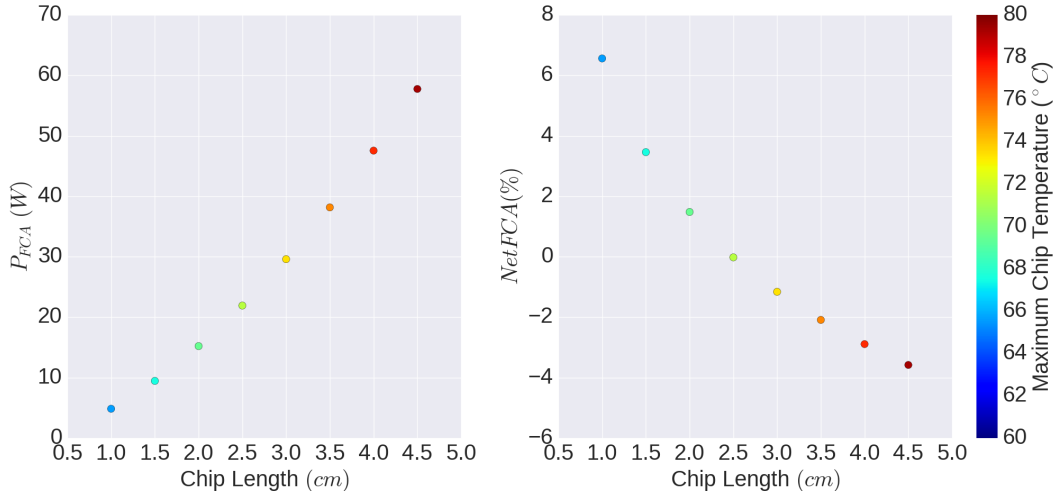


Figure 5·5: (a) P_{FCA} and (b) $NetFCA(\%)$ versus chip length at heat flux=20 W/cm^2, velocity=2.5 m/s, T_{inlet}=60°C, chip width=chip length, β=0.1, and κ=0.013. Color map shows the maximum temperature and it is the same for both plots.

leading to higher power generation. On the other hand, $NetFCA(\%)$ drops with increasing chip length. This is due to the fact that when channels get longer, the rise of liquid temperature from the inlet to the outlet of the chip is higher, leading to higher chip temperatures (i.e., T_{max} rises from 65 to 80°C). This increase in chip temperature translates to higher leakage power. Moreover, as the channel length increases, pumping power also increases linearly. The rates of increase in leakage and pumping powers surpass the rate of increase in generated power as the channels get longer. Hence, from 1 cm chip to the 4.5 cm chip, $NetFCA(\%)$ falls from 6.6% to -3.6%. The contrasting trend for the two metrics shows that the choice of chip length should be made according to the target achievement from a system. For example, if the system performance is limited by the total power that can be delivered through the PCB, then longer chips will be preferable as they generate more power. On the other hand, if the aim is to design a self-sustaining system, which generates a large portion of its power consumption regardless of the absolute amount, then smaller

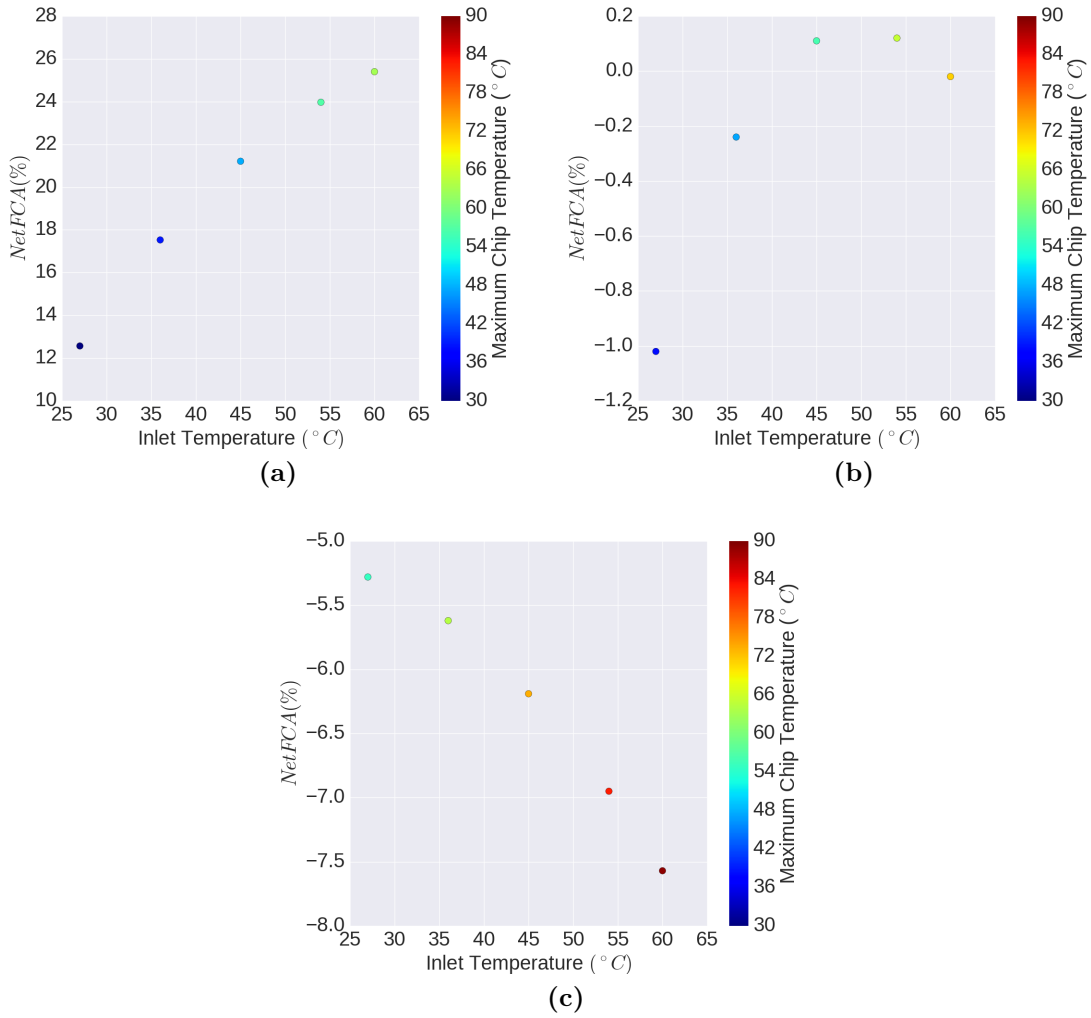


Figure 5·6: $NetFCA(\%)$ versus maximum chip temperature for varying liquid inlet temperatures at heat flux of (a) 5 W/cm^2, (b) 20 W/cm^2, and (c) 50 W/cm^2. The other parameters are set as velocity=2.5 m/s, chip width=chip length=2.5 cm, β=0.1, and κ=0.013.

chips are more suitable.

Impact of Inlet Temperature: In Figures 5·6(a) and (b), we examine the impact of inlet temperature for three different heat flux cases corresponding to 5 W/cm^2, 20 W/cm^2, and 50 W/cm^2. All other parameters are fixed as follows: velocity=2.5 m/s, chip width=chip length=2.5 cm, β=0.1, and κ=0.013. As demonstrated by the

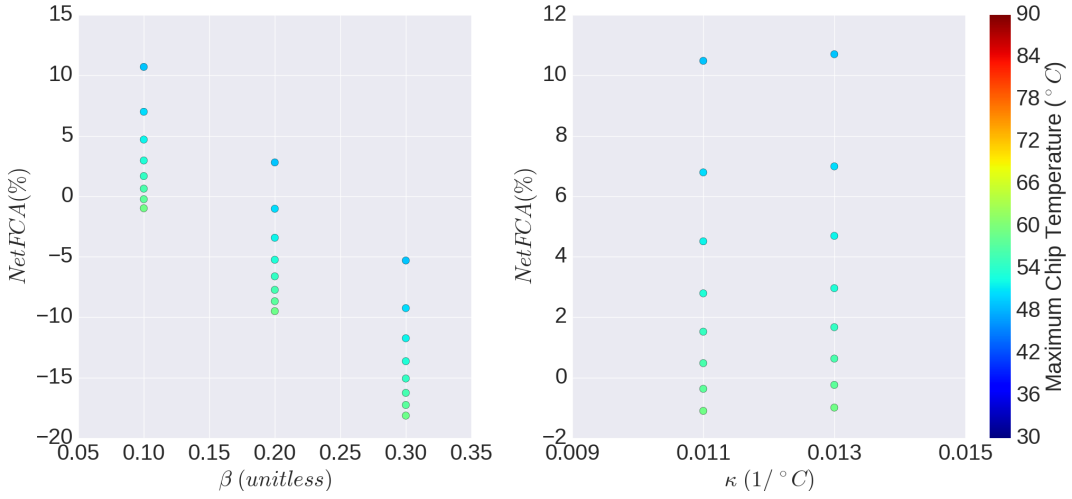


Figure 5·7: $NetFCA(\%)$ versus maximum chip temperature for (a) varying β when $\kappa = 0.013$ and (b) varying κ when $\beta = 0.1$ at heat flux=15 W/cm^2. For the other parameters, we use velocity=2.5 m/s, T_{inlet}=45°C, and show data corresponding to all chip lengths in the plots.

Figures, the inlet temperature dependency is coupled to the heat flux of the chip. For low heat flux such as 5 W/cm^2, increasing the inlet temperature leads to a higher $NetFCA(\%)$, since the power generation increases due to higher diffusion rates and the chip is already cool even at the highest inlet temperature setting. As the heat flux rises to medium and high levels, the dependency of $NetFCA(\%)$ on the inlet temperature changes direction. At 50 W/cm^2 heat flux, $NetFCA(\%)$ is negative and monotonically decreases with higher inlet temperature. This effect can be explained by the impact of temperature dependent leakage. For the 50 W/cm^2 case, $P_{leak,subth}$ increases from 25.3 W to 39.2 W between the lowest and highest inlet temperatures, while the P_{FCA} changes only by 5.7 W from 16.9 W to 22.6 W. This shows that leakage effects dominate $NetFCA(\%)$ for hotter chips, thus, in order to achieve more efficient operation, we need to keep the inlet temperature as low as possible.

Impact of Leakage Parameters: We next show the impact of technology depen-

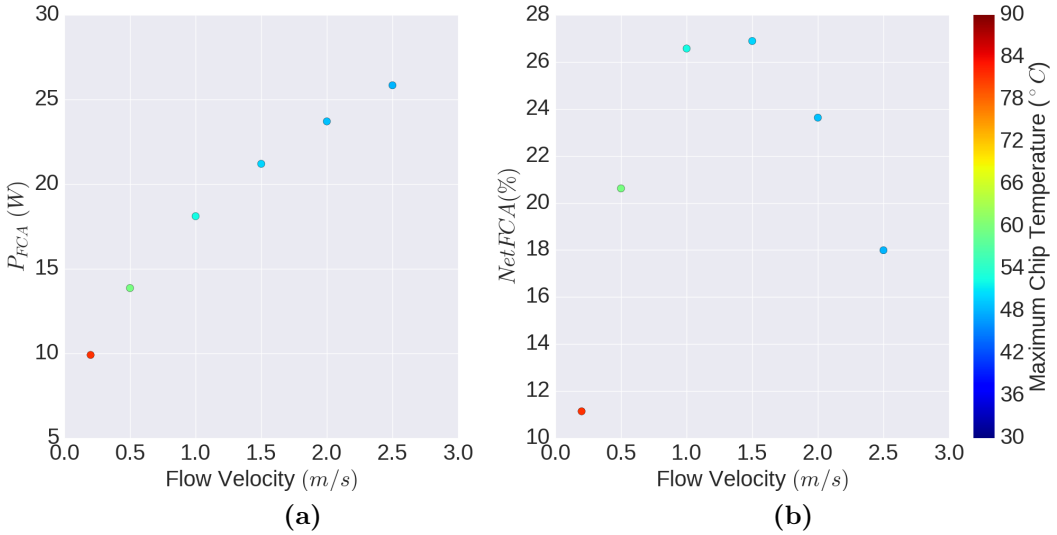


Figure 5·8: (a) P_{FCA} and (b) $NetFCA(\%)$ versus maximum chip temperature for varying flow velocity values at heat flux=5 W/cm^2. For the other parameters, we use chip width=chip length=2.5 cm, T_{inlet}=45°C, β=0.1, and κ=0.013. Color map is the same for both plots.

dent leakage parameters. In Figure 5·7(a), we plot $NetFCA(\%)$ for changing β (i.e., percentage of temperature dependent leakage power) when κ=0.013 and in Figure 5·7(b) we plot $NetFCA(\%)$ for changing κ (i.e., the scaling factor on the exponent) when β=0.1. For the other parameters, we use heat flux=15 W/cm^2, velocity=2.5 m/s, T_{inlet}=60°C, and show data corresponding to all chip lengths in the plots. Low β and high κ values represent a low-leakage system. Figure 5·7(a) indicates that $NetFCA(\%)$ has a large dependence on β and a small dependence on κ. While the useful power percentage changes between 0 to 10% for the low leakage system (i.e., β=0.1), it drops down to negative values between -5 to -20% for the high leakage system (i.e., β=0.3). On the other hand, we observe from Figures 5·7(b) that the data points for different κ values are very close to each other when all other parameters are kept the same.

Impact of Flow Velocity: We show the impact of flow velocity in Figures 5·8(a)

and (b). As we mentioned before and as indicated in Figure 5·8(a), FCA power increases with flow velocity and saturates at high values. However, $NetFCA(\%)$ shows a concave curve due to the quadratic relationship between flow velocity and pumping power, as illustrated in Figure 5·8(b). Similar to the previous discussion on chip length, the right choice of velocity depends on the target metric to be optimized. When maximum power generation is required, then the highest flow velocity should be picked. Otherwise, if the $NetFCA(\%)$ is the metric to be maximized for the system in hand, then the best velocity 1.5 m/s and how to determine this velocity becomes the question to be answered in an optimization framework.

Examples of Real Systems for FCA Integration:

We conclude our design space analysis by providing a table of platforms that have similar heat fluxes and chip sizes to those of the real processor systems. For each platform, we provide the maximum values we can achieve for the metrics P_{FCA}, $FCA(\%)$, and $NetFCA(\%)$ given the corresponding heat flux level and chip area for that platform. The target platforms we examine include mobile processors, FPGAs, and MPSoCs. Mobile processors fall into the small, low-heat-flux-chip category, for which, chip sizes of 1cmx1cm and heat fluxes of 5-6 W/cm^2 has been reported on a Snapdragon SoC (Yueh et al., 2015). FPGAs consume very low power (in the range of 5-10 W/cm^2) and in general have larger die size. For example, recent work describes 2.5D integration of multiple Xilinx FPGAs on a 3.5cmx3.5cm silicon interposer (Chaware et al., 2012). For Intel quad-core Core i7-940 platform, total power consumption was reported in prior work (Paterna and Reda, 2013) as 73 W on a 2.63 cm^2 die, corresponding to 27.8 W/cm^2 heat flux. Similarly prior work on FCAs (Sridhar et al., 2014) focus on POWER7+ processor, which has a peak power density of 26.7 W/cm^2 with a size of 2.13cmx2.65cm.

Table 5.1 presents the results of our evaluation involving these corresponding real

Platforms	Heat Flux Range	Size Range	Corresponding Heat Flux in Our Analysis	Corresponding Area in Our Analysis	Max P_{FCA} (W)	Max FCA (%)	Max NetFCA (%)
Mobile Snapdragon	5-6 W/cm^2	1cmx1cm	5 W/cm^2	1cmx1cm	4.8 W	76%	55.0%
FPGAs on Interposer	5-10 W/cm^2	3.5cmx3.5cm	10 W/cm^2	3.5cmx3.5cm	37.6 W	30.7%	9.28%
Intel i7-940	27.8 W/cm^2	2.63 cm^2	30 W/cm^2	1.75cmx1.5cm	11.1 W	14%	-0.21%
IBM POWER7+	26.7 W/cm^2	2.13cmx2.65cm	30 W/cm^2	2.25cmx2.5cm	19.9 W	11.7%	-2.14%

Table 5.1: Examples of real systems for FCA integration and the corresponding power generation metrics.

platforms. As indicated in the table, in terms of the amount of power generation, FPGAs are the most promising platforms with a maximum P_{FCA} of $\sim$38 W. On the other hand, when considering the net power percentage, i.e., a measure of the self-sustaining capability that FCA provides, mobile platforms are the most promising ones owing to their small size and low heat fluxes.

Chapter 6

Conclusions and Future Directions

6.1 Summary of Major Contributions

In order to reach exascale computing, we need dramatic improvements in processor energy efficiency. High power densities leading to elevated on-chip temperatures have been among the major limiting factors against such efficiency improvements. This thesis has claimed that the desired energy efficiency levels can only be achieved through a temperature-centric co-design of the cooling and computing subsystems, where cooling and computing elements are mutually customized with awareness of each other's characteristics. An essential pre-requisite to such a co-design approach is fast and accurate thermal modeling of the novel cooling techniques. These models should then be encapsulated in tools that enable developing thermally aware design-time and runtime optimization techniques.

This thesis has first addressed the need for thermal modeling tools that would be used in the design and optimization of future processors adopting cutting-edge cooling solutions. To this end, we have provided novel methods to model advanced cooling techniques, namely PCM, liquid cooling, and TECs, in compact thermal simulators. We have evaluated the accuracy and speed of our proposed models by comparing their results against multi-physics simulations as well as testbed measurements. Thermal models proposed as part of this thesis provide a modular simulation environment, where the user can plug-in a single or a combination of cooling modules on the same platform and explore a wider design space with reasonably low effort.

The second major body of this thesis has focused on optimization of the cooling and computing systems to maximize energy efficiency under temperature constraints. We demonstrated that when designing sprinting policies in systems with PCM-based cooling, being "aware" of the PCM state is a key element in achieving performance gains substantially larger than these achievable by prior methods. We have proposed *adaptive sprinting*, which is driven by the observation that PCM melts at different rates across the chip due to heterogeneous on-chip heat distribution and that the sprinting ability of the individual cores will depend on the percentage of the unmelted PCM around them. *Adaptive sprinting* policy tracks the PCM state at runtime and utilizes this information to decide on the number, location, and V/F levels of the sprinting cores. Comparison against the state-of-the-art sprinting policies shows that *adaptive sprinting* improves performance by 29% and saves energy by 22%.

This thesis then has focused on mitigating high density hot spots using localized hybrid cooling techniques. We have proposed *LoCool* to maximize the energy efficiency in hybrid-cooled systems combining TECs and liquid microchannels. *LoCool* is a cooling system optimizer that co-optimizes the liquid flow rate and TEC bias current to minimize the total cooling power for a given temperature constraint. We have shown that a hybrid design optimized with *LoCool* can save cooling power by up to 28% compared to a design that uses liquid cooling only. We have finally demonstrated the importance of cooling design optimization using the proposed *LoCool*. Our analysis shows that if not optimized, the same hybrid design may lead to higher cooling power consumption than the liquid-cooled design in at least 80% of the cases.

The scope of this thesis extends to emerging technologies and early exploration of their benefits and tradeoffs. FCA technology has been proposed as a promising solution to the power delivery and temperature challenges in future processors. In FCA, fuel cells are pumped through microchannels to deliver cooling and on-chip

power generation. This thesis has provided a thorough analysis of the potential target platforms, the corresponding design parameters as well as the tunable runtime parameters that will maximize the net power generation under thermal constraints on FCA-based systems. We have shown that in smaller low-power chips, up to 76% of the total power can be generated on-chip using FCA. On the other hand, for power-hungry processors, FCA can generate power that is equivalent to the temperature dependent leakage power, reducing its negative effects on the power delivery.

Significance of the Thesis Contributions

The work presented as part of this thesis improves the energy efficiency of the processors by tackling the temperature problem in more effective ways than the state-of-the-art. By applying the proposed optimization and runtime management techniques, we can maintain the same temperature constraints while achieving a substantially higher performance.

Prior work shows that PCM-based cooling combined with *computational sprinting* can provide up to 16x speedup on a 16-core system running parallel workloads (Raghavan et al., 2012). The benefits of the PCM on extending the sprinting duration has also been shown experimentally in prior work (Raghavan et al., 2013). When running the same application, we can sprint for 6x longer time when using PCM in comparison to air. Similarly, PCM-based sprinting can last 3x longer compared to sprinting with a copper container filled with water, allowing to complete 3x more work until reaching a thermal limit. Our proposed *adaptive sprinting* policy exploits the heterogeneous melting of the PCM to further extend this sprinting duration and provides 29% higher performance than the best performing sprinting policy without any additional cost compared to prior techniques.

On systems where high density hot spots are the limiting factors against achieving higher efficiency, we have argued that hybrid cooling can effectively address the

problem. A hybrid cooling design optimized with our proposed *LoCool* algorithm can handle much higher power densities than a system using liquid cooling only. For example, liquid cooling can reduce the hot spot temperature down to $80°C$ for a hot spot with 1.3 kW/cm^2 heat flux. Consuming the same cooling power, a hybrid design optimized with *LoCool* can achieve the same thermal constraint for a hot spot with 1.5 kW/cm^2 heat flux. As prior work shows, there is a near-linear relationship between average power consumption and throughput in real systems (Tuncer et al., 2014; Hankendi and Coskun, 2013). Based on this relationship, we can estimate that the hybrid cooled system can potentially achieve at least 15% higher throughput while meeting the thermal constraints and requiring the same cooling power.

On systems with FCA, the benefits of power generation can be translated to performance gains using a similar relationship between power consumption and throughput (Tuncer et al., 2014; Hankendi and Coskun, 2013). For example, for the mobile system in Table 5.1, we can generate ∼76% of the total power on-chip using FCAs (see Max $FCA(\%)$ column in Table 5.1). It means that for the same total wall-power budget, we can do 76% more work in comparison to a system without FCA. On the other hand, if we consider the work done, we can complete the same amount of work by drawing only 24% of the total power externally. In terms of throughput per external watts, it corresponds to 4.16x improvement. Similarly, for the higher power commercial processor systems such as Intel i7-940 and IBM POWER7+, using the generated power coming from FCA, we can get 14% and 11.7% more performance given the same wall-power requirement, respectively.

These substantial performance/throughput improvements, which come at essentially no additional cost compared to their closest state-of-the-art solution, can be further boosted using a co-design approach. In such a design approach, computing and cooling subsystems would be carefully tailored for the specific platform in hand to

achieve potentially several times more performance gain. We believe this thesis paves the way for developing system-level design automation tools to achieve this ambitious vision. To this end, next, we describe specific directions that could immediately follow this thesis.

6.2 Future Research Directions and Open Problems

6.2.1 Thermal Modeling of Two-Phase Cooling and a System-Level Simulation Framework

In this thesis, we have focused on electronic cooling with PCM, TEC, and single-phase liquid cooling. Two-phase cooling has been gaining attention owing to its ability to handle high heat fluxes with low pumping requirements. In two-phase cooling, the fluid goes through phase change from liquid to vapor, harnessing the high heat storage ability of vaporization. Examples of two-phase cooling methods include two-phase microchannel cooling (Schultz et al., 2016; Thome, 2004), thin film evaporation (Zhu et al., 2016), and nanoporous evaporation (Lu et al., 2015). The challenges with two-phase cooling include minimization of the flow instabilities while enhancing the critical heat flux to achieve higher heat dissipation. Thermal models for two-phase cooling exist, however, most of them are not compact models (Zhu et al., 2016; Zhu et al., 2014) and they do not allow exploration of hybrid designs combining two-phase cooling with other methods (Zhu et al., 2016; Zhu et al., 2014; Sridhar et al., 2013). Thus, inclusion of a two-phase cooling model is a next step towards exploration and optimization of this cooling solution.

Our future directions in the modeling domain also includes combining those thermal models with system-level optimization libraries under a unified simulation framework. This framework enables the simulation of runtime aspects of computing systems, such as workload scheduling, job allocation, migration, DVFS, and sprinting

policies. The significance of the proposed framework is that it closes the feedback loop between processor temperature and runtime decisions that rely on temperature under a dynamic workload scenario. In this way, the described framework provides much more functionality than mere implementation of thermal models. In this thesis, we have implemented an initial version of this framework and utilized it to develop and evaluate computational sprinting policies. The ultimate goal is to develop the necessary infrastructure for this framework to be compatible with various other design-time and runtime techniques. We are also planning to make our software available for other researchers to use and serve the electronic design automation community this way.

6.2.2 Hybrid Cooling Optimizer for Heterogeneous Architectures

In this thesis, we have provided an optimization algorithm to minimize power in hybrid cooling systems with TECs and liquid microchannels for steady-state operation. The dynamic power/performance characteristics of the applications require such hybrid optimization algorithm to adapt to the runtime changes for optimum operation. In our ongoing work, we focus on improving the proposed optimizer, *LoCool*, to respond to the changes in heat flux levels during runtime operation.

Our hybrid cooling work so far has targeted platforms with localized high density hot spots and we have chosen liquid cooling and TECs as our cooling modules. This initial investigation of hybrid systems reflects the importance of heterogeneity in the heat distribution of current and next generation processors. This heterogeneity is not limited to the heat distribution, but also extends to the architectural heterogeneity owing to the integration of different computing elements on a single die, such as CPUs, GPUs and FPGAs (Burt, 2016; Caulfield et al., 2016; George et al., 2016). These computing elements show diverse characteristics in terms of the power consumption, performance, and area as well as the applications running on them. Thus, the next step in that domain will be to design an optimizer such that, given a platform with

heterogeneous computing elements, it would first select the best combination of cooling methods to be used for each computing element and then optimize the system for maximum efficiency.

6.2.3 Optimization of Systems with Integrated FCAs

In this thesis, we have analyzed various aspects involved in the design of systems with FCA technology, considering the tradeoffs between temperature, leakage, and generated power. Based on this analysis, we have proposed target systems and properties that will benefit from FCA the most. The immediate question to answer in the next step is how to select these design parameters to maximize FCA power generation given a platform with temperature, area, and total system power constraints. For example, if we have the choice to design a server processor, should we design the system with a large number of low-power cores and have a larger die with low heat flux; or should we have a smaller chip with less number of higher performance and power-hungry cores? Assuming these two systems achieve the same overall throughput, in which of these systems, FCA power generation will be higher leading to lower net wall power consumption?

Future work in this domain will aim to answer these questions and propose an optimization framework for FCAs. The inputs to the framework are the area, performance, power consumption cap, maximum temperature constraints and a set of core architectures to choose from. The output is a design that specifies the chip length, chip width, liquid flow rate, and the selected core types to maximize throughput per Watt. This optimization approach will help researchers to unearth the full potential of FCA integration and provide insight towards self-sustaining computing systems.

References

Alawadhi, E. and Amon, C. H. (2003). PCM thermal control unit for portable electronic devices: experimental and numerical studies. *IEEE Transactions on Components and Packaging Technologies*, 26(1):116–125.

ANSYS (2017). ANSYS. http://www.ansys.com.

Bass, D. and King, I. (2017). Microsoft pledges to use ARM server chips, threatening Intel's dominance. `https://tinyurl.com/gq7hexv`.

Bienia, C. (2011). *Benchmarking Modern Multiprocessors*. PhD thesis, Princeton University.

Binkert, N. L. et al. (2006). The M5 simulator: Modeling networked systems. *IEEE Micro*, 26(4):52–60.

Brunschwiler, T., Sridhar, A., Ong, C., and Schlottig, G. (2015). Benchmarking study on the thermal management landscape for 3D ICs: From back-side to volumetric heat removal. *ASME International Electronic Packaging Technical Conference and Exhibition*, 1:V001T09A069.

Burt, J. (2016). Intel begins shipping xeon chips with fpga accelerators. `https://tinyurl.com/loxoevd`.

Caulfield, A. et al. (2016). A cloud-scale acceleration architecture. In *IEEE/ACM International Symposium on Microarchitecture*, pages 1–13.

Chaware, R., Nagarajan, K., and Ramalingam, S. (2012). Assembly and reliability challenges in 3D integration of 28nm FPGA die on a large high density 65nm passive interposer. In *IEEE Electronic Components and Technology Conference*, pages 279–283.

Chowdhury, I., Prasher, R., Lofgreen, K., Chrysler, G., Narasimhan, S., Mahajan, R., Koester, D., Alley, R., and Venkatasubramanian, R. (2009). On-chip cooling by superlattice-based thin-film thermoelectrics. *Nature Nanotechnology*, 4(4):235–238.

COMSOL (2017). Comsol Multiphysics Software. `http://www.comsol.com`.

Conway, P. et al. (2009). Blade computing with the AMD Opteron Processor ($magny - cours$). `http://bit.ly/1LVbJXn`.

Coskun, A., Meng, J., Atienza, D., and Sabry, M. (2011). Attaining single-chip, high-performance computing through 3D systems with active cooling. *IEEE Micro*, 31(4):63–75.

Coskun, A. K., Atienza, D., Rosing, T. S., Brunschwiler, T., and Michel, B. (2010). Energy-efficient variable-flow liquid cooling in 3D stacked architectures. In *Design, Automation and Test in Europe (DATE)*, pages 111–116.

Coskun, A. K., Ayala, J. L., Atienza, D., Rosing, T. S., and Leblebici, Y. (2009a). Dynamic thermal management in 3D multicore architectures. In *Design, Automation and Test in Europe (DATE)*, pages 1410–1415.

Coskun, A. K., Rosing, T. S., Ayala, J., and Atienza, D. (2009b). Modeling and dynamic management of 3D multicore systems with liquid cooling. In *IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, pages 60–65.

Coskun, A. K., Rosing, T. S., and Whisnant, K. (2007). Temperature aware task scheduling in MPSoCs. In *Design, Automation and Test in Europe (DATE)*, pages 1659–1664.

Coskun, A. K., Rosing, T. S., Whisnant, K. A., and Gross, K. C. (2008a). Temperature-aware MPSoC scheduling for reducing hot spots and gradients. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 49–54.

Coskun, A. K., Rosing, T. v., Whisnant, K. A., and Gross, K. C. (2008b). Static and dynamic temperature-aware scheduling for multiprocessor socs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 16(9):1127–1140.

Coskun, A. K., Strong, R., Tullsen, D. M., and Rosing, T. S. (2009c). Evaluating the impact of job scheduling and power management on processor lifetime for chip multiprocessors. In *ACM Special Interest Group on Measurement and Evaluation (SIGMETRICS)*, pages 169–180.

Dang, B., Bakir, M. S., Sekar, D. C., Jr, C. R. K., and Meindl, J. D. (2010). Integrated microfluidic cooling and interconnects for 2D and 3D chips. *IEEE Transactions on Advanced Packaging*, 33(1):79–87.

Escher, W., Michel, B., and Poulikakos, D. (2010). A novel high performance, ultra thin heat sink for electronics. *International Journal of Heat and Fluid Flow*, 31(4):586 – 598.

Fourmigue, A., Beltrame, G., and Nicolescu, G. (2014). Efficient transient thermal simulation of 3d ics with liquid-cooling and through silicon vias. In *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*, pages 1–6.

George, A. D., Herbordt, M. C., Lam, H., Lawande, A. G., Sheng, J., and Yang, C. (2016). Novo-g#: Large-scale reconfigurable computing with direct and programmable interconnects. In *IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–7.

Gruener, W. (2008). IBM cools 3D chips with integrated water channels. `http://www.tomshardware.com/news/IBm-research,5604.html`.

Hale, D. et al. (1971). *Phase Change Materials Handbook*. Lockheed Missiles & Space Company.

Hall, S. A. and Kopcsay, G. V. (2014). Energy-Efficient Cooling of Liquid-Cooled Electronics Having Temperature-Dependent Leakage. *Journal of Thermal Science and Engineering Applications*, 6(1):011008/1–011008/12.

Hankendi, C. and Coskun, A. K. (2013). Energy-efficient server consolidation for multi-threaded applications in the cloud. In *International Green Computing Conference (IGCC)*, pages 1–8.

Hankendi, C., Reda, S., and Coskun, A. (2013). vCap: Adaptive power capping for virtualized servers. In *International Symposium on Low Power Electronics and Design (ISLPED)*, pages 415–420.

Hu, Y., Chen, S., Peng, L., Song, E., and Choi, J.-W. (2013). Effective thermal control techniques for liquid-cooled 3D multi-core processors. In *International Symposium onQuality Electronic Design (ISQED)*, pages 8–15.

IBM (2010). IBM POWER7+. `https://tinyurl.com/yl4w5bv`.

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

Jayaseelan, R. and Mitra, T. (2009). Dynamic thermal management via architectural adaptation. In *ACM/IEEE Design Automation Conference (DAC)*, pages 484–489.

JEDEC (2009). Failure mechanisms and models for semiconductor devices, jedec publication jep122e. `http://www.jedec.org/`.

Kandlikar, S. G. and Bapat, A. V. (2007). Evaluation of jet impingement, spray and microchannel chip cooling options for high heat flux removal. *Heat Transfer Engineering*, 28(11):911–923.

Kaplan, F. and Coskun, A. (2015). Adaptive sprinting: How to get the most out of phase change based passive cooling. In *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 37–42.

Kaplan, F., De Vivero, C., Howes, S., Arora, M., Homayoun, H., Burleson, W., Tullsen, D., and Coskun, A. (2014). Modeling and analysis of phase change materials for efficient thermal management. In *IEEE International Conference on Computer Design (ICCD)*, pages 256–263.

Kim, N. S., Austin, T., Baauw, D., Mudge, T., Flautner, K., Hu, J. S., Irwin, M. J., Kandemir, M., and Narayanan, V. (2003). Leakage current: Moore's law meets static power. *Computer*, 36(12):68–75.

Kjeang, E., McKechnie, J., Sinton, D., and Djilali, N. (2007). Planar and three-dimensional microfluidic fuel cell architectures based on graphite rod electrodes. *Journal of Power Sources*, 168(2):379 – 390.

Ladenheim, S., Chen, Y.-C., Mihajlović, M., and Pavlidis, V. (2016). IC thermal analyzer for versatile 3-D structures using multigrid preconditioned krylov methods. In *International Conference on Computer-Aided Design (ICCAD)*, pages 123:1–123:8.

Lee, C.-C. and Groot, J. D. (2006). On the thermal stability margins of high-leakage current packaged devices. In *Electronics Packaging Technology Conference*, pages 487–491.

Lee, P.-S., Garimella, S. V., and Liu, D. (2005). Investigation of heat transfer in rectangular microchannels. *International Journal of Heat and Mass Transfer*, 48(9):1688 – 1704.

Li, S., Ahn, J. H., Strong, R., Brockman, J., Tullsen, D., and Jouppi, N. (2009). McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 469–480.

Lingamneni, S., Asheghi, M., and Goodson, K. (2014). A parametric study of microporous metal matrix-phase change material composite heat spreaders for transient thermal applications. In *IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITHERM)*.

Liu, X.-X., Liu, Z., Tan, S.-D., and Gordon, J. (2012). Full-chip thermal analysis of 3d ics with liquid cooling by gpu-accelerated gmres method. In *International Symposium on Quality Electronic Design (ISQED)*, pages 123–128.

Lu, Z., Narayanan, S., and Wang, E. N. (2015). Modeling of evaporation from nanopores with nonequilibrium and nonlocal effects. *Langmuir*, 31(36):9817–9824. PMID: 26322737.

Lu, Z., Salamon, T. R., Narayanan, S., Bagnall, K. R., Hanks, D. F., Antao, D. S., Barabadi, B., Sircar, J., Simon, M. E., and Wang, E. N. (2016). Design and modeling of membrane-based evaporative cooling devices for thermal management of high heat fluxes. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 6(7):1056–1065.

Meng, J., Kawakami, K., and Coskun, A. (2012). Optimizing energy efficiency of 3-D multicore systems with stacked DRAM under power and thermal constraints. In *Design Automation Conference (DAC)*, pages 648–655.

Mills, A., Farid, M., Selman, J., and Al-Hallaj, S. (2006). Thermal conductivity enhancement of phase change materials using a graphite matrix. *Applied Thermal Engineering*, 26(14-15):1652 – 1661.

Narendra, S. G. and Chandrakasan, A. P. (2010). *Leakage in Nanometer CMOS Technologies*. Springer Publishing Company, Incorporated, 1st edition.

Ogoh, W. and Groulx, D. (2012). Effects of the heat transfer fluid velocity on the storage characteristics of a cylindrical latent heat energy storage system: a numerical study. *Heat and Mass Transfer*, 48(3):439–449.

Park, S., Park, J., Shin, D., Wang, Y., Xie, Q., Pedram, M., and Chang, N. (2013). Accurate modeling of the delay and energy overhead of dynamic voltage and frequency scaling in modern microprocessors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 32(5):695–708.

Paterna, F. and Reda, S. (2013). Mitigating dark-silicon problems using superlattice-based thermoelectric coolers. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, pages 1391–1394.

Pozo, R. and Miller, B. (2017). SciMark 2.0. `http://http://math.nist.gov/scimark2/index.html`.

Qian, H., Chang, C.-H., and Yu, H. (2013). An efficient channel clustering and flow rate allocation algorithm for non-uniform microfluidic cooling of 3D integrated circuits. *Integration, the VLSI Journal*, 46(1):57 – 68.

Raghavan, A., Emurian, L., Shao, L., Papaefthymiou, M., Pipe, K. P., Wenisch, T. F., and Martin, M. M. (2013). Computational sprinting on a hardware/software testbed. In *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 155–166.

Raghavan, A., Luo, Y., Chandawalla, A., Papaefthymiou, M., Pipe, K. P., Wenisch, T. F., and Martin, M. M. K. (2012). Computational sprinting. In *IEEE Symposium on High Performance Computer Architecture (HPCA)*, pages 1–12.

Reda, S., Cochran, R., and Coskun, A. (2012). Adaptive power capping for servers with multithreaded workloads. *IEEE Micro*, 32(5):64–75.

Reddy, J. N. (1993). *An introduction to the finite element method*, volume 2. McGraw-Hill New York.

Ruch, P., Brunschwiler, T., Escher, W., Paredes, S., and Michel, B. (2011). Toward five-dimensional scaling: How density improves efficiency in future computers. *IBM Journal of Research and Development*, 55(5):15:1–15:13.

Ruch, P., Brunschwiler, T., Paredes, S., Meijer, I., and Michel, B. (2013). Roadmap towards ultimately-efficient zeta-scale datacenters. In *International Conference on High Performance Computing Simulation (HPCS)*, pages 161–163.

Sabry, M. M., Coskun, A. K., Atienza, D., Rosing, T. S., and Brunschwiler, T. (2011). Energy-efficient multiobjective thermal control for liquid-cooled 3-D stacked architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 30(12):1883–1896.

Sabry, M. M., Sridhar, A., Atienza, D., Ruch, P., and Michel, B. (2014). Integrated microfluidic power generation and cooling for bright silicon MPSoCs. In *Design, Automation Test in Europe (DATE)*, pages 1–6.

Sabry, M. M., Sridhar, A., Meng, J., Coskun, A. K., and Atienza, D. (2013). Greencool: An energy-efficient liquid cooling design technique for 3-D MPSoCs via channel width modulation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 32(4):524–537.

Sahu, V., Fedorov, A. G., Joshi, Y., Yazawa, K., Ziabari, A., and Shakouri, A. (2012). Energy efficient liquid-thermoelectric hybrid cooling for hot-spot removal. In *IEEE Semiconductor Thermal Measurement and Management Symposium (SEMITHERM)*, pages 130–134.

Sahu, V., Joshi, Y. K., Fedorov, A. G., Bahk, J. H., Wang, X., and Shakouri, A. (2015). Experimental characterization of hybrid solid-state and fluidic cooling for thermal management of localized hotspots. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 5(1):57–64.

Schultz, M., Yang, F., Colgan, E., Polastre, R., Dang, B., Tsang, C., Gaynes, M., Parida, P., Knickerbocker, J., and Chainer, T. (2016). Embedded two-phase cooling of large three-dimensional compatible chips with radial channels. *ASME Journal of Electronic Packaging*, 138(2):021005–1–021005–5.

Shafique, M., Garg, S., Henkel, J., and Marculescu, D. (2014). The EDA challenges in the dark silicon era. In *Design Automation Conference (DAC)*, pages 1–6.

Shah, R. K. and London, A. L. (1978). *Laminar flow forced convection in ducts: A source book for compact heat exchanger analytical data*. New York: Academic Press.

Shao, L., Raghavan, A., Emurian, L., Papaefthymiou, M. C., Wenisch, T. F., Martin, M. M., and Pipe, K. P. (2014). On-chip phase change heat sinks designed for computational sprinting. In *Semiconductor Thermal Measurement and Management Symposium*, pages 29–34.

Sharma, C. S., Tiwari, M. K., Michel, B., and Poulikakos, D. (2013). Thermofluidics and energetics of a manifold microchannel heat sink for electronics with recovered hot water as working fluid. *International Journal of Heat and Mass Transfer*, 58(12):135 – 151.

Sharma, C. S., Tiwari, M. K., Zimmermann, S., Brunschwiler, T., Schlottig, G., Michel, B., and Poulikakos, D. (2015). Energy efficient hotspot-targeted embedded liquid cooling of electronics. *Applied Energy*, 138(C):414–422.

Shi, B. and Srivastava, A. (2014). Optimized micro-channel design for stacked 3-d-ics. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 33(1):90–100.

Skach, M., Arora, M., Hsu, C.-H., Li, Q., Tullsen, D. M., Tang, L., and Mars, J. (2015). Thermal time shifting: leveraging phase change materials to reduce cooling costs in warehouse-scale computers. In *International Society for Computers and their Applications (ISCA)*, pages 439–449. ACM.

Skadron, K., Stan, M., Huang, W., Velusamy, S., Sankaranarayanan, K., and Tarjan, D. (2003a). Temperature-aware microarchitecture. In *International Society for Computers and their Applications (ISCA)*, pages 2–13.

Skadron, K., Stan, M. R., Huang, W., Velusamy, S., Sankaranarayanan, K., and Tarjan, D. (2003b). Temperature-aware microarchitecture. In *International Society for Computers and their Applications (ISCA)*, pages 2–13.

Sridhar, A., Madhour, Y., Atienza, D., Brunschwiler, T., and Thome, J. (2013). STEAM: A fast compact thermal model for two-phase cooling of integrated circuits. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 256–263.

Sridhar, A., Sabry, M. M., Ruch, P., Atienza, D., and Michel, B. (2014). PowerCool: Simulation of integrated microfluidic power generation in bright silicon

MPSoCs. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 527–534.

Sridhar, A., Vincenzi, A., Ruggiero, M., Brunschwiler, T., and Atienza, D. (2010a). 3D-ICE: Fast compact transient thermal modeling for 3D ICs with inter-tier liquid cooling. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 463–470.

Sridhar, A., Vincenzi, A., Ruggiero, M., Brunschwiler, T., and Atienza, D. (2010b). Compact transient thermal model for 3D ICs with liquid cooling via enhanced heat transfer cavity geometries. In *International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, pages 1–6.

Srinivas, V. S. S. and Ananthasuresh, G. K. (2006). Analysis and topology optimization of heat sinks with a phase-change material on COMSOL multiphysics platform. In *COMSOL Users Conference*, pages 1–7.

Srinivasan, J., Adve, S. V., Bose, P., Rivers, J., and Hu, C.-K. (2003). RAMP: A model for reliability aware microprocessor design. Technical Report IBM-RC23048(W0312-122), IBM Research.

Stupar, A., Drofenik, U., and Kolar, J. (2010). Application of phase change materials for low duty cycle high peak load power supplies. In *International Conference on Integrated Power Electronics Systems (CIPS)*, pages 1–11.

Tan, F. and Fok, S. C. (2007). Thermal management of mobile phone using phase change material. In *Electronics Packaging Technology Conference*, pages 836–842.

Taylor, R. A. and Solbrekken, G. L. (2008). Comprehensive system-level optimization of thermoelectric devices for electronic cooling applications. *IEEE Transactions on Components and Packaging Technologies*, 31(1):23–31.

Thome, J. R. (2004). Boiling in microchannels: a review of experiment and theory. *International Journal of Heat and Fluid Flow*, 25(2):128 – 139.

Thoziyoor, S., Muralimanohar, N., Ahn, J. H., and Jouppi, N. P. (2008). CACTI 5.1. Technical report, HP Laboratories.

Tilli, A., Bartolini, A., Cacciari, M., and Benini, L. (2012). Don't burn your mobile!: safe computational re-sprinting via model predictive control. In *International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, pages 373–382.

Tuncer, O., Vaidyanathan, K., Gross, K., and Coskun, A. K. (2014). CoolBudget: Data center power budgeting with workload and cooling asymmetry awareness. In *IEEE International Conference on Computer Design (ICCD)*, pages 497–500.

Vivero, C. D., Kaplan, F., and Coskun, A. K. (2015). Experimental validation of a detailed phase model on a hardware testbed. *ASME International Electronic Packaging Technical Conference and Exhibition*, 1:V001T09A086.

Xie, H., Ali, A., and Bhatia, R. (1998). The use of heat pipes in personal computers. In *IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITHERM)*, pages 442–448.

Yazawa, K., Ziabari, A., Koh, Y. R., Shakouri, A., Sahu, V., Fedorov, A. G., and Joshi, Y. (2012). Cooling power optimization for hybrid solid-state and liquid cooling in integrated circuit chips with hotspots. In *IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITHERM)*, pages 99–106.

Yoo, D. and Joshi, Y. (2004). Energy efficient thermal management of electronic components using solid-liquid phase change materials. *IEEE Transactions on Device and Materials Reliability*, 4(4):641–649.

Yueh, W., Wan, Z., Joshi, Y., and Mukhopadhyay, S. (2015). Experimental characterization of in-package microfluidic cooling on a system-on-chip. In *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 43–48.

Zanini, F., Atienza, D., Coskun, A. K., and Micheli, G. D. (2009). Optimal multiprocessor SoC thermal simulation via adaptive differential equation solvers. In *IFIP International Conference on Very Large Scale Integration (VLSI-SoC)*, pages 139–146.

Zhao, D., Homayoun, H., and Veidenbaum, A. V. (2013). Temperature aware thread migration in 3D architecture with stacked DRAM. In *International Symposium on Quality Electronic Design (ISQED)*, pages 80–87.

Zhu, Y., Antao, D. S., Chu, K., Chen, S., Hendricks, T. J., Zhang, T., and Wang, E. N. (2016). Surface structure enhanced microchannel flow boiling. *ASME Journal of Heat Transfer*, 138(9):091501–091501–13.

Zhu, Y., Antao, D. S., Xiao, R., and Wang, E. N. (2014). Real-time manipulation with magnetically tunable structures. *Advanced Materials*, 26(37):6442–6446.

CURRICULUM VITAE

Fulya Kaplan

Education

Ph.D., Boston University, 05/2017
Electrical and Computer Engineering Department
Advisor: Professor Ayse K. Coskun
Dissertation Title: "Improving Processor Efficiency through Thermal Modeling and Runtime Management of Hybrid Cooling Strategies"

B.S., Middle East Technical University, 06/2011
Electrical and Electronics Engineering Department

Professional Experience

Sandia National Laboratories, Albuquerque, NM, U.S., 05/2015 to 08/2015
Research Intern, *Supervisor:* Dr. Vitus J. Leung
Discrete Math & Optimization Department – Development of SST, a large scale HPC data center simulator.

Advanced Micro Devices Inc., Boxborough, MA, U.S., 01/2013 to 08/2013
Co-Op Intern, *Supervisor:* Prof. Wayne Burleson & Manish Arora
Fast Forward Project – Thermal modeling and analysis of Phase Change Materials

Research Experience

Performance & Energy-Aware Computing Laboratory, Boston University, Boston, MA, U.S., 09/2011 to 05/2017
Research Assistant, *Advisor:* Prof. Ayse K. Coskun
Thermal modeling and management of advanced cooling strategies; simulation and energy management techniques for HPC data centers.

Embedded Systems Laboratory, EPFL, Switzerland, 09/2016 to 12/2016
Research Intern, *Supervisor:* Prof. David Atienza
Design space exploration and runtime optimization of microfluidic cooling and power generation.

Teaching Experience

Boston University Summer Challenge, Summer 2014

EC311: Introduction to Logic Design, Spring 2012

EC571: Digital VLSI Circuit Design, Fall 2011

Book Chapters

1. Tiansheng Zhang, **Fulya Kaplan**, and Ayse K. Coskun. "Thermal Modeling and Management in 3D Stacked Systems". In *Physical Design for 3D Integrated Circuits.* Editors: Aida Todri-Sanial, and Chuan Seng Tan. CRC Press, (ISBN: 978-1-498-71036-7), pp. 229-244, 2015.

Refereed Journal Publications

1. Jie Meng, Eduard Llamosí, **Fulya Kaplan**, Chulian Zhang, Jiayi Sheng, Martin Herbordt, Gunar Schirner and Ayse K. Coskun. "Communication and Cooling Aware Job Allocation in Data Centers for Communication-Intensive Workloads". In *Elsevier Journal of Parallel and Distributed Computing*, Volume 96, pp. 181-193, October 2016.

2. Jie Meng, Samuel McCauley, **Fulya Kaplan**, Vitus J. Leung, and Ayse K. Coskun. "Simulation and Optimization of HPC Job Allocation for Reducing Communication and Cooling Cost". In *Elsevier Sustainable Computing: Informatics and Systems*, Volume 6, pp. 48-57, June 2015.

Refereed Conference Publications

1. **Fulya Kaplan**, Sherief Reda and Ayse K. Coskun. "Fast Thermal Modeling of Liquid, Thermoelectric sand Hybrid Cooling". In *The Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITHERM)*, May 2017.

2. **Fulya Kaplan**, Ozan Tuncer, Vitus J. Leung, Scott K. Hemmert and Ayse K. Coskun. "Unveiling the Interplay Between Global Link Arrangements and Network Management Algorithms on Dragonfly Networks". In *Proceedings of International Symposium on Cluster, Cloud, and Grid Computing (CCGrid)*, May 2017.

3. **Fulya Kaplan** and Ayse K. Coskun. "Adaptive Sprinting: How to Get the Most Out of Phase Change Based Cooling". In *Proceedings of International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 37-42, July 2015.

4. Charlie De Vivero, **Fulya Kaplan** and Ayse K. Coskun. "Experimental Validation of a Detailed Phase Change Model on a Hardware Testbed". In *Proceedings of ASME International Electronic Packaging Technical Conference and Exhibition (InterPack)*, pp. V001T09A086, July 2015.

5. **Fulya Kaplan**, Charlie De Vivero, Samuel Howes, Manish Arora, Houman Homayoun, Wayne Burleson, Dean Tullsen and Ayse K. Coskun. "Modeling and Analysis of Phase Change Materials for Efficient Thermal Management". In *Proceedings of International Conference on Computer Design (ICCD)*, pp. 256-263, October 2014.

6. **Fulya Kaplan**, Jie Meng and Ayse K. Coskun. "Optimizing Communication and Cooling Costs in HPC Data Centers via Intelligent Job Allocation". In *Proceedings of International Green Computing Conference (IGCC)*, pp. 1-10, June 2013.

7. J. Meng, **F. Kaplan**, M. Hsieh, A. Coskun. "Topology-aware Reliability Optimization for Multiprocessor Systems". In *Proceedings of International Conference on Very Large Scale Integration (VLSI-SoC)*, pp. 243-248, October 2012.

Refereed Workshop Publications

1. **Fulya Kaplan** and Ayse K. Coskun. "Adaptive Sprinting for Systems with Phase Change Based Cooling". In *Proceedings of Boston Area Architecture Workshop (BARC)*, January 2016.

Patents

1. **Fulya Kaplan**, Manish Arora, Wayne P. Burleson, Indrani Paul and Yasuko Eckert. "Control of Thermal Energy Transfer for Phase Change Material in Data Center". Pub. No. US 2016/0338230 A1, November 2016.

2. **Fulya Kaplan**, Manish Arora, Indrani Paul and Wayne P. Burleson. "Predictive Management of Heterogeneous Processing Systems". Pub. No. US 2016/0077871 A1, March 2016.