ABSTRACT

Title of dissertation: ARCHITECTURAL-PHYSICAL CO-DESIGN OF 3D CPUS WITH MICRO-FLUIDIC COOLING

Caleb Serafy, Doctor of Philosophy, 2016

Dissertation directed by: Professor Ankur Srivastava
Department of Electrical Engineering

The performance, energy efficiency and cost improvements due to traditional technology scaling have begun to slow down and present diminishing returns. Underlying reasons for this trend include fundamental physical limits of transistor scaling, the growing significance of quantum effects as transistors shrink, and a growing mismatch between transistors and interconnects regarding size, speed and power. Continued Moore's Law scaling will not come from technology scaling alone, and must involve improvements to design tools and development of new disruptive technologies such as 3D integration. 3D integration presents potential improvements to interconnect power and delay by translating the routing problem into a third dimension, and facilitates transistor density scaling independent of technology node.

Furthermore, 3D IC technology opens up a new architectural design space of heterogeneously-integrated high-bandwidth CPUs. Vertical integration promises to provide the CPU architectures of the future by integrating high performance proces-

sors with on-chip high-bandwidth memory systems and highly connected network-on-chip structures. Such techniques can overcome the well-known CPU performance bottlenecks referred to as memory and communication wall.

However the promising improvements to performance and energy efficiency offered by 3D CPUs does not come without cost, both in the financial investments to develop the technology, and the increased complexity of design. Two main limitations to 3D IC technology have been heat removal and TSV reliability. Transistor stacking creates increases in power density, current density and thermal resistance in air cooled packages. Furthermore the technology introduces vertical through silicon vias (TSVs) that create new points of failure in the chip and require development of new BEOL technologies. Although these issues can be controlled to some extent using thermal-reliability aware physical and architectural 3D design techniques, high performance embedded cooling schemes, such as micro-fluidic (MF) cooling, are fundamentally necessary to unlock the true potential of 3D ICs.

A new paradigm is being put forth which integrates the computational, electrical, physical, thermal and reliability views of a system. The unification of these diverse aspects of integrated circuits is called Co-Design. Independent design and optimization of each aspect leads to sub-optimal designs due to a lack of understanding of cross-domain interactions and their impacts on the feasibility region of the architectural design space. Co-Design enables optimization across layers with a multi-domain view and thus unlocks new high-performance and energy efficient configurations. Although the co-design paradigm is becoming increasingly necessary in all fields of IC design, it is even more critical in 3D ICs where, as we show, the inter-

layer coupling and higher degree of connectivity between components exacerbates the interdependence between architectural parameters, physical design parameters and the multitude of metrics of interest to the designer (*i.e.* power, performance, temperature and reliability). In this dissertation we present a framework for multi-domain co-simulation and co-optimization of 3D CPU architectures with both air and MF cooling solutions. Finally we propose an approach for design space exploration and modeling within the new Co-Design paradigm, and discuss the possible avenues for improvement of this work in the future.

ARCHITECTURAL-PHYSICAL CO-DESIGN OF 3D CPUs WITH MICRO-FLUIDIC COOLING

by

Caleb Serafy

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2016

Advisory Committee:
Professor Ankur Srivastava, Chair/Advisor
Professor Donald Yeung
Professor Joseph JaJa
Professor Manoj Franklin
Professor Alan Sussman

© Copyright by
Caleb Serafy
2016

Acknowledgments

I would like to thank my advisor, Professor Ankur Srivastava for the support and guidance he has provided throughout my time in the Ph.D. program at the University of Maryland. Professor Srivastava has always been very available to meet and discuss research while at the same time allowing his students to foster self sufficiency and creative critical thinking on their own. Professor Srivastava demands the highest quality of work from his students, but in return offers reliable support both financially and technically, resulting in a very strong and fruitful advisor-student relationship that facilitates significant contributions to the research community.

I would also like to thank Donald Yeung for the many hours we have spent together discussing research and for his many insights and suggestions regarding how to apply our EDA research base with problems of interest in the architectural community. Identifying and advancing the state of the art at the crossover between the two disciplines is the fundamental motivation behind this dissertation.

Furthermore I would like to thank Professor Ankur Srivastava, Professor Donald Yeung, Professor Joseph JaJa, Professor Manoj Franklin and Professor Alan Sussman for their time to serve on this committee and their valuable technical feedback on the content of this dissertation. I would also like to thank Professor Avram Bar-Cohen, Professor Uzi Vishkin, Professor Yogendra Joshi, Professor Sudhakar Yalamanchili and all of their respective students for their technical contributions to the work put forth in this dissertation.

I would be remiss not to thank my wonderful colleagues. First I should thank my senior colleagues Dr. Bing Shi and Professor Domenic Forte for their guidance and friendship as I began by academic career and now as I transition into the industry. Second I thank my current colleagues, Tiantao Lu, Chongxi Bao, Zhiyuan Yang, Yang Xie and Yuntao Liu. I thank you for all the great technical work we have collaborated on, and the fruitful and interesting research discussions we have had. I am grateful for the lifelong friendships and professional relationships I have developed during my time in this group.

Finally I thank my lovely wife Kacee for all her encouragement, support and self-sacrifice to make this dissertation possible. While I worked long hours at the lab Kacee has done more than her share to help provide for our family and take care of our two beautiful daughters. I thank my parents for raising me to appreciate academia, inspiring me to pursue doctoral studies, and providing moral and financial support throughout my studies.

Table of Contents

List of Tables vii

List of Figures viii

List of Abbreviations xi

List of Publications xi

1 Introduction 1
- 1.1 Advantages of 3D Integration . . . 3
- 1.2 Thermal and Reliability Issues . . . 6
- 1.3 3D IC Co-Design . . . 7
- 1.4 Thesis Outline . . . 9

2 3D CPUs: Background and Motivation 12
- 2.1 Three-Dimensional Integration . . . 12
- 2.2 Memory Wall . . . 14
- 2.3 3D Memories . . . 15
  - 2.3.1 Wide-IO . . . 16
  - 2.3.2 Hybrid Memory Cube . . . 17
- 2.4 Memory-on-Logic 3D CPU . . . 18
  - 2.4.1 Capacity Limitations . . . 19
- 2.5 3D Super-Mesh NOC . . . 20
  - 2.5.1 3D Super-Mesh TSV Requirements . . . 23
  - 2.5.2 3D NOC-Bus Hybrid . . . 24
- 2.6 Thermal Issues . . . 24
- 2.7 Reliability Issues . . . 27
- 2.8 Micro-Fluidic Cooling . . . 28

3 3D CPU Co-Simulation Co-Optimization Flow 31
- 3.1 Architectural Design Space . . . 33
- 3.2 Performance Simulation . . . 34
  - 3.2.1 Benchmarks . . . 36

3.3 DRAM Latency Model . . . 36
3.3.1 MC Queuing Delay . . . 37
3.3.1.1 Derivation . . . 37
3.4 Power/Area Estimation . . . 38
3.4.1 Pumping Power . . . 39
3.5 Core Netlist . . . 40
3.6 Wire Delay Model . . . 42
3.7 Reliability Model . . . 43
3.8 Thermal Model . . . 46
3.8.1 Leakage Model . . . 47
3.9 Floorplan Optimization . . . 48
3.9.1 Floorplan Representation . . . 50
3.9.2 Simulated Annealing Approach . . . 51
3.9.3 Speeding Up Simulation Time . . . 52
3.9.4 Core Tiling and NOC Design . . . 53
3.9.5 Example . . . 54
3.10 Cooling Optimization . . . 56
3.10.1 Microchannel Placement Representation . . . 57
3.10.2 Simulated Annealing Approach . . . 58
3.10.3 Example . . . 58
3.10.4 Microchannel Cost Model . . . 61
3.11 Simultaneous Optimization . . . 64

4 Architectural Opportunities of Micro-Fluidically Cooled 3D CPUs 64
4.1 2D vs. 3D CPUs and the need for MF cooling . . . 65
4.1.1 Performance . . . 68
4.1.2 Temperature . . . 68
4.1.3 Thermally Feasible Performance . . . 74
4.1.4 Power . . . 74
4.2 Frequency Scaling with Micro-Fluidics . . . 78
4.2.1 Design Space and Benchmarks and Metrics . . . 79
4.2.2 Core and Frequency Scaling . . . 80
4.2.3 Scaling Trends . . . 81
4.3 Summary . . . 85

5 Architectural-Physical Co-Design of Micro-Fluidically Cooled 3D CPUs 86
5.1 Thermal-Reliability Aware Architectural-Physical DSE . . . 87
5.1.1 Feasibility Region . . . 88
5.1.2 Optimal Performance . . . 92
5.1.3 Reliability Constraint Sensitivity . . . 94
5.2 Thermal-Bandwidth Trade-offs in MF Cooled 3D CPUs . . . 96
5.2.1 Bandwidth Requirements . . . 99
5.2.2 Memory Controller TSV Density . . . 99
5.2.3 Router TSV Density . . . 100
5.2.4 TSV Density Requirement . . . 100

5.2.5 Bandwidth Capacity . . . 100
5.2.6 Pin Fin Thermal Model . . . 101
5.2.7 Experimental Setup . . . 104
5.2.8 Architectural Parameter Sensitivity . . . 106
5.2.9 Heatsink Parameter Sensitivity . . . 106
5.2.10 Results . . . 108
5.3 Summary . . . 112

6 Design Space Modeling for Physically Constrained 3D CPUs 114
6.1 Previous Work . . . 117
6.2 Contributions . . . 119
6.3 Modeling and Simulation Technique . . . 121
6.3.1 SS-ANOVA Modeling . . . 122
6.3.2 Choosing Model Terms . . . 123
6.3.3 Adding Simulation Points . . . 125
6.3.4 Stopping Criteria . . . 126
6.4 Experimental Setup . . . 127
6.4.1 Architectural Design Space . . . 127
6.4.2 Software Benchmarks . . . 128
6.4.3 Discovery Metrics . . . 129
6.4.4 Modeling and Simulation Parameters . . . 130
6.4.5 Evaluation Metrics . . . 132
6.4.6 Comparison to Other Techniques . . . 133
6.5 Results . . . 134
6.5.1 Design Space Characterization . . . 135
6.5.2 "Optimal" Discovery . . . 137
6.5.2.1 Robustness to Constraint Tightness . . . 139
6.5.3 "Pareto" Discovery . . . 142
6.5.4 Overhead of modeling approach . . . 143
6.6 Summary . . . 143

7 Conclusions and Future Work 145
7.1 Future Work . . . 147
7.1.1 Expansion of Co-Design Scope . . . 148
7.1.1.1 Power Delivery . . . 149
7.1.1.2 Signal Integrity . . . 150
7.1.2 Fine-Grained Design and Integration . . . 151
7.1.3 Runtime Management . . . 152

Bibliography 154

List of Tables

2.1 Comparison of 3D mesh and 3D super-mesh NOC [1] 22

3.1 Architectural parameters 35
3.2 2D vs. 3D DRAM Bus 37
3.3 Micro-fluidic system parameters 40
3.4 CPU core component properties 42
3.5 Transistor and interconnect parameters for 45 nm technology [2] . . . 43
3.6 Thermal model material properties 47

4.1 Study 1: Architectural Design Space 67
4.2 Study 2: Architectural Design Space 79
4.3 Maximum benchmark performance s.t. thermal constraint 81

5.1 Study 3: Architectural Design Space 87
5.2 Micro-fluidic pin-fin heatsink dimensions 97
5.3 Micro-fluidic pin-fin thermal model parameters 103
5.4 Study 4: Architectural Design Space 105
5.5 Normalized Co-design Results 111

6.1 Architectural design space (baseline architecture shown in bold). . . . 128
6.2 Simulated Workloads 129

List of Figures

1.1 (a) Transistor cost [3] (b) wire/gate delay [4] (c) wire/gate power [5] . 4
1.2 Relationship graph for 3D CPU metrics and design variables 7

2.1 3D IC cross section 13
2.2 Memory wall [6]. Multi-core trends plotted for different amounts of workload parallelism. 15
2.3 Stacked DRAM architecture 19
2.4 NOC (left) 2D mesh (right) 3D mesh [7] 21
2.5 Vertical connections in a column of 3D super-mesh routers 23
2.6 Trapped heat effect 25
2.7 Thermal map of (a) processor layer, (b) bottom DRAM layer and (c) top DRAM layer 26
2.8 TSV CTE miss-match stress field 27
2.9 Micro-fluidic heatsink in memory-on-logic 3D CPU 30

3.1 Simulation flow 34
3.2 CPU core component netlist with net widths notated. 41
3.3 TSV EM reliability model 45
3.4 Thermal resistance grids for fluid and solid materials 47
3.5 Thermal-leakage relationship 48
3.6 Example thermally unaware floorplan with MF cooling 54
3.7 Example thermally aware floorplan with MF cooling 55
3.8 Temperature and power density of air cooled floorplan 59
3.9 Temperature and channel distribution using uniform MF heatsink. . . 59
3.10 Temperature and channel distribution using optimized MF heatsink. . 60
3.11 Microchannel cost model example 62

4.1 Average DRAM latency vs. number of memory controllers [8] 67
4.2 Performance vs. MCs and frequency (a) 2D CPU (c) 3D CPU 69
4.3 Temperature vs. MCs and frequency of air cooled 2D CPU 71
4.4 Temperature vs. MCs and frequency (a) air cooled 3D CPU (b) MF cooled 3D CPU 72
4.5 Best achievable performance subject to thermal constraints 73

4.6 Power dissipation vs. MCs and frequency of air cooled 2D CPU . . . 76
4.7 Power dissipation vs. MCs and frequency (a) air cooled 3D CPU (b) MF cooled 3D CPU . . . 77
4.8 3D CPU (a) performance (b) energy efficiency vs. frequency with air cooling and MF cooling . . . 82
4.9 3D CPU (a) temperature (b) power vs. frequency with air cooling and MF cooling . . . 84

5.1 3D CPU design space performance . . . 88
5.2 Thermal feasibility region (shown in white) . . . 89
5.3 Reliability feasibility region (shown in white) . . . 89
5.4 Thermal-reliability feasibility region (shown in white) . . . 90
5.5 Co-design results . . . 92
5.6 Performance improvement due to reliability-aware FP . . . 95
5.7 Micro-fluidic pin-fin cooling of a single layer in a 3D-IC . . . 97
5.8 Control volume around one pin . . . 102
5.9 Normalized metrics of 3D CPU architectural design space . . . 105
5.10 Maximum feasible performance and energy efficiency vs. pin pitch . . . 107
5.11 Thermal feasibility region (shown in white) . . . 109
5.12 Bandwidth feasibility region (shown in white) . . . 109
5.13 Thermal-bandwidth feasibility region (shown in white) . . . 110

6.1 Modeling and simulation technique . . . 121
6.2 Distribution of (a) performance (b) temperature in design space . . . 135
6.3 Temperature vs. performance of entire design space . . . 136
6.4 Optimality of identified design. . . . 138
6.5 Additional simulations required when $T_{violation}$ is reduced from 85 °C to 65 °C. . . . 140
6.6 Accuracy of identified Pareto set. . . . 141

7.1 PDN model in a 3D IC . . . 149
7.2 TSV-TSV coupling circuit model . . . 151

List of Abbreviations

BEOL	Back End of Line
RC	Resistance/Capacitance
HMC	Hybrid Memory Cube
PRAM	Phase-Change RAM
MRAM	Magnetic RAM
MC	Memory Controller
NUMA	Non-Uniform Memory Access
CMP	Chip Multi-Processor
NOC	Network on Chip
CTE	Coefficient of Thermal Expansion
MF	Micro-Fluidic
PPAT	Performance, Power, Area and Timing
M2S	Multi2Sim
IPC	Instructions per Clock
AR	Aspect Ratio
RAT	Register Alias Table
ALU	Arithmetic Logic Unit
IFU	Instruction Fetch Unit
LSU	Load Store Unit
MMU	Memory Management Unit
TLB	Translation Look-aside Buffer
EM	Electromigration
PDN	Power Delivery Network
PDF	Probability Density Function
TCG	Transitive Closure Graph
ROUT	Router
EX	Execution Unit
IPnS	Instructions per Nanosecond
BIPS	Billion Instructions per Second
EDP	Energy Delay Product
Freq	Frequency
T	Thermal
R	Reliability
BW	Bandwidth
DSE	Design Space Exploration
SS-ANOVA	Smoothing Spline Analysis of Variance
ROI	Region of Interest

List of Publications

Journal Publications

J1. Y. Xie, C. Bao, **C. Serafy**, T. Lu, A. Srivastava and M. Tehranipoor, "Security and Vulnerability Implications of 3D ICs", *IEEE Transactions on Multi-Scale Computing Systems*, Accepted March 2016

J2. **C. Serafy**, Z. Yang, Y. Hu, A. Srivastava and Y. Joshi, "Thermo-Electric Co-design of 3D CPUs and Embedded Micro-fluidic Pin-fin Heatsinks", *IEEE Design and Test*, February 2016

J3. **C. Serafy**, A. Bar-Cohen, A. Srivastava and D. Yeung, "Unlocking the True Potential of 3D CPUs with Micro-Fluidic Cooling", *IEEE Transactions on VLSI Systems*, July 2015

J4. **C. Serafy** and A. Srivastava, "TSV Placement and Shield Insertion for TSV-TSV Coupling Reduction in 3-D Global Placement", *IEEE Transactions on CAD: Special Issue on Physical Design Techniques for Advanced Technology Nodes*, January 2015

J5. **C. Serafy**, B. Shi and A. Srivastava, "A Geometric Approach to Chip-Scale TSV Shield Placement for the Reduction of TSV Coupling in 3D-ICs", *Integration, the VLSI Journal by Elsevier: VLSI for the New Era*, December 2013

Journal Publications (Under Review)

R1. T. Lu, **C. Serafy**, Z. Yang, S.K. Lim and A. Srivastava, "3D ICs: Design Methods and Tools", *IEEE Transactions on CAD*, Submitted March 2016

Conference Publications

C1. T. Lu, **C. Serafy**, Z. Yang and A. Srivastava, "Voltage Noise Induced DRAM Soft Error Reduction Technique for 3D-CPUs", *International Symposium on Low Power Electronics and Design (ISLPED)*, August 2016

C2. Z. Yang, **C. Serafy** and A. Srivastava, "ECO Based Placement and Routing Framework for 3D FPGAs with Micro-fluidic Cooling", *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, May 2016

C3. **C. Serafy**, T. Lu and A. Srivastava, "Thermal-Reliability Physical Co-Optimization During Architectural Design Space Exploration of 3D-CPUs", *Government Microcircuit Applications and Critical Technology Conference (GOMACTech)*, March 2016

C4. **C. Serafy**, A. Srivastava, A. Bar-Cohen and D. Yeung, "Design Space Exploration of 3D CPUs and Micro-Fluidic Heatsinks with Thermo-Electrical-Physical Co-Optimization", *International Technical Conference and Exhibition on Packaging and Integration of Electronic and Photonic Microsystems (InterPACK)*, July 2015

C5. **C. Serafy**, A. Srivastava and D. Yeung, "Unlocking the True Potential of 3D CPUs with Micro-Fluidic Cooling", *International Symposium on Low Power Electronics and Design (ISLPED)*, August 2014

C6. **C. Serafy**, A. Srivastava and D. Yeung, "Continued Frequency Scaling in 3D ICs through Micro-fluidic Cooling", *IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, May 2014

C7. **C. Serafy** and A. Srivastava, "Coupling-Aware Force Driven Placement of TSVs and Shields in 3D-IC layouts", *ACM International Symposium on Physical Design (ISPD)*, April 2014

C8. **C. Serafy**, B. Shi, A. Srivastava and D. Yeung, "High Performance 3D Stacked DRAM Processor Architectures with Micro-Fluidic Cooling", *IEEE International 3D Systems Integration Conference (3D-IC)*, October 2013

C9. **C. Serafy** and A. Srivastava, "Online TSV Health Monitoring and Built-in Self-Repair to Overcome Aging", *IEEE Symposium on Defect and Fault Tolerance (DFT)*, October 2013

C10. B. Shi, **C. Serafy** and A. Srivastava, "Co-Optimization of TSV Assignment and Micro-Channel Placement for 3D-ICs", *Great Lakes Symposium on VLSI (GLSVLSI)*, May 2013

C11. **C. Serafy**, B. Shi and A. Srivastava, "A Geometric Approach to Chip-Scale TSV Shield Placement for the Reduction of TSV Coupling in 3D-ICs", *Great Lakes Symposium on VLSI (GLSVLSI)*, May 2013

Magazine Articles

M1. **C. Serafy** and A. Srivastava, "Leakage Power: Physical Mechanisms and Possible Solutions", *Electronics Cooling*, December 2014

Chapter 1: Introduction

CMOS technology has for the last half century taken advantage of aggressive technology scaling, resulting in faster and more densely packed transistors that have provided exponential increases in computing capacity. Over the years, the consumer market for semiconductors has come to expect such a rate of growth to continue far into the future. However, today transistor scaling is approaching fundamental physical and economic limits, and already the rate of increase in computing power and performance has begun to slow.

Vertical integration (3D ICs) is an emerging technology which promises to reinvigorate Moore's Law performance scaling by reducing interconnect power and delay, and facilitating new heterogeneous computer architectures such as stacked memory-on-logic CPUs [9–11]. Additionally, logic-on-logic stacking can create more highly connected circuits and increase inter-core communication bandwidth in multi-core CPUs [7, 12, 13]. Stacking memory-on-logic can provide a high-bandwidth memory interface to the processor [9, 14], overcoming the memory wall [6] and facilitating the processing in memory paradigm [11].

Thus 3D integration brings the potential of many advantages both at the circuit and architectural level. However these advantages come with a cost in terms of physical constraints and increased dependencies between CPU components and across metric domains. The chief limitation associated with 3D ICs is thermal in nature [8, 14–16]. Vertical stacking inherently increases power flux while inter-layer dielectrics significantly increase the thermal resistance of the stack. Other limitations come from the introduction of through silicon vias (TSVs) which introduce new failure modes [17–19] and sources of noise coupling [20–24] while increasing the impedance of the power delivery network [25, 26].

Increased thermal insulation makes 3D IC temperature a much more highly coupled function of CPU architecture, performance and power [8, 27]. Furthermore it is well known that critical path delay, leakage power and reliability are strong functions of temperature, creating an interconnected network of metrics that all influence each other. Although the same fundamental relationships exist in 2D ICs, the higher connectivity, and spatial coupling between stacked components exacerbate these inter-dependencies in 3D to such an extent that simultaneous modeling and optimization is a must [27–32].

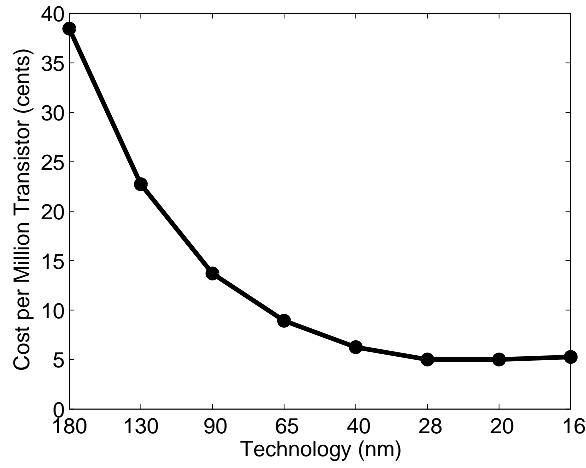
In this dissertation we explore the potential of 3D CPU architectural opportunities and evaluate the associated challenges (*e.g.*, thermal and reliability issues) and their implications on the architectural feasibility space. We propose a co-design paradigm to design 3D CPUs to maximize their performance and/or energy ef-

ficiency under physical constraints and finally propose a modeling and simulation methodology for high dimensionality design space exploration of the 3D CPU design space.

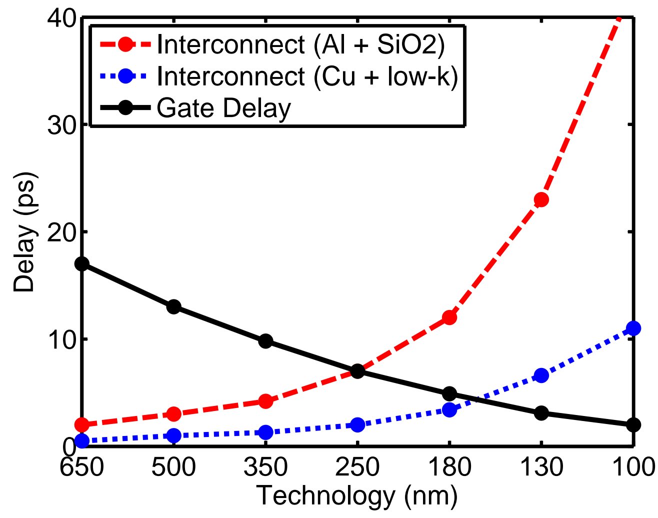
1.1 Advantages of 3D Integration

As transistor sizes approach atomic scale, quantum effects that have traditionally been insignificant begin to significantly effect behavior. Moreover transistor size is fundamentally limited by the dimensions of the atoms used to construct them. Additionally, the traditional scaling trend of manufacturing cost per transistor (Figure 1.1(a)) is expected to stall out very soon, removing a significant economic incentive to invest in future technology nodes [3].

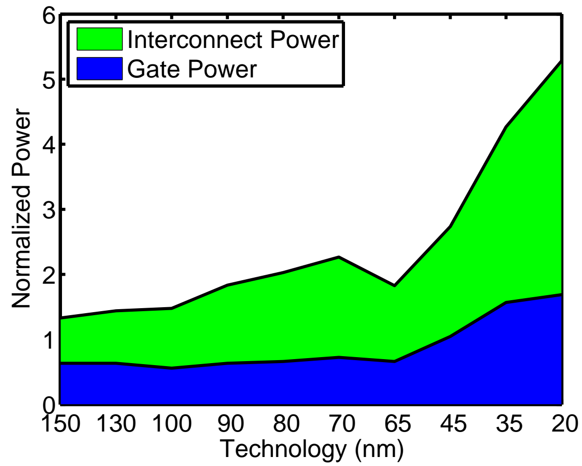
Another issue causing Moore's Law scaling to end is the growing gap in performance and power efficiency of transistors vs. interconnect [4, 5]. Figures 1.1(b) and 1.1(c) show the trends of transistor and interconnect delay and power respectively as technology has advanced. Transistors are clearly increasing in speed due to smaller input capacitance whereas interconnect is decreasing in speed due to smaller more resistive wires, and more wire-wire parasitic capacitance [33]. For similar reasons, chip-scale transistor power remains nearly flat over time while interconnect power is increasing at a much faster rate [5]. Closing the gap between transistors and wires is necessary to continue historical scaling trends of power and performance over time.



(a)



(b)



(c)

Figure 1.1: (a) Transistor cost [3] (b) wire/gate delay [4] (c) wire/gate power [5]

Engineers are aggressively investigating new technologies and paradigm shifts that can continue to provide the market with the growth it expects, even as technology scaling has begun to stall out. Transistors have traditionally been laid out in a two dimensional plane on a silicon wafer. One technique to improve transistor and interconnect density without the use of technology scaling is to pack transistors into three dimensional space, resulting in what are called three-dimensional integrated circuits (3D ICs). In addition to increasing transistor density, which can increase circuit performance and reduce power consumption, 3D integration can theoretically reduce interconnect length by a factor of $\sqrt{N}$ where N is the number of stacked layers [34]. Assuming optimal buffer insertion, this would reduce wire delay and power proportionally [35].

Another advantage of vertical integration is chip level integration of circuits manufactured in disparate technologies, referred to as heterogeneous integration. This allows circuits such as analog sensors, MEMs, RF, DRAM, and CMOS to all be integrated together, extending the system on a chip (SoC) paradigm to many new applications. Not only can heterogeneous integration make new SoC designs feasible, it can improve the quality of current SoC designs, by allowing different components of the design to be fabricated in a manufacturing process optimized for that specific component. Circuits that are traditionally fabricated as separate chips and connected using an interposer or PCB can be vertically integrated with TSVs, greatly increasing the bandwidth between these chips, and opening up opportunity to redesign how such circuits interact with one another, possibly increasing performance and/or decreasing power consumption.

1.2 Thermal and Reliability Issues

Temperature and reliability are two of the most important challenges associated with 3D ICs. Other challenges include signal integrity and power delivery [26]. Thermal challenges arise from the increased power flux inherent to 3D stacking. High temperatures can cause timing violations by increasing transistor and interconnect resistance, and excessively high temperatures can even cause permanent physical damage to the chip. Thus chip temperature plays a critical roll in both soft and hard error reliability. Temperature significantly effects leakage power. Increased power leads to higher current density which can cause electromigration and IR voltage drop in the power delivery network (PDN). Furthermore temperature fluctuations can cause TSV defect formation from thermal cycling and so called TSV pop-out and delamination [36].

Although traditional 2D circuits can address the thermal and related reliability issues by attaching a large heatsink to the back side of the chip to dissipate the heat to the environment, this approach is not applicable to 3D ICs. An attached heatsink can only remove significant heat from the top layer, as other layers are sandwiched between electrical isolation layers composed of SiO_2 which block heat dissipation and cause high temperatures [27, 28]. We refer to this as the trapped heat effect (Figure 2.6). Micro-fluidic cooling is a promising technology for localized embedded cooling that can overcome the trapped heat effect and scale cooling capacity with

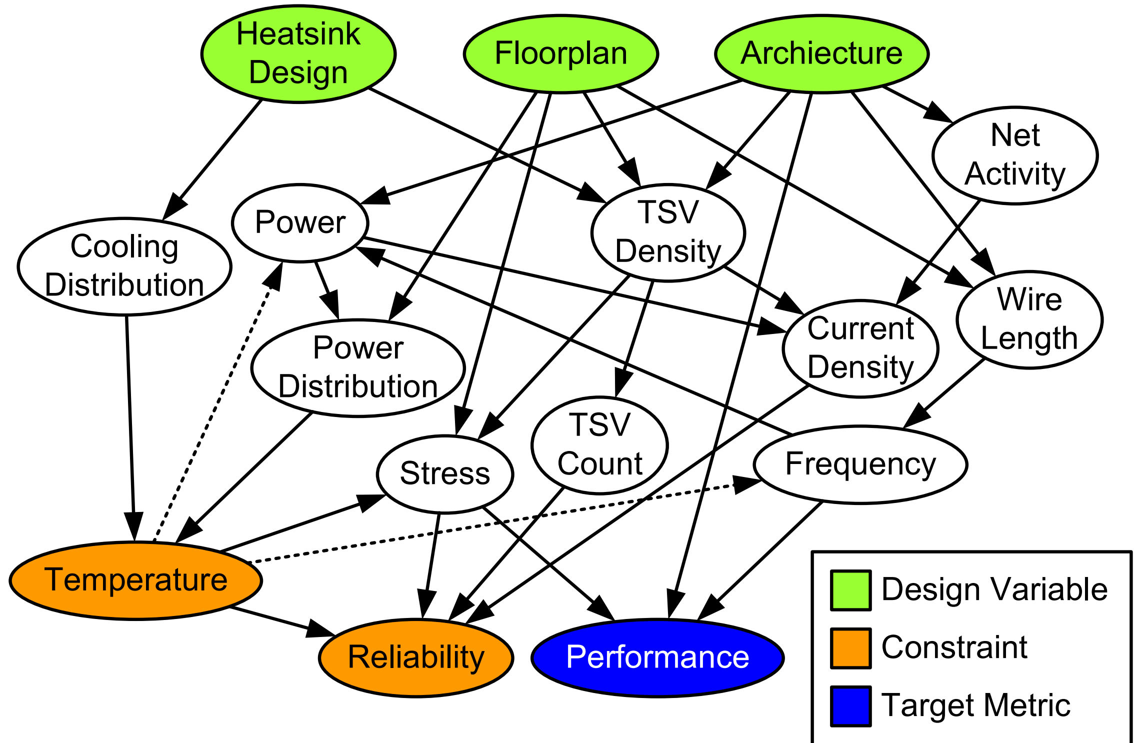


Figure 1.2: Relationship graph for 3D CPU metrics and design variables

number of layers. In our work we examine the power, performance, thermal and reliability interdependence and show the massive potential of micro-fluidically cooled and multi-objective co-design in 3D CPUs.

1.3 3D IC Co-Design

In the previous sections we have discussed the physical design challenges (*e.g.*, temperature and reliability) and the architectural opportunities of 3D integration. Traditionally the physical and architectural designs are performed independently in sequence using different levels of abstraction. Moreover, even within the physical design domain, design problems are tackled sequentially, and cross-domain opti-

mizations are not usually considered. A new paradigm which integrates the computational, electrical, physical, thermal and reliability views of the system is gaining steam. This unification of diverse aspects of the overall integrated system is called Co-design. Co-design enables optimizations across different layers of the design hierarchy which are not possible through a conventional top down design approach thereby unlocking new high performance configurations.

In the remainder of this dissertation we use 3D CPUs as a case study to exemplify the interdependence of the physical and architectural design spaces. We use a novel simulation flow which integrates placement, temperature and reliability design challenges into a unified framework for architectural-physical optimization and analysis (Chapter 3).

Figure 1.2 illustrates the cause and effect relationships from some chosen design variables to the optimization and constraint metrics of interest. The figure clearly illustrates the interdependence between the terminal and intermediate nodes, and no metric of interest can be determined without simultaneous consideration of all design variables. The interconnectedness of this relationship graph strongly motivates the need for the co-design paradigm. Isolating any subset of graph nodes from Figure 1.2 requires cutting many edges. In other words estimates calculated from a subset of design metrics, variables and objective functions suffer from comprised accuracy due to the high connectivity in the graph and large loss of information when graph edges are removed.

Furthermore, we observe that the relationship graph contains cycles, which imply nested loops within a simulation flow. An example is the interdependence of temperature and leakage power. Leakage power increases as temperature elevates, and likewise temperature will rise when leakage power increases. Iterative simulations are required to accurately capture such inter-dependencies. Co-design design space exploration (DSE) is a computationally intensive problem due to both optimization loops and nested simulation loops within the evaluation flow of a single design candidate.

1.4 Thesis Outline

In this thesis we first provide some in depth background information on 3D CPUs in Chapter 2. This includes details on the architectural advantages of 3D integration, the physical design issues and micro-fluidic cooling. In Chapter 3 we introduce the simulation flow used to estimate metrics of interest for a given 3D CPU architecture, including performance, power, temperature and reliability. Furthermore we introduce here the physical design optimization loops evaluated in Chapter 5.

Chapter 4 evaluates the advantages in performance and energy efficiency that can be achieved by 3D CPUs. Our first study shows significant performance potential, but this potential is not realized with traditional air cooling, and MF cooling is required to unlock the benefits of high-bandwidth stacked memory. In our second study we consider how micro-fluidic cooling and 3D memory-on-logic stacking can

revitalize the classic frequency scaling paradigm in parallel with the current core scaling model. Some of the major reasons frequency scaling came to an end was temperature and memory bandwidth issues, which are largely overcome by memory-on-logic stacking and MF cooling.

Chapter 5 evaluates the effectiveness of physical co-design towards expanding the 3D CPU architectural design space feasibility region and thus unlocking new high-performance high-energy-efficient CPU architectures of the future. Physical design of both the logic and the heatsink are explored subject to simultaneous and interrelated temperature and reliability constraints. One interesting result is that temperature and reliability optimization can be at conflict with one another, which seems counter-intuitive, and further justifies the need for a co-design approach that is aware of the intricate trade-offs between multiple design variables.

Another study reported in this chapter investigates the fundamental trade-off between cooling capacity and inter-layer bandwidth (*i.e.* TSV density) in a MF cooled 3D IC. We show that using a generic heatsink design geared towards minimizing temperature or maximizing TSV density only leads to significant performance sub-optimality, and a co-design approach is necessary to discover the best heatsink parameters for each architectural design point.

Chapter 6 introduces a modeling and simulation scheme to bring the co-design framework discussed in previous chapters into practical use on large multi-dimensional problems. The 3D CPU co-simulation framework introduced in Chapter 3 covers a wide array of different simulations and model, and thus consumes a non-trivial amount of compute resources. Exhaustive application of this simu-

lation flow over a large industry-scale design space may not be computationally feasible. Thus we propose a methodology to accurately predict the design space and identify regions of interest (*e.g.*, optimal-feasible region or Pareto optimal front) while simulating only a small percentage of the design space. Our results show high accuracy compared to randomized or modeling-only approaches, and makes the co-design paradigm developed in this dissertation practically applicable to real design problems.

Finally Chapter 7 concludes the dissertation with a summary of the work completed, and some recommendations for future work. Avenues for continuation of the work begun in this dissertation include integration of additional design metrics and models, a hierarchical co-design framework to progress from high-level to detailed design, efficient methods of cutting the co-design graph to balance design time with quality, and the integration of runtime management approaches into the co-design framework.

Chapter 2: 3D CPUs: Background and Motivation

3D Stacking is an emerging technology which offers many new opportunities for high performance CPU architectures. The memory wall [9] is a known hurtle to future performance and power scaling, and 3D integration is a promising technology to overcome it. Stacked memory circuits are already in commercial production [37, 38] and heterogeneous memory-on-logic CPUs are being aggressively researched and prototyped [14, 27, 39]. Moreover, communication overheads in both power and delay have become more and more significant as we have entered the age of big data. This is the so-called communication wall [40]. 3D CPUs offer new solutions such as high-bandwidth on-chip processing-in-memory [11, 41, 42] and highly connected 3D NOC topologies [13, 27, 43]. Finally we discuss some of the physical challenges associated with 3D CPUs, potential solutions, and the need for a co-design paradigm to optimize for strong architectural-physical interactions inherent to 3D CPUs.

2.1 Three-Dimensional Integration

3D ICs are formed by stacking multiple layers of traditional (2D) ICs one atop the other. Some nets in the 3D circuit span multiple layers, and must be connected with vertical interconnects. The most prominent type of vertical interconnect is

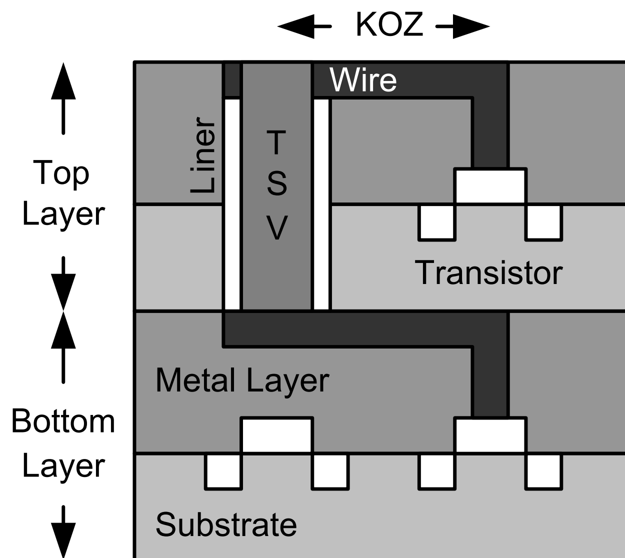


Figure 2.1: 3D IC cross section

called the through silicon via (TSV). TSVs are vertical columns of metal that pass through the silicon substrate and connect the horizontal metal wires in adjacent IC layers, as shown in Figure 2.1. TSVs are used to deliver both signals and power between layers of a 3D IC. Because a TSV passes through the substrate, transistors and TSVs cannot coexist at that same location in the same layer. Hence TSV placement effects the positions of transistors and the length of wires, which determine the overall delay of a circuit.

TSVs pass through the electrically charged and conductive silicon substrate, and so they must be surrounded by a layer of insulating material to decouple them from the substrate. This layer of insulation is called the liner, and is typically made of silicon dioxide (SiO_2). There exists a minimum spacing between TSVs and other features such as transistors and other TSVs, which must be enforced in order to guarantee proper functionality of the chip. This minimum spacing is called the keep

out zone (KOZ) and is determined by the precision of the manufacturing process and TSV effects such as thermally-induced stress around a TSV due to the mismatch in thermal expansion of the silicon, the liner, and the TSV [44].

Vertical integration is a promising new technology and can continue transistor density scaling as technology scaling slows down due to physical limitations. Beyond transistor density scaling, 3D integration brings other unique advantages. Because each layer in a chip stack is manufactured independently, 3D integration can facilitate heterogeneous integration by manufacturing different layers with disparate manufacturing processes. Vertical integration also increases the overall connectivity of a system by decreasing the average distance between system components, thus decreasing global wirelengths, critical path delays and interconnect power. By implementing a circuit in N layers, the global wirelength can be reduced by up to a factor of $\sqrt{N}$ [34].

2.2 Memory Wall

The so-called memory wall describes the limitation put on processor performance and energy efficiency due to a lack of high-bandwidth, high-density low-power DRAM circuits. The term was originally coined to describe the gap in CPU and memory performance, as shown in Figure 2.2. An initial solution to this gap was the addition of cache memory on chip to hide the DRAM latency, but caches are limited in size due to silicon area and leakage power constraints. Moreover as the multi-core paradigm has matured, memory bandwidth has become a limitation not

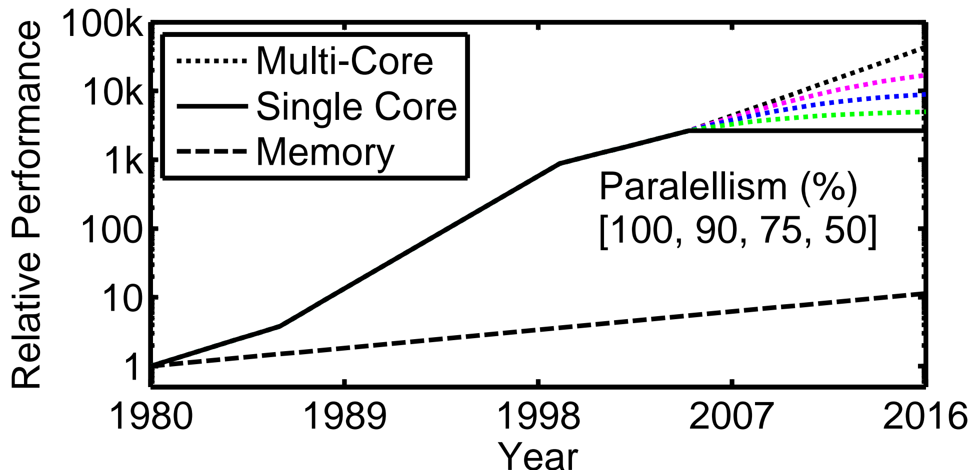


Figure 2.2: Memory wall [6]. Multi-core trends plotted for different amounts of workload parallelism.

just due to DRAM speed, but also due to increased memory access rates as more cores operate in parallel. The memory wall is a key obstacle in the climb towards next generation computing: both mobile and exascale supercomputing.

2.3 3D Memories

3D integration is an enabling technology to further the three memory design goals: higher density, higher bandwidth, and lower power. Vertical stacking inherently increases memory density within a fixed footprint area, and heterogeneous integration facilitates high speed, and/or very wide TSV memory buses which dissipate considerably less power than their off-chip counterparts.

Two main strategies have been employed towards bringing 3D memory into the commercial market. One focuses on speed using very high speed differentially signaled serial interconnects. Although this strategy increases absolute power, the power efficiency (bandwidth per Watt) is much improved. An example of such an

architecture is Micron's Hybrid Memory Cube (HMC) [37]. Alternatively a wide parallel bus can be pursued taking advantage of the tremendous interconnect density offered by TSV technology [37]. This strategy can massively improve memory bandwidth without increasing power, or alternatively provide very low power operation at nominal performance. An example of such an architecture is Samsung's Wide-IO DRAM [38].

2.3.1 Wide-IO

The Wide-IO memory architecture consists of 4 independent channels each with a 128 bit data bus. Each channel contains four 64 Mb arrays, for a total capacity of 1Gb per layer. The Wide-IO memory can deliver peak bandwidth up to 12.8 GB s^{-1}, 4x higher than the equivalent LPDDR2 device, while increasing bandwidth per Watt of IO power by more than 10x [38]. The Wide-IO 2 specification has been released by JEDEC and makes many significant improvements [45]. The number of channels can be increased from 4 to 8, the density ranges from 8 to 32 Gb and the peak bandwidth tops out at 34 (4 channel) or 68 (8 channel) GB s^{-1}. Moreover the operating voltage is reduced from 1.2 to 1.1 V, providing even lower power. Wide-IO 2 is expected to surpass the performance of LPDDR4 in 3D stacked devices [45].

Wide-IO memory is intended to be integrated directly on top of logic using TSVs. This approach is ideal for density and power, but has thermal implications. Wide-IO is expected to be used in high-end smart phones, but in the absence of embedded active cooling schemes may not be thermally feasible in a server or supercomputer environment [46].

2.3.2 Hybrid Memory Cube

The HMC is connected to the CPU through a board-level high speed differential serial interface [37]. However the cube itself is composed of stacks of DRAM on top of a layer of CMOS. This heterogeneous integration allows for optimized common logic circuits such as decoders and memory controllers while maintaining the memory density characteristics of stacked DRAM. HMC facilitates a distributed architecture called "Far" mode [37] where multiple HMCs are connected together to form a memory network for scalable high capacity memory systems. HMC moves the memory controller to the DRAM module itself rather than the core in order to efficiently realize such a scaled architecture.

The HMC significantly improves DRAM latency by reducing memory controller queuing delays and providing more memory parallelism though independent bank operation. Experimental data from first generation HMC prototype reports DRAM bandwidth of 128GB s^{-1}while dissipating 11 W, improving bandwidth per Watt more than 3.5x over DDR4 [37].

Analysis by TSMC [46] shows that Wide-IO 2 brings the best of both worlds by providing performance parity with DDR4 while matching LPDDR4 in power dissipation. On the other hand the HMC is a revolutionary new memory architecture that pushes performance, power and price to new extremes.

2.4 Memory-on-Logic 3D CPU

Heterogeneous 3D integration can provide massive bandwidth improvements between CPU core logic and memory. Non-CMOS technologies such as DRAM, phase-change RAM (PRAM) and magnetic RAM (MRAM) [47] can be stacked directly on top of logic cores. Stacked memory-on-logic DRAM architectures are a natural solution to the memory wall problem as they can offer high-bandwidth, low-latency, low-power interconnects between memory and CPU. Increases in bandwidth and power efficiency come from reduction in interconnect length (*i.e.* RC parasitics) and massively increased integration density of TSVs as compared to off-chip PCB traces [9, 27]. TSV integration can facilitate many more memory controller (MC) modules to increase memory access parallelism at the expense of increased power, temperature and area [8, 9, 12].Studies have shown that the performance improvements due to main memory stacking can be up to 2x [8, 9]. Stacked DRAM is considered to be one of the primary advantages of 3D CPUs [9, 39]. A cross section of a stacked DRAM memory-on-logic 3D CPU is shown in Figure 2.3.

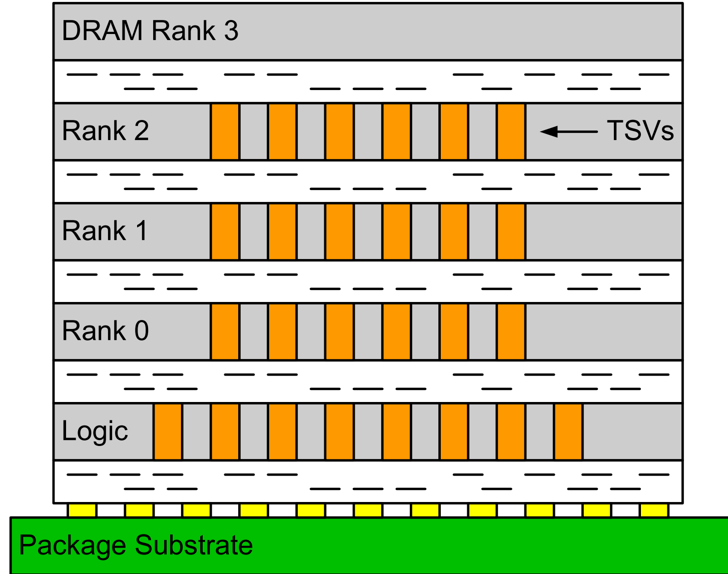


Figure 2.3: Stacked DRAM architecture

2.4.1 Capacity Limitations

The capacity of on-chip DRAM is limited to only a few GB [11, 27]. Thus most computing systems require both on and off-chip DRAM. On-chip DRAM could be leveraged as cache or a non-uniform memory access (NUMA) paradigm can be applied [48] to manage both on and off-chip DRAM as a unified main memory. Even within a stacked DRAM module, non-uniform access constraints may need to be applied due to non-uniform power delivery capacity in the 3D stack [49]. Such NUMA systems require memory swap controllers to keep hot memory pages in low-latency portions of the memory [48, 49].

Studies have shown the effectiveness of using stacked DRAM for additional cache rather than main memory. DRAM cache can offer large capacity compared to an SRAM cache of the same area [50] while maintaining higher bandwidth and lower latency compared to main memory [51]. Moreover hot page migration into a DRAM

cache can be done at the cache line granularity whereas NUMA stacked memory systems must swap memory at the page granularity, which is both inefficient and requires OS support [48].

However there are two main limitations to DRAM cache: the tag array would be unreasonably large for standard (*e.g.*, 64 MB) cache line sizes, and off-chip main memory cannot provide the necessary bandwidth to use significantly larger cache line sizes. Jiang *et al.* [51] proposed a hot-page filtering technique to efficiently manage the DRAM bandwidth to leverage performance improvements of up to 25% from a 128 MB DRAM cache. Loh [50] leveraged the DRAM row buffer hardware to further increase DRAM cache performance by 29% by employing an adaptive multi-queue policy. On the other hand, Chou *et al.* [48] presented a low overhead technique that allows NUMA stacked memory to achieve cache-line level data migration, outperforming both DRAM cache and traditional NUMA stacked memory.

2.5 3D Super-Mesh NOC

Traditionally, communication between caches, cores and IO devices has been accomplished using a bus architecture. A bus is a shared communication fabric where communication is broadcast to all bus nodes. While such an architecture is fast, it has been shown to scale poorly when the number of bus nodes surpasses roughly 10 [13] due to bus contention in the shared fabric. Today's chip multiprocessors (CMPs) already have more than 10 cores, and are expected to continue scaling to hundred or even thousands of nodes [52]. Thus the network on chip (NOC) has

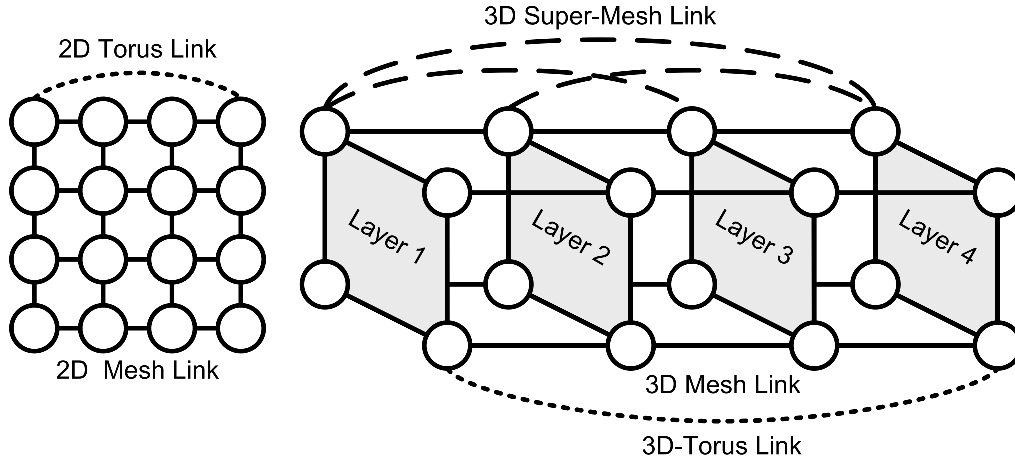


Figure 2.4: NOC (left) 2D mesh (right) 3D mesh [7]

become standard communication fabric in modern multi-core architectures. NOC-s use a packetized routing network. Thus many communication packets can be simultaneously passed through the network across independent router links.

The standard NOC topology has been a 2D mesh where nodes are spread uniformly in two dimensions and each router connects to its four Manhattan neighbors as well as its local node [7, 13]. However in many-core systems, whether distributed or integrate on chip, inter-core communications delays have begun to dominate [11, 53–55]. This is called the communication wall. The extension of the mesh topology into 3D has been shown to provide significant improvements in latency, throughput and energy efficiency [7, 43]. However, due to the mismatch in vertical (hundreds of microns) and horizontal (millimeters) length of inter-core router links, more innovative NOC topologies that provide higher connectivity in the vertical direction have also been proposed [7, 12, 13].

One simple extension that can be applied to either 2D or 3D mesh topologies is the torus ring. The torus adds a connection between the first and last node in each row and column of a mesh. This modification reduces the diameter (*i.e.* worst case distance) of the NOC, but introduces non-uniform delay hops which complicate routing algorithms. However this can be significantly offset by use of a folded torus topology. In general torus topology has less latency but consumes more power [56].

In the vertical direction, the motivation behind the torus architecture can be further extended to include connecting all nodes in a vertical column due to the relatively small distance between nodes on adjacent layers. Circuit analysis estimates that multilayer routing channels can traverse up to four layers in the vertical direction with the same delay as a horizontal connection between adjacent cores [1,57]. The 3D super-mesh topology was introduced in [27] which connects each pair of network nodes in a vertical column with a dedicated router link. Performance improvements and power and area overheads versus standard 3D-NOC are shown in Table 2.1. Mesh, torus and super-mesh topologies are illustrated in Figure 2.4.

Table 2.1: Comparison of 3D mesh and 3D super-mesh NOC [1]

Metric	3D super-mesh	3D mesh	Ratio
IPC	29.3	25.3	1.16
Average Latency (cycles)	42.9	49.4	0.87
Total CPU Power (W)	315	284	1.11
Total CPU Area (m^2m)	1580	1516	1.04

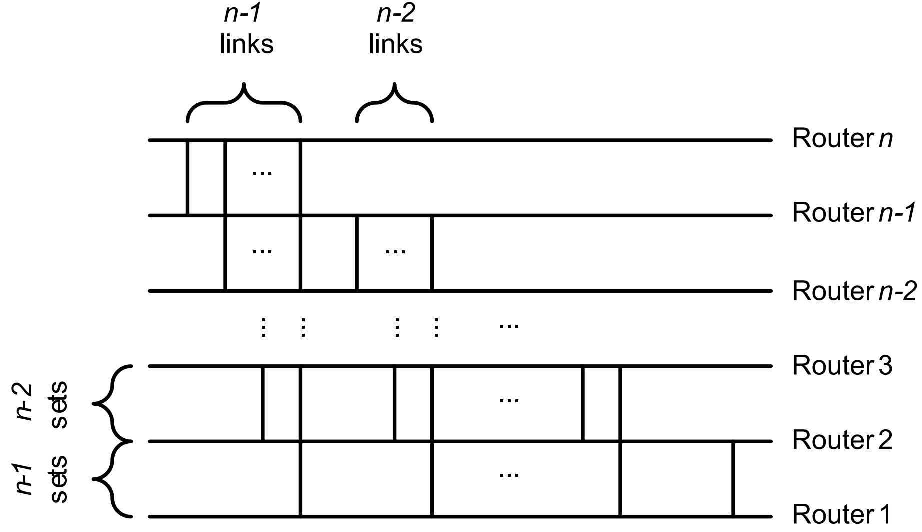


Figure 2.5: Vertical connections in a column of 3D super-mesh routers

2.5.1 3D Super-Mesh TSV Requirements

In a 3D CPU with a 3D super-mesh NOC on n logic layers, each router requires $n-1$ vertical links to directly connect to all routers above and below it. Each vertical connection between layer i and layer j requires a TSV between all adjacent layers from i to j. Hence, the total number of TSVs that passes between layer i and layer $i+1$ in a vertical column of 3D super-mesh NOC routers is given in Equation (2.1) as T_{ROUT} and illustrated in Figure 2.5. W_{link} is the bit width of the router link. In the studies presented in this dissertation $W_{link} = 128$ bits.

$$T_{ROUT}(i) = W_{link} i(n-i) \tag{2.1}$$

2.5.2 3D NOC-Bus Hybrid

A hybrid structure for 3D NOC has been proposed in [13]. A traditional 2D mesh is used in each layer, but a subset of the routers on each layer are connected to a vertical bus that allows broadcast communication between all routers in a vertical column. This approach achieves full communication between all layers in the vertical direction while minimizing the number of ports (and thus the power and area) of each router. The number of nodes on each vertical bus is equal to the number of layers in the NOC which is typically less than 10 [58], implying that bus is a reasonable communication fabric in the vertical direction. Results show that the proposed 3D NOC-bus hybrid structure applied to a shared banked L2 cache outperforms a 2D NOC. Moreover it is shown that cache line mitigation is much less common in the 3D NOC due to higher connectedness between nodes, and even with cache line mitigation turned off in the 3D NOC, it still outperforms 2D [13].

2.6 Thermal Issues

The chief challenge associated with 3D integration is thermal management. Thermal challenges in 3D ICs are twofold. Unlike technology scaling, 3D integration increases transistor density without reducing the power per transistor. This results in increased power flux as more layers are stacked. Exacerbating this problem, the dielectrics between functional layers have relatively low thermal conductivity, and significantly diminish heat flow from stacked layers to the heat sink in traditional air-cooling schemes. The cooling capacity on each layer of an air-cooled 3D IC degrades

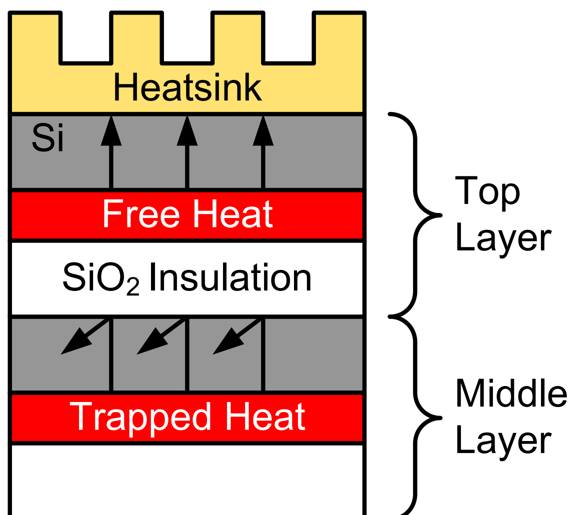


Figure 2.6: Trapped heat effect

as the layer moves farther away from the heatsink, therefore large thermal gradients form in the vertical direction [27]. We call this phenomenon the trapped heat effect (Figure 2.6) and it can result in extremely high peak temperatures [59, 60].

Figure 2.7 shows an example thermal profile for a 3D CPU with two DRAM layers stacked on a 16-core multiprocessor layer (Section 2.4). We observe a large thermal gradient both within a layer and across vertical layers. We also observe significant thermal coupling from the processor layer to the neighboring DRAM layer, even though the DRAM layer has very low power density. This phenomenon leads to increased DRAM leakage and requires shorter refresh periods in memory-on-logic 3D CPUs [61], which has performance implications.

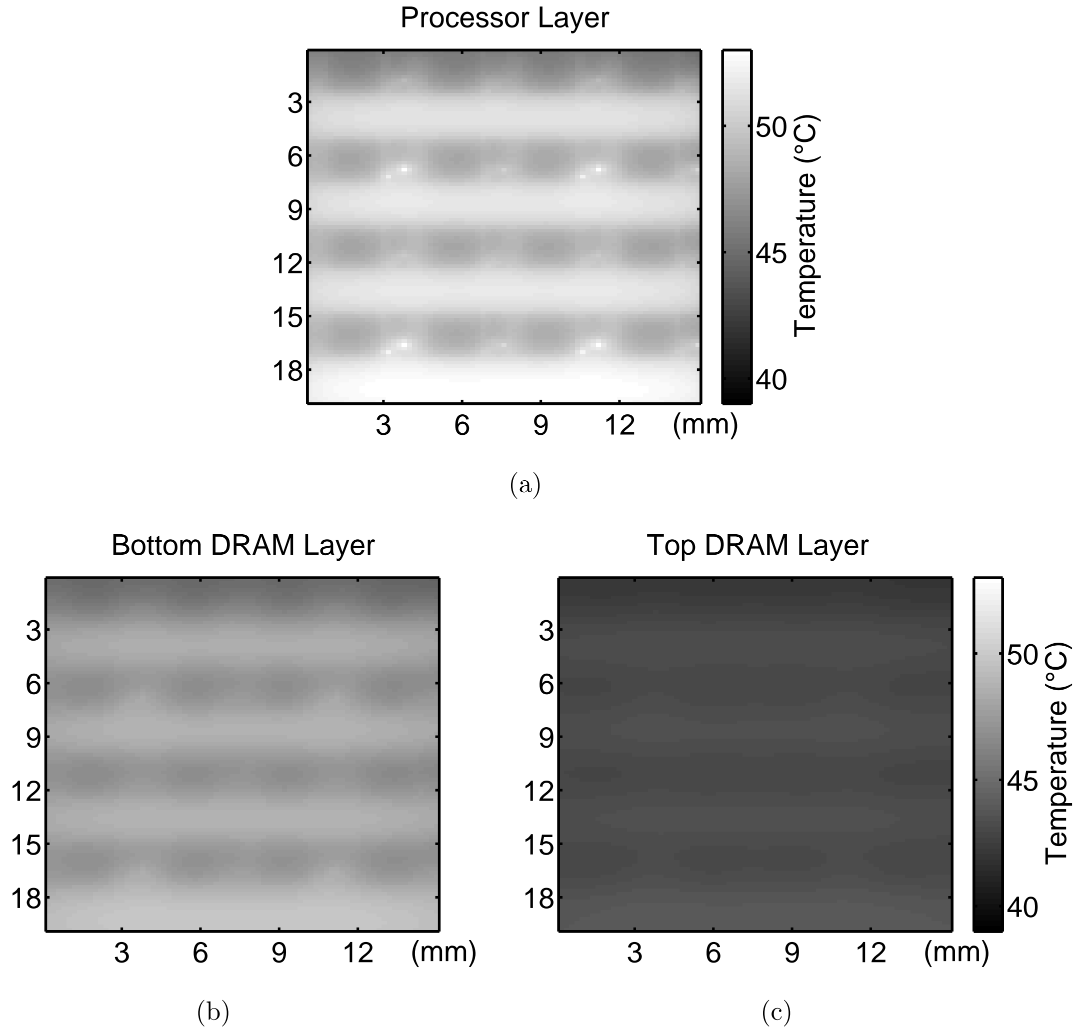


Figure 2.7: Thermal map of (a) processor layer, (b) bottom DRAM layer and (c) top DRAM layer

The high temperatures associated with air cooled 3D ICs cause high leakage power (thus reducing the energy efficiency and possibly resulting in thermal runaway [62]), increased transistor and wire delay (thus degrading performance), and reduced chip reliability (Section 2.7). A promising solution to the thermal issue comes from embedded active cooling technology such as micro-fluidic cooling (Section 2.8).

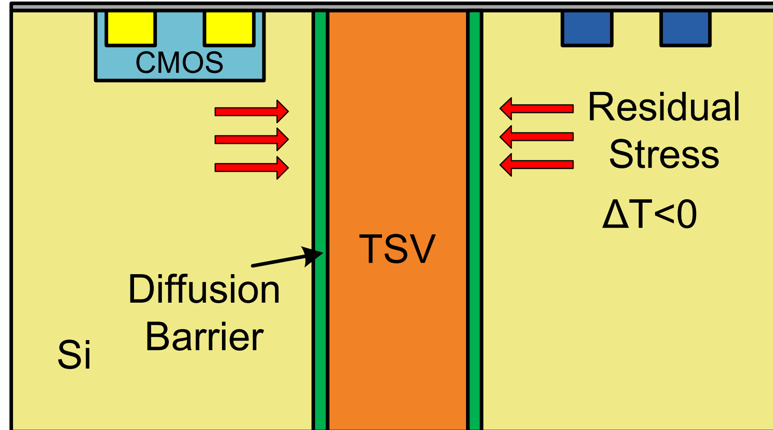


Figure 2.8: TSV CTE miss-match stress field

2.7 Reliability Issues

Most reliability concerns specific to 3D ICs are related to TSVs, which introduce several new failure modes. Many TSV reliability degradations are fundamentally caused by thermal and stress issues [17, 18, 63]. The thermal issue comes from the fact that the stacked structure increases the power density without providing a sufficient heat removal path (Section 2.6). The stress issue is due to significant differences in the coefficient of thermal expansion (CTE) between TSVs (*e.g.*, copper 17.7 MK^{-1}) and the silicon substrate (3.05 MK^{-1}). When TSVs are cooled down from high manufacturing temperature to room temperature, negative thermal load is applied creating compressive and tensile stress inside TSVs and neighboring substrate areas [44]. This phenomenon is illustrated in Figure 2.8. TSV stress not only affects reliability, but is also shown to influence transistor mobility and thus circuit performance [64].

TSV-induced reliability losses include: TSV electromigration [19,65,66], TSV stress migration [17, 18, 63, 67], TSV oxide breakdown [68], TSV thermal cycling [69–71] and TSV stress-induced material fracture [72–74]. TSV electromigration and stress-migration cause TSV's metal atoms to migrate, gradually altering material density and resistance, and eventually causing TSVs to form short or open-circuits. Electromigration moves atoms by transfer of momentum from flowing electrons, whereas stress-migration moves atoms along stress gradients. TSV oxide breakdown occurs when the electrical field inside the TSV barrier layer exceeds its threshold, destroying the electrical isolation between TSVs and the substrate. Thermal cycling shortens a TSV's lifetime by introducing TSV defects through thermal fatigue. Material fracture, initiated by manufacturing imperfections (*e.g.*, voids inside TSVs) and accelerated in high stress environments, may lead to delaminations or cracks around the TSV structure. All the above mentioned TSV failures are exacerbated at elevated temperature [63].

2.8 Micro-Fluidic Cooling

Micro-fluidic (MF) cooling is a promising technology for cooling ICs with high power flux. DARPA's Intra/Interchip Enhanced Cooling (ICECool) Program [75] has been investigating and prototyping such cooling systems for both high-flux 2D ICs (*e.g.*, high gain RF amplifier arrays) and 3D CPUs. By pumping coolant into the substrate of the chip, the resistive path through the oxide layers and chip package are short-circuited, providing significantly lowered transistor junction temperatures

[27, 59]. Moreover, MF cooling channels can be etched into the substrate of each layer in a 3D stack before bonding, providing equal cooling capacity to all layers and removing vertical thermal gradients [27, 60]. Finally, the high conductance of water coupled with the active heat movement due to fluid pumping velocity provide massively increased cooling capacity as compared to traditional air cooling [16].

Although general purpose CPUs have not generally required active cooling in the past, 3D stacking and the trapped heat effect will significantly increase thermal resistance. Enhanced cooling will be necessary to sustain the high power density of modern CPU architectures implemented in 3D IC technology [8]. Solutions such as DVFS have been proposed to control temperature in air cooled 3D CPUs, but at the expense of performance [14, 76].

A MF heatsink is created by fabricating microchannels in the silicon substrate of each layer in a 3D IC. A microchannel is a small channel (generally 10s to 100s of µmin dimension [77]) etched into the silicon substrate. These microchannels are created with the intention of pumping fluid through them in order to cool each layer of the chip [60]. The fluid enters the system at a low temperature and as it flows through each channel, heat is conducted through the silicon substrate into the fluid and then pumped out of the system. This concept is illustrated in Figure 2.9.

Micro-fluidic cooling comes with some overheads. One such overhead is the additional power required to pump the fluid. In previous work, methods for reducing pumping power have been investigated, such as nonuniform microchannel distribution [59] and dynamic control of fluid flow rate [78, 79]. The results of the studies presented in this dissertation [8, 27–29] show that the pumping power used

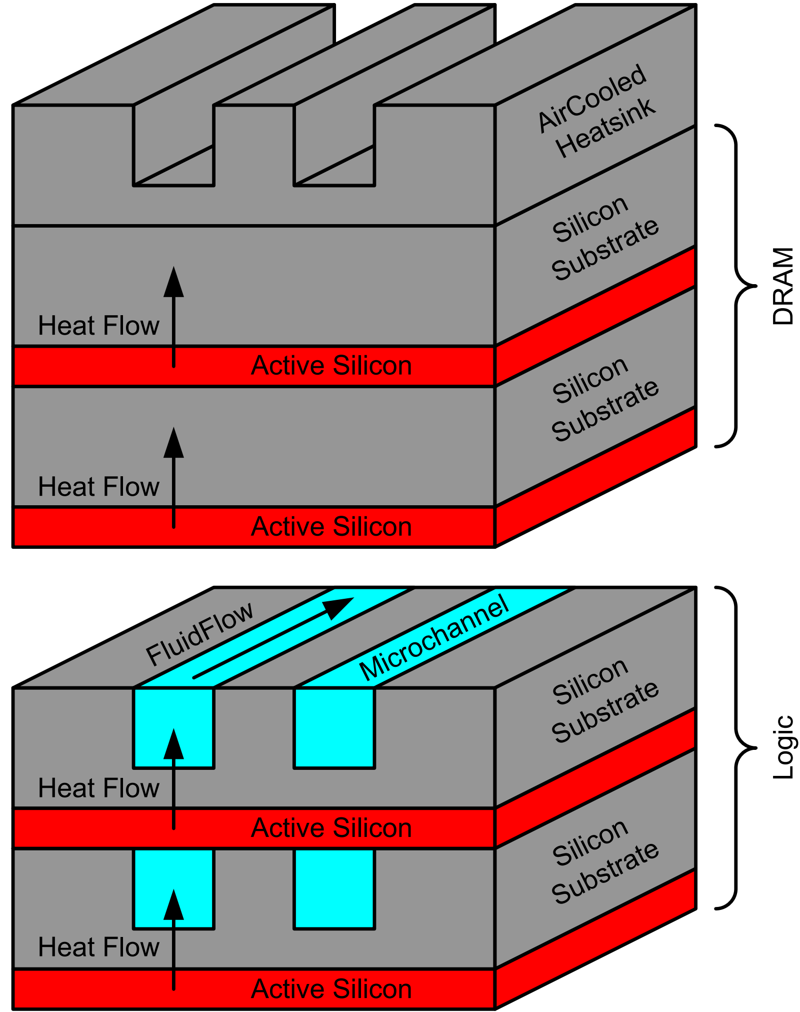


Figure 2.9: Micro-fluidic heatsink in memory-on-logic 3D CPU

to implement a MF heatsink is more than accounted for by the leakage power reduction that is a result of temperature reduction. Another overhead to MF cooling is that adding microchannels to a 3D IC requires a thicker substrate. This requires both the length and diameter of TSVs to increase in order to maintain a specific TSV aspect ratio defined by the manufacturing process, which increases the area overhead of TSVs. Typical 3D IC thinned silicon substrates have thickness in the 50um range while micro-channels would require thicker substrate (in the 150-200um range) [59]. TSVs and microchannels cannot coexist in the same space, so adding

micro-fluidic cooling to a design also constrains where TSVs can be placed, and the placement of microchannels and TSVs must be co-designed [30, 31, 80]. We investigate this trade-off between cooling capacity and vertical interconnect density (*i.e.* vertical signal bandwidth) in Section 5.2.

Chapter 3: 3D CPU Co-Simulation Co-Optimization Flow

3D integration technology brings the opportunity for new computer architectures, however such drastic changes to the conventional computing paradigm require new architectural models of 3D CPU performance, power, area and timing (PPAT). The 3D PPAT modeling challenges can be broadly broken down into the following categories.

- **Memory Hierarchy:** Stacked memory architectures have significantly different memory hierarchy topologies due to more fine grained integration with TSV technology. CPU-DRAM communication may take place over multiple independent communication channels which could be point-to-point, bus or a hybrid of both [27]. Each communication channel can be wider and/or clocked faster using high-density low-impedance on-chip interconnects. PPAT simulations must be configured to model the power and performance of such unconventional memory hierarchies. Moreover heterogeneous integration facilitates on-chip cache and/or main memory technologies such as DRAM, MRAM and

PRAM, all of which require complex memory controller designs [47]. Models of these technologies and their controllers are not included in most 2D PPAT simulation frameworks which assume on-chip SRAM and off-chip DRAM. Finally, due to drastically reduced parasitics, memory-on-chip integration could facilitate a reemergence of large parallel interfaces as opposed to high speed serial communication for low-power designs [38]. The whole spectrum of interface implementations must have available models within a 3D PPAT simulator for proper trade-off analysis.

- **Communication Networks:** Like the memory hierarchy, inter-core communication can leverage similar benefits from 3D integration. NOCs in 2D CPUs usually follow typical topologies such as 2D mesh and torus. However the expansion of cores into the third dimension in logic-on-logic architectures introduces new 3D NOC topologies. These 3D networks are more highly connected offering higher bandwidth and reduced logical distance between nodes (*i.e.* number of hops), but require more complex routers and thus dissipate more power and may introduce larger router delays. Additionally, the vertical distance between nodes is often much less (*e.g.*, 10x) than the horizontal distance. Asymmetric NOC topologies with larger router radix in the vertical direction can take advantage of this physical asymmetry (*e.g.*, 3D supermesh [27]). Thus a 3D PPAT simulator must have the capability of simulating customized asymmetrical NOCs and the associated physical implementations of the routers and drivers.

- **Fine Grained Integration:** One of the main advantages of 3D integration is the reduction to wire length due to fine grained integration. The reduction in length to the longest wires in a large circuit (*e.g.*, a CPU function block) can approach $\sqrt{n}$ where n is the number of layers across which the circuit is split [34]. Power, delay and area estimates for circuits with regular structure (*e.g.*, memory elements) can be estimated analytically using technology and topology parameters (although 3D implementation significantly increases the design space of the topology parameters to be considered [81]). However, highly complex and customized circuits (*e.g.*, ALU) are hard to estimate analytically. For 2D CPU analysis, empirical models have been fit to real CPU circuits in the market [2]. Since 3D CPUs are still in the research and development stage, similar data does not exist. Developing models for 3D function unit PPAT is a challenging and open problem.

The simulation flow used to evaluate the 3D CPU design space explored in the following chapters is shown in Figure 3.1. We provide a detailed description of each step in the simulation flow in the following sections.

3.1 Architectural Design Space

The studies presented in Chapters 4 and 5 involve exhaustive simulation across a set of computer architectural variables. Table 3.1 enumerates the fixed architectural parameters across all studies. The three study variables (number of cores, CPU clock rate and number of memory controllers) take on different ranges in different

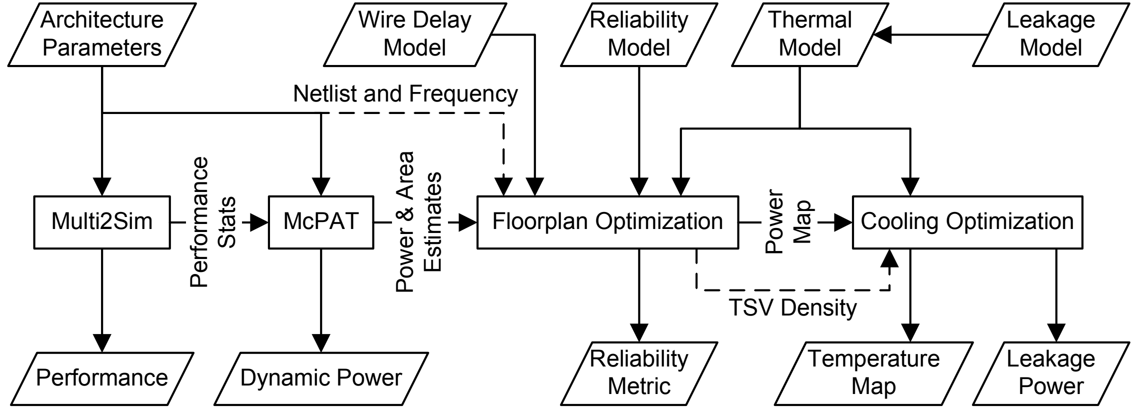


Figure 3.1: Simulation flow

studies, and are thus enumerated in their respective sections. In these chapters we maintain a relatively small architectural design space to accommodate exhaustive simulation. However, in Chapter 6 we expand the scope and dimensionality of our architectural design space and apply modeling techniques to feasibly estimate the metrics of interest across a large combinational space of architectural variables.

3.2 Performance Simulation

Performance simulation is performed by Multi2Sim (M2S) [82], a cycle accurate CPU simulator. Architectural parameters are passed to the simulator through configuration files that include number of cores, number of function units within cores, pipeline width, buffer/queue/register size, cache size/associativity/latency, network-on-chip (NOC) topology/latency, branch predictor size and type *etc.* Cache and register (*e.g.*, register file, register alias table (RAT) and branch target buffer) latencies are determined using CACTI [81, 83] to provide realistic architectural setups to the simulator. DRAM latency is calculated as explained in Section 3.3 and

Table 3.1: Architectural parameters

Parameter	Value
Cores	**See Study Details**
Clock Rate	**See Study Details**
Memory Controllers	**See Study Details**
Technology	45 nm
Branch Predictor	4k Entry 2-Level
Issue	Out of Order
Reorder Buffer	64 entries
Fetch/Dec/Issue Width	4
Functional Units	4 IALU, 1 IMult, 2 FPALU, 1 FPMult
Physical RF	80 Int, 40 FP
BTB Size	1024 entries
Return Addr. Stack	32 entries
Load/Store Queue	20 entries
Private L1 I/D Cache	256 Sets per Core, 2-Way, 64B Block (32 kB per Core) @ 2 cycle
Shared L2 Cache	512 Sets per Core, 16-Way, 64B Block (512 kB per Core) @ 7 cycles
NOC type	3D Super-Mesh
NOC link latency	3 cycles
DRAM bus width	64 B
DRAM bus speed	Core clock rate
DRAM capacity	1 GB/layer $\times$ 4 layers = 4 GB

NOC topology/latency is calculated as explained in Section 3.9. M2S simulates the execution of an x86 binary file on the described CPU. The simulator outputs a list of performance statistics such as IPC, memory reads, writes, hits and misses, branch prediction rate, number of instructions that access each type of execution unit, reads and writes to buffers, queues and RAT *etc.*

3.2.1 Benchmarks

The studies presented in the subsequent chapters evaluate an architectural-physical design space across a suite of benchmark workloads. All benchmarks used in our work come from the SPLASH-2 [84] and PARSEC [85] benchmark suites. These benchmarks are standard for evaluating the results of architectural research on CMPs [14, 86–90].

3.3 DRAM Latency Model

Although DRAM latency depends on many transient factors, many performance simulators, including M2S, simply model memory latency as a constant average value. We propose a model for the average memory latency time, comprised of five different steps in the DRAM access procedure, starting at the time a last level cache (L2 cache in this work) miss is detected. We estimate the average duration of each step as a function of the architectural parameters. The five steps are as follows: (1) MC Queuing Delay, (2) Memory Address Translation, (3) Address Transfer Delay, (4) DRAM Core Access (5) Data Transfer Delay. Step (1) is the only step that is a strong function of the architectural variables considered in these studies. Steps (2) through (5) are modeled as a constant delay of 5 cycles [91], 1 DRAM bus cycle [57], 32 ns [9] and b DRAM bus cycles [57] respectively, where b is the cache line width divided by the DRAM bus width. DRAM bus width and frequency are given in Table 3.2.

Table 3.2: 2D vs. 3D DRAM Bus

Integration	Bus Width	Bus Frequency
2D Off-Chip DRAM	64 bits	200 MHz
3D Stacked DRAM	512 bits	Core Frequency

3.3.1 MC Queuing Delay

The memory controller queuing delay represents the amount of time a memory request spends waiting in the memory controller queue. This value depends on the number of memory controllers (*i.e.* consumers of memory requests) and the number of cores (*i.e.* producers of memory requests). The work by Awasthi *et al.* [86] reports that the increase in queuing delay from a single core to a 16 core processors is about 8x. Dong *et al.* [91] reported that a configuration with 4 cores and one MC has a queuing latency of 116 cycles. We linearly extrapolate these two observations to model queuing delay as a function of $\#core$, and assume that memory requests are uniformly distributed across the address space[1], such that queuing delay is inversely proportional to the number of MCs. Thus we model MC queuing delay T_Q with Equation (3.1).

$$T_Q = \frac{388 \text{ ns}}{\#MC} \times \left[1 + \left(\#core \times \frac{1 - 1/8}{16 - 1}\right) - \left(16 \times \frac{1 - 1/8}{16 - 1}\right)\right] \tag{3.1}$$

3.3.1.1 Derivation

We can solve $T_Q(\#core) = T_Q(d) + m(\#core - d)$ as a linear function of $\#core$ using the following two observations:

[1] This assumption was validated in [14].

1. $T_Q(4) = 116$ ns

2. $\frac{T_Q(16)}{T_Q(1)} = 8$

Observation 2 can be rearranged as $T_Q(1) = \frac{1}{8}T_Q(16)$. Thus $m = \frac{T_Q(16)-T_Q(1)}{16-1} = T_Q(16)\frac{1-\frac{1}{8}}{16-1}$. Allowing $d = 16$ we can write $T_Q(\#core) = T_Q(16)+T_Q(16)\frac{1-\frac{1}{8}}{16-1}(\#core-16) = T_Q(16)[1 + \frac{1-\frac{1}{8}}{16-1}(\#core - 16)]$.

All that is left is to solve for $T_Q(16)$ by solving $m = \frac{T_Q(4)-T_Q(1)}{4-1} = \frac{T_Q(16)-T_Q(4)}{16-4}$. Substituting Observation 1 ($T_Q(4) = 116$ ns) and rearranged Observation 2 ($T_Q(1) = \frac{1}{8}T_Q(16)$) yields $m = \frac{116\ \text{ns}-\frac{1}{8}T_Q(16)}{4-1} = \frac{T_Q(16)-116\ \text{ns}}{16-4}$ which when solved yields $T_Q(16) = 388$ ns.

3.4 Power/Area Estimation

Dynamic and leakage power are estimated along with the total area of each CPU component by McPAT [2], a power and area estimation tool commonly used in computer architecture research [14,92–95]. The architectural parameters are used to estimate the leakage power at nominal temperature using internal transistor-level models of CPU components. Likewise these transistor models also estimate the energy-per-access (*e.g.*, read, write or decode) and total area of each component. The combination of access counts from Multi2Sim and energy-per-access estimates from McPAT yield dynamic power. Dynamic and leakage power estimates are applied to an optimized floorplan topology to generate a power density map. The power density map is consumed by the thermal model, which internally applies thermal-leakage scaling (Section 3.8.1).

Transistor level power and area models of regular structures such as caches, registers etc. are provided internally through CACTI [83]. Power and area models of complex combinational logic such as ALUs and decoders are generated by applying curve fitting to empirical data collected from real CPUs. Cacti has been expanded to estimate 3D memory implementations [81], but development fine-grain 3D combinational logic blocks is an area of future work, and in this dissertation 2D function blocks are used[2].

3.4.1 Pumping Power

The micro-fluidic heatsink's simulated for this work consist of straight microchannels with non-uniform spacing between channels. The minimum pitch between channels is double the channel width W, however many channels are spaced considerably farther apart than the minimum pitch. The power required to pump fluid through a microchannels, P_{pump} is defined in Equations (3.2) through (3.6) [59], where N is the number of microchannels, f is the fluid flow rate, Δp is the pressure drop across each microchannel, γ is a function of microchannel aspect ratio ($AR = W/H$), μ is the viscosity of fluid flow, L is the length of the channel, v is the fluid velocity, D_h is the hydraulic diameter of the channel, W is the width and H is the height of the microchannel. Specific values used in the work reported here are given in Table 5.2.

[2]We do allow the memory controller and execution unit to be split across two layers at subcomponent boundaries (*e.g.*, FPU-IFU boundary in execution unit or Front-end-back-end boundary of the memory controller [2]). The effects on power and area of such a coarse-grained split are assumed to be negligible.

Table 3.3: Micro-fluidic system parameters

Var	Value	Name	Var	Value	Name
W	100 µm	Width	μ	653 µPa s	Viscosity
H	200 µm	Height	P_{pump}	2 mW per layer	Pumping Power

In our study we assume a constant pumping power P_{pump}. Thus a reduction in the number of channels N results in increased pressure drop and fluid velocity in the remaining channels, which increases the local heat transfer coefficient of each channel [96]. Our heatsink optimization scheme (Section 3.10) finds the optimal trade-off between number (and location) of channels vs. heat transfer coefficient of each channel. The pumping power used to provide micro-fluidic cooling in our studies is more than made up for by reductions in thermally induced leakage power due to reduced chip temperatures [27–29].

$$P_{pump} = N f \Delta p \tag{3.2}$$

$$f = WHv \tag{3.3}$$

$$\Delta p = 2\gamma\mu L v D_h^{-2} \tag{3.4}$$

$$\gamma = 4.7 + 19.64 \times \frac{(AR^2 + 1)}{(AR + 1)^2} \tag{3.5}$$

$$D_h = \frac{2WH}{W + H} \tag{3.6}$$

3.5 Core Netlist

Each CPU core consists of a set of interconnected components as shown in Figure 3.2. The bit width of each connection in the netlist is annotated in the figure, and the associated utilization of each net is calculated from the Multi2Sim

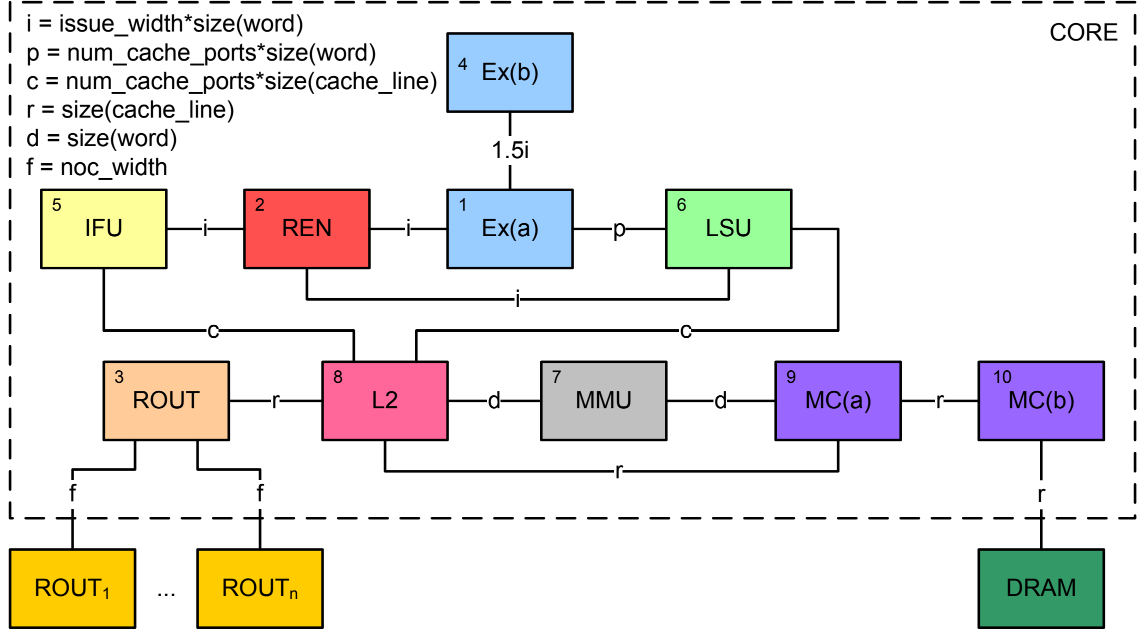


Figure 3.2: CPU core component netlist with net widths notated.

performance statistics (Section 3.2). Details of each CPU component are given in Table 3.4. The execution unit and memory controller are large components, and are allowed to be pipelined and/or split into two sub-components which can be placed on separate layers of the 3D stack (multi-layer)[2]. The instruction fetch unit (IFU) contains the branch predictor and the instruction cache. The execution unit contains integer and floating point function units along with the register file and the reorder buffer. The load store unit (LSU) contains the load store queues and the data cache and the memory management unit (MMU) contains the translation look-aside buffers (TLBs). Core routers are connected in a 3D super-mesh topology (Section 2.5). More detailed descriptions of each CPU component can be found in [2].

Table 3.4: CPU core component properties

	Name	Comments
IFU	Instruction Fetch Unit	
REN	Rename Unit	
EX	Execution Unit	Multi-layer
LSU	Load Store Unit	
ROUT	Router	Inter-core
L2	L2 Cache	Shared
MMU	Memory Mgmt. Unit	
MC	Memory Controller	Multi-layer, Inter-core, Shared

As shown in the figure, the router and the memory controller are the only components that communicate outside of the core (inter-core), either with other cores or with the DRAM. The L2 cache and memory controller components are slices of a larger component that services multiple cores (shared). The L2 cache is a single shared cache with a local slice associated with each core, whereas each memory controller can service two, four, or eight L2 cache slices, depending on the total number of memory controllers. Using the wire delay model (Section 3.6), we calculate the maximum allowed center-to-center distance between each connected component for the target clock frequency to prevent timing violations. These distance constraints are used to create a timing-feasible floorplan (Section 3.9).

3.6 Wire Delay Model

We calculate the wire delay per unit length using Equation (3.7) from [35]. The variables $a = 0.4$ and $b = 0.7$ are fitting parameters taken from [35], and the variables r, c, r_0, c_0 and c_p are respectively the wire resistance per unit length, wire capacitance per unit length, output resistance of a minimum-size inverter, in-

put capacitance of a minimum-size inverter and parasitic output capacitance of a minimum-sized inverter. These values were extracted from the McPAT source code and are given in Table 3.5. Given these parameters the delay per unit length calculated by Equation (3.7) is 81 ps mm^{-1}. The wire delay model is used to insure timing feasibility during floorplan creation (Section 3.9).

$$\frac{d}{l} = 2\sqrt{rcr_0c_0}\left(b + \sqrt{ab(1 + \frac{c_p}{c_0})}\right) \tag{3.7}$$

Table 3.5: Transistor and interconnect parameters for 45 nm technology [2]

variable	value	variable	value
r	0.36 Ω μm^{-1}	c	0.28 fF μm^{-1}
r_0	10.9 kΩ	c_0	0.85 fF
c_p	0.31 fF		

3.7 Reliability Model

Our reliability model focuses on TSV electromigration (EM), one of the 3D CPU's critical failure modes [18, 19, 63, 65–67, 69, 97]. As more power dissipating device layers are stacked vertically, power flux increases dramatically. However, 3D power delivery network (PDN) is limited by the number of power pins (*i.e.* C4 bumps) which is a function of the footprint area of the chip, and does not increase as more layers are stacked [25, 26]. This leads to a significant increase in PDN's current density in 3D CPUs. Furthermore, the stacking structure generates thermal hotspot in areas of high power (and current) density [59]. The increases in both current density and temperature accelerate TSV EM. In addition, the immature TSV

fabrication process induces structural defects such as voids inside TSVs [97], which also degrade TSV's EM reliability. As TSVs consume many placement/routing resources, it is hard to make post-layout EM fixes (*i.e.* redundant wires/vias) without significant area overhead and redesign effort [18, 30, 31, 63, 98].

In the proposed reliability model each TSV's EM lifetime is considered as a random variable, where the randomness is caused by TSV manufacturing [99]. We model each TSV's failure probability density function (PDF) using a Weibull distribution. Each Weibull distribution is determined by a shape parameter k and a scale parameter λ. We assume that TSV EM failure rate is constant over time (therefore $k = 1$). The scale parameter λ, is determined by TSV's mean-time-to-failure (MTTF). Specifically, λ is calculated based on classic Black's equation [100] as shown in Equation (3.8).

$$\lambda = MTTF_{EM} \propto (J_{avg})^{-2} e^{\frac{E_a}{k_b T}}. \tag{3.8}$$

J_{avg} is the average DC current density, E_a is activation energy, k_b is Boltzmann's constant, and T is absolute temperature in degrees Kelvin. In cases where AC signal is concerned, J_{avg} is its equivalent DC current density [101]. Higher current density and temperature shorten the expected EM lifetime of TSVs, according to Equation (3.8).

For reliability estimation, each TSV must be assigned a point in space at which to measure the temperature. Signal TSVs within a 3D net are uniformly distributed inside its feasible region. A 3D net's feasible region is determined such that the in-

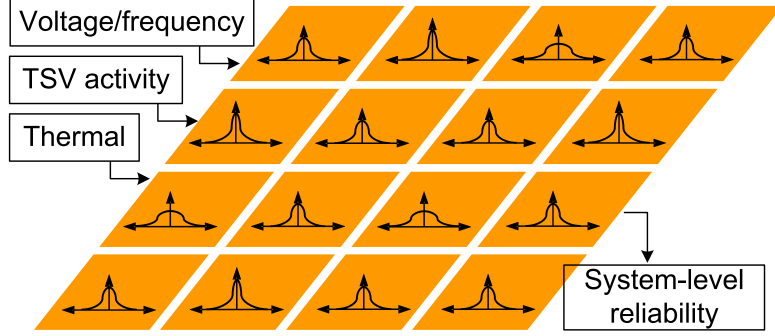


Figure 3.3: TSV EM reliability model

terconnect timing constraint between the connecting blocks is not violated using the 3D net wirelength model from [21]. Figure 3.3 illustrates our system-level EM reliability modeling approach. Based on typical 3D-CPU applications, TSV activities (messaging between logic blocks and/or memory blocks) can be acquired from performance simulation (Section 3.2). Combined with voltage/frequency information, the TSV activities are translated into transient currents by modeling the capacitive load's charging/discharging behavior. The transient current is subsequently converted to its equivalent DC current density distribution [101]. This DC current density distribution and the thermal profile define a failure PDF for each TSV.

System's EM reliability (R_{EM}) is defined as the probability that none of the TSVs fail before the target lifetime of has elapsed. R_{EM} can be expressed using Equation (3.9), where P_{EM} is the probability that a 3D-CPU fails before target lifetime, and P^{i}_{EM} is the probability of the i$^{\text{th}}$ TSV fails before target lifetime.

$$R_{EM} = 1 - P_{EM} = 1 - \prod_{i \in TSV} (1 - P^{i}_{EM}). \tag{3.9}$$

3.8 Thermal Model

Once the chip floorplan has been constructed (Section 3.9) and component power estimation is complete (Section 3.4), we have a power density map for each tier of the 3D stack. Power density maps are converted into thermal maps using our compact thermal model [59]. A 3D grid is constructed representing the physical structure of the 3D IC. Each tier in the chip stack is divided into sub-layers: silicon substrate (with or without microchannels), active silicon, interconnect and passivation. Likewise the power map is discretized into a 3D grid and the total power of each power grid is assigned to the respective physical grid in the active silicon sub-layer (all other sub-layers have zero power).

Then each physical grid is converted to an electrical circuit representation as shown in Figure 3.4. Power is modeled as a current source and thermal resistance is modeled as electrical resistance. The voltage at the center of each circuit grid represents the temperature of the respective physical grid. This technique takes advantage of the thermal-electrical duality, similar to HotSpot [102]. Thermal resistances are evaluated based on material properties and dimensions of the respective physical grid using the technique in [59]. Material properties and dimensions of different sub-layers are listed in Table 3.6. When modeling a MF heatsink, the circuit model contains both solid and fluid grids. The resistance of a fluid grid depends on material properties and fluid flow rate [96].

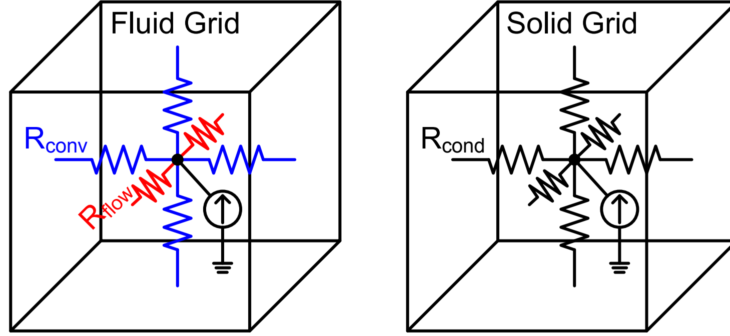


Figure 3.4: Thermal resistance grids for fluid and solid materials

Table 3.6: Thermal model material properties

Sub-Layer	Thickness (μ m)	Material	Conductivity (W m^{-1} K^{-1})
Top Substrate	995	Si	148
Microchannel Substrate	200	Si	148
Microchannel Fluid	200	H_2O	0.58
Thinned Substrate	55	Si	148
Active Silicon	5	Si	148
Interconnect	15	SiO_2+Cu	2.25
Passivation	15	SiO_2	1.4

3.8.1 Leakage Model

McPAT reports a base leakage value for each CPU component which is estimated at a fixed temperature T_0. To obtain more accurate leakage power estimates, which take into account leakage power's strong dependence on temperature, we iteratively solve our thermal model and then scale leakage estimates at each grid based on the estimated temperature of that grid after the previous iteration. We repeat this process until the change in temperature between two iterations is less than some threshold (*e.g.*, 1 °C). The thermal leakage scaling model is extracted from McPAT source code [2] (Figure 3.5).

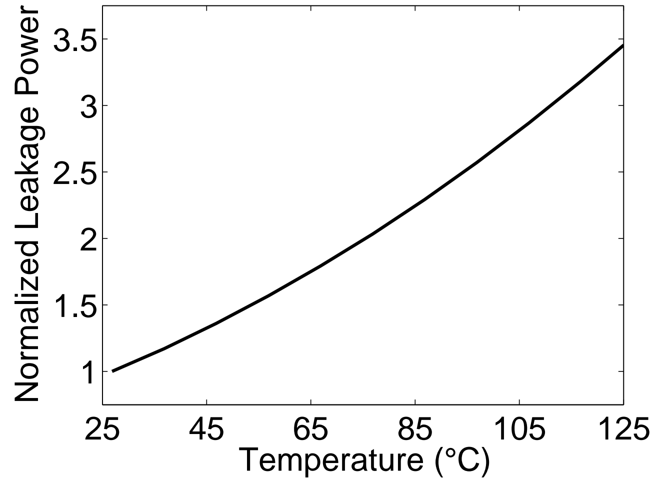


Figure 3.5: Thermal-leakage relationship

3.9 Floorplan Optimization

For each architectural configuration, we run a thermal-reliability aware floorplanner to create an optimized CPU floorplan for that architecture[3]. Floorplans are optimized iteratively using feedback from the thermal (Section 3.8) and reliability (Section 3.7) models while estimating timing feasibility using the netlist (Figure 3.2) and wire delay model (Section 3.6). A fundamental trade-off exists between timing, reliability and temperature. Placing high power components closer together can reduce wire delay and negative slack, but will increase hot-spot temperatures [27]. Likewise, splitting components across layers can reduce power density and thus

[3]Some of the studies here disable floorplan optimization and use a fixed topology, while others use modified objective functions. The algorithm presented here is the fully comprehensive method proposed in this dissertation at large, while other versions are considered for comparison and sensitivity analysis.

remove hotspots, but introduces additional TSVs which increase probability of failure [103]. Thus the timing, reliability and thermal profile must be simultaneously co-optimized during floorplanning.

The power dissipation and net activity of each component is averaged across all benchmark workloads when evaluating the thermal and reliability profile for floorplan optimization. The area of each component is given by McPAT (Section 3.4) and each component is assumed to be laid out as a rectangle. Net activities are derived from Multi2Sim (Section 3.2) and net widths are annotated in Figure 3.2.

Our approach optimizes the floorplan of a single CPU core, and then tiles that single-core floorplan in order to generate a chip level floorplan with the correct number of cores. Floorplan optimization at chip-scale would have been computationally infeasible, so the problem is reduced to floorplan optimization of a single core. However the thermal effects of core tiling and stacking are captured in the embedded thermal and reliability models. Cores are allowed (but not required) to be distributed across multiple layers.

Thermally aware floorplan optimization reduces peak temperature by optimizing the vertical and planar power density to reduce hot-spots, as well as moving high power components closer to the fluid inlets where maximum cooling potential exists [27]. However, timing violations are modeled (Section 3.6) throughout the optimization flow, and only timing feasible floorplans are accepted. Reliability aware floorplan optimization improves MTTF by preventing high activity nets to span across layers, and by minimizing the number of TSVs in general [103].

3.9.1 Floorplan Representation

We use transitive closure graphs (TCGs) [104] to represent the physical relationship between CPU components on each logic layer. A 3D floorplan can be represented as a set of n TCGs, where n is the number of layers in the 3D stack. We call such a set a 3DTCG. A simulated annealing approach is used to search the solution space of 3DTCGs, and a nested simulated annealing loop is used to optimize the component aspect ratios (AR) for each 3DTCG considered.

Given a 3DTCG with the area and AR of each component, a unique 3D floorplan is constructed. Then the chip area, thermal profile, MTTF and netlist wirelengths of that floorplan are evaluated. The objective of the floorplanning algorithm is to find an optimized floorplan for each architecture which minimizes area, peak temperature, and negative slack and maximizes lifetime. It may be hard or even impossible to find a floorplan that meets both thermal, reliability and timing constraints when considering an aggressive 3D CPU architectural design. High quality physical design optimization of the floorplan can significantly increase the feasibility region of an evaluated architectural design space, which will ultimately result in the selection of more optimal design points [1, 103].

3.9.2 Simulated Annealing Approach

Simulated annealing is used to search the solution space of 3DTCG topologies and CPU component aspect ratios. The annealing operations used for the simulated annealing of the 3DTCG are the original four intra-layer annealing operations from [104] (rotate, swap, move and reverse), plus the inter-layer swap from [105] and the inter-layer move from [106] (referred to as "Change Layer" in that paper).

The objective function used for simulated annealing of the 3DTCGs is given in Equation (3.10), where A is the total area of the core (Section 3.4), S is the total negative slack, T is the maximum temperature from the thermal model (Section 3.8) and R is the reliability metric (Section 3.7). The negative slack on each net is the wire delay (Section 3.6) on that net minus one cycle delay. Wirelength between two components is measured as the Manhattan distance between the center point of each component.

$$OBJ = c_1A + c_2S + c_3T - c_4R \tag{3.10}$$

The nested simulated annealing loop for determining aspect ratio of each component chooses a random component and scales its AR by a value randomly chosen from a normal distribution with $\mu = 1$ and $\sigma = 0.1$. Aspect ratio for each component is constrained by the equation $\frac{1}{5} < AR < 5$. The objective function used for the aspect ratio simulated annealing is $OBJ = c_1A + c_2S$.

3.9.3 Speeding Up Simulation Time

Because a temperature profile is required to evaluate the objective function at each iteration of the 3DTCG simulated annealing algorithm, the thermal model must be evaluated many times. The full chip-scale thermal model would be too time consuming to evaluate on each iteration, so instead we evaluate the thermal profile of a $2 \times 2 \times k$ core tiling and use this as an indicator of the true chip-scale temperature profile,where k is the number of core layers. This approach can make thermal simulation up to 30-50x faster than the evaluation of the full chip-scale model while still modeling the thermal effects of core stacking and the junction where cores abut in the horizontal direction. The correlation coefficient between the maximum temperature observed by chip-scale vs. reduced model is 80%. Thus thermal simulation of a reduced core tiling is a practical and accurate way of approximating temperature in the thermally aware floorplanning algorithm.

Likewise the reliability model is applied to the same $2 \times 2 \times k$ tiling of the floorplan. The thermal and reliability estimates of this reduced tiling do not provide reliable estimates of absolute temperature and lifetime, but do provide accurate estimates of the relative ordering between floorplan candidates, making this technique suitable for unconstrained optimization.

Removing thermal and reliability terms from the objective function and reformulating them as constraints would invalidate the proposed simulation speed up technique, and significantly increase the optimization runtime. However this would

remove the need for proper choices of weighting factors to drive the trade-off between conflicting optimization terms. The comparison and trade-offs of these two schemes is left to future work.

3.9.4 Core Tiling and NOC Design

To generate the final chip floorplan, the core floorplan is replicated on an $i \times j \times k$ grid such that $ijk = n$ where n is the total number of cores. The dimensions of a single core floorplan are defined as $width_{core}$ and $height_{core}$ respectively (determined by single-core floorplan optimization). The values i, j and k are chosen such that:

- Total area per layer ($iwidth_{core}jheight_{core}$) is less than $A_{max} = 400$ m^{2}m.
- Total number of layers is minimized.
- Layer aspect ratio ($^{iwidth_{core}}/_{jheight_{core}}$) is close to unity.

NOC topology is defined as an $i \times j \times k$ 3D super-mesh [7] (Section 2.5) and NOC latency is defined as the wire delay of length $\max(width_{core},\ height_{core})$ (Section 3.6). NOC topology and latency are fed back into the performance simulator to get accurate inter-core communication simulations [4].

[4]Floorplan and NOC design are required to define NOC parameters for performance simulation. McPAT is run once to generate area estimates before performance simulation, and then again to generate power estimates after performance simulation. The initial area estimations are enough to generate an estimate of NOC latency, assuming a perfectly square core floorplan with no white-space.

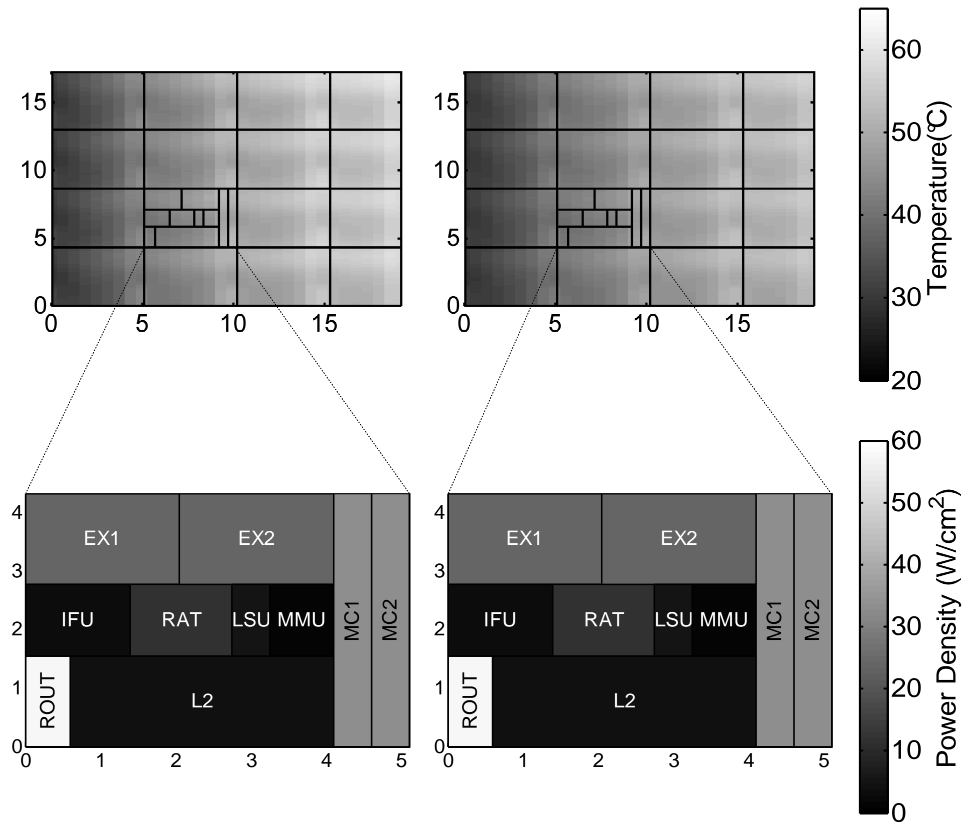


Figure 3.6: Example thermally unaware floorplan with MF cooling

3.9.5 Example

Figures 3.6 and 3.7 illustrate an example floorplan result[5] with and without thermal awareness, and the resulting thermal and power maps. This example is from a 32-core 16 MC 3D CPU running `ocean` at 2.4 GHz with micro-fluidic cooling. We see that thermally unaware floorplanning results in less total chip area and a more square chip outline, however this floorplan has significantly higher temperatures.

[5]Dimensions shown in mm.

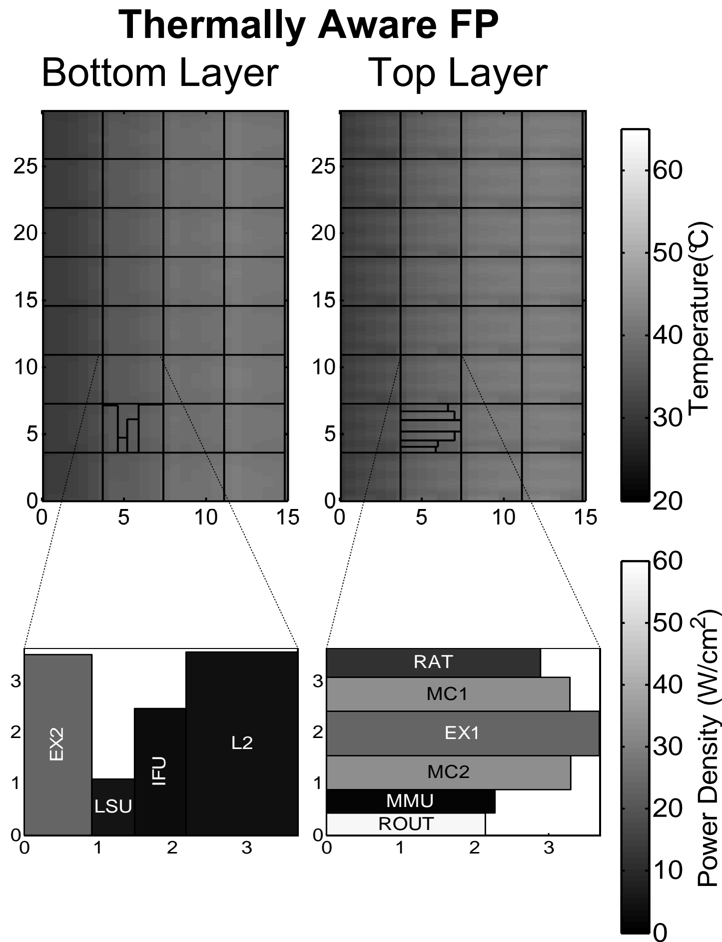


Figure 3.7: Example thermally aware floorplan with MF cooling

Note that fluid flow direction in this figure is from left to right and the pumping power is fixed. The thermally aware floorplan is able to improve chip temperature using a number of techniques.

First, shifting the chip dimensions towards a more tall and narrow chip outline allows for the fabrication of more microchannels and reduces the length of each channel, which significantly increases the cooling capacity of the micro-fluidic heatsink by reducing the thermal wake effect [107]. Second, the function unit with the highest power density (ROUT) is surrounded by low power units or dead-space on all sides, allowing for more lateral heat spreading and reducing hotspot temperatures.

In the thermally unaware floorplan, the router in one core abuts the MC in the neighboring core, leading to hotspots. More importantly, the thermally aware floorplan splits cores across two layers, preventing vertical stacking of hotspots. In the fixed floorplan routers are stacked vertically, leading to significant hotspot heating. Finally, compared to the thermally unaware floorplan, the thermally aware floorplan allocates more total power to the top layer and less to the bottom layer. This is due to the significantly larger thermal resistance between the ambient temperature (at the top of the chip stack) and the bottom layer, as compared to the top layer[6].

3.10 Cooling Optimization

The final step in our analysis approach for DSE of 3D CPUs with micro-fluidic heatsinks is to consider optimized non-uniform heatsink designs. Due to the non-uniform nature of the generated power map after floorplan optimization, the optimal microchannel distribution in the micro-fluidic heatsinks is also non-uniform when subjected to a constant pumping power. Simply placing microchannels uniformly at minimum pitch (the default heatsink design in this work) is inefficient as cooling potential is distributed to hot-spots and cold-spots equally. In addition to the nonuniform power density profile on each layer, one must also consider the nonuniform thermal resistance between each layer and the ambient, due to inter-layer resistances. Thus microchannels are more valuable when placed between layers that are far from the top (ambient interface) of the chip, where thermal resistance is high.

[6]The bottom and sides of the chip stack are adiabatic.

Like floorplan optimization, heatsink optimization is performed for each architectural configuration, and is optimized using a simulated annealing algorithm with feedback from the thermal model. The chip-scale power map consists of a tiling of single-core power maps. We take advantage of this by optimizing the heatsink configuration for a single core stack and then tile the optimized microchannel configuration for the final solution. A core stack is a single core that is tiled in the vertical direction as many times as it would be in the true chip-scale layout (*i.e.* k times). In other words, the microchannel placement on different layers of the stack can be different, but in the planar direction it is tiled. Tiling of microchannels in the vertical direction is inefficient because of the strong dependence of thermal resistance on layer depth. As in floorplan optimization, thermal evaluation of heatsink design points is carried out on a 2x2xk tiling of cores such that the thermal interface between adjacent cores is molded accurately, while speeding up simulation time.

3.10.1 Microchannel Placement Representation

Microchannels are assumed to be straight channels of constant width which extend along the entire length of the chip from inlet to outlet. Thus, channel placement can be represented as a two-dimensional placement problem, the two dimensions being vertical (*i.e.* in the direction of layer stacking) and horizontal (perpendicular to the direction of flow). We represent the placement of channels as a binary matrix $\mathbf{B}$, which has k rows and $W_{chip}/\Delta x$ columns, where W_{chip} is the width of the chip perpendicular to the direction of flow, and Δx is the width of a grid in

the thermal model (Section 3.8). In our thermal model it is assumed that $\Delta x = W$, where W is the width of a microchannel. If $b_{y,x} = 1$, then grid x on layer y contains a microchannel, and if $b_{y,x} = 0$, it does not. All channels must be separated by at least one non-channel grid (*i.e.* channel wall must have nonzero width). Thus if $b_{y,x} = 1$, then $b_{y,x+1} = b_{y,x-1} = 0$.

3.10.2 Simulated Annealing Approach

Simulated annealing is used to explore the solution space of matrix $\mathbf{B}$. Two annealing operations can be applied to $\mathbf{B}$ during simulated annealing optimization: add or remove a channel. The initial solution is uniform channels with minimum pitch. All entries in $\mathbf{B}$ which are candidates for channel insertion or removal are identified. If a channel is being added, a random candidate is chosen and the solution is updated. If a channel is being removed, a ranking is imposed on existing channels using our microchannel cost model (Section 3.10.4), and a candidate is selected from the bottom q^{th} percentile. In these studies we set $q = 25\%$. The objective function used to evaluate annealing moves is $OBJ = T$, where T is the maximum temperature from the thermal model (Section 3.8).

3.10.3 Example

Figures 3.8 through 3.10 exemplify how micro-channel placement optimization can reduce on chip temperatures for a given floorplan and a fixed pumping power. Figure 3.8 shows the power density and associated temperature maps of 32-core

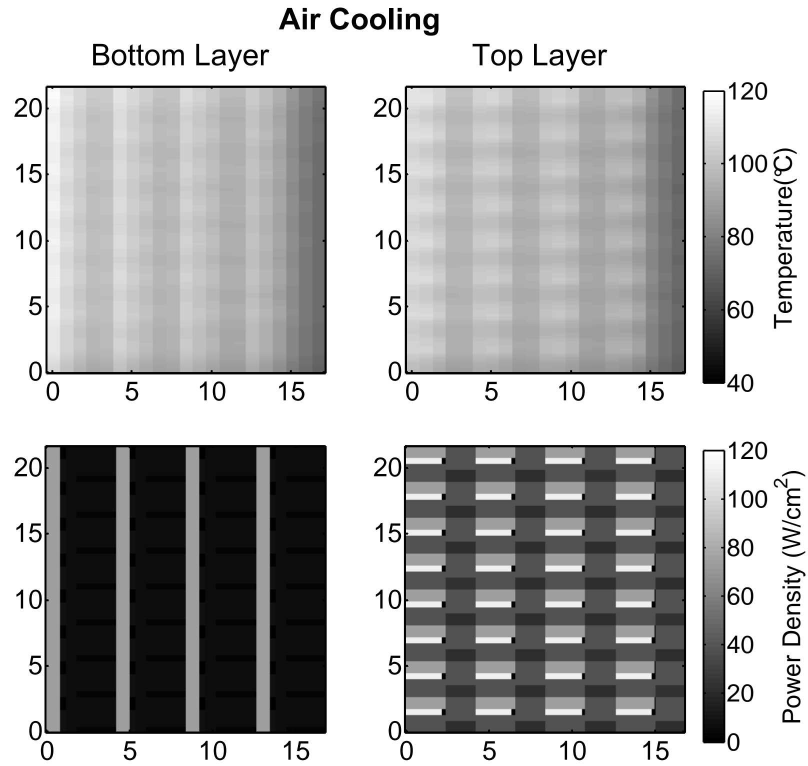


Figure 3.8: Temperature and power density of air cooled floorplan

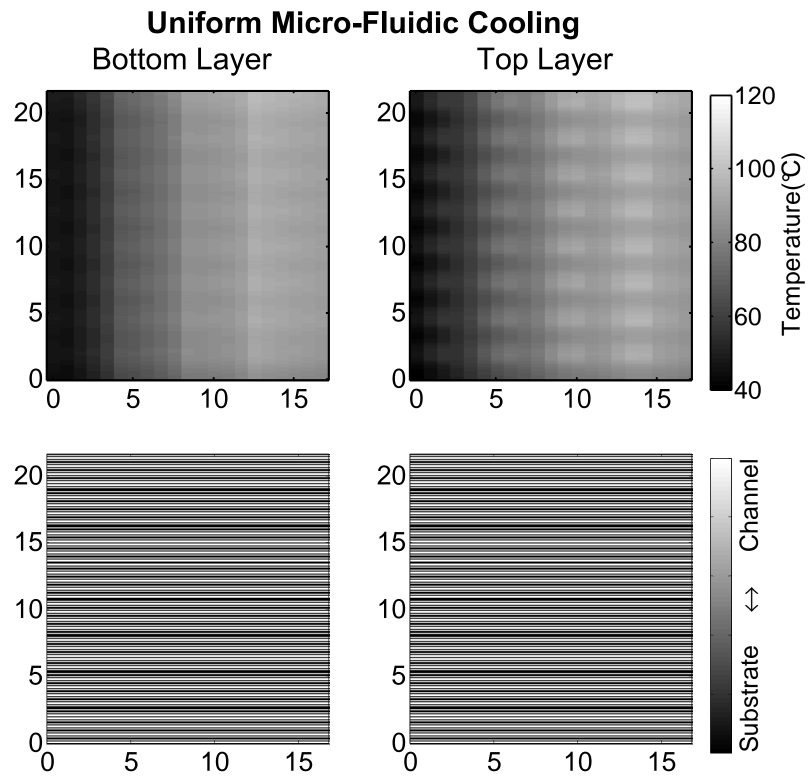


Figure 3.9: Temperature and channel distribution using uniform MF heatsink.

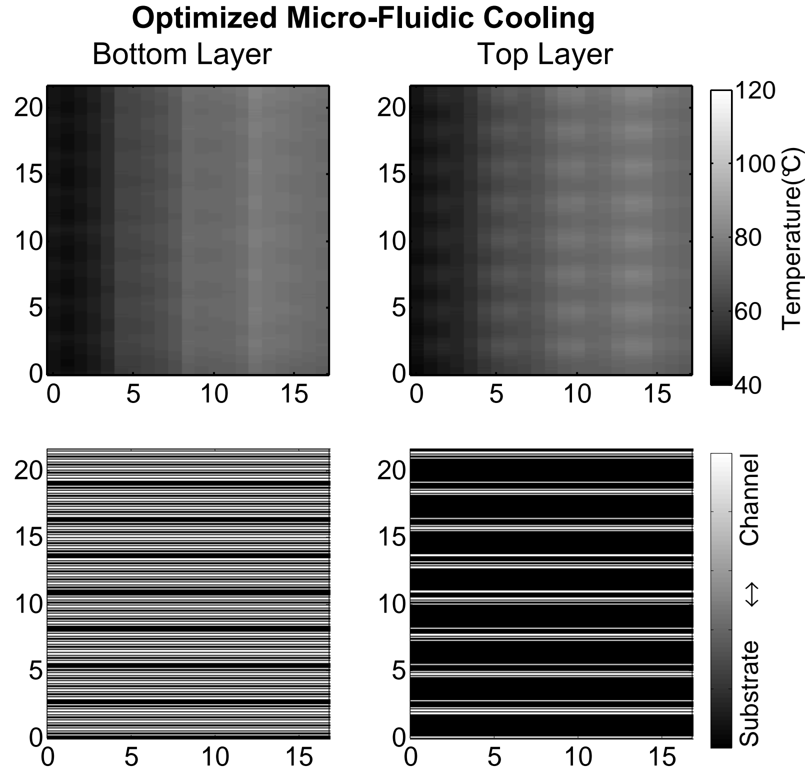


Figure 3.10: Temperature and channel distribution using optimized MF heatsink.

3D CPU using air cooling. Each core spans two layers and the tiling topology is $4 \times 8 \times 1$. The dynamic power density is fixed regardless of cooling scheme, although the leakage power does change with the temperature when uniform and optimized MF heatsinks are applied. Figures 3.9 and 3.10 show the temperature and associated microchannel placement vectors of a uniform and optimized MF heatsink respectively.

We observe that the reduction in peak temperature is only marginal from air cooling to uniform MF cooling, whereas the reduction due to an optimized MF heatsink is substantial. The basic mechanism of improvement in this example is as follows: by removing microchannels on the top layer that run through areas of low power density, more cooling capacity can be delivered to the bottom layer,

which has much higher thermal resistance and suffers from thermal coupling with the high-power top layer. Although the microchannel distribution on the bottom layer remains generally uniform, the top layer only has channels running under the thin strips of high-power-density components. Since many less channels are used in the Optimized MF heatsink, the fluid velocity is increased, counteracting the thermal wake effect and greatly improving heatsink cooling capacity, while still keeping channels in place under local hotspots.

3.10.4 Microchannel Cost Model

In order to reduce convergence time of our simulated annealing approach, we define a cost model of microchannels such that removing channels with lower cost are more likely to improve the objective function. The basic idea is to quantify the amount of power being sunk by each channel, and remove the channels that are sinking the least power. The formulation for our cost model is given below, and illustrated in Figure 3.11.

1) Sum Power: Since **B** is a two dimensional variable, we must create a corresponding two-dimensional representation of the three-dimensional power map. Since each channel sinks power from all sources along the direction of flow, it makes sense to sum the power map along the flow direction. However, one must take into account the decreasing cooling capacity of a microchannel along the direction of flow due to an increase in fluid temperature (*i.e.* the thermal wake effect [107]). Thus the power generated near the outlet is more critical in determining peak temperature

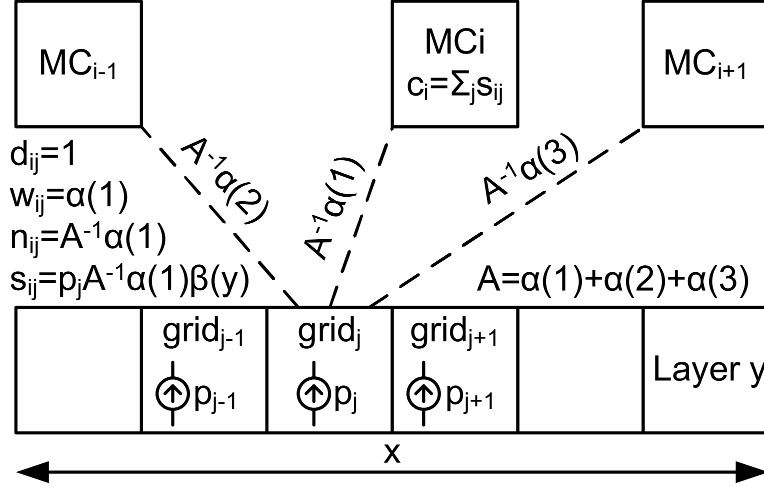


Figure 3.11: Microchannel cost model example

than the power located near the inlet because it is subject to less cooling. When summing the power map along the direction of flow the power is scaled by some function σ which increases along the direction of flow. Scaled power matrix $\mathbf{P}$ is created such that $p_{y,x} = \sum_z power_{y,x,z}\sigma(z)$ where $power$ is the three dimensional power map such that the third dimension runs along the direction of flow. In our study we set $\sigma(z) = 1 + 0.5(z - 1)$.

2) Enumerate Microchannels and Grids: We enumerate each microchannel in $\mathbf{B}$ and each power grid in $\mathbf{P}$ such that the ith microchannel is represented by b_{y_i,x_i} and the jth power grid is has power p_{y_j,x_j}.

3) Evaluate Distance: Generate distance matrix $\mathbf{D}$ such that $d_{i,j} = |x_i - x_j| + \lambda|y_i - y_j|$ is the distance between the ith microchannel and the jth power grid. The coefficient λ is the relative weighting between vertical and horizontal distance, and can be adjusted to model the amount of thermal coupling between layers. In our study $\lambda = 1$.

4) Weight: Using the distance matrix **D** we create a weight matrix **W** which represents the relative thermal conductance from each power grid to each microchannel. We convert **D** to **W** by mapping each element using some function α which decreases with distance. Thus $w_{i,j} = \alpha(d_{i,j})$. In our study α is a Gaussian function centered at 0 with a standard deviation of 2. After determining the values of **W** the normalized matrix **N** is generated such that the sum of weights between each grid and all channels equals one: $n_{i,j} = {w_{i,j}}/{\sum_i w_{i,j}}$. Thus all grids have the same total influence on the outcome of the cost model, but the relative influence on each channel is determined by distance.

5) Scale: Finally a scale matrix **S** is created representing the total power sunk by each channel from each grid. The values of this matrix depend on the position weights from the previous step and the total power in a grid. However, as stated earlier, the thermal resistance to ambient of the layers deep in the stack is more than those near the top, making the power in these layers more critical to peak temperature. To model this, power matrix **P** is scaled by some function β which is an increasing function of layer depth. Thus $s_{i,j} = n_{i,j} p_{y_j,x_j} \beta(y_j)$. In our study we define $\beta(y) = 1 + 0.5(y - 1)$. The final channel cost vector **c** is generated by summing **S** across all grids: $c_i = \sum_j s_{i,j}$. The cost vector is used to determine the set of channels considered for removal during each iteration of the simulated annealing algorithm.

3.11 Simultaneous Optimization

One would assume that floorplan and heatsink optimization would need to be done simultaneously, or in a nested loop to avoid convergence to a local minimum. Initially that approach was implemented, but upon comparison of the nested optimization to the sequential method proposed in the paper, we observed that sequential optimization resulted in very similar quality results as the nested optimization, and significantly reduced the simulation runtime.

Chapter 4: Architectural Opportunities of Micro-Fluidically Cooled 3D CPUs

This chapter presents the results of two studies undertaken to quantify the potential architectural opportunities presented by 3D IC technology using a stacked memory-on-logic processor. In the first study (Section 4.1) we show that indeed significant speedup can be achieved, but as expected this speedup is significantly thermally limited by the trapped heat effect. However we show that MF cooling can overcome the thermal issues and thus realize the true potential of the 3D CPU architectures under consideration. In the second study (Section 4.2) we explore the potential return to a frequency scaling scheme in light of the reduced memory wall inherent to stacked memory processors, and the reduced leakage power and chip

temperatures achieved with micro-fluidic cooling. We find that the energy efficiency scaling trend vs. frequency is actually reversed when MF cooling is applied. Finally we summarize this chapter in Section 4.3.

4.1 2D vs. 3D CPUs and the need for MF cooling

Chapter 2 introduced a number of architectural opportunities brought on by 3D technology, as well as some of the associated challenges. Thermal management was identified as a primary limitation of 3D integration and micro-fluidic (MF) cooling was introduced as a promising potential solution. In this study we begin with the simplest type of 3D CPU: a stacked DRAM memory integrated on top of a traditional 2D multi-core processor. We ask two fundamental questions in this study: What are the potential performance improvements offered by this architecture, and what are the thermally feasible improvements. Furthermore, regarding the second question we investigate how the switch from air cooling to MF cooling will affect the thermal feasibility, and push the 3D memory-on-logic architecture closer to realization of it's true potential.

As discussed in Section 2.4 the primary performance benefit of memory-on-logic stacking comes from higher memory bandwidth [9, 27, 39]. In our study we increase the memory bus frequency to match the CPU core frequency and expand the bus bit witch to match that of the L2 cache line (Table 3.2). Although these two extensions do improve memory bandwidth significantly, they do not fully leverage the additional CPU-DRAM interconnect density offered by TSV technology. To

explore architectural designs with even more bandwidth we consider increasing the number of memory controllers (MCs), allowing parallel memory access and thus scaling memory bandwidth proportional to the number of MCs.

Although additional MCs can also be added to traditional 2D CPUs with off-chip DRAM, they will not benefit from more than a few MCs due to off-chip bandwidth constraints imposed by IO pin count limitations [9,108,109]. On the other hand, memory-on-logic 3D CPUs achieve monotonic (albeit diminishing) speedup as more MCs are added due virtually unlimited[1] CPU-DRAM integration density. Memory latency vs. number of MCs is shown in Figure 4.1 in a traditional 2D off-chip DRAM configuration and a memory-on-logic 3D CPU. This data was generated for a 16-core CPU using the simulation infrastructure and DRAM models introduced in Chapter 3. As more MCs compete for a fixed number of IO pins in a traditional DRAM CPU, the transfer delay from our latency mode (Section 3.3) begins to dominate as it increases proportional to the number of MCs[2]. This makes MC scaling beyond 8 inefficient, whereas the DRAM latency with on-chip vertical integration shows significant gains all the way up to 32.

In the this study we sweep the number of MCs and the clock frequency of a traditional 2D CPU and a memory-on-logic 3D CPU and evaluate the performance, power and temperature. We observe thermal violations in the 3D CPU with air cooling, so we evaluate the potential improvements to thermally feasible performance offered by applying a MF heatsink. The architectural design space considered in

[1]Feasible TSV integration density is many orders of magnitude higher than the density required for any reasonable number of memory controllers.

[2]DRAM bus width per MC is total IO pins (64) divided by total number of MCs.

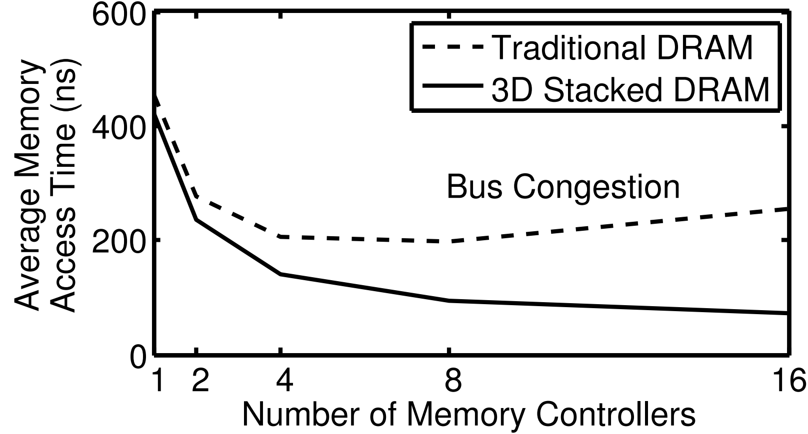


Figure 4.1: Average DRAM latency vs. number of memory controllers [8]

this study is given in Table 4.1. In this study the floorplan topology was fixed and uniform microchannel placement was used. The effects of physical optimizations are introduced in Section 5.1.

Table 4.1: Study 1: Architectural Design Space

Cores	16
Clock Rate	{2.4, 2.6, 3.0, 3.2, 3.4} GHz
Memory Controllers	{1, 2, 4, 8, 16, 32}

We conclude that memory-on-logic architectures do bring significant potential performance improvements, but are thermally infeasible with traditional air cooling. In fact, 3D stacking actually reduces the feasible performance compared to traditional off-chip DRAM when air cooling is applied because the trapped heat effect requires total chip power to be scaled down significantly. However MF cooling is able to realize the potential benefits of 3D CPUs by removing thermal violations. We also show that MF cooling significantly reduces leakage power, more than making up for the required MF pumping power, and begging the question of how MF cooling effects energy efficiency scaling trends, which we investigate in Section 4.2.

4.1.1 Performance

Throughout this dissertation we measure performance by the average number of committed instructions per nanosecond (IPnS) which is equivalent to billions of instructions per second (BIPS). Figure 4.2 shows the performance of our target processor with a variable number of memory controllers and clock rates. On average the peak performance for a 3D CPU is 1.62x the peak performance of a 2D CPU within the studied design space. Although 3D integration offers the potential for significant speedups, these improvements can only be feasibly realized if the heat generated as a result of the increased power flux and thermal resistance can be sufficiently removed form the chip. It is important to note that performance improvements result from both reduced latency at a fixed number of MCs, and the ability to leverage more MCs and thus access multiple DRAM ranks in parallel.

4.1.2 Temperature

Figures 4.3 and 4.4 show the peak temperature of our target processor configurations. In this work we assume the thermal violation temperature is 85 °C, which is shown as a horizontal black line in each figure. The number annotated above each bar represents the maximum performance (across all different MC configurations) that does not violate the thermal constraint for each frequency/benchmark pair.

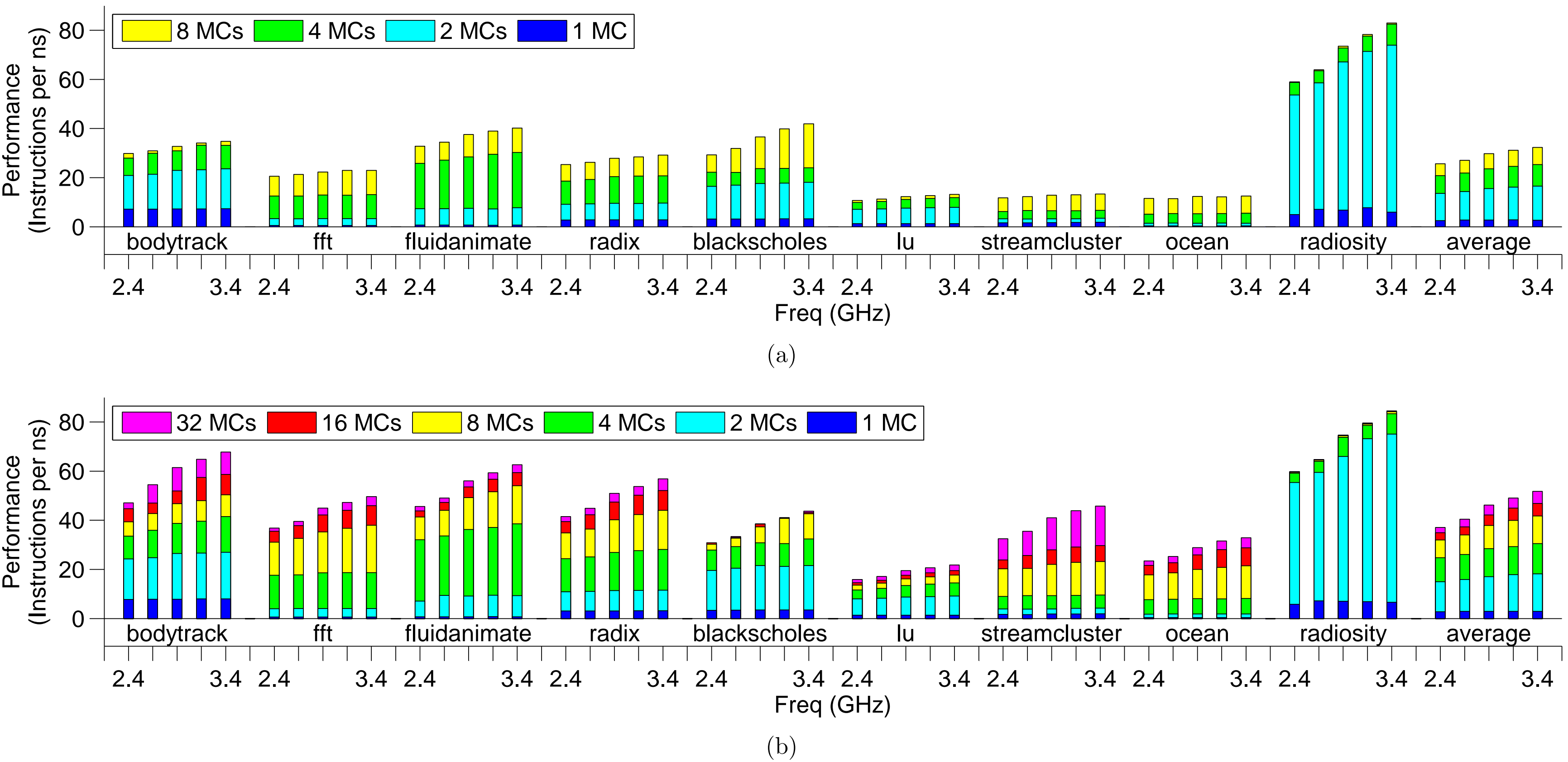


Figure 4.2: Performance vs. MCs and frequency (a) 2D CPU (c) 3D CPU

In the 2D case adding more memory controllers did not significantly increase the temperature of the chip (Figure 4.3). This is because the generated heat has a low thermal resistance path to the heatsink (Section 3.8). Thus no thermal violations occur, and the optimal number of MCs can be implemented without considering any new cooling methods. However the performance gains are limited.

In the 3D case, when the chip is air cooled (Figure 4.4(a)) the peak temperature often surpasses the thermal constraint, and thus the peak performance cannot be achieved. The maximum achievable performance of an air cooled 3D system is in most cases actually less than that of a 2D IC. This is because adding more MCs to a 3D IC increases the peak temperature drastically (which is not the case for 2D), meaning that in most cases the 2D IC can use more MCs than the air cooled 3D IC, causing the 3D IC to get worse performance.

We know from the performance plots (Figure 4.2) that 3D ICs are capable of achieving much greater performance, and this motivates the need for more aggressive cooling techniques in order to achieve the performance increases potentially offered by 3D integration. When micro-fluidic cooling is applied (Figure 4.4(b)) the peak temperatures are all brought to below the temperature threshold, and the great performance increases offered by 3D integration can be thermally realized. Thus, aggressive cooling has enabled more aggressive architectural configurations. On average, the MF cooled 3D CPU's maximum achievable performance is 2.4x greater than the maximum achievable performance of an air cooled 3D CPU and 1.6x greater than the maximum achievable performance of an air cooled 2D CPU.

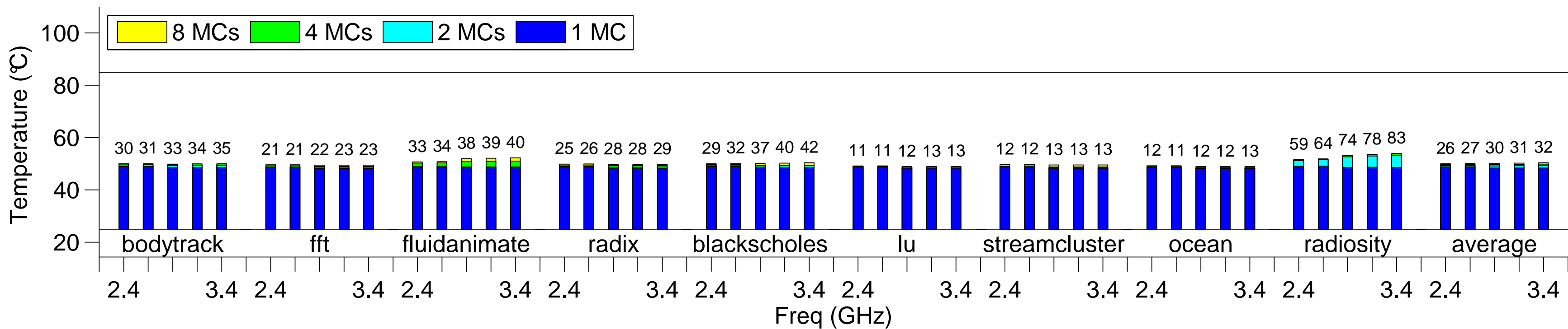


Figure 4.3: Temperature vs. MCs and frequency of air cooled 2D CPU

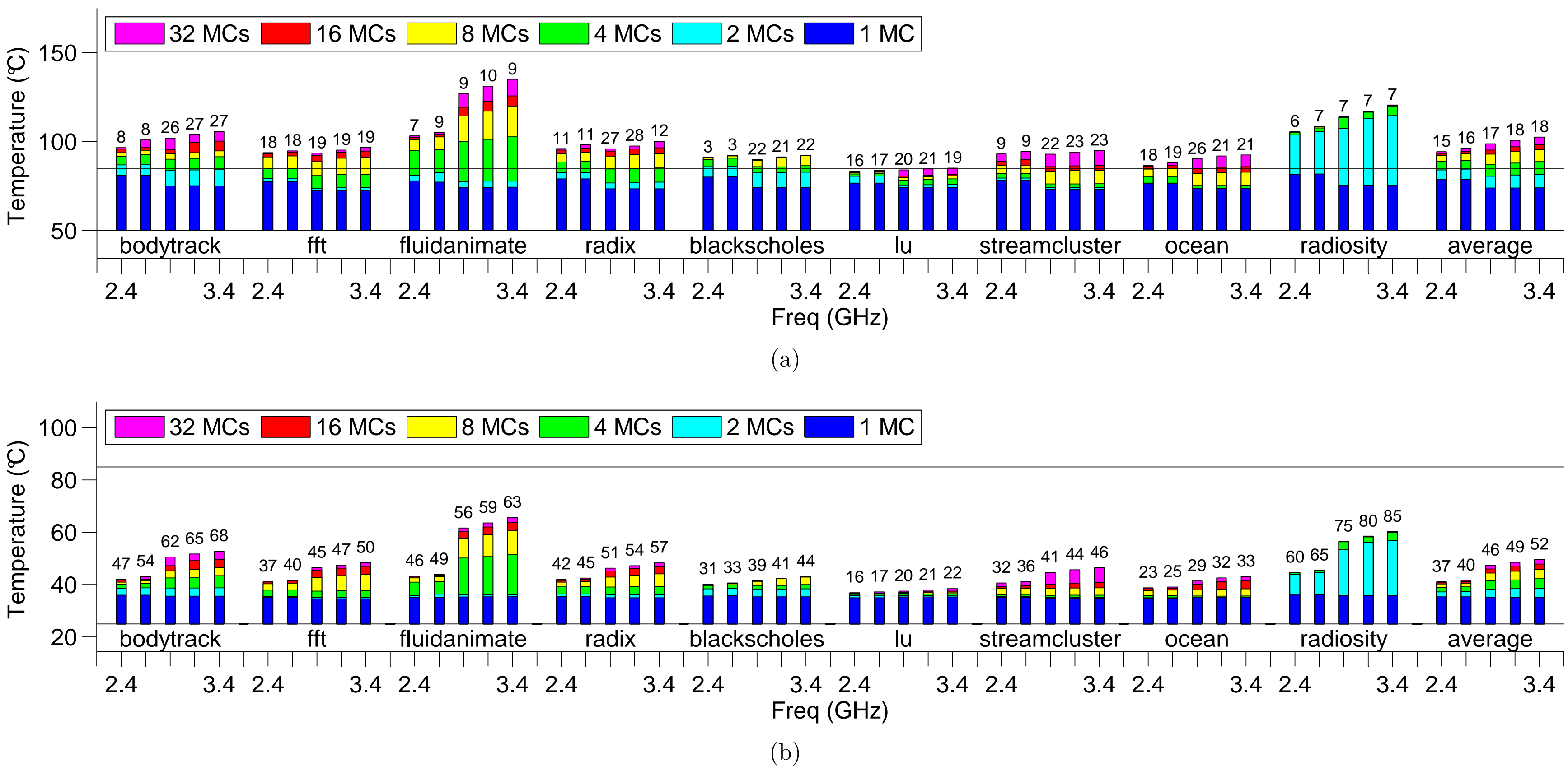


Figure 4.4: Temperature vs. MCs and frequency (a) air cooled 3D CPU (b) MF cooled 3D CPU

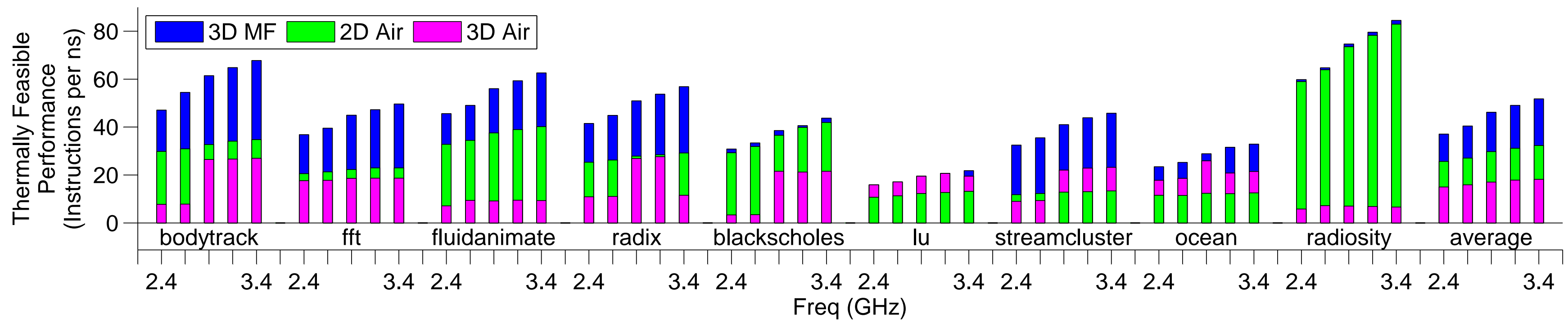


Figure 4.5: Best achievable performance subject to thermal constraints

4.1.3 Thermally Feasible Performance

The maximum performance subject to thermal constraints (*i.e.* the annotations in Figures 4.3 and 4.4) is plotted in Figure 4.5. When air cooling is used 3D and 2D CPUs alternatively outperform each other depending on the workload. In general 3D CPUs have better performance than 2D CPUs when the number of MCs is the same. However, for most benchmarks 2D CPUs can thermally accommodate more MCs, allowing them to outperform an air cooled 3D CPU. But for the low power benchmarks (*e.g.*, `lu`, `streamcluster` and `ocean`) the 3D temperature is low enough even with air cooling to take advantage of the additional bandwidth offered by memory-on-logic stacking. When thermal concerns are alleviated with MF cooling, 3D CPUs always perform best.

It can be observed in Figure 4.5 that average performance improves very little with respect to frequency in an air cooled 3D CPU. Due to thermal constraints, there must be a trade-off between frequency and the number of memory controllers to maintain a safe temperature. With MF cooling or a traditional 2D layout, enough temperature slack exists in the system that both frequency scaling and increased number of memory controllers can be leveraged for higher performance.

4.1.4 Power

Dynamic power remains the same regardless of heatsink type. However, Figures 4.6 and 4.7 show that adding MF cooling actually decreases the total power dissipation dramatically. This is because the leakage power is strongly dependent

on temperature and the temperature reduction due to liquid cooling reduces the leakage power. On average micro-fluidic cooling can reduce 3D IC leakage power by 20.9W, which easily justifies the extra power used to pump the fluid through the microchannels (less than 1 W). Furthermore, it begs the question of how MF cooling effects energy efficiency scaling trends, which are examined in Section 4.2.

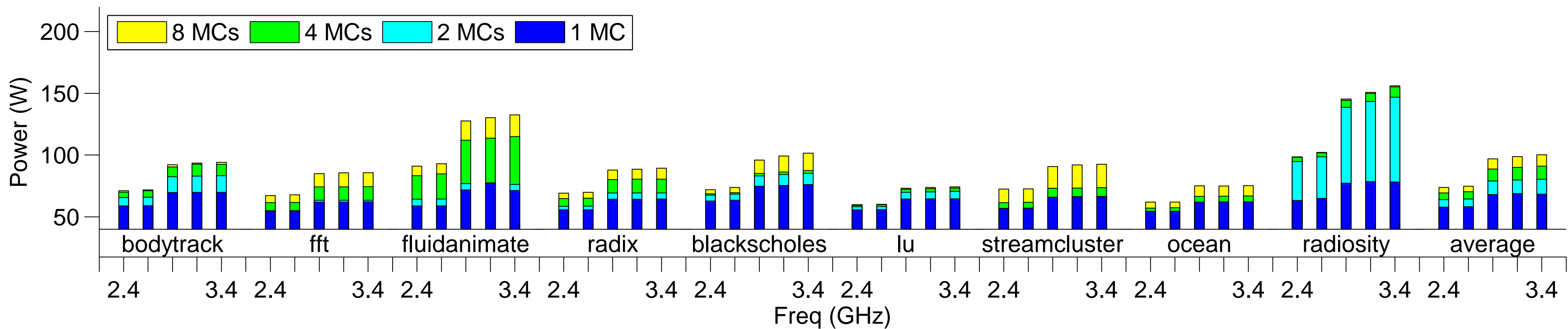


Figure 4.6: Power dissipation vs. MCs and frequency of air cooled 2D CPU

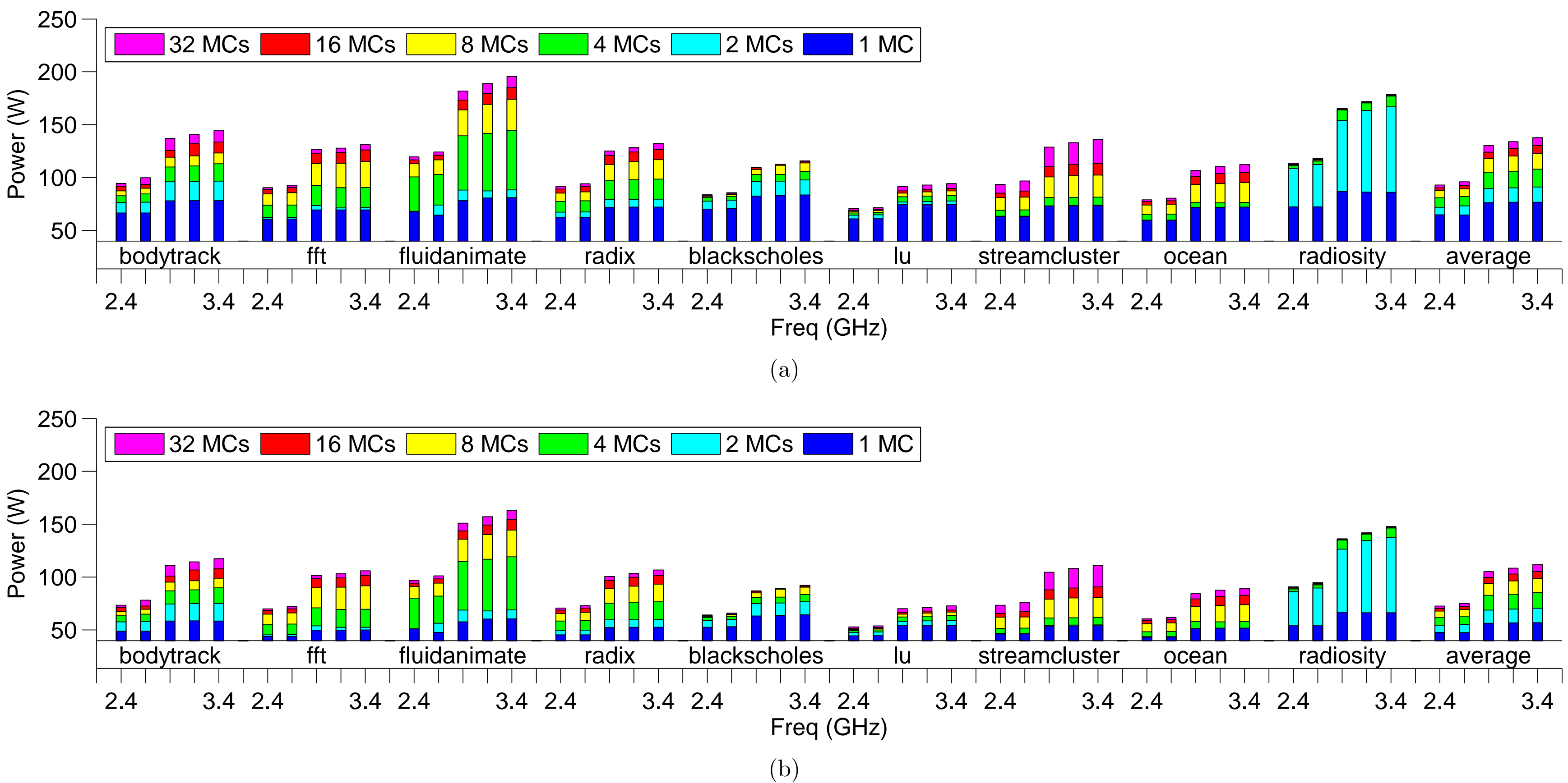


Figure 4.7: Power dissipation vs. MCs and frequency (a) air cooled 3D CPU (b) MF cooled 3D CPU

4.2 Frequency Scaling with Micro-Fluidics

Since the 1980s Moore's Law performance scaling was traditionally achieved through constant increases to CPU frequency, made possible by similar reductions in capacitance and voltage through technology scaling. However the increase in power, and therefore temperature, associated with frequency scaling became unsustainable in the mid 2000s [110]. One of the biggest problems was the exponential increase in leakage power as temperatures increased, causing energy efficiency to plummet past a few GHz [111]. Another big issue with frequency scaling was the ever increasing memory wall gap between processor and memory performance (Section 2.2) [110].

In Section 4.1 we observed a large reduction in leakage power and temperature due to the application of MF cooling. Additionally we observed a significant performance improvement due to increased memory bandwidth when memory-on-logic stacking was applied. These two observations cause us to reexamine the feasibility and efficiency of further frequency scaling in 3D CPUs with MF cooling.

In this study we first argue that frequency scaling is a more versatile scaling trend than the core scaling that has come to replace it. We sample the parallelism of a group of benchmarks and show that only those with very large degrees of parallelism will benefit from core scaling, whereas all workloads benefit from frequency scaling. However with traditional air cooling, both core and frequency scaling are limited in 3D CPUs. Next we compare air cooled and MF cooled 3D CPUs and their associated scaling trends with respect to temperature, power and energy efficiency.

4.2.1 Design Space and Benchmarks and Metrics

The design space swept in this study includes the number of cores (*i.e.* core scaling) and the clock rate (*i.e.* frequency scaling). The specific values simulated are given in Table 4.2. Different workloads exhibit different performance/power/temperature trade-offs across these different variables, and the highest performance thermally feasible design point is identified for each benchmark. In this study the floorplan topology was fixed and uniform microchannel placement was used. The effects of these physical optimizations are introduced in Section 5.1.

Table 4.2: Study 2: Architectural Design Space

Cores	{16, 32, 64}
Clock Rate	{2.4, 3.0, 3.6} GHz
Memory Controllers	0.5 per Core

Each benchmark (except for `ferret`, which has a unique data pipeline) has some period of sequential execution that occurs on a single processing core, followed by a period of parallel execution distributed across all cores. The ratio of parallel execution time to total execution time[3] is denoted α. According to Amdahl's law, the amount of speedup offered by using n cores (compared to a single core) is shown in Equation (4.1).

$$\frac{Performance(n)}{Performance(1)} = \frac{n}{n - \alpha(n-1)} \tag{4.1}$$

[3]Benchmarks were terminated after 540M instructions if they had not already finished to maintain reasonable simulation time.

In the architectures simulated here, adding more cores also changes the size and distribution of the L2 cache as well as increasing the average distance between routers in the NOC, causing performance to depend on other factors beyond Amdahl's law. Nevertheless, benchmarks with a large α value often achieve optimal performance with more cores, whereas benchmarks with a low α value often achieve optimal performance with a smaller number of cores. The α value and highest performing core count for each benchmark is tabulated in Table 4.3.

In this work we measure performance by the average number of committed instructions per nanosecond (IPnS) and energy efficiency by the reciprocal of the energy delay product (EDP).

4.2.2 Core and Frequency Scaling

For each benchmark, we find the highest performing architectural configuration that does not violate the peak temperature constraint of 85 °C. The results of this experiment are shown in Tables 4.3.

We observe that with air cooling both the number of cores and the frequency is severely limited. With the application of MF cooling, every benchmark except `radix` is able to achieve its optimal number of cores. Moreover, only `swaptions` pursues core scaling over frequency scaling, and this is because `swaptions` is nearly 100% parallel. The main conclusion from this data is that even when thermal constraints are mitigated (*e.g.*, by applying MF cooling), the amount of potential improvement due to core scaling has an established upper limit inherent to the

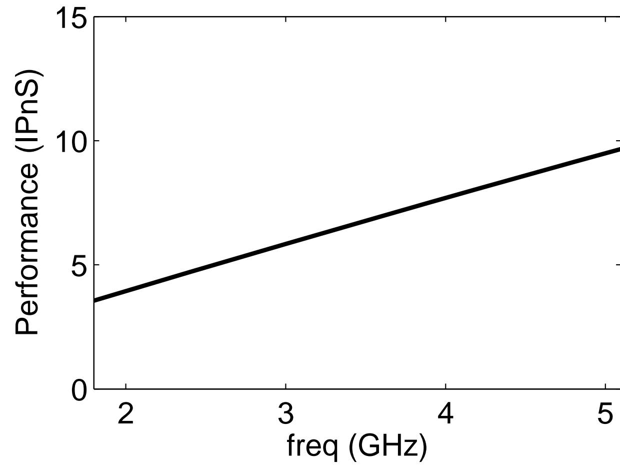
Table 4.3: Maximum benchmark performance s.t. thermal constraint

Benchmark	α (%)	Opt. #Cores	Air Cooled			MF Cooled			Inc. IPnS
			#Core	Freq	IPnS	#Core	Freq	IPnS	
Swaptions	99.8	64	16	3.0	35.1	64	3.0	119.6	3.41x
Radix	99.8	64	16	3.0	34.9	32	3.6	51.8	1.48x
Barnes	98.8	64	16	3.0	27.4	64	3.6	70.0	2.56x
FMM	98.7	32	16	3.0	24.5	32	3.6	42.6	1.74x
Water-spatial	93.2	64	16	3.0	40.5	64	3.6	67.1	1.66x
Water-nsquared	93.0	16	16	3.0	32.4	16	3.6	38.4	1.19x
FFT	74.3	64	16	3.0	6.2	64	3.6	7.6	1.23x
Raytrace	71.9	16	16	3.0	1.9	16	3.6	2.1	1.15x
Fluidanimate	35.7	16	16	3.0	4.7	16	3.6	5.5	1.18x
Dedup	29.2	16	16	3.6	1.3	16	3.6	1.3	0.00x
Facesim	0.0	16	16	2.4	4.8	16	3.6	7.0	1.48x
Radiosity	0.0	16	16	3.0	2.5	16	3.6	3.0	1.19x
Ferret	-	32	16	3.0	4.6	32	3.6	5.5	1.20x
Average									1.57x

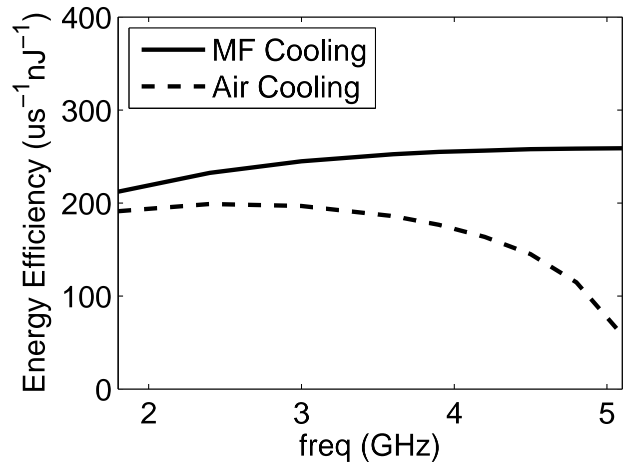
parallelism (α) in the workload. On the other hand frequency scaling can continue to push performance for any arbitrary workload, until the thermal constraint is hit. With MF cooling and 3D memory-on-logic stacking we expect that frequency scaling once again becomes a viable strategy, at least in the short term.

4.2.3 Scaling Trends

To further investigate the frequency scaling trends of 3D CPUs, we fixed the number of cores (32) and performed a detailed frequency sweep on a sequential benchmark (`facesim`). The sequential nature of the benchmark eliminates the possibility of improving its performance through core scaling, and leads us to view frequency scaling as the only avenue for benchmark speedup. We compare the frequency scaling trends of an air cooled vs. MF cooled 3D CPU.



(a)



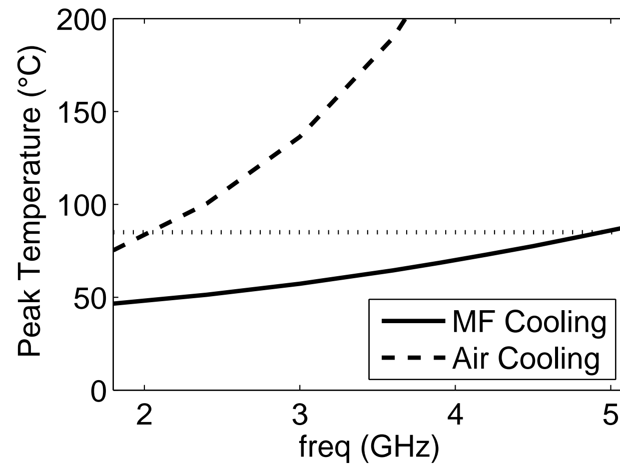
(b)

Figure 4.8: 3D CPU (a) performance (b) energy efficiency vs. frequency with air cooling and MF cooling

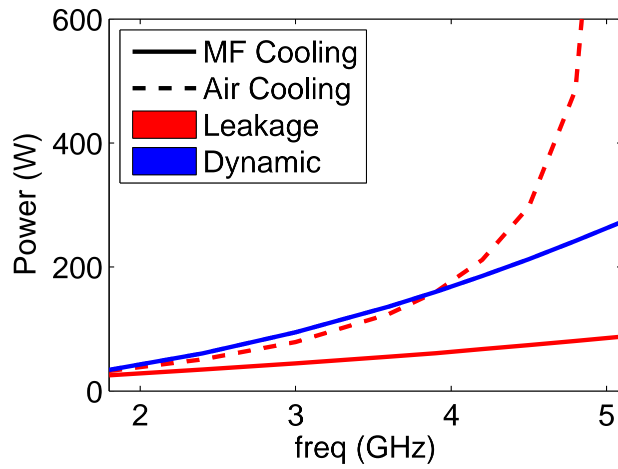
It is obvious that frequency scaling will improve performance roughly linearly with frequency (Figure 4.8(a)), but what is interesting is how power, temperature and energy efficiency scale using different types of heatsinks. Figure 4.8(b) shows that air cooled 3D CPUs will become energy inefficient beyond 3-4 GHzwhereas MF cooled 3D CPUs will continue to be energy efficient beyond 5 GHz. This is an interesting result because the traditionally frequency scaling paradigm ended around 3 GHzwhich has good agreement with the simulation data. This implies the possibility of MF cooling providing a realignment back to frequency scaling, or the application of frequency and core scaling in tandem for future computer architectures.

Figure 4.9(a) shows the thermal scaling trends. We can see that air cooled 3D CPUs become thermally infeasible beyond 2 GHzwhereas MF cooling can push thermal feasibility out to nearly 5 GHz. One advantage of 3D integration is core scaling independent of technology scaling by applying logic-on-logic stacking. However this will yield similar thermal scaling trends to frequency scaling due to increased power flux, and will likewise require aggressive active cooling solutions such as MF cooling.

Finally, Figure 4.9(b) shows the power scaling trends. Two important observations can be made about air cooled 3D CPUs. First, they generally have large amounts of leakage, roughly 50% up to 4 GHz. Beyond this point the thermal runaway phenomenon [62] causes the leakage and temperature to quickly increase without bound in a positive feedback loop. Moreover, leakage power scales at the same rate as dynamic power, reducing energy efficiency as clock rates increase. MF cooling not only removes the thermal runaway issue (in the range of frequencies



(a)



(b)

Figure 4.9: 3D CPU (a) temperature (b) power vs. frequency with air cooling and MF cooling

simulated), but also causes leakage power to scale slower than dynamic, leading to more efficient systems and improving the effectiveness of dynamic power control schemes like clock gating [112].

4.3 Summary

In this Chapter we have quantitatively investigated some of the architectural opportunities offered by memory-on-logic 3D CPUs with micro-fluidic cooling. We consider the memory bandwidth advantages of 3D stacked memory and identify the need for embedded active cooling to realize the theoretical gains of such a system. Furthermore we consider the scaling trends of 3D CPUs with MF cooling and show that frequency scaling may once again emerge (in conjunction with core scaling) as a viable avenue for performance scaling of future CPUs cooled with micro-fluidics.

Section 4.1 made the case for memory-on-logic 3D CPUs by demonstrating their potential speedup over traditional 2D CPUs with off-chip DRAM, but showed that those improvements could only be thermally realized with embedded active cooling such as MF cooling due to the high power flux of the core logic layer and the trapped head effect of the stacked DRAM. Speedup was achieved by increasing the clock speed and bit width of the memory bus using high density TSV integration, and increasing the number of dedicated memory controllers allowing for parallel memory access.

Section 4.2 built on some of the findings from Section 4.1 and evaluated the frequency scaling trends of power, temperature and energy efficiency when using 3D CPUs with MF cooling. Two major factors in the switch to multi-core paradigm were excessive power and heat, and the memory wall. We show that the power and heat scaling issue can be significantly curbed with embedded MF cooling, and that the memory wall can be overcome with high bandwidth on-chip DRAM integration. The scaling trends of temperature and leakage power are significantly linearized by application of MF cooling, and moreover, the energy efficiency continues to rise in an MF cooled 3D CPU as frequency is increased up to 5 GHzwhereas the energy efficiency of an air cooled CPU begins to decrease past 3-4GHz.

Chapter 5: Architectural-Physical Co-Design of Micro-Fluidically Cooled 3D CPUs

In this chapter we present results from the application of our proposed co-design flow. Section 5.1 applies the proposed scheme across a 3D CPU design space with different physical optimizations, objective functions, and physical constraints. Section 5.2 investigates a fundamental trade-off between TSV density (*i.e.* inter-layer communication bandwidth) and the cooling capacity of a MF heatsink. Specifically we target a pin-fin heatsink. Compared to microchannel MF heatsinks,

pin-fin MF heatsinks are known to have higher cooling capacity, but are more restrictive on TSV density and placement [113]. Section 5.3 concludes this chapter with a summary.

5.1 Thermal-Reliability Aware Architectural-Physical DSE

In this study we investigate the effects of the floorplan (Section 3.9) and cooling (3.10) optimization schemes on the feasibility region of a 3D CPU design space. In addition to the thermal constraints imposed in Chapter 4 we also incorporate the reliability model from Section 3.7 and impose a reliability constraint on the design space. We combine the design variable spaces considered in the two previous studies in Chapter 4. This results in a three-dimensional design space of cores, MCs and frequency, as enumerated in Table 5.1.

Table 5.1: Study 3: Architectural Design Space

Cores	{16, 32, 64}
Clock Rate	{2.4, 3.0, 3.6} GHz
Memory Controllers	{0.125, 0.25, 0.5} per Core

Thus we perform 3D memory-over-logic processor DSE across a combined design space of architectural parameters, floorplan topology and MF heatsink design, subject to thermal and reliability metrics. The optimization metric is performance measured in instructions per nanosecond (IPnS, a.k.a. BIPS). We use a variable reliability threshold of $0.00 \leq \alpha \leq 0.99$ such that the probability the CPU fails

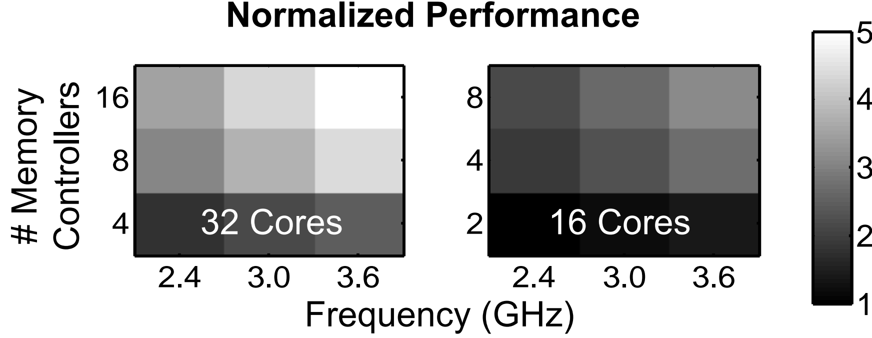


Figure 5.1: 3D CPU design space performance

before target lifetime is less than or equal to $1 - \alpha$. For sensitivity analysis, we also investigate the effects of ignoring one or more of the floorplan objective terms and sweeping the tightness the reliability constraint.

5.1.1 Feasibility Region

First we explore the feasibility region of the design space. An architecture is considered feasible if for all benchmarks the thermal and reliability constraints are met. Although the entire design space from Table 5.1 was considered in this evaluation, we found that no 64-core architectures could meet both thermal and reliability constraints, so the 64-core architectures were trimmed from the design space for this section[1]. Figure 5.1 illustrates the normalized performance of the trimmed design space, evaluated over a set of parallel benchmarks from Splash-2 [84] and PARSEC [85] benchmark suites. Performance values for each benchmark were normalized to the 16-core 2 MC 2.4 GHzarchitecture before averaging across all benchmarks.

[1]However in Section 5.1.2 we consider the optimal architecture of each benchmark individually (as was done in Section 4.2) and the 64-core architectures are included in those results.

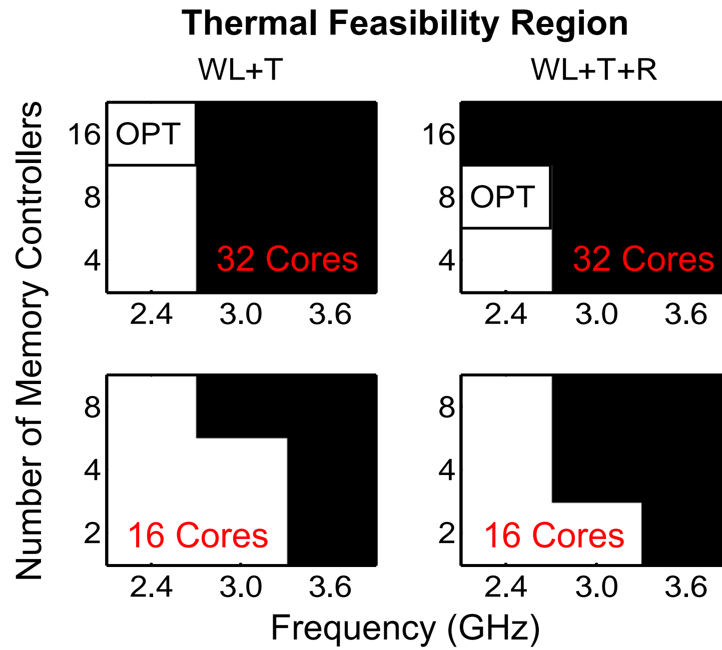


Figure 5.2: Thermal feasibility region (shown in white)

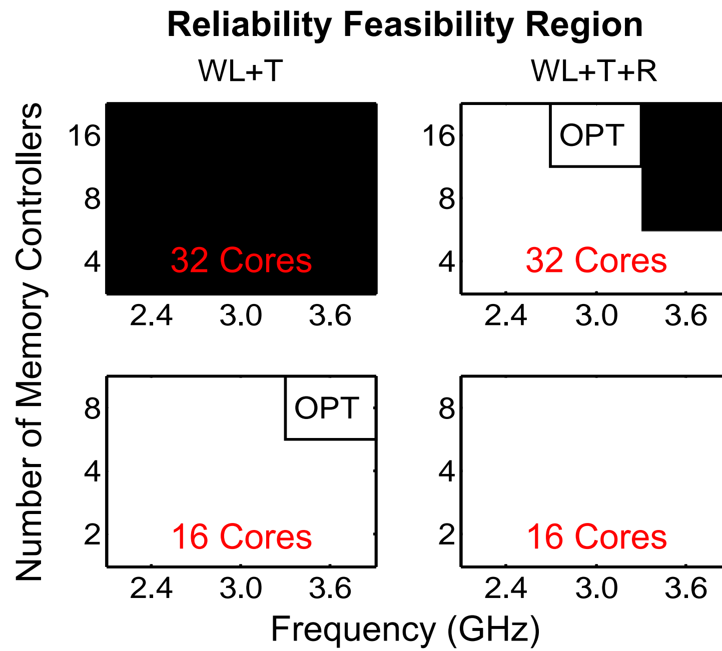


Figure 5.3: Reliability feasibility region (shown in white)

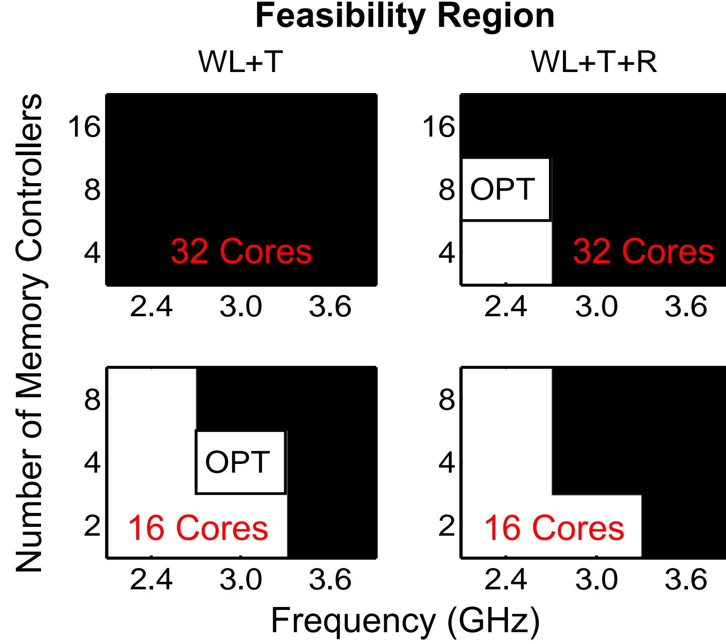


Figure 5.4: Thermal-reliability feasibility region (shown in white)

Figurea 5.2 through 5.4 show the feasibility region of the design space. Feasible architectures are shown in white, infeasible architectures are shown in black and the highest performing feasible architecture is marked with "OPT". The thermal (Figure 5.2) and reliability (Figure 5.3) feasibility regions are evaluated separately and their intersection defines the true thermal-reliability feasibility region (Figure 5.4). Thermal feasibility is defined as maximum on-chip temperature less than $T_{violation} = 85°\text{C}$. Reliability feasibility was defined as $P_{fail}(t_{target}) < \alpha$ where $\alpha = 99\%$ is the reliability confidence and $t_{target} = 3$ years is the lifetime target.

Two floorplan objective functions are considered. The first only includes wirelength[2] and temperature ($WL + T$), whereas the second also includes reliability ($WL + T + R$). The results in this figure assume MF cooling with uniform microchannel placement.

Looking at the thermal feasibility region, we observe that the addition of reliability to the floorplan objective function causes the thermal feasibility region to contract, resulting in reduced optimal performance. However, the addition of reliability to the floorplan objective massively expands the reliability feasibility region and the true thermal-reliability feasibility region which increasing the optimal performance significantly.

This result exposes an interesting potential trade-off between temperature and reliability in 3D CPUs. Although increased temperature increases the probability of failure of a single TSV, it is quite possible that thermally optimized floorplans contain more 3D nets (*i.e.* more cuts in the inter-layer partition) in order to optimize the distribution of power. In some cases the increase in number of TSVs will outweigh the reduction in temperature when considering the net effect on system reliability.

Overall we conclude that even though one would assume optimization of thermal and reliability metrics to go hand in hand, this is in fact not the case. Optimization for temperature only is significantly suboptimal due to splitting too many 3D nets to get fine-grained power density matching against the thermal resistance of

[2] In this context wirelength consists of the combination of area A and total negative slack S from Equation (3.10).

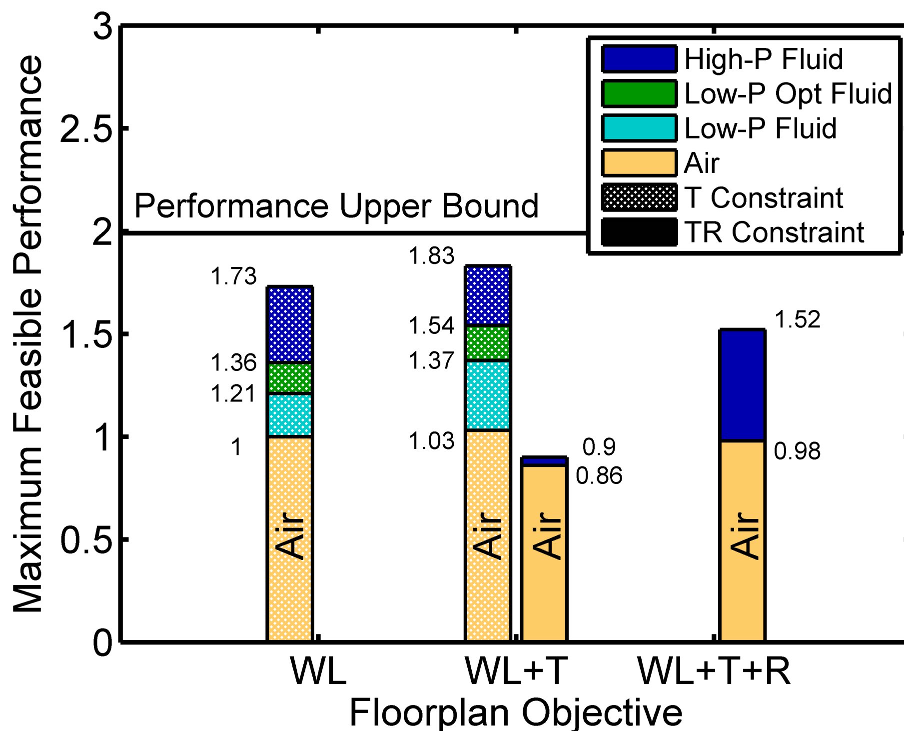


Figure 5.5: Co-design results

each stack layer. Conversely, consideration of the reliability objective in optimization increases hot-spot temperature, and awareness of both metrics is necessary to maximize the intersection of the thermal and reliability feasibility region.

5.1.2 Optimal Performance

The optimal feasible performance of the investigated architectural design space is plotted in Figure 5.5. This data is generated by finding the optimal feasible performance of each benchmark separately, and normalizing against the base case before averaging the results across all benchmarks. In this study the base case

is as follows: air cooling, thermal-reliability unaware floorplanning (WL), and no reliability constraint (*i.e.* $\alpha = 0$). Three floorplan objectives are used to generate the data, each one adding an additional term to the objective function.

The data is obtained using two different constraints: thermal (T Constraint) and thermal-reliability (TR Constraint). These two constraints are defined by setting $\alpha = 0$ and $\alpha = 0.99$ respectively. The unconstrained performance of the design space is notated as an upper bound. Likewise, four different cooling schemes are considered: high-pumping-power uniform MF cooling (High-P Fluid), low-pumping-power optimized MF cooling (Low-P Opt Fluid), low-pumping-power uniform MF cooling (Low-P Fluid) and traditional air cooling (Air). Low-pumping-power MF cooling uses 5x less pumping power, and optimized MF cooling uses the microchannel placement optimization technique described in Section 3.10.

Comparing the first (leftmost) two bars in the figure, we can see that without reliability constraints, thermally-aware floorplanning improves thermally feasible performance between 3% and 13% depending on the cooling method applied. Additionally one can observe that none of the considered cooling techniques are able to thermally unlock the entire design space, and the improvement in performance due to increasing MF cooling power 5x is less than 2x. Finally, microchannel placement optimization can provide significant performance improvements while maintaining a constant pumping power, thus greatly increasing the power efficiency of the MF heatsink.

Comparing the middle two bars we observe that the massive improvement to the thermal feasibility region provided by MF cooling becomes a moot point when reliability constraints are included. However, by comparing the last (rightmost) two bars we see that reliability-aware floorplanning can once again unlock the performance potential of MF cooling. Reliability feasibility does not significantly affect the potential performance of an air-cooled 3D CPU since the architectural design points which would benefit from the expanded reliability feasibility region are still thermally infeasible. The conclusion here is that aggressive cooling is required to thermally unlock 3D CPU performance, but must also be accompanied by reliability aware physical design to realize the potential gains brought by the new cooling technology.

5.1.3 Reliability Constraint Sensitivity

Finally we repeat the above analysis for different values of α and compare the performance ratio between reliability aware ($WL + T + R$) and reliability unaware ($WL + T$) designs. The improvement in average feasible performance is shown in Figure 5.6. We observe that the performance improvement due to reliability awareness in floorplanning increases as the reliability constraint tightens because reliability becomes a more significant factor in determining physical feasibility.

Moreover we observe that the performance improvement due to reliability awareness is significantly less when air cooling is used because many design points are thermally limited. Due to a very small thermal feasibility region, reliability aware

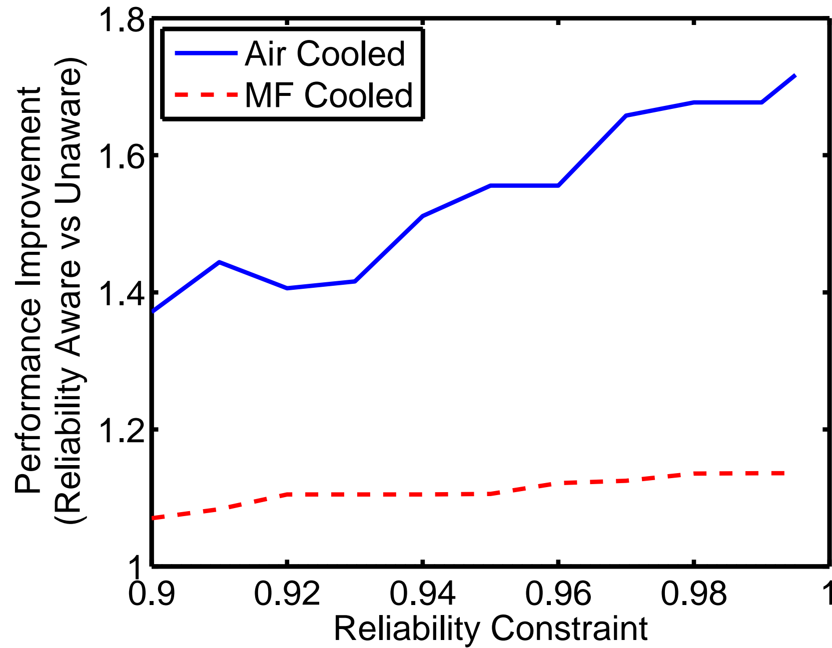


Figure 5.6: Performance improvement due to reliability-aware FP

design has little effect on the physical feasibility region, and thus offers only marginal improvement. On the other hand when MF cooling is used the improvement due to reliability-aware floorplanning is quite large since reliability is the dominating factor determining physical feasibility.

The conclusion is that the effectiveness of certain optimization schemes, such as reliability-aware floorplanning, will depend on other design choices, such as heatsink type, and the design specifications, such as reliability constraint. This further motivates the need for a holistic co-design paradigm.

5.2 Thermal-Bandwidth Trade-offs in MF Cooled 3D CPUs

In the previous studies we have investigated the trade-offs between performance, temperature and reliability across an architectural physical design space. In those studies constraints on TSV integration density did not come into play because the microchannel MF heatsink can accommodate sufficient integration density to support the architectures investigated in this dissertation[3]. However, other types of MF heatsinks exist, which offer better cooling at the expense of reduced TSV integration density [113, 115]. In this study we investigate one such heatsink design: the micro-fluidic pin-fin heatsink. In this section we present a study that shows that a heatsink designed for maximum cooling will actually limit the architectural design space due to inter-layer bandwidth constraints more so than a heatsink that provides worse cooling in order to accommodate higher TSV density.

Micro-fluidic pin-fin heatsinks (Figure 5.7) pump fluid through cavities etched into the silicon substrate of each layer in a 3D chip stack. The fluid cavities are etched around cylindrical islands of silicon called pin-fins. Pin-fins provide a physical, electrical and thermal interconnection between adjacent layers in the chip stack, and provide a path for heat transfer from the silicon into the fluid.Unlike microchannel heatsinks, pin-fin cooling pumps all fluid through a single connected cavity, and has been shown to provide better cooling performance compared to a micro-channel heatsink when fluid velocity is high [113, 115].

[3]However the inter-layer integration density required for more fine-grained 3D circuits may see limitations due to micro-channel heatsinks. Moreover TSV-microchannel conflicts impose constrains on detailed gate-level placement [30, 31, 114]

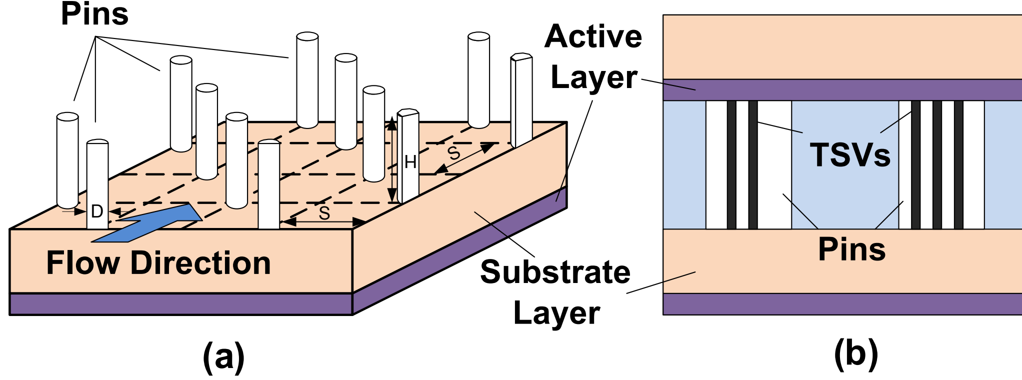


Figure 5.7: Micro-fluidic pin-fin cooling of a single layer in a 3D-IC

Two of the most important geometric parameters that determine the cooling capacity of a micro-fluidic pin-fin heat sink are the pin diameter D and pitch S [113, 116], which are illustrated in Figure 5.7. The pin pitch determines the number of pins per unit area, and the pin diameter determines the surface area of each pin. Increasing pin diameter or decreasing pitch increases the total surface area between fluid and silicon substrate, increasing heat conduction, but also increases the resistance to flow, causing fluid velocity to drop when a constant pressure drop is enforced between fluid inlet and outlet. The micro-fluidic pin-fin heatsink parameters explored in this paper are enumerated in Table 5.2.

Table 5.2: Micro-fluidic pin-fin heatsink dimensions

Variable	Value	Unit	Description
S	$\{250, 300, \ldots, 600\}$	µm	Pin Pitch
D	75	µm	Pin Diameter
H	100	µm	Pin Height

Past work [113] has shown that micro-fluidic pin-fin heatsink parameters can be optimized to improve cooling capacity, but have not considered how such optimizations affect architectural design constraints such as vertical interconnect den-

sity. Furthermore that work only considered one fixed micro-architecture, and did not consider how optimal heatsink parameters change under different architectural design choices.

One drawback associated with micro-fluidic cooling in general is the resource conflict that emerges between TSVs and fluid cavities. Since TSVs cannot pass through the fluid cavities, the location and density of vertical interconnects is determined by the design of the cooling system, such as pin-fin or microchannel diameter and pitch. In other words, TSVs can not be placed through the fluid cavity. In a pin-fin MF heatsink, TSVs are generally more constrained because more of the chip area is dedicated to the fluid cavity [115]. In such a heatsink, TSVs can only pass through the pins themselves (Figure 5.7).

Past work [30, 31] has shown that this resource conflict can restrict the placement of TSVs, leading to increased wirelength and thus critical path delay, but has not considered how the resource conflict can affect micro-architectural design choices.

Our results show there exists a trade-off between maximum TSV density and cooling capacity of the micro-fluidic heatsink. Since different 3D CPU architectures require varying amounts of vertical interconnect density, the cooling solution for each architecture should be designed to maximize cooling while accommodating sufficient TSV bandwidth (BW). We show that naïve application of fixed micro-fluidic heatsink designs will severely limit the feasible design space for 3D CPUs and result in the selection of suboptimal designs.

5.2.1 Bandwidth Requirements

The bandwidth requirement of a 3D CPU architecture is defined as the maximum TSV density required by the architecture. In this study we simulate single-layer cores, so TSVs are only required for extra-core communication: 1) communication between memory controllers and DRAM, and 2) communication between routers. An extension of this study which is left to future work would be to include multi-layer cores and the TSV density requirements associated with these intra-core vertical nets.

5.2.2 Memory Controller TSV Density

The number of DRAM buses passing through layer i in a vertical column of memory controllers (MCs) is i: the number of MCs contained on all layers below and including layer i. Thus the logic layer with the highest MC TSV density is always the top layer, layer n. The minimum TSV density required for communication between the MCs and the DRAM DT_{MC} is given in Equation (5.1), where W_{bus} is the DRAM bus width, A_{TSV} is the area of a single TSV and A_{MC} is the total area of a single memory controller. In this work W_{bus} is assumed to be 512 bits (64 bytes).

$$DT_{MC} = nW_{bus}A_{TSV}\frac{1}{A_{MC}} \tag{5.1}$$

5.2.3 Router TSV Density

The number of TSVs between layer i and $i+1$ in a vertical column of routers was defined in Equation (2.1). Thus the minimum TSV density requirement for router communication, DT_{ROUT}, is given in Equation (5.2) where A_{ROUT} is the total area of a single router.

$$DT_{ROUT} = \max_{i=\{1,2,\ldots,(n-1)\}} T_{ROUT}(i) A_{TSV} \frac{1}{A_{ROUT}} \tag{5.2}$$

5.2.4 TSV Density Requirement

The overall TSV density requirement of a 3D CPU DT is the larger of the two aforementioned density requirements, as expressed in Equation (5.3). In this study we assume TSV pitch is 10 µm, making $A_{TSV} = 100$ ^{2}µm. Other area values used in this study are: $A_{MC} = 8.660$ ^{2}mm and $A_{ROUT} = 0.924$ ^{2}mm which are obtained from McPAT [2] (Section 3.4).

$$DT = \max(DT_{MC}, DT_{ROUT}) \tag{5.3}$$

5.2.5 Bandwidth Capacity

The pin-fin structure not only affects cooling, but also the maximum bandwidth capacity of a micro-fluidic pin-fin heatsink. The bandwidth capacity is defined as the maximum TSV density supported by the heatsink. The maximum TSV density supported by a pin-fin heatsink with pin diameter D and pin pitch S is DP as

defined in Equation (5.4). The first two terms in the equation represent the cross sectional area of a pin divided by the total area between adjacent pins. Y is the TSV yield, which is the amount of pin area which can contain TSVs. In this work we assume $Y = 0.8$ due to the circular shape of pin fins which results in wasted area around the edge.

$$DP = \frac{\pi}{4}\frac{D^2}{S^2}Y \tag{5.4}$$

5.2.6 Pin Fin Thermal Model

The thermal model introduced in Section 3.8 was for a microchannel MF heatsink. In this study we use a different thermal model to model the pin-fin MF heatsink. The model was developed by our collaborators at Georgia Institute of Technology [113] with whom we preformed this study. The pin-fin MF heatsink model is explained in the remainder of this section.

The 3D stack is discretized into multiple control volumes, each modeling the temperature around one pin. Figure 3.4 shows the energy flows in a single control volume. Energy balance analysis is conducted for each control volume to evaluate the thermal map of the system.

Each control volume is assumed to have a uniform fluid temperature T_f and a uniform silicon temperature T_s. The energy equation for the solid components of a control volume is given in Equation (5.5), where q_{gen} is the energy generation rate

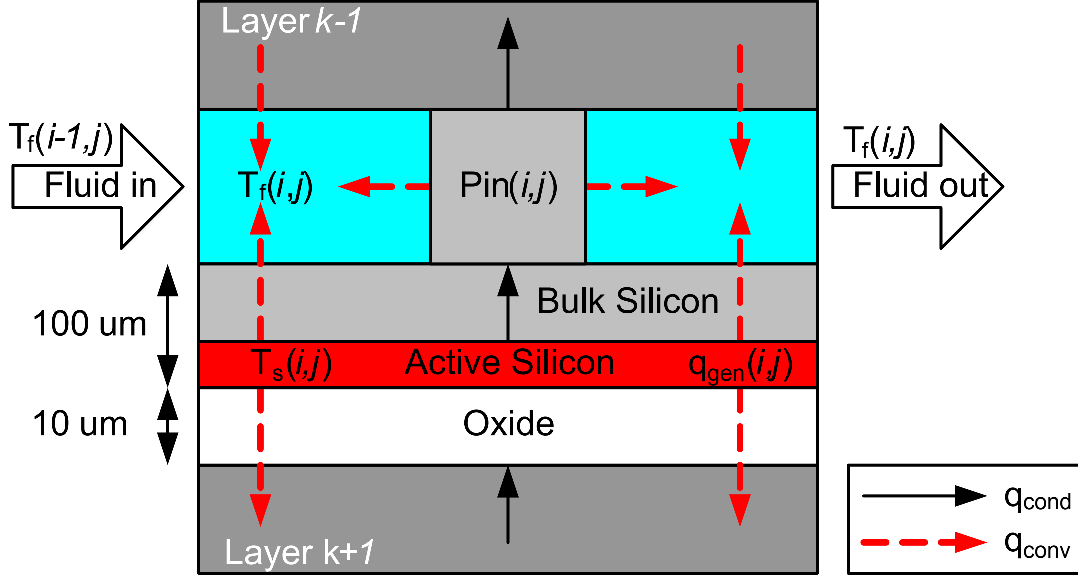


Figure 5.8: Control volume around one pin

obtained from the power map, q_{cond} is the heat conduction from neighboring control volumes and q_{conv} is the heat transferred by convection between the solid and the fluid.

$$q_{gen} = q_{cond} + q_{conv} \tag{5.5}$$

The energy balance equation for the fluid is given in Equation (5.6), where $\dot{m}$ is the mass flow rate, C_p is the specific heat capacity of the fluid, and $T_f(i-1,j)$ is the fluid temperature of the upstream neighbor control volume.

$$q_{conv} = \dot{m}C_p\left(T_f(i,j) - T_f(i-1,j)\right) \tag{5.6}$$

A system of equations is obtained by applying energy balance analysis to each control volume, and the system is solved simultaneously. Heat convection terms are defined using fluid heat transfer coefficient h_f, which is given in Equation (5.7), where Nu is Nusselt number which we estimate using the equations in [113], and k_f is the thermal conductivity of the fluid.

$$h_f = Nuk_f D \tag{5.7}$$

In this study the fluid is assumed to be water. Table 5.3 gives a list of parameter values used in the thermal model. Some parameters are temperature dependent, so their default value (calculated at 25 °C) is given in the table, and temperature dependent scaling factors from [117] are applied within the model. Heat conduction from the chip stack into the environment is modeled as a heat transfer coefficient between the ambient temperature and the top and bottom of the chip stack.

Table 5.3: Micro-fluidic pin-fin thermal model parameters

Variable	Value	Unit	Description
T_{amb}	40	°C	Ambient temperature
T_{fin}	25	°C	Fluid inlet temperature
h_{bot}	10	W m^{-2} K^{-1}	Heat transfer coefficient at layer n
h_{top}	562	W m^{-2} K^{-1}	Heat transfer coefficient at layer 1
k_{Si}	149	W m^{-1} K^{-1}	Thermal conductivity of silicon
k_{Ox}	1.4	W m^{-1} K^{-1}	Thermal conductivity of oxide
$\rho_f(25)$	1000	kg m^{-3}	Fluid density at 25 °C
$k_f(25)$	0.5573	W m^{-1} K^{-1}	Fluid thermal conductivity at 25 °C
$C_p(25)$	4200	J kg^{-1} K^{-1}	Fluid specific heat capacity at 25 °C
$\mu_f(25)$	1.53	mPa s	Fluid dynamic viscosity at 25 °C
Δp	1500	Pa	Pressure drop from inlet to outlet

5.2.7 Experimental Setup

In the following sections we discuss our experiment and results. First we discuss our methodology and characterize the design space (Section 5.2.8). Next we characterize the effect of pin-fin pitch S on the thermal and bandwidth feasibility of the design space. Finally we introduce two naïve schemes for choosing a heatsink design and compare them to our proposed co-design methodology for choosing the heatsink design that optimally balances thermal and bandwidth (*i.e.* inter-tier communication density) design constraints. We compare the feasibility region and maximum feasible performance and energy efficiency using the three heatsink design methodologies.

We exhaustively simulate all unique combinations of the architectural design variables in Table 5.4 using 12 parallel software workloads from the SPLASH-2 [84] and PARSEC [85] benchmark suites. For each architecture-benchmark pair we evaluate the performance (instructions per unit time) and power using the evaluation methodology from Chapter 3. For this study we use a fixed single-layer core floorplan topology. For a given architecture-benchmark pair, the performance is normalized to the performance of the baseline architecture (64-core, 32 MC, 3.6 GHz). Normalized performance is averaged across all benchmarks to yield a single performance number for each CPU architecture. Similarly, the dynamic and leakage power of each CPU component of a CPU design is averaged across all benchmarks yielding a single

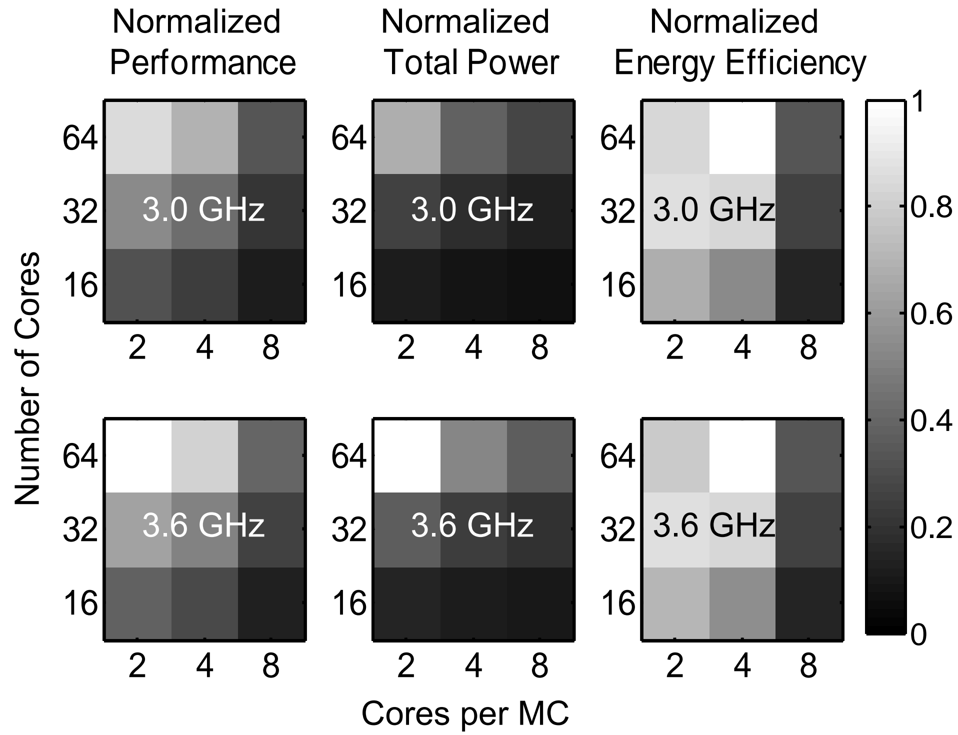


Figure 5.9: Normalized metrics of 3D CPU architectural design space

power map for each architectural design point. This power map is fed into the pin-fin thermal simulator (Section 5.2.6) to generate a unique thermal map and leakage power estimate for each heatsink design enumerated in Table 5.2.

Table 5.4: Study 4: Architectural Design Space

Cores	{16, 32, 64}
Clock Rate	{3.0, 3.6} GHz
Memory Controllers	{0.125, 0.25, 0.5} per Core

5.2.8 Architectural Parameter Sensitivity

The normalized performance, total power and energy efficiency of our CPU designs are shown in Figure 5.9[4]. As number of cores increases, both performance and power increase drastically, due to the highly parallel nature of the simulated workloads. Likewise as cores per MC decreases (*i.e.* number of MCs increases for a fixed number of cores) power and performance increase due to higher memory bandwidth and parallel memory access, leading to higher core utilization. These trends are more or less the same for both frequencies, with the higher frequency offering higher performance at the expense of higher power. We calculate the energy efficiency of each design point as $\frac{performance^2}{power}$ which is similar to the inverse of the energy-delay-product (EDP) metric.

5.2.9 Heatsink Parameter Sensitivity

Each cooling design has a unique cooling capacity and maximum bandwidth capacity. The cooling capacity is modeled using the pin-fin thermal model (Section 5.2.6) and the maximum BW capacity is modeled in Equation (5.4). Likewise each CPU architectural design has a unique bandwidth requirement as modeled in Equation (5.3). A heatsink-architecture pair is considered to be thermally feasible if the maximum temperature is less than $T_{violation} = 85°C$. A heatsink-architecture

[4] Total power and energy efficiency depend on leakage and micro-fluidic pumping power, which is a function of heatsink design. However the trends did not substantially change across heatsink designs, so only the data generated by our proposed co-design methodology is shown in the figure.

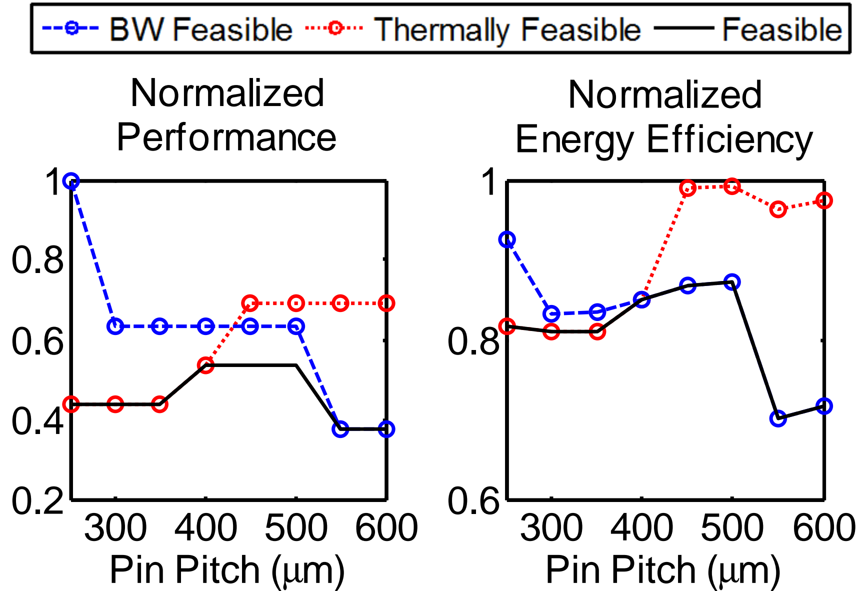


Figure 5.10: Maximum feasible performance and energy efficiency vs. pin pitch

pair is considered to be bandwidth feasible if the required TSV capacity is met by the heatsink (*i.e.* $DP \geq DT$). Only heatsink-architecture pairs that meet both feasibility constraints are considered as feasible design choices.

Figure 5.10 shows the maximum feasible performance and energy efficiency within the architectural design space as a function of the micro-fluidic heatsink pin pitch. We plot the maximum performance (energy efficiency) subject to BW and thermal constraints separately and then show the maximum performance (energy efficiency) subject to both constraints. We see that both metrics peak somewhere in between the maximum and minimum pin pitch where the optimal balance is struck between thermal and bandwidth feasibility regions.

In this study, the intersection of the thermal and bandwidth feasibility region-s is largest between 400 and 500 μm, thus unlocking more high performance and energy efficient 3D CPU architectures. Note that when different architectural pa-

rameters and physical parameters such as floorplan are considered, the optimal pin pitch value may change, but the fundamental trade-off between cooling and bandwidth as a function of pin pitch will remain and require co-design optimization.

5.2.10 Results

Finally, we analyze the architectural design space using three schemes for assigning a separate heatsink design to each architectural design point. The first two schemes are examples of naïve methods that might be used in absence of a comprehensive co-design methodology. These involve simply designing the heatsink independent of the logic architecture. Thus they apply the same heatsink parameters across the design space. The third scheme is our proposed co-design method, which designs a unique heatsink for each CPU architecture in order to maximize feasible performance or energy efficiency. The considered schemes are as follows:

1. **"Max Cooling"**: Choose a fixed heatsink design for all architectures that minimizes peak temperature.

2. **"Max BW"**: Chose a fixed heatsink design for all architectures that maximizes bandwidth capacity (*i.e.* pin density).

3. **"Co-design"**: Choose a separate heatsink design for each architecture that minimizes leakage power[5] while maintaining thermal and BW feasibility.

[5] We minimize leakage power to maximize energy efficiency since dynamic power and performance are not affected by heatsink design.

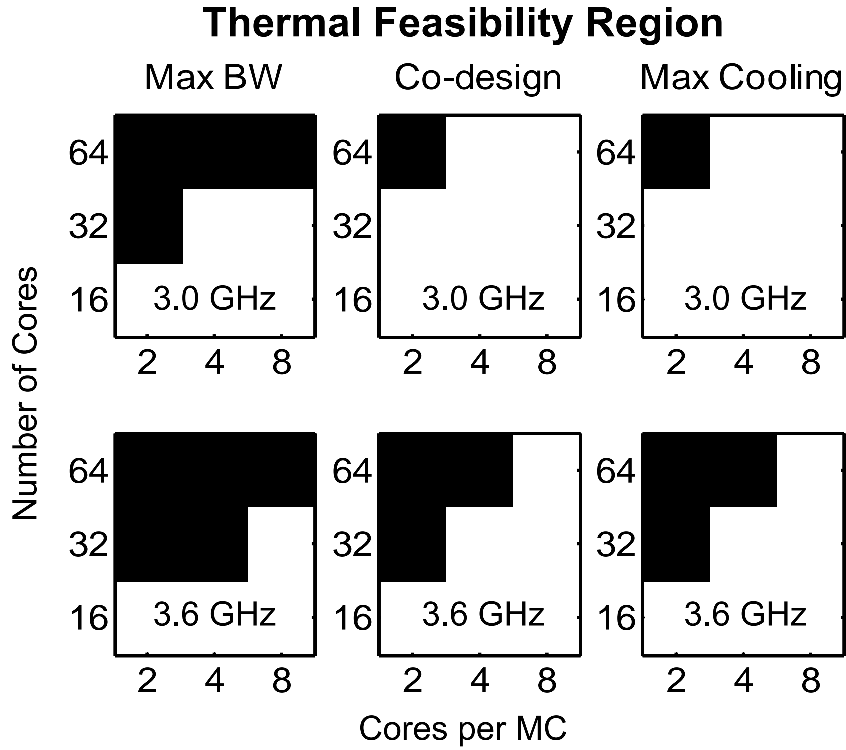


Figure 5.11: Thermal feasibility region (shown in white)

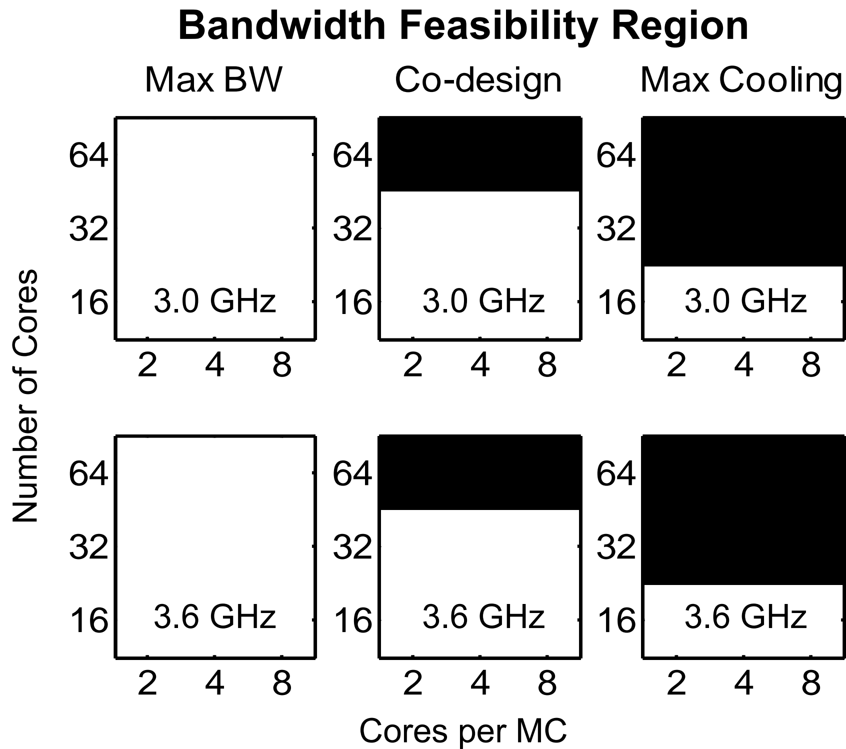


Figure 5.12: Bandwidth feasibility region (shown in white)

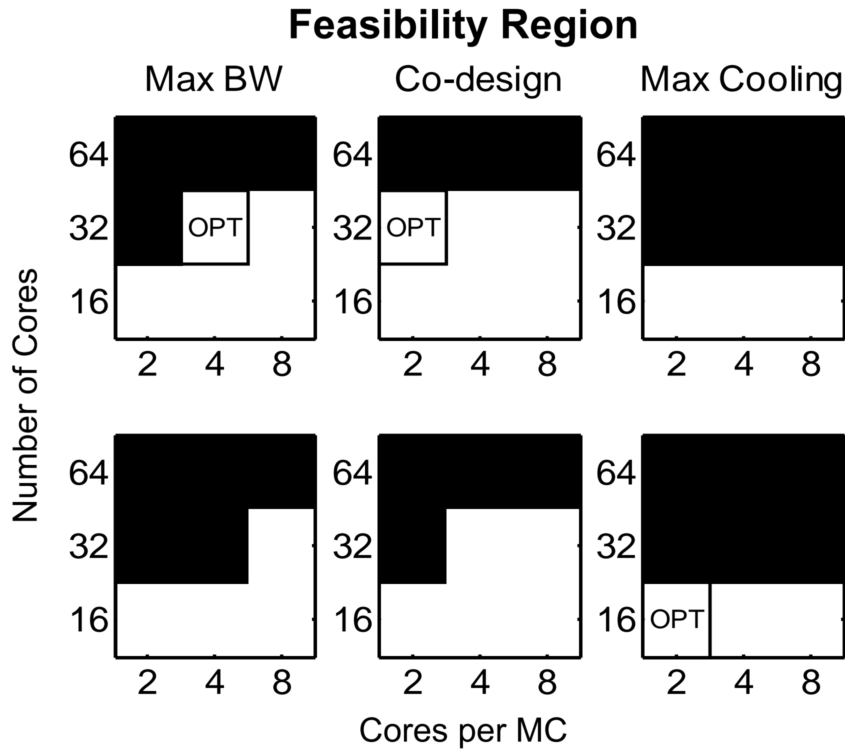


Figure 5.13: Thermal-bandwidth feasibility region (shown in white)

Figure 5.11 and 5.12 respectively show the thermal and bandwidth feasibility region of the architectural design space using the three schemes discussed above. We can observe that "Max BW" makes the entire design space bandwidth feasible, but offers a very small thermal feasibility region. Alternatively, "Max Cooling" offers a large thermal feasibility region but a very restrictive bandwidth feasibility region. "Co-design" is able to match the thermal feasibility of "Max Cooling" while drastically increasing the bandwidth feasibility region, leading to the largest overall feasibility region among the three schemes. Thus the "Co-design" scheme unlocks more high performance and energy efficient designs than the two naïve schemes. The optimal feasible architectural design under each heatsink design scheme is designated

as "OPT" in Figures 5.11 through 5.13. The optimal design is determined by cross-referencing the feasibility regions with the performance and energy efficiency results shown in Figure 5.9[6].

Table 5.5: Normalized Co-design Results

Metric	Max Cooling	Max BW	Co-design
Optimal Performance	0.70x	0.81x	1.00x
Optimal Energy Efficiency	0.82x	0.94x	1.00x
Optimal Number of Cores	16	32	32
Optimal Cores per MC	2	4	2
Optimal Frequency (GHz)	3.6	3.0	3.0
Chosen Pin Pitch (μm)	600	250	500

A comparison of the maximum feasible performance and energy efficiency of the architectural design space using the three heatsink design schemes is shown in Table 5.5. Numbers in this table have been normalized to "Co-design". The results show that co-design of 3D CPU architecture and micro-fluidic pin-fin heatsink can achieve significant improvements by optimally balancing the trade-off between TSV density and cooling capacity. The optimal design points are enumerated in the table, and illustrated in Figures 5.11 through 5.13.

We observe that "Max Cooling" in fact achieves the worst performance and energy efficiency because the TSV density is so restricted as to not allow core stacking (*i.e.* the number of cores was restricted to only 16, which is the maximum that can be accommodated on one layer). Although the additional cooling did facilitate higher frequency, it was not able to achieve good performance due to limits on core scaling.

[6] In our study the same design is optimal in both performance and efficiency, however it is certainly possible (even likely) that two different designs could have been optimal in the two different metrics if a different physical or architectural design space were considered.

Alternatively, “Max BW” was unable to accommodate sufficient MCs due to thermal constraints. “Co-design” chooses a heatsink pin-fin pitch in between the pitch chosen by the naïve schemes, thus providing sufficient cooling to accommodate many MCs and maintaining sufficient bandwidth to accommodate core stacking.

5.3 Summary

In this chapter we introduce the physical optimization algorithms discussed in Chapter 3 into our evaluation of the 3D CPU architectural design space. Section 5.1 introduces reliability constraints on top of thermal constraints and studies their effect on the feasibility region of the CPU design space at hand. The impact of different floorplan objective functions is reported and the conclusion is that all metrics of interest (in this case temperature and reliability) must be considered simultaneously during physical design to select the optimal feasible architectural design point. Furthermore the microchannel heatsink optimization technique from Section 3.10 is evaluated and shown to offer significant cooling improvements for a fixed pumping power, and blindly increasing pumping power with a uniform MF heatsink is shown to be inefficient.

Section 5.2 examines the trade-off between TSV bandwidth and cooling capacity which is inherent to MF heatsinks, especially pin-fin MF heatsinks. The optimal heatsink design will be a different for different architectural and physical CPU de-

signs with their unique cooling and TSV density requirements. We show that a simple fixed heatsink design focusing on maximizing either cooling or bandwidth will fail to realize the true potential of the design space at hand.

Chapter 6: Design Space Modeling for Physically Constrained 3D CPUs

Design space exploration (DSE) involves the evaluation of a multitude of design choices prior to detailed implementation. Such a technique is necessary to identify regions of interest in the design space and perform educated trade-off analysis of conflicting objectives. In its simplest form, DSE can be performed by exhaustively simulating the entire design space. However as CPU designs become ever more complex in the pursuit of Moore's law performance scaling, the DSE problem has become increasingly intractable as the design space grows combinatorially in the number of design parameters. Exhaustive simulation across such large design spaces is inefficient and potentially infeasible or unaffordable in terms of runtime.

Past work has attempted to overcome the computational infeasibility of exhaustive simulation in two ways. One is to reduce simulation time by orders of magnitude using techniques such as host-compiled simulation [118] or statistical simulation [119]. Although these approaches can make exhaustive simulation possible, the accuracy of such fast simulation techniques is reduced, and the applicability of the techniques is limited in scope. Another approach to the DSE problem is to

simulate only a small subset of the the full design space and use modeling techniques to predict the properties of un-simulated designs. Modeling approaches [120–123] have shown promising results on large architectural design spaces.

Vertical integration of circuits (3D ICs) moves the architectural design problem into uncharted territory where traditional domain knowledge and designer intuition may no longer apply. Moreover, past work [12, 29] has shown that 3D-CPU architectural design choices have a profound impact on physical properties such as power, area and temperature and significant portions of the 3D CPU design space can be infeasible due to physical constraint violations. 3D integration significantly complicates the DSE problem as follows:

- 3D integration brings many new architectural opportunities that significantly compound the intractability of exhaustive simulation.
- The effects of these new architectures on the design trade-off space are currently not well understood.
- 3D ICs are more thermally sensitive to architectural changes than equivalent 2D chips due to their physical structure [27, 29].
- 3D ICs can eliminate communication bottlenecks that are inherent in 2D ICs, making performance and power more sensitive to architectural changes [8].

- *Ad hoc* fixes late in the design cycle due to poor architectural design choices can be more costly in 3D ICs because of higher interconnectivity and density of circuit components and resource conflicts between transistors and vertical vias [30, 31, 114].

Physically aware DSE is becoming more important, especially in the context of 3D ICs. Past work [29, 103, 124] has examined the effect of physical constraints on a CPU design space, but has only done so with exhaustive simulation over a small design space. On the other hand, the literature on design space modeling [120–123] has only attempted to model optimization variables such as performance or energy efficiency with no consideration of physical constraints.

In this Chapter we introduce a modeling and simulation technique for 3D CPUs. The proposed technique models physical properties (*e.g.*, power, area and temperature) and traditional optimization metrics (*e.g.*, instructions per second or energy-delay-product). The technique uses these models to direct simulation effort towards user-defined regions of interest in the design space for the purpose of identifying interesting trends such as the Pareto optimal trade-off curve. Our models accurately predict the performance and temperature of a diverse 3D CPU design space and identify the optimal feasible design point (Pareto optimal design set) with 100% (98%) accuracy while simulating less than 2% (5%) of the design space.

This Chapter is laid out as follows. Section 6.1 gives a detailed overview of related work and Section 6.2 enumerates the contributions this work makes to the research effort. Section 6.3 introduces our modeling and simulation approach for identifying the design space region of interest to the designer and accurately estimating optimization metrics and physical properties while only simulating a small subsection of the space. Section 6.4 explains the experimental setup of our studies, and Section 6.5 presents the results which demonstrate the effectiveness and accuracy of our DSE modeling and simulation technique using two case studies. Finally, Section 6.6 concludes the chapter with a summary.

6.1 Previous Work

As the CPU design space has become increasingly large, exhaustive simulation has become computationally infeasible. Methodologies to facilitate large scale DSE have taken two orthogonal approaches: drastically reduce simulation time or produce models of un-simulated design points using simulation data from a small subset of the design space. The works by Genbrugge and Eeckhout [119] and Perelman *et al.* [125] attempt to significantly reduce simulation time with statistical simulation, which entails constructing a short code sequence that is representative of a full workload. Other work by Gandhi *et al.* [118] uses host-compiled simulation, which natively executes workloads that have been annotated with performance and power data generated offline using system models. Both techniques massively reduce simulation time, but at the cost of reduced accuracy and limited applicability.

Design space modeling likewise trades off accuracy for increased simulation time by omitting simulation of certain design points and instead estimating those points using modeling techniques. Historically, design space modeling techniques [120–123] have used uniform random sampling to build models of the entire design space. However there is a missed opportunity here. A significant advantage of modeling approaches is the ability to control the accuracy of the model in different regions of the design space, which we refer to as directed simulation. This is important because it is often the case that accuracy of the simulations is only important in a small subset of the design space, such as the Pareto front for the design objectives at hand, or the region of physically feasible design points. Directed simulation can improve the efficiency of a design space modeling technique by achieving sufficient model accuracy in the region of interest while using significantly less simulations as compared to random sampling.

Different modeling techniques have been proposed to accurately estimate the properties of a design space. Early work by Joseph *et al.* [123] used linear regression to model instructions per cycle (IPC) across a 23-variable CPU design space. However only two factors of each variable were considered, and the accuracy of the generated models was not reported. Later that year two similar works by Lee and Brooks [122] and İpek *et al.* [121] applied spline regression and artificial neural network models to similar problems, yielding average errors less than 10% and maximum error around 50%. More recent work by Jia *et al.* [120] applied spline regression to GPUs. This technique reduced maximum error to around 15% and had average error in the single-digit range.

Past work has had significant limitations. Most work has attempted only to build models of the design space and not to apply those models in an efficient manner to solve design space exploration problems of interest to a designer. Moreover no work until now has attempted to use modeling to estimate the physical feasibility region of the design space, or to provide a generic and systematic framework for solving a multitude of DSE problems involving discovery of a region of interest in the design space. Our proposed technique leverages the observation that it is inefficient to model the entire design space when only a small subset of the design space is physically feasible, or many of the design points represent low quality configurations that should be trimmed from the design space.

Finally, past work has only been applied to traditional computer architectures where a large amount of domain knowledge and intuition exists. 3D CPUs are a new frontier of computer architecture research and their design will rely much more heavily on statistical modeling than designer intuition. Moreover, physical constraints, especially thermal, are well known to be one of the primary limitations to the potential performance and efficiency of new 3D CPU architectures [15, 27]. Proper consider of physical feasibility constraints during DSE must be incorporated in order to properly design the 3D CPUs of the future.

6.2 Contributions

This work makes the following contributions:

- We propose a design space modeling and simulation technique that builds regression models to identify the region of the design space that is of interest to the designer and predict optimization metrics and physical properties within that region while only simulating a small subset of the space.

- To the best of our knowledge our work is the first to apply design space modeling techniques to 3D CPUs. 3D CPU design is expected to rely more on design space modeling than traditional CPU architectures due to a lack of designer experience and intuition regarding this emerging technology and architectural paradigm.

- To the best of our knowledge our work is the first to apply design space modeling to physical properties such as temperature to predict the feasibility region of a design space. This is extremely important for designing 3D CPUs which are known to be heavily thermally constrained [15, 29].

- Unlike past work, our proposed modeling and simulation methodology is expendable to any arbitrary design objective and associated metrics (*e.g.*, power, performance, area, timing, temperature) and is able to maximize the efficiency of optimization through directed simulation.

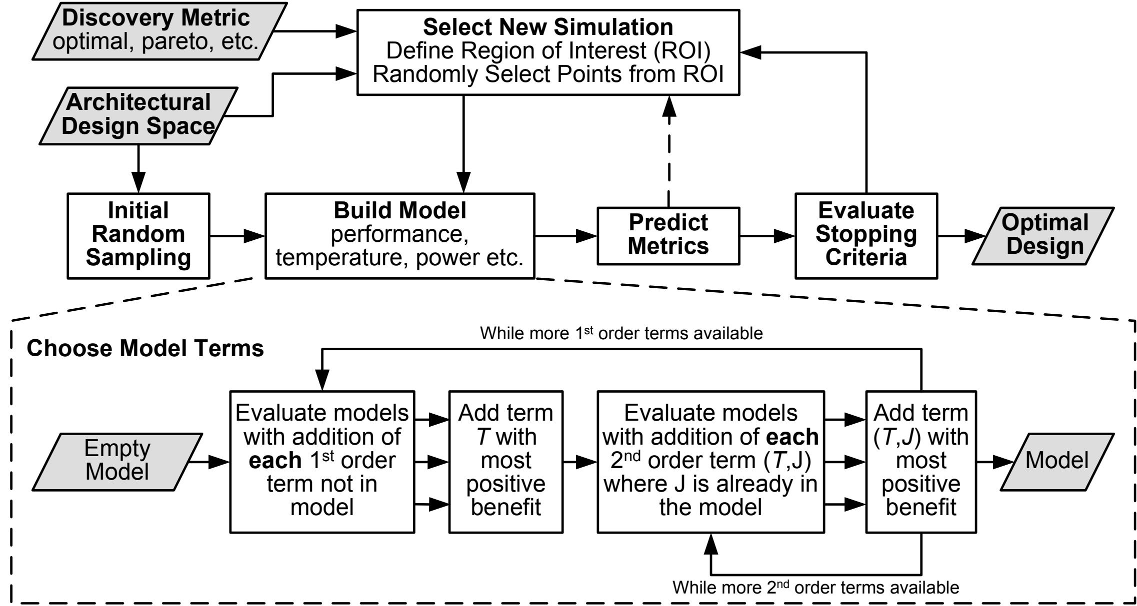


Figure 6.1: Modeling and simulation technique

6.3 Modeling and Simulation Technique

In this section we introduce our modeling and simulation technique for 3D CPU DSE subject to physical constraints. We use the smoothing spline analysis of variance (SS-ANOVA) [126] modeling technique to build models for each design parameters of interest (*e.g.*, performance, temperature and power) as a composition of cubic spline functions evaluated on combinations of design variables (*i.e.* model terms). First we give some background on SS-ANOVA modeling and then describe our technique for building models of the 3D CPU architectural design space with a limited number of simulations. Figure 6.1 illustrates the overall flow of our modeling and simulation technique, and details are given in the subsections below. The basic flow is an iterative back-and-forth between model building and choosing new simulation points based off the constructed model predictions.

6.3.1 SS-ANOVA Modeling

A spline is a piecewise polynomial function [126]. In this work we consider cubic splines, which are piecewise cubic functions. Splines are both differentiable and continuous at the piecewise boundaries which are called knots [126]. The smoothing spline is a technique to smooth noisy data by fitting a spline function to the data. Analysis of variance (ANOVA) is a statistical technique for analyzing the underlying source of variations in a population [126]. Multi-factor ANOVA can be used to generate models of an observed data set as a function of some underlying properties of each observation. An observation f can be modeled as a function of the variables $\mathbf{x} = x_1, x_2, \ldots, x_n$ as shown in Equation (6.1) [126]. SS-ANOVA limits the functions $\{f_1, \ldots, f_n, f_{1,2}, \ldots, f_{1,2,\ldots,n}\}$ to be spline functions which operate on some subset of the variables in $\mathbf{x}$. Each unique subset of input variables is called a term, and the order of a term is the number of members in the subset. c is the trivial function on the 0th order term (*i.e.* a scalar value).

$$f(\mathbf{x}) = c + \sum_{j=1}^{n} f_j(x_j) + \sum_{j=1}^{n} \sum_{k=j+1}^{n} f_{j,k}(x_j, x_k) + \ldots + f_{1,2,\ldots,n}(x_1, x_2, \ldots, x_n) \tag{6.1}$$

In this work we use the `gss` [127] package for the statistical computing environment `R` [128] to generate a unique smoothing spline model for each design property of interest. To generate each model, `gss` requires a set of simulation data and a set of model terms. However, choosing the appropriate simulation points and model

terms are nontrivial problems. The choice of model terms and simulations points strongly affects the quality of the model and suboptimal choices have a high cost in terms of total simulation time and model complexity. Our iterative technique for model term and simulation point selection and is explained in detail in the following subsections.

6.3.2 Choosing Model Terms

The maximum number of terms (*i.e.* unique subsets of all model variables) associated with n variables is 2^n. However as a rule of thumb a model is unreliable when the number of terms is greater than $s/20$ [129] where s is the number of simulated points. If too many model terms are used, the model can suffer from over-fitting, making it very accurate with respect to the observed data, but a poor predictor of the un-simulated data we wish to predict. Thus the number of model terms must be kept relatively small in order to maintain model accuracy when the number of simulations is small. The intended goal of the modeling and simulation approach is to build accurate models while requiring only a small number of simulations, so avoidance of the over-fitting problem is of critical importance.

The coefficient of determination (R^2) is a commonly used metric to evaluate how well a model fits the data [130]. However R^2 monotonically increases as new terms are added to a model [120]. Thus optimization of R^2 itself would inevitably lead to inclusion of all model terms, unnecessarily complicating the model and potentially causing over-fitting. Adjusted R^2 ($\bar{R}^2$) [131] (Equation (6.2)) scales R^2

relative to the number of model terms, m, and the number of data points, s. Thus if an additional model term is added that only marginally improves R^2, $\bar{R}^2$ will decrease, indicating that the added term has reduced the quality of the model. Separate models (using separate sets of model terms) are built for each design property of interest, so a separate $\bar{R}^2$ value is calculated for each model.

$$\bar{R}^2 = 1 - (1 - R^2)\frac{s-1}{s-m-1} \tag{6.2}$$

We use a forward selection $\bar{R}^2$ based technique to select the terms in the model. The model building technique is similar to the technique used in [120], and is shown in the bottom half of Figure 6.1. Starting with an empty model we consider each model consisting of one first order term. We evaluate the $\bar{R}^2$ metric for each model and accept the one with the largest value. We then consider adding each remaining first order term and accept the terms that increase the quality of the model by at least θ. Model terms are added in decreasing order of model improvement, and model improvement is reevaluated each time any term is added to the model.

Every time a new first order term is added to the model, we consider all second order interaction terms created by combining the new first order term with any other first order terms already in the model. Amongst all new second order terms generated this way we add any that cause the model quality to improve by at least θ. Second order terms are added to the model in a nested loop in decreasing order of model improvement. The model is complete once all first order terms have been added to the model, or when adding any new first order terms causes model

quality to improve less than θ. We limit our model to terms of order two and below, although the proposed model building approach could easily be extended to include terms of arbitrary order. High order interactions are seldom significant [126] so limiting the order of our model is expected to reduce the complexity of the model and the model building procedure without incurring significant losses in accuracy.

6.3.3 Adding Simulation Points

The designer defines a discovery metric, which determines the point(s) in the design space they are interested in accurately identifying. Some examples of potential discovery metrics are the optimal design point subject to a set of constraints (*e.g.*, design space optimization), or the set of Pareto optimal designs (*e.g.*, trade-off analysis). The optimality metric (*e.g.*, performance or energy efficiency), constraints (*e.g.*, temperature, power, area or timing) and Pareto metrics (*e.g.*, temperature-performance trade-off curve) are defined by the designer. The goal of our proposed modeling and simulation technique is to identify these points by iteratively predicting them and concentrating simulator effort around the predicted point(s) to improve the accuracy of the prediction.

Initial models are built using a random sampling of η simulation points from the design space. Using the model predictions[1], the predicted design point(s) of interest are identified. However, due to model error, the identified point(s) are not necessarily the true points of interest. Luckily, the true points of interest are likely

[1] Design points that have already been simulated use real simulation metrics rather than predictions from models to improve accuracy of the method.

to be close to the predicted points of interest. Thus a region of interest (ROI) is defined which contains the design points which are close to the predicted point(s) of interest, and additional simulation effort is concentrated towards this ROI to improve model fidelity in that region. The ROI is defined as the design points close to the predicted point(s) of interest, however the concrete definition of closeness will necessarily be a function of the discovery metric. Section 6.4 introduces the specific discovery metrics and associated ROI definitions used for the case studies presented in this chapter.

Each iteration of the flow identifies χ new design points from the predicted ROI and queues them for simulation. Once the simulations are performed, the model is rebuilt and the process repeats. If the initial model mispredicts the ROI, additional simulation effort in the mispredicted region will reduce model residuals in that region and cause the newly predicted ROI to move away from its original mispredicted region towards the true ROI. Thus as the modeling and simulation flow iterates, predictions of the design point(s) of interest converge towards their true values. The process terminates when a defined stopping criteria has been met.

6.3.4 Stopping Criteria

Stopping criteria could involve reaching a maximum number of simulations, or a sustained convergence in predictions of ROI and/or point(s) of interest across multiple iterations. Since we are considering different discovery metrics with different definitions of point(s) of interest and ROI, we simply set the stopping criteria to

terminate when the total number of simulations reaches ζ. However we investigate the trade-off between number of simulations and optimality of our selected design space in Section 6.5, and the point at which prediction convergence is achieved can be observed *post hoc* in the results.

6.4 Experimental Setup

In this section we describe the experimental setup to evaluate the effectiveness of the modeling and simulation technique introduced in Section 6.3. In the following subsections we introduce the 3D CPU design space, the discovery metrics and associated ROI definitions considered in our case studies and the metrics we use to measure the success of our approach. Results are presented and discussed in Section 6.5.

6.4.1 Architectural Design Space

Our study searches the architectural design space in Table 6.1. Variables with values in brackets can take on any of the bracketed values, and the cross product of all variable values represents the complete design space. The architectural design space in Table 6.1 contains 4374 unique design points.

Table 6.1: Architectural design space (baseline architecture shown in bold).

Variable	Value(s)
Technology Node	32 nm
Number of cores (*core*)	{**8**, 16, 32}
Memory controllers	*core*{**1/2**, 1/4, 1/8}
Clock frequency	{**2.4**, 3.0} GHz
NOC width	128 bits
L2 cache size (per core)	{**256**, 512, 1024} kB
L2 cache associativity	{**4**, 8, 16}
L1 cache size (per core)	{**16**, 32, 64} kB
L1 cache associativity	1
Pipeline width	{**2**, 4, 6}
Branch predictor	Tournament
Local history table	1024 8-bit entries
Global predictor	4096 2-bit entries
BTB size	32 kB
BTB associativity	1
Reorder buffer length (*rob*)	{**96**, 128, 160}
Issue queue length	$0.4rob$
Load-store queue length	$0.5rob$
Fetch queue length	64
Int architectural registers	$0.67rob$
FP architectural registers	$0.33rob$
RAT size	*rob* 8-bit entries
DRAM size	4 GB
Cache line size	64 B
DRAM bus width	64 B

6.4.2 Software Benchmarks

Each architectural design point is evaluated using a set of software workloads from the SPLASH-2 [84] and PARSEC [85] benchmark suites. The performance of each design point is defined as the average normalized performance across all benchmarks and the maximum temperature for each design point is the maximum

Table 6.2: Simulated Workloads

SPLASH-2	PARSEC
`water-nsquared`	`blackscholes`
`fft`	`fluidanimate`
`radix`	`dedup`
	`swaptions`

temperature amongst all benchmarks. The specific benchmark programs used for this study are given in Table 6.2. The inputs and parameters used for each benchmark are the default settings recommended in the Multi2Sim documentation [82].

6.4.3 Discovery Metrics

The goal of our DSE study is to identify the design point(s) of interest as defined by the discovery metric chosen by the designer. Two discovery metrics are considered as case studies in this paper, but our proposed methodology is applicable to any arbitrary discovery metric. The discovery metrics considered here are:

- **"Optimal"**: design point with highest normalized performance subject to thermal constraint $temp_p < T_{violation}$.
- **"Pareto"**: Pareto optimal set of design points in thermal-performance space.

Thus the modeled design parameters are performance and temperature.

Each discovery metric defines an accompanying ROI of radius $\phi = (\phi_{perf},\ \phi_{temp})$. The ROI for the "Optimal" and "Pareto" discovery metrics are given in Equations (6.3) and (6.4)[2] respectively, where $perf_i$ and $temp_i$ are the performance and

[2]Pareto optimal points are the set of points such that no other point is better in all metrics of interest. Equation (6.4) presents a ϕ-relaxed definition of Pareto optimality that includes all points such that no other point is better by a degree of ϕ in all metrics of interest.

temperature of design point i and Ω is the design space. Design point p is the predicted optimal feasible point for the discovery metric "Optimal". The defined ROI is the set of points within distance ϕ of the identified point(s) of interest, and setting $\phi = (0\%,\ 0°C)$ causes the ROI to degenerate into a set containing only the identified point(s) themselves. The nominal thermal constraint is $T_{violation} = 85°C$, however the impact on our results due to reduced $T_{violation}$ is studied in Section 6.5.

$$ROI_{Optimal} = \{i \in \Omega \mid \left|\frac{perf_i - perf_p}{perf_p}\right| \leq \phi_{perf} \ \wedge \ |temp_i - temp_p| \leq \phi_{temp}\} \quad (6.3)$$

$$ROI_{Pareto} = \{i \in \Omega \mid \forall_{(j \neq i) \in \Omega}\, perf_j(1-\phi_{perf}) \leq perf_i \vee (temp_j + \phi_{temp}) \geq temp_i\} \quad (6.4)$$

6.4.4 Modeling and Simulation Parameters

The modeling and simulation technique introduced in Section 6.3 can be parametrized to make trade-offs between simulation time and optimality of the selected design point. In this study we use the following parameters:

- We sample $\eta = 40$ simulation points at random from the design space to build the initial model. The parameter η should be large enough to generate an initial model with reasonable accuracy in order to yield a reasonable approxi-

mation of ROI. However a large value of η would degrade the efficiency of the method as it degenerates towards random sampling. Letting $\eta = 40$ was found to be the smallest number of simulations that would allow the `gss` package to generate models without causing software errors, and larger values degraded efficiency.

- The threshold for accepting new model terms is $\bar{R}^2_{new} - \bar{R}^2_{current} > \theta = 0$. By increasing θ, the model complexity could be reduced at the expense of model quality.

- We use ROI radius of $\phi = (8\%,\ 4°C)$ when the discovery metric is "Optimal" and $\phi = (5\%,\ 3°C)$ when the discovery metric is the "Pareto". Larger values of ϕ prevent convergence to local minima, but generally increase the number of simulations. The values chosen were determined experimentally to make good tradeoffs between these two properties.

- We iteratively simulate chosen design points in increments of $\chi = 5$. Small values of χ increase the number of iterations and thus the number of times model building must be performed. Moreover the new model is unlikely to change much if χ is very small since only one or two new simulations does not significantly change the input to the model builder. However excessively large values of χ will spend too much simulation effort in the current estimation of ROI when potentially the prediction of ROI will change substantially after the next iteration. The value $\chi = 5$ was found experimentally to provide a good trade-off between these two concerns.

- We use a nominal stopping criteria of $\zeta = 200$ simulations. The trade-off of optimality vs. number of simulations is investigated in Section 6.5. The value $\zeta = 200$ represents nearly 5% of the total design space. Simulation of significantly more points would degrade the usefulness of the proposed method, whose intended goal is to only simulate a very small subset of the space. Moreover we find that our proposed method achieves very accurate results with less than 200 simulations.

6.4.5 Evaluation Metrics

The goal of the experiment is to identify the design point(s) defined by the discovery metric, while minimizing the total number of simulations performed. Thus the primary metrics used to evaluate the quality of our technique will be the accuracy of the identification, the number of simulations performed and the runtime overhead of the modeling technique. The accuracy of identification is defined as the distance of the identified point(s) from the actual point(s) of interest (which were obtained by exhaustive simulation solely for the purpose of evaluation).

When the discovery metric is "Optimal", the distance between the identified point and the true solution is quantified as optimality, which is the ratio $\frac{perf_p}{perf_o}$ where p is the predicted optimal feasible point and o is the true optimal feasible point (determined by exhaustive simulation).

When the discovery metric is "Pareto", the distance between the identified points and the true Pareto set is quantified as accuracy, which is the average Pareto optimality of the predicted Pareto set. The Pareto optimality of design point k is determined by finding the smallest value of ϕ such that k is included in the ROI. Specifically, the Pareto optimality of k is α_k and the smallest value of ϕ that includes k in the ROI is $\phi = (1 - \alpha_k)(100\%,\ 60°\text{C}$[3]$)$.

In general the optimality/accuracy of the predicted point(s) will increase as more simulations are performed, eventually degenerating into the exhaustive simulation . The net speedup of our technique consists of the reduction in total number of simulations minus the runtime overhead of building the models. However we will show in Section 6.5 that the modeling overhead is negligible compared to the reduction in necessary simulations due to application of our approach.

6.4.6 Comparison to Other Techniques

The rudimentary technique to which our technique could be compared is exhaustive simulation. However one can conceive of a less rigorous random sampling approach to DSE in which some portion of the solution space is sampled at random and the best design amongst the sampled designs is selected[4]. Additionally we could consider a less sophisticated modeling-only version of our proposed technique that uses SS-ANOVA model building to predict the design point(s) of interest, but simply uses random sampling to provide data to the model builder. The modeling-only

[3] 60 °C was roughly the thermal range of the design space considered in this work as shown in Figure 6.3.

[4] Exhaustive simulation is simply a degenerative case of random sampling where the simulated portion of the solution space is the entire space.

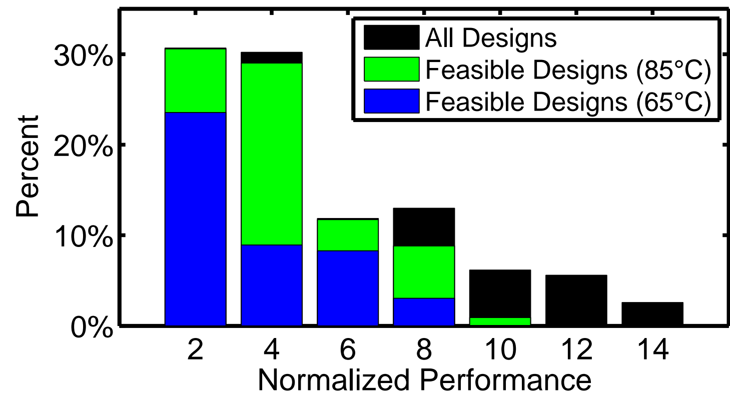
approach is representative of design space modeling techniques proposed in past work [120–123]. The advantage of a modeling-only technique is that it only requires models to be built once, but we will show that the time spent building models is insignificant compared to the savings in simulation time achieved by our proposed modeling and simulation technique. In Section 6.5 we compare the trade off curves of simulation count vs. quality for the three aforementioned techniques:

- **Proposed**: modeling and directed simulation
- **Modeling-Only**: modeling and random simulation (representative of past work [120–123])
- **Random Sampling**: no modeling and random simulation

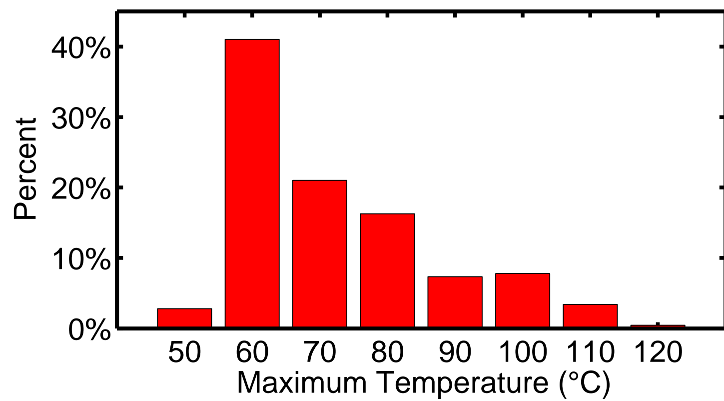
Since all techniques involve randomized sampling to some degree (*e.g.*, building the initial model in our proposed technique), experiments are replicated multiple times.

6.5 Results

In this section we describe the results of our experiments. First we provide some characterization of the design space explored in our study, and then we compare the quality of the different methodologies described in Section 6.4.6 for the "Optimal" and "Pareto" discovery metrics.



(a)



(b)

Figure 6.2: Distribution of (a) performance (b) temperature in design space

6.5.1 Design Space Characterization

We begin by examining the properties of the design space. Exhaustive simulation was performed for the purpose of evaluation, as the design points of interest must be identified before the quality of the considered techniques can be evaluated. Exhaustive simulation took weeks to perform using university servers, further motivating the strong need for techniques such as the one proposed in this paper in

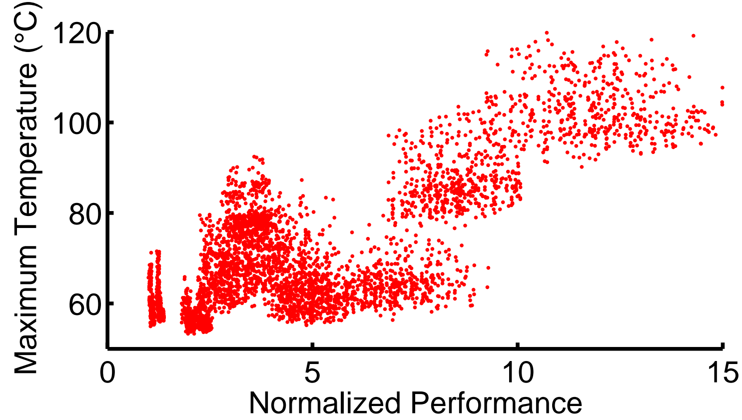


Figure 6.3: Temperature vs. performance of entire design space

order to reduce simulation time significantly below that of exhaustive design space simulation. We provide some statistics of the design space properties in order to give context for the results of this study.

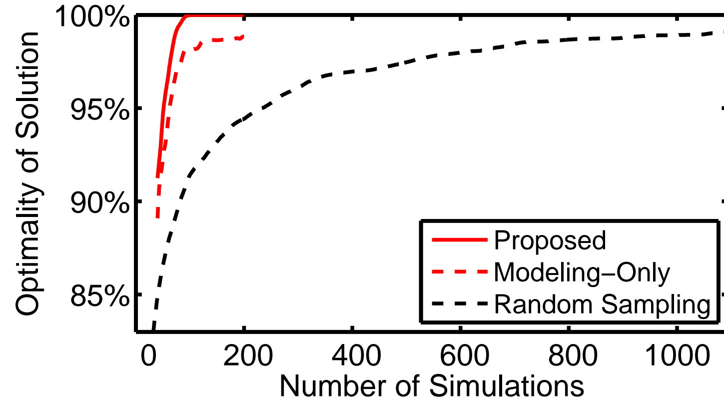
Figure 6.2(a) shows the distribution of normalized performance across all architectural design points. We can see that the design space is biased heavily towards the low-performance region. Furthermore thermal feasibility constraints bias the design space even further as the constraints tighten (*i.e.* $T_{violation}$ is reduced). This implies that random sampling is not a very good technique for discovering the "Optimal" design point since the probability of randomly sampling a high-performance thermally-feasible design point is low. The more biased the performance distribution is towards low-performance design points, the less effective random sampling will be for finding the "Optimal" design point, and the greater the need for directed simulation. Likewise Figure 6.2(b) shows the distribution of temperature. From this figure we can see how different values of $T_{violation}$ will affect the size of the feasibility region of the design space.

Figure 6.3 shows a scatter plot of the performance and temperature of each design point in the design space. We can see that identification of both the optimal feasible design point and the Pareto optimal design set without exhaustive simulation is non-trivial. The vast majority of design points in the design space are far from the point(s) of interest using either discovery metric. Moreover the correlation between performance and temperature is weak, motivating the need for independent models of each design property.

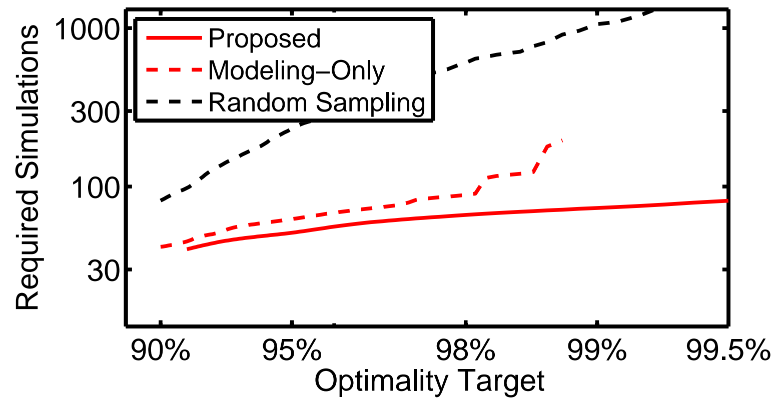
6.5.2 "Optimal" Discovery

There exists a fundamental trade-off between the number of simulations and the quality of the identified solution. We compare the random sampling and modeling-only technique to our proposed modeling and simulation technique and show that our technique is far better both in terms of quality of trade-off and the reliability of the approach. First we evaluate the techniques using the "Optimal" discovery metric.

Figure 6.4(a) tracks the optimality of the evaluated techniques as they iteratively add additional simulation points. We observe that modeling alone is a large contributor to the optimality of the identified point. With only 1% of the solution space sampled (roughly 40 points), the two modeling techniques can already identify a solution within 90% of the optimal, 38% closer to the optimal point than the prediction made by random sampling. However the true power of the proposed technique becomes clear as the number of samples increases. The random sam-



(a)



(b)

Figure 6.4: Optimality of identified design.

pling techniques, both with and without modeling, quickly improve the optimality of the predicted design as more simulation points are added, but then eventually flatten out as additional sampling is unable to significantly improve the quality of the prediction. However this diminishing returns phenomena is not observed in our proposed modeling and simulation technique. By using models to direct simulation effort on each iteration towards the ROI, the technique is able to make roughly linear

improvements to prediction accuracy for each additional simulation. Our proposed technique is able to identify the optimal feasible design point while simulating less than 2% (roughly 80 points) of the entire design space.

Figure 6.4(b) re-examines the data from the perspective of the number of simulations required to reach an optimality target. The data plotted here is on a log-log axis, meaning polynomial relationships will appear as a straight lines whose slope is proportional to the polynomial degree. An interesting result is that even if only 90% accuracy is required, the application of model building still reduces the total simulation time by roughly 2x compared to random sampling (saving over 100 simulation-hours in our study). This gap increases superlinearly as the optimality target increases. Furthermore as the optimality target tightens beyond 98% the slope of the trendline for the modeling-only technique significantly increases as the technique begins to degenerate into random sampling. On the other hand our proposed technique shows no such degeneration.

6.5.2.1 Robustness to Constraint Tightness

The previous results were evaluated at $T_{violation} = 85°C$. However as Figure 6.2(a) shows, reducing $T_{violation}$ significantly reduces the size of the thermal feasibility region. It is expected that this will reduce the quality of the random sampling technique significantly, but it is unclear how shrinking the feasibility region will affect the techniques that use model building. The fundamental question here is how the size of the feasibility region affects the quality of the different tech-

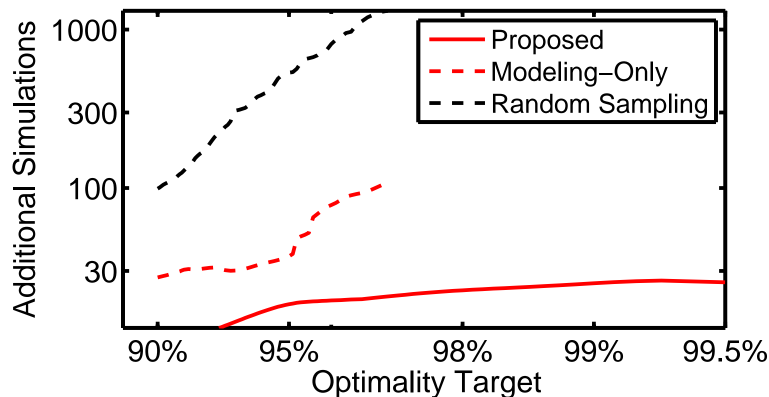
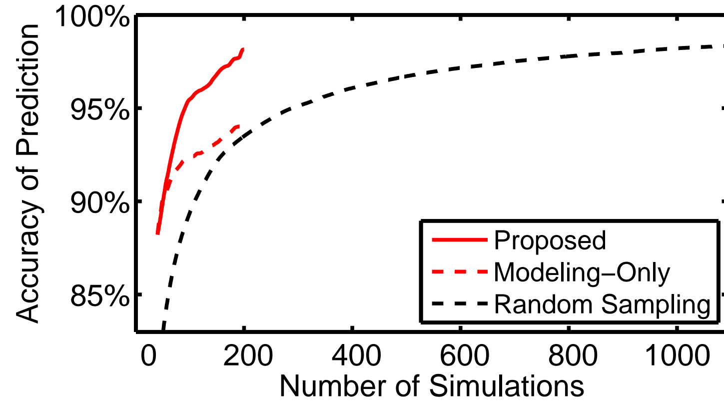


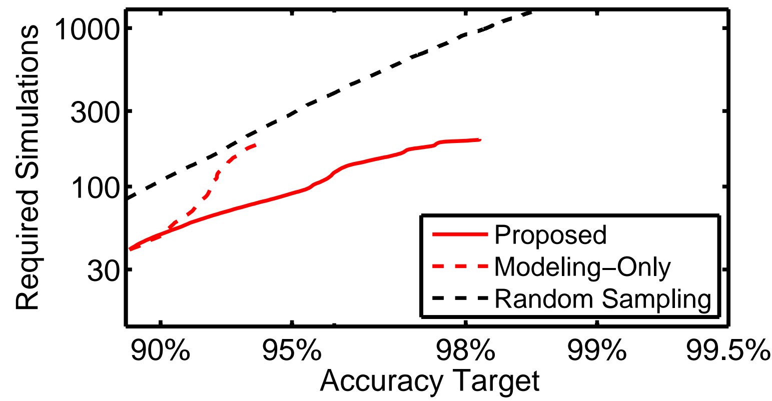
Figure 6.5: Additional simulations required when $T_{violation}$ is reduced from 85 °C to 65 °C.

niques. Although that question is investigated in this study by simply tightening the thermal constraints, it is logically equivalent to considering a lower-performance heatsink which would cause many design points to become thermally infeasible due to elevated temperatures. Moreover, heterogeneous integration in 3D ICs may introduce thermal constraints at substantially lower temperatures than those used for CMOS logic. Reduction in $T_{violation}$ was a simple way to consider the effect of design space constraints without requiring re-simulation of the entire solution space.

Figure 6.5 plots the number of additional simulations required when $T_{violation}$ changes from 85 °C to 65 °C. We notice that the number of additional simulations required for our proposed method is less than 30 (¡1% of the entire design space) and moreover, remains roughly constant as the optimality target is tightened. On the other hand random sampling and modeling-only both require superlinearly increasing amounts of additional simulations in order to meet optimality targets. Although model-building in and of itself does significantly reduce the amount of overhead compared to random sampling, the point at which additional simulation effort begins to



(a)



(b)

Figure 6.6: Accuracy of identified Pareto set.

show diminishing returns now occurs when optimality target reaches roughly 95%, reducing the scalability of this approach in heavily constrained design spaces. The conclusion is that our proposed technique is nearly independent of the size of the design space feasibility region due to the application of directed sampling, whereas techniques that use random sampling become less effective as the feasibility region shrinks.

6.5.3 "Pareto" Discovery

Figures 6.6(a) and 6.6(b) show the accuracy of the considered methods when the "Pareto" discovery metric is applied. Although the general trends and relative ordering of the method results are similar to the "Optimal" case, there are some significant differences. The most obvious difference is that the quality of both model-based techniques is reduced. Identification of a set of Pareto points is a more challenging problem and it makes sense that more simulation would be required to identify the true Pareto design set. However the relative improvement of our proposed technique vs. the modeling-only technique is substantially increased, indicating the increased need for directed simulation for more complex design space modeling and exploration problems such as identification of the Pareto design set.

Another interesting difference is that modeling-only is degenerating into random sampling much sooner than it did for the "Optimal" discovery metric. The conclusion here is that models built with random sampling can approximating a single design much better than the relative ordering of all design points. Directed simulation towards the ROI is of utmost importance for estimation of the Pareto design set, even for rather loose accuracy targets.

Finally we observe that random sampling has roughly the same trade-off curve whether predicting a single optimal feasible point or the entire Pareto optimal set. However the modeling-based approaches both perform significantly better for the "Optimal" discovery metric, which is the simpler problem[5]. This implies that ran-

[5]In fact the "Optimal" discovery metric problem is a sub-problem of the "Pareto" discovery metric problem, but with significantly reduced complexity.

dom sampling (and by extension exhaustive sampling) is failing to take advantage of the significantly different degrees of problem complexity to efficiently find a solution. Our technique is able to take advantage of the reduced complexity across all accuracy targets, and a modeling-only approach is able to take the same advantage when the accuracy target is low.

6.5.4 Overhead of modeling approach

There is obviously some runtime overhead for building the model in the proposed modeling approaches. We observed that the time consumed building models in our proposed approach was less than the time consumed to simulate a single design point ($<$ 0.025% of the design space). Figure 6.4(b) clearly shows that this overhead is negligible compared to the savings in number of required simulations compared to random sampling.

6.6 Summary

In this chapter we propose a modeling and simulation technique to apply the co-simulation and co-optimization techniques explored in the previous chapters to a large design space where exhaustive simulation of the architectural design space is not computationally feasible. We use smoothing spline ANOVA to build models of the metrics of interest across the entire design space using simulation data from only a small subset of the space. We iteratively build models and use these models to choose new simulations that will improve the accuracy of the model in the region

of interest to the designer, such as the optimal feasible design point or the Pareto optimal front. Our proposed methodology is applied to an eight-dimensional 3D CPU design space and tasked to discover the optimal feasible point and the Pareto optimal set of designs. Using less than 5% of the design space, we are able to identify both objectives with an accuracy of over 98%.

Chapter 7: Conclusions and Future Work

In Chapter 1 we introduce 3D integration as a promising new technology that promises to overcome some of the fundamental roadblocks to CPU performance scaling, such as interconnect power and delay dominance, the slowdown of economic incentives for technology scaling, and the physical fundamental limits of technology scaling due to quantum effects. We cite thermal and reliability concerns as first tier limitations to 3D IC technology, and discuss the fundamental interconnectedness of many metrics of interest and physical constraints in modern ICs. This interconnectedness is only exacerbated by 3D stacking and we introduce the co-design paradigm as a systematic methodology for addressing the simultaneous modeling and optimization of many design metrics and their interdependence on each other as well as design variables.

In Chapter 2 we explain 3D integration technology and provide more detailed analysis of the potential opportunities of 3D CPUs including massive memory bandwidth and highly connected on-chip inter-core communication networks. Such architectural advancements offer an opportunity to overcome the memory- and communication-wall. We detail the thermal and reliability concerns in 3D integration and introduce micro-fluidic cooling as a potential solution.

Chapter 3 introduces the co-simulation co-optimization flow used to evaluate a given architectural-physical design space throughout the many experiments presented in this dissertation. The flow models performance, power, timing, reliability and temperature. This chapter also introduces the physical optimization loops evaluated in Chapter 5 which can be driven by objective functions composed of arbitrary combinations of simulated design metrics.

Chapter 4 presents the results of two studies that quantitatively show the potential performance opportunities of stacked memory-on-logic CPUs and the associated need for micro-fluidic cooling. The first experiment finds that 3D stacking has the potential to improve performance significantly, but without proper cooling may actually reduce performance in order to meet thermal constraints. The second experiment explores the possibility of a return to a frequency scaling paradigm in parallel with the current core-scaling scheme in place today. This is made by the combination of high bandwidth architectures and micro-fluidic cooling.

In Chapter 5 we apply the physical optimization algorithms introduced in Chapter 3 and demonstrate the need for and advantages of simultaneous simulation and optimization of a multitude of design metrics, and the impact of their interdependence. We also introduce a new trade-off unique to MF cooled 3D ICs, which is between inter-layer via density (*i.e.* inter-layer bandwidth) and cooling capacity.

Finally Chapter 6 brings together the co-design simulation scheme and proposes a way to realistically apply it across a real-world design space where exhaustive simulation is not computationally feasible. We propose a modeling and simulation framework that is able to apply the co-design paradigm over a large design space

while only simulating a small subset of design points. Our method can discover the user-defined architectural regions of interest with over 98% accuracy while only requiring simulation of 5% of the design space.

7.1 Future Work

This dissertation significantly advances the emerging co-design paradigm, and represents a prototype of application of co-design in a holistic and comprehensive simulation and optimization framework. However, being an emerging design paradigm coupled with an exciting new technology, there are obviously many exciting avenues for future work in this field. Significant expansion of the scope of our work can be achieved by introducing models of heretofore un-modeled phenomena and improving (*e.g.*, adding granularity and inter-metric coupling) the existing models. Furthermore, an open research question how to efficiently model interaction relationships to best balance design time with quality. The extension of the co-design paradigm to low level detailed design will inevitable be introduced in future research, however our work sets the groundwork with a comprehensive high-level abstract implementation. Finally, our work investigates the application of the co-design paradigm to design-time decision making, but it can equally be applied to run-time management, and the interaction and simultaneous application of these two domains will certainly be the ultimate goal of the research effort begun in this dissertation.

7.1.1 Expansion of Co-Design Scope

The work presented in this dissertation has covered significant ground towards an implementation of the co-design paradigm. However it is by no means exhaustive. There are other significant interconnected design challenges and metrics that are not considered here, such as power delivery and signal integrity. In reality the co-design relation graph presented in Figure 1.2 is only a sub-graph of the true scope of the interconnected relationships involved in chip design. Due to the finite nature of compute resources and the need to find efficient trade-offs between design time and design quality, not every relationship can be considered in a real implementation of the co-design paradigm. However the decision of which relationships to model and which to ignore is domain specific, and as of yet there is no methodology in place to quantitatively decide how to construct the co-design simulation structure (*i.e.* to choose the sub-graph of the true global relationship graph to include in a co-design implementation). Development of such a methodology would be a significant contribution to be made by future work in this area, and would significantly advance the work towards industrial-scale applicability for arbitrary design problems.

In the following subsections we discuss two important design problems that are expected to limit the further advancement of 3D IC technology if the thermal and reliability concerns can be overcome. Modeling and optimization of these design problems would be a logical next step in expanding the scope of the proposed co-design framework put forth in this dissertation.

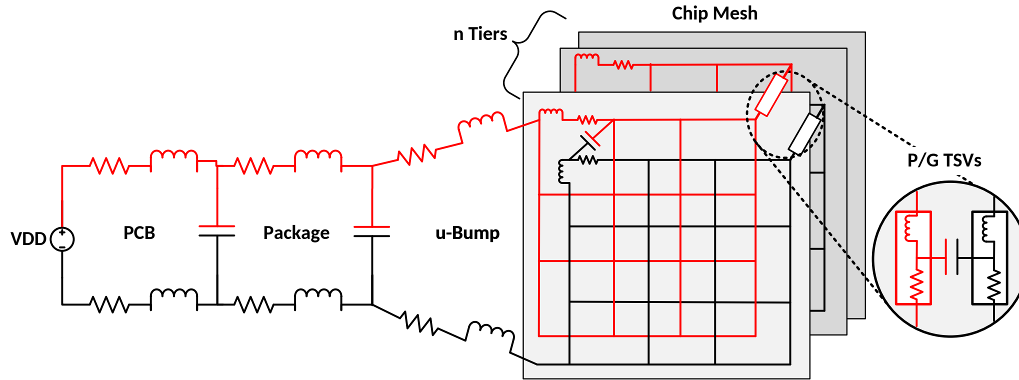


Figure 7.1: PDN model in a 3D IC

7.1.1.1 Power Delivery

In a 3D IC, power is delivered from off-chip package through C4 bumps and then distributed vertically through power TSVs. Figure 7.1 illustrates a 3D PDN circuit model, which consists three parts: PCB, Package and On-chip circuits. The on-chip circuit is modeled as a meshed RLC network capturing the voltage distribution in both vertical and planar directions.

The vertical structure of a 3D PDN brings several new challenges. First, as 3D integration enables stacking multiple functional layers vertically, power scales volumetrically with the product of footprint area and number of layers. However, the number of power delivery pins (*i.e.* the power delivery capacity) is a function of footprint area only. This imbalance between power supply and demand makes maintenance of high quality voltage rails a challenging problem. Second, the parasitics of power/ground TSVs affect the resonant frequency of each layer thus influencing the power noise characteristics in 3D ICs. As the current draw in 3D ICs has significant spacial variation, the PDN noise shows great variation spatially. Third,

the stacking structure of 3D ICs enables power noise from one layer to couple in neighboring layers. For example, when CPUs at different layers share the same PDN, one active CPU core can affect the voltage level of another core on a different layer. Fourth, in an air-cooled 3D IC, the heatsink and the power delivery pins are almost always on opposite ends of the chip stack. This means there is a trade off in that the chip layer with the most cooling capacity (*i.e.* closest to the heatsink) will also be the layer with the worst power integrity, and vice versa. This necessitates aggressive management and design methodologies considering both power delivery and temperature.

7.1.1.2 Signal Integrity

Another design challenge in 3D ICs is to ensure signal voltage noise is maintained within design margins. Cross coupling between switched devices can cause increased leakage/short circuit currents and possibly result in digital glitches that affect circuit behavior or cause incorrect computations. In addition to the traditional sources of coupling noise (wires and transistors), TSVs provide a new coupling source in 3D ICs. TSVs have the potential to be more problematic than planar wires since they are much larger, and surrounded by a much thinner insulation layer [20,21]. TSVs can easily couple into the conductive silicon substrate through the thin oxide liner around the TSV [23]. From there the voltage noise can couple into other TSVs or transistors through the conductive substrate.

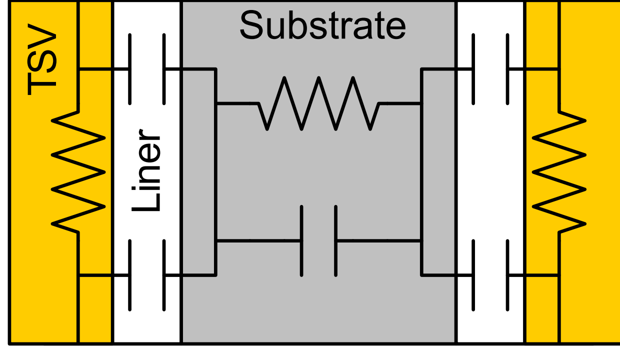


Figure 7.2: TSV-TSV coupling circuit model

Figure 7.2 shows a circuit model of coupling between two TSVs. TSV coupling is most strongly affected by liner capacitance which is independent of the distance between TSVs [23]. Thus, TSV coupling is not efficiently mitigated by increasing TSV pitch. Liu *et al.* [23] show that increasing TSV pitch from 1 μmto 20 μm(20x increase) only reduced TSV coupling from 255 mVto 225 mV(12% reduction).

We have done extensive work on modeling and reducing TSV-TSV coupling noise [20, 21, 132, 133], but this work is at this point outside the scope of this dissertation since it operates at the global placement layer of abstraction. However by applying the co-design paradigm to more fine-grained detailed physical design (Section 7.1.2) our past work on TSV coupling could be easily integrated into the co-design paradigm.

7.1.2 Fine-Grained Design and Integration

The work in this dissertation has attacked the co-design problem at a high level of abstraction. The architectural design knobs considered were macro-architectural parameters and the physical design space consisted of high-level abstract floorplan-

ning. A significant avenue for future work is to consider micro-architectural design variables and/or more detailed physical design such as global placement. A promising approach going forward would be to add a fine-grained co-design scheme as a hierarchical level under the high-level co-design flow presented in this dissertation. Thus this future work would be a direct vertical extension of the current work.

Fine grained co-design would fundamentally require fine-grained models at both the physical and architectural level. Although such models do exist for traditional 2D CPUs, to our knowledge no generally accepted low level models have been put forth for 3D CPUs, and this is an area of ongoing research. For this reason the current work in this dissertation has considered coarse-grained integration of either vertical stacking of traditional 2D CPU layouts, or folding of high level function blocks across layers. However theoretical and experimental work has shown that the true advantage of 3D integration comes when circuits are split across layers at a fine granularity [15, 134]. Development of fine-grained physical models for 3D CPU function blocks would be a significant contribution to the advancement of the co-design paradigm and would facilitate a hierarchical co-design approach to go all the way from architectural design space exploration to tape-out physical layouts.

7.1.3 Runtime Management

This dissertation has only considered a design-time solution space. However the co-design paradigm could equally be applied to the design of runtime management policies and algorithms. For example, traditional dynamic voltage and

frequency scaling considers only core performance and temperature (or power). But these policies also affect reliability, power integrity, DRAM refresh rate *etc.* The co-design principle tells us we should simultaneously consider all the effects of a given runtime policy or decision in order to choose the optimal operating conditions at any given point in time.

Similar to design-time architectural decisions, runtime architectural decisions such as turning on/off certain cores, memory controllers, regions of cache *etc.* can be made using the co-design paradigm. Such adaptive architectures will become necessary in the future due to the Dark Silicon effect [87]. Even micro-fluidic heatsinks can benefit from runtime control [79]. Although the placement and dimensions of fluid cavities are determined at design time, the fluid flow rate can be toggled, especially in conjunction with DVFS and task migration techniques, and micro-values can be designed to give runtime control of which cavities fluid is pumped through [135].

Runtime management is an orthogonal but not an independent means of chip co-design. The scope of runtime techniques available are inherently decided at design time, and the existence of adaptive control can allow co-design methodologies to target average rather than worst case design, opening up significant average performance improvements while still guaranteeing worst case viability.

Bibliography

[1] C. Serafy, Z. Yang, Y. Hu, A. Srivastava, and Y. Joshi. Thermo-electric co-design of 3d cpus and embedded micro-fluidic pin-fin heatsinks. *Design Test, IEEE*, PP(99):1–1, 2015.

[2] Sheng Li, Jung Ho Ahn, Richard D Strong, Jay B Brockman, Dean M Tullsen, and Norman P Jouppi. Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Microarchitecture, 2009. MICRO-42. 42nd Annual IEEE/ACM International Symposium on*, pages 469–480. IEEE, 2009.

[3] Benjamin Sutherland. No moore? a golden rule of microchips appears to be coming to an end. *The Economist*, 2013.

[4] Toshihiko Osada and Milt Godwin. International technology roadmap for semiconductors. 1999.

[5] Nir Magen, Avinoam Kolodny, Uri Weiser, and Nachum Shamir. Interconnect-power dissipation in a microprocessor. In *Proceedings of the 2004 International Workshop on System Level Interconnect Prediction*, SLIP '04, pages 7–13, New York, NY, USA, 2004. ACM.

[6] J Hennessy and D Patterson. Memory hierarchy design. *Computer Architecture: A Quantitative Approach*, pages 390–525, 2011.

[7] B. Feero and P.P. Pande. Performance evaluation for three-dimensional networks-on-chip. In *VLSI, 2007. ISVLSI '07. IEEE Computer Society Annual Symposium on*, pages 305–310, March 2007.

[8] C. Serafy, Bing Shi, A. Srivastava, and D. Yeung. High performance 3d stacked dram processor architectures with micro-fluidic cooling. In *3D Systems Integration Conference (3DIC), 2013 IEEE International*, pages 1–8, Oct 2013.

[9] Gabriel H. Loh. 3d-stacked memory architectures for multi-core processors. In *Proceedings of the 35th Annual International Symposium on Computer Architecture*, ISCA '08, pages 453–464, Washington, DC, USA, 2008. IEEE Computer Society.

[10] G.L. Loi, B. Agrawal, N. Srivastava, Sheng-Chih Lin, T. Sherwood, and K. Banerjee. A thermally-aware performance analysis of vertically integrated (3-d) processor-memory hierarchy. In *Design Automation Conference, 2006 43rd ACM/IEEE*, pages 991–996, 2006.

[11] S.H. Pugsley, J. Jestes, Huihui Zhang, R. Balasubramonian, V. Srinivasan, A. Buyuktosunoglu, A. Davis, and Feifei Li. Ndc: Analyzing the impact of 3d-stacked memory+logic devices on mapreduce workloads. In *Performance Analysis of Systems and Software (ISPASS), 2014 IEEE International Symposium on*, pages 190–200, March 2014.

[12] Caleb Serafy, Ankur Srivastava, and Donald Yeung. Unlocking the true potential of 3d cpus with micro-fluidic cooling. In *Proceedings of the 2014 International Symposium on Low Power Electronics and Design*, ISLPED '14, pages 323–326, New York, NY, USA, 2014. ACM.

[13] Feihui Li, Chrysostomos Nicopoulos, Thomas Richardson, Yuan Xie, Vijaykrishnan Narayanan, and Mahmut Kandemir. Design and management of 3d chip multiprocessors using network-in-memory. In *Proceedings of the 33rd Annual International Symposium on Computer Architecture*, ISCA '06, pages 130–141, Washington, DC, USA, 2006. IEEE Computer Society.

[14] Jie Meng, K. Kawakami, and A.K. Coskun. Optimizing energy efficiency of 3-d multicore systems with stacked dram under power and thermal constraints. In *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*, pages 648–655, 2012.

[15] Gabriel H. Loh, Yuan Xie, and Bryan Black. Processor design in 3d die-stacking technologies. *Micro, IEEE*, 27(3):31–48, May 2007.

[16] Yue Zhang, A. Dembla, Y. Joshi, and M.S. Bakir. 3d stacked microfluidic cooling for high-performance 3d ics. In *ECTC'12*, pages 1644–1650, May 2012.

[17] Tiantao Lu and Ankur Srivastava. Detailed electrical and reliability study of tapered tsvs. In *Physical Design for 3D Integrated Circuits*, pages 39–52. CRC Press, 2015.

[18] Tiantao Lu, Zhiyuan Yang, and Ankur Srivastava. Electromigration-aware placement for 3d-ics. In *Proceedings of the 2016 international symposium on Quality Electronic Design*. ACM, 2016.

[19] Jiwoo Pak, Mohit Pathak, Sung Kyu Lim, and David Z Pan. Modeling of electromigration in through-silicon-via based 3d ic. In *Electronic Components and Technology Conference (ECTC), 2011 IEEE 61st*, pages 1420–1427. IEEE, 2011.

[20] Caleb Serafy, Bing Shi, and Ankur Srivastava. A geometric approach to chip-scale TSV shield placement for the reduction of TSV coupling in 3d-ics. *Integration, the VLSI Journal*, (0):–, 2013.

[21] C. Serafy and A. Srivastava. Tsv replacement and shield insertion for tsv-tsv coupling reduction in 3-d global placement. *IEEE TCAD*, 34(4):554–562, April 2015.

[22] J. Cho, E. Song, K. Yoon, J.S. Pak, W. Kim, J. J. Lee, H. Lee, et al. Modeling and analysis of through-silicon via (tsv) noise coupling and suppression using a guard ring. *Components, Packaging and Manufacturing Technology, IEEE Trans. on.*

[23] Chang Liu, Taigon Song, Jonghyun Cho, Joohee Kim, Joungho Kim, and Sung Kyu Lim. Full-chip tsv-to-tsv coupling analysis and optimization in 3d ic. In *Proceedings of the 48th Design Automation Conference*, DAC '11, pages 783–788, New York, NY, USA, 2011. ACM.

[24] Taigon Song, Chang Liu, Yarui Peng, and Sung Kyu Lim. Full-chip multiple tsv-to-tsv coupling extraction and optimization in 3d ics. In *Proceedings of the 50th Annual Design Automation Conference*. ACM.

[25] Jun So Pak, Joohee Kim, Jonghyun Cho, Kiyeong Kim, Taigon Song, Seungyoung Ahn, Junho Lee, Hyungdong Lee, Kunwoo Park, and Joungho Kim. Pdn impedance modeling and analysis of 3d tsv ic by using proposed p/g tsv array model based on separated p/g tsv and chip-pdn models. *Components, Packaging and Manufacturing Technology, IEEE Transactions on*, 1(2):208–219, 2011.

[26] Runjie Zhang, Kaushik Mazumdar, Brett H. Meyer, Ke Wang, Kevin Skadron, and Mircea Stan. A cross-layer design exploration of charge-recycled power-delivery in many-layer 3d-ic. In *Proceedings of the 52Nd Annual Design Automation Conference*, DAC '15, pages 133:1–133:6, New York, NY, USA, 2015. ACM.

[27] C. Serafy, A. Bar-Cohen, A. Srivastava, and D. Yeung. Unlocking the true potential of 3-d cpus with microfluidic cooling. In *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, volume 24, pages 1515–1523, April 2016.

[28] C. Serafy, A. Srivastava, and D. Yeung. Continued frequency scaling in 3d ics through micro-fluidic cooling. In *Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), 2014 IEEE Intersociety Conference on*, pages 79–85, May 2014.

[29] Caleb Serafy, Ankur Srivastava, Avram Bar-Cohen, and Donald Yeung. Design space exploration of 3d cpus and micro-fluidic heatsinks with thermo-electrical-physical co-optimization. In *Proceedings of the ASME 2015 International Technical Conference and Exhibition on Packaging and Integration of Electronic and Photonic Microsystems*. ASME, 2015.

[30] Zhiyuan Yang and Ankur Srivastava. Co-placement for pin-fin based micro-fluidically cooled 3d ics. In *ASME 2015 International Technical Conference and Exhibition on Packaging and Integration of Electronic and Photonic Microsystems collocated with the ASME 2015 13th International Conference on Nanochannels, Microchannels, and Minichannels*, pages V001T09A036–V001T09A036. American Society of Mechanical Engineers, 2015.

[31] Zhiyuan Yang and Ankur Srivastava. Physical co-design for micro-fluidically cooled 3d ics. In *Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), 2016 IEEE Intersociety Conference on*. IEEE, 2016.

[32] Avram Bar-Cohen, Ankur Srivastava, and Bing Shi. Thermo-electrical co-design of three-dimensional integrated circuits: challenges and opportunities. *Computational Thermal Sciences: An International Journal*, 5(6), 2013.

[33] Mark T Bohr et al. Interconnect scaling-the real limiter to high performance ulsi. In *International Electron Devices Meeting*, pages 241–244. INSTITUTE OF ELECTRICAL & ELECTRONIC ENGINEERS, INC (IEEE), 1995.

[34] J.W. Joyner, P. Zarkesh-Ha, and J.D. Meindl. A stochastic global net-length distribution for a three-dimensional system-on-a-chip (3d-soc). In *ASIC/SOC Conference, 2001. Proceedings. 14th Annual IEEE International*, pages 147–151, 2001.

[35] Ralph HJM Otten and Robert K Brayton. Planning for performance. In *DAC*, DAC '98, pages 122–127, New York, NY, USA, 1998. ACM, ACM.

[36] Kuan H. Lu, Suk-Kyu Ryu, Qiu Zhao, Xuefeng Zhang, Jay Im, Rui Huang, and Paul S. Ho. Thermal stress induced delamination of through silicon vias in 3-d interconnects. In *Electron. Compon. and Tech. Conf. (ECTC), 2010 Proc. 60th*, pages 40 –45, June 2010.

[37] J Thomas Pawlowski. Hybrid memory cube (hmc). In *Hot Chips*, volume 23, 2011.

[38] Jung-Sik Kim, Chi Sung Oh, Hocheol Lee, Donghyuk Lee, Hyong Ryol Hwang, Sooman Hwang, Byongwook Na, Joungwook Moon, Jin-Guk Kim, Hanna Park, Jang-Woo Ryu, Kiwon Park, Sang Kyu Kang, So-Young Kim, Hoyoung Kim, Jong-Min Bang, Hyunyoon Cho, Minsoo Jang, Cheolmin Han, Jung-Bae Lee, Joo Sun Choi, and Young-Hyun Jun. A 1.2 v 12.8 gb/s 2 gb mobile wide-i/o dram with 4×128 i/os using tsv based stacking. *Solid-State Circuits, IEEE Journal of*, 47(1):107–116, Jan 2012.

[39] Dae Hyun Kim, K. Athikulwongse, M. Healy, M. Hossain, Moongon Jung, I. Khorosh, G. Kumar, Young-Joon Lee, D. Lewis, Tzu-Wei Lin, Chang Liu, S. Panth, M. Pathak, Minzhen Ren, Guanhao Shen, Taigon Song, Dong Hyuk Woo, Xin Zhao, Joungho Kim, Ho Choi, G. Loh, Hsien-Hsin Lee, and Sung Kyu Lim. 3d-maps: 3d massively parallel processor with stacked memory. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, pages 188–190, Feb 2012.

[40] Michael Gschwind. Blue gene/q: design for sustained multi-petaflop computing. In *Proceedings of the 26th ACM international conference on Supercomputing*, pages 245–246. ACM, 2012.

[41] Y Eckert, Nuwan Jayasena, and G Loh. Thermal feasibility of die-stacked processing in memory. In *Proceedings of the 2nd Workshop on Near-Data Processing*, 2014.

[42] Dongping Zhang, Nuwan Jayasena, Alexander Lyashevsky, Joseph L Greathouse, Lifan Xu, and Michael Ignatowski. Top-pim: throughput-oriented programmable processing in memory. In *Proceedings of the 23rd international symposium on High-performance parallel and distributed computing*, pages 85–98. ACM, 2014.

[43] V. F. Pavlidis and E. G. Friedman. 3-d topologies for networks-on-chip. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 15(10):1081–1090, Oct 2007.

[44] Bing Shi and Ankur Srivastava. Thermal stress aware 3d-ic statistical static timing analysis. In *Proceedings of the 23rd ACM international conference on Great lakes symposium on VLSI*, GLSVLSI '13, pages 281–286, New York, NY, USA, 2013. ACM.

[45] JEDEC. Wide i/o 2 (wideio2) (jesd229-2). August 2014.

[46] Joel Hruska. Beyond ddr4: The differences between wide i/o, hbm, and hybrid memory cube. *ExtremeTech [online]*, 2015.

[47] Xiaoxia Wu, Jian Li, Lixin Zhang, Evan Speight, Ram Rajamony, and Yuan Xie. Hybrid cache architecture with disparate memory technologies. In *Proceedings of the 36th Annual International Symposium on Computer Architecture*, ISCA '09, pages 34–45, New York, NY, USA, 2009. ACM.

[48] Chiachen Chou, Aamer Jaleel, and Moinuddin K. Qureshi. Cameo: A two-level memory organization with capacity of main memory and flexibility of hardware-managed cache. In *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-47, pages 1–12, Washington, DC, USA, 2014. IEEE Computer Society.

[49] Manjunath Shevgoor, Jung-Sik Kim, Niladrish Chatterjee, Rajeev Balasubramonian, Al Davis, and Aniruddha N Udipi. Quantifying the relationship between the power delivery network and architectural policies in a 3d-stacked memory device. In *Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 198–209. ACM, 2013.

[50] G.H. Loh. Extending the effectiveness of 3d-stacked dram caches with an adaptive multi-queue policy. In *Microarchitecture, 2009. MICRO-42. 42nd Annual IEEE/ACM International Symposium on*, pages 201–212, Dec 2009.

[51] Xiaowei Jiang, N. Madan, Li Zhao, M. Upton, R. Iyer, S. Makineni, D. Newell, D. Solihin, and R. Balasubramonian. Chop: Adaptive filter-based dram caching for cmp server platforms. In *High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*, pages 1–12, Jan 2010.

[52] Shekhar Borkar. Thousand core chips: a technology perspective. In *Proceedings of the 44th annual Design Automation Conference*, pages 746–749. ACM, 2007.

[53] Keren Bergman, Gilbert Hendry, Paul Hargrove, John Shalf, Bruce Jacob, K. Scott Hemmert, Arun Rodrigues, and David Resnick. Let there be light!: The future of memory systems is photonics and 3d stacking. In *Proceedings of the 2011 ACM SIGPLAN Workshop on Memory Systems Performance and Correctness*, MSPC '11, pages 43–48, New York, NY, USA, 2011. ACM.

[54] Syed Minhaj Hassan, Sudhakar Yalamanchili, and Saibal Mukhopadhyay. Near data processing: Impact and optimization of 3d memory system architecture on the uncore. In *2015 International Symposium on Memory Systems (Memsys 2015)*, October 2015.

[55] Stephen Jarvis, Steven Wright, and Simon D Hammond. *High Performance Computing Systems. Performance Modeling, Benchmarking and Simulation: 4th International Workshop, PMBS 2013, Denver, CO, USA, November 18, 2013. Revised Selected Papers*, volume 8551. Springer, 2014.

[56] M. Mirza-Aghatabar, S. Koohi, S. Hessabi, and M. Pedram. An empirical investigation of mesh and torus noc topologies under different routing algorithms and traffic models. In *Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on*, pages 19–26, Aug 2007.

[57] I. Savidis and E.G. Friedman. Closed-form expressions of 3-d via resistance, inductance, and capacitance. *Electron Devices, IEEE Transactions on*, 56(9):1873–1881, 2009.

[58] A.W. Topol, D.C.La Tulipe, L. Shi, D.J. Frank, K. Bernstein, S.E. Steen, A. Kumar, G.U. Singco, A.M. Young, K.W. Guarini, and M. Ieong. Three-dimensional integrated circuits. *IBM Journal of Research and Development*, 50(4.5):491–506, July 2006.

[59] Bing Shi, Ankur Srivastava, and Peng Wang. Non-uniform micro-channel design for stacked 3d-ics. In *Proceedings of the 48th Design Automation Conference*, DAC '11, pages 658–663, New York, NY, USA, 2011. ACM.

[60] M.S. Bakir, C. King, D. Sekar, H. Thacker, B. Dang, Gang Huang, A. Naeemi, and J.D. Meindl. 3d heterogeneous integrated systems: Liquid cooling, power delivery, and implementation. In *Custom Integrated Circuits Conference, 2008. CICC 2008. IEEE*, pages 663–670, 2008.

[61] Mrinmoy Ghosh and Hsien-Hsin S. Lee. Smart refresh: An enhanced memory controller design for reducing energy in conventional and 3d die-stacked drams. In *Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 40, pages 134–145, Washington, DC, USA, 2007. IEEE Computer Society.

[62] Bing Shi and Ankur Srivastava. Dynamic thermal management considering accurate temperature-leakage interdependency. *Cooling of Microelectronic and Nanoelectronic Equipment: Advances and Emerging Research*, page 43, 2014.

[63] Tiantao Lu and Ankur Srivastava. Electrical-thermal-reliability co-design for tsv-based 3d-ics. In *ASME 2015 International Technical Conference and Exhibition on Packaging and Integration of Electronic and Photonic Microsystems collocated with the ASME 2015 13th International Conference on Nanochannels, Microchannels, and Minichannels*, pages V001T09A037–V001T09A037. American Society of Mechanical Engineers, 2015.

[64] Jae-Seok Yang, Krit Athikulwongse, Young-Joon Lee, Sung Kyu Lim, and David Z. Pan. Tsv stress aware timing analysis with applications to 3d-ic layout optimization. In *Proceedings of the 47th Design Automation Conference*, DAC '10, pages 803–806, New York, NY, USA, 2010. ACM.

[65] T. Frank, S. Moreau, C. Chappaz, L. Arnaud, P. Leduc, A. Thuaire, and L. Anghel. Electromigration behavior of 3d-ic tsv interconnects. In *Electron. Compon. and Tech. Conf. (ECTC), 2012 IEEE 62nd*, pages 326 –330, 29 2012-June 1 2012.

[66] YC Tan, Cher Ming Tan, XW Zhang, Tai Chong Chai, and DQ Yu. Electromigration performance of through silicon via (tsv)–a modeling approach. *Microelectronics Reliability*, 50(9):1336–1340, 2010.

[67] Zhaohui Chen, Zhicheng Lv, Xuefang Wang, Yong Liu, and Sheng Liu. Modeling of electromigration of the through silicon via interconnects. In *Electronic Packaging Technology & High Density Packaging (ICEPT-HDP), 2010 11th International Conference on*, pages 1221–1225. IEEE, 2010.

[68] Cathal Cassidy, Jochen Kraft, Sara Carniello, Frederic Roger, Hajdin Ceric, Anderson Pires Singulani, Erasmus Langer, and Franz Schrank. Through silicon via reliability. *Device and Materials Reliability, IEEE Transactions on*, 12(2):285–295, 2012.

[69] T Frank, Stéphane Moreau, C Chappaz, Patrick Leduc, L Arnaud, Aurélie Thuaire, E Chery, F Lorut, L Anghel, and G Poupon. Reliability of tsv interconnects: Electromigration, thermal cycling, and impact on above metal level dielectric. *Microelectronics Reliability*, 53(1):17–29, 2013.

[70] P Kumar, I Dutta, and MS Bakir. Interfacial effects during thermal cycling of cu-filled through-silicon vias (tsv). *Journal of electronic materials*, 41(2):322–335, 2012.

[71] Chukwudi Okoro, John W Lau, Fardad Golshany, Klaus Hummler, and Yaw S Obeng. A detailed failure analysis examination of the effect of thermal cycling on cu tsv reliability. *Electron Devices, IEEE Transactions on*, 61(1):15–22, 2014.

[72] Juergen Auersperg, Dietmar Vogel, Ellen Auerswald, Sven Rzepka, and Bernd Michel. Nonlinear copper behavior of tsv for 3d-ic-integration and cracking risks during beol-built-up. In *Electronics Packaging Technology Conference (EPTC), 2011 IEEE 13th*, pages 29–33. IEEE, 2011.

[73] David Z Pan, Sung Kyu Lim, Krit Athikulwongse, Moongon Jung, Joydeep Mitra, Jiwoo Pak, Mohit Pathak, and Jae-seok Yang. Design for manufacturability and reliability for tsv-based 3d ics. In *Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific*, pages 750–755. IEEE, 2012.

[74] Zhen Zhang. Guideline to avoid cracking in 3d tsv design. In *Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), 2010 12th IEEE Intersociety Conference on*, pages 1–5. IEEE, 2010.

[75] Avram Bar-Cohen[1], Joseph J Maurer, and Jonathan G Felbinger. Darpas intra/interchip enhanced cooling (icecool) program. In *CS MANTECH Conference, May 13th-16th*, 2013.

[76] W. Yun, Jongpil Jung, Kyungsu Kang, and Chong-Min Kyung. Temperature-aware energy minimization of 3d-stacked l2 dram cache through dvfs. In *SoC Design Conference (ISOCC), 2012 International*, pages 475–478, Nov 2012.

[77] Bing Shi, Caleb Serafy, and Ankur Srivastava. Co-optimization of tsv assignment and micro-channel placement for 3d-ics. In *ACM Great Lakes Symposium on VLSI*.

[78] A.K. Coskun, J.L. Ayala, D. Atienza, and T.S. Rosing. Modeling and dynamic management of 3d multicore systems with liquid cooling. In *Very Large Scale Integration (VLSI-SoC), 2009 17th IFIP International Conference on*, pages 35–40, 2009.

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

[80] Bing Shi, Caleb Serafy, and Ankur Srivastava. Co-optimization of tsv assignment and micro-channel placement for 3d-ics. In *Proc. of the 23rd ACM Int. Conf. on Great Lakes Symp. on VLSI*, GLSVLSI '13, pages 337–338, New York, NY, USA, 2013. ACM.

[81] Y. F. Tsai, F. Wang, Y. Xie, N. Vijaykrishnan, and M. J. Irwin. Design space exploration for 3-d cache. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 16(4):444–455, April 2008.

[82] Rafael Ubal, Julio Sahuquillo, Salvador Petit, and Pedro Lopez. Multi2sim: A simulation framework to evaluate multicore-multithreaded processors. In *Computer Architecture and High Performance Computing, 2007. SBAC-PAD 2007. 19th International Symposium on*, pages 62–68, 2007.

[83] Premkishore Shivakumar and Norman P Jouppi. Cacti 3.0: An integrated cache timing, power, and area model. Technical report, Technical Report 2001/2, Compaq Computer Corporation, 2001.

[84] Steven Cameron Woo, Moriyoshi Ohara, Evan Torrie, Jaswinder Pal Singh, and Anoop Gupta. The splash-2 programs: Characterization and methodological considerations. In *Proceedings of the 22Nd Annual International Symposium on Computer Architecture*, volume 23 of *ISCA '95*, pages 24–36, New York, NY, USA, 1995. ACM, ACM.

[85] Christian Bienia, Sanjeev Kumar, Jaswinder Pal Singh, and Kai Li. The parsec benchmark suite: Characterization and architectural implications. In *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques*, PACT '08, pages 72–81, New York, NY, USA, 2008. ACM.

[86] Manu Awasthi, David W Nellans, Kshitij Sudan, Rajeev Balasubramonian, and Al Davis. Handling the problems and opportunities posed by multiple on-chip memory controllers. In *Proceedings of the 19th international conference on Parallel architectures and compilation techniques*, pages 319–330. ACM, 2010.

[87] Hadi Esmaeilzadeh, Emily Blem, Renee St Amant, Karthikeyan Sankaralingam, and Doug Burger. Dark silicon and the end of multicore scaling. In *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*, pages 365–376. IEEE, 2011.

[88] Wim Heirman, Souradip Sarkar, Trevor E. Carlson, Ibrahim Hur, and Lieven Eeckhout. Power-aware multi-core simulation for early design stage hardware/software co-optimization. In *Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques*, PACT '12, pages 3–12, New York, NY, USA, 2012. ACM.

[89] W. J. Song, S. Mukhopadhyay, and S. Yalamanchili. Managing performance-reliability tradeoffs in multicore processors. In *2015 IEEE International Reliability Physics Symposium*, pages 3C.1.1–3C.1.7, April 2015.

[90] Michael Moeng and Rami Melhem. Applying statistical machine learning to multicore voltage & frequency scaling. In *Proceedings of the 7th ACM International Conference on Computing Frontiers*, CF '10, pages 277–286, New York, NY, USA, 2010. ACM.

[91] Xiangyu Dong, Yuan Xie, N. Muralimanohar, and N.P. Jouppi. Simple but effective heterogeneous main memory with on-chip memory controller support. In *High Performance Computing, Networking, Storage and Analysis (SC), 2010 International Conference for*, pages 1–11, 2010.

[92] Ming-Yu Hsieh, Arun Rodrigues, Rolf Riesen, Kevin Thompson, and William Song. A framework for architecture-level power, area, and thermal simulation and its application to network-on-chip design exploration. *SIGMETRICS PER*, 38(4):63–68, March 2011.

[93] Hadi Esmaeilzadeh, Adrian Sampson, Luis Ceze, and Doug Burger. Architecture support for disciplined approximate programming. In *Proceedings of the Seventeenth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS XVII, pages 301–312, New York, NY, USA, 2012. ACM.

[94] Jingwen Leng, Tayler Hetherington, Ahmed ElTantawy, Syed Gilani, Nam Sung Kim, Tor M. Aamodt, and Vijay Janapa Reddi. Gpuwattch: Enabling energy optimizations in gpgpus. In *Proceedings of the 40th Annual International Symposium on Computer Architecture*, ISCA '13, pages 487–498, New York, NY, USA, 2013. ACM.

[95] R. Sheikh, J. Tuck, and E. Rotenberg. Control-flow decoupling: An approach for timely, non-speculative branching. *IEEE Transactions on Computers*, 64(8):2182–2203, Aug 2015.

[96] Y. Zhang, A. Dembla, and M. S. Bakir. Silicon micropin-fin heat sink with integrated tsvs for 3-d ics: Tradeoff analysis and experimental testing. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 3(11):1842–1850, Nov 2013.

[97] T. Frank, C. Chappaz, P. Leduc, L. Arnaud, F. Lorut, S. Moreau, A. Thuaire, R. El Farhane, and L. Anghel. Resistance increase due to electromigration induced depletion under tsv. In *Reliability Physics Symposium (IRPS), 2011 IEEE International*, pages 3F.4.1–3F.4.6, April 2011.

[98] Jason Cong and Guojie Luo. A 3D physical design flow based on Open Access. In *International Conference on Communications, Circuits and Systems*. IEEE, 2009.

[99] Tiantao Lu and Ankur Srivastava. Detailed electrical and reliability study of tapered tsvs. In *3D Systems Integration Conference (3DIC), 2013 IEEE International*, pages 1–7. IEEE, 2013.

[100] J.R. Black. Mass transport of aluminum by momentum exchange with conducting electrons. In *Reliability Physics Symposium*, pages 1 – 6, 2005.

[101] J. Pak, S. K. Lim, and D. Z. Pan. Electromigration-aware routing for 3d ics with stress-aware em modeling. In *2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 325–332, Nov 2012.

[102] Wei Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M.R. Stan. Hotspot: a compact thermal modeling methodology for early-stage vlsi design. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 14(5):501–513, 2006.

[103] Caleb Serafy, Tiantao Lu, and Ankur Srivastava. Thermal-reliability physical co-optimization during architectural design space exploration of 3d-cpus. In *GOMACTech*, 2016.

[104] Jai-Ming Lin and Yao-Wen Chang. Tcg: a transitive closure graph-based representation for non-slicing floorplans. In *Design Automation Conference, 2001. Proceedings*, pages 764–769, 2001.

[105] Jason Cong, Jie Wei, and Yan Zhang. A thermal-driven floorplanning algorithm for 3d ics. In *ICCAD'04*, pages 306–313. IEEE, 2004.

[106] Jill HY Law, Evangeline FY Young, and Royce LS Ching. Block alignment in 3d floorplan using layered tcg. In *GLSVLSI'06*, pages 376–380. ACM, 2006.

[107] A. Ortega, S. Ramanathan, J. D. Chicci, and J. L. Prince. Thermal wake models for forced air cooling of electronic components. In *Semiconductor Thermal Measurement and Management Symposium, 1993. SEMI-THERM IX., Ninth Annual IEEE*, pages 63–74, Feb 1993.

[108] A. Kagi, J. R. Goodman, and D. Burger. Memory bandwidth limitations of future microprocessors. In *Computer Architecture, 1996 23rd Annual International Symposium on*, pages 78–78, May 1996.

[109] Jaehyuk Huh, D. Burger, and S. W. Keckler. Exploring the design space of future cmps. In *Parallel Architectures and Compilation Techniques, 2001. Proceedings. 2001 International Conference on*, pages 199–210, 2001.

[110] Rajkumar Buyya, Christian Vecchiola, and S Thamarai Selvi. *Mastering cloud computing: foundations and applications programming*. Newnes, 2013.

[111] Joel Hruska. The death of cpu scaling: From one core to many–and why were still stuck. *ExtremeTech [online]*, 2012.

[112] Tiantao Lu and Ankur Srivastava. Gated low-power clock tree synthesis for 3d-ics. In *Proceedings of the 2014 International Symposium on Low Power Electronics and Design*, ISLPED '14, pages 319–322, New York, NY, USA, 2014. ACM.

[113] Zhimin Wan, He Xiao, Yogendra Joshi, and Sudhakar Yalamanchili. Co-design of multicore architectures and microfluidic cooling for 3d stacked ics. *Microelectronics Journal*, 2014.

[114] Dae Hyun Kim, Krit Athikulwongse, and Sung Kyu Lim. A study of through-silicon-via impact on the 3d stacked ic layout. In *Proceedings of the 2009 International Conference on Computer-Aided Design*, ICCAD '09, pages 674–680, New York, NY, USA, 2009. ACM.

[115] B. A. Jasperson, Y. Jeon, K. T. Turner, F. E. Pfefferkorn, and W. Qu. Comparison of micro-pin-fin and microchannel heat sinks considering thermal-hydraulic performance and manufacturability. *IEEE Transactions on Components and Packaging Technologies*, 33(1):148–160, March 2010.

[116] Yoav Peles, Ali Koar, Chandan Mishra, Chih-Jung Kuo, and Brandon Schneider. Forced convective heat transfer across a pin fin micro heat sink. *International Journal of Heat and Mass Transfer*, 48(17):3615 – 3627, 2005.

[117] Frank P Incropera. *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

[118] Darshan Gandhi, Andreas Gerstlauer, and Lidiya John. Fastspot: Host-compiled thermal estimation for early design space exploration. In *Quality Electronic Design (ISQED), 2014 15th International Symposium on*, pages 625–632. IEEE, 2014.

[119] Davy Genbrugge and Lieven Eeckhout. Chip multiprocessor design space exploration through statistical simulation. *Computers, IEEE Transactions on*, 58(12):1668–1681, 2009.

[120] Wenhao Jia, Kelly Shaw, Margaret Martonosi, et al. Stargazer: Automated regression-based gpu design space exploration. In *Performance Analysis of Systems and Software (ISPASS), 2012 IEEE International Symposium on*, pages 2–13. IEEE, 2012.

[121] Engin İpek, Sally A McKee, Rich Caruana, Bronis R de Supinski, and Martin Schulz. *Efficiently exploring architectural design spaces via predictive modeling*, volume 40. ACM, 2006.

[122] Benjamin C Lee and David M Brooks. Accurate and efficient regression modeling for microarchitectural performance and power prediction. In *ACM SIGPLAN Notices*, volume 41, pages 185–194. ACM, 2006.

[123] PJ Joseph, Kapil Vaswani, and Matthew J Thazhuthaveetil. Construction and use of linear regression models for processor performance analysis. In *High-Performance Computer Architecture, 2006. The Twelfth International Symposium on*, pages 99–108. IEEE, 2006.

[124] Yingmin Li, Benjamin Lee, David Brooks, Zhigang Hu, and Kevin Skadron. Cmp design space exploration subject to physical constraints. In *High-Performance Computer Architecture, 2006. The Twelfth International Symposium on*, pages 17–28. IEEE, 2006.

[125] Erez Perelman, Greg Hamerly, Michael Van Biesbrouck, Timothy Sherwood, and Brad Calder. Using simpoint for accurate and efficient simulation. In *Proceedings of the 2003 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '03, pages 318–319, New York, NY, USA, 2003. ACM.

[126] Chong Gu. *Smoothing spline ANOVA models*, volume 297. Springer Science & Business Media, 2013.

[127] Chong Gu. Smoothing spline anova models: R package gss. *Journal of Statistical Software*, 58(5):1–25, 2014.

[128] Brian D Ripley. The r project in statistical computing. *MSOR Connections*, 1(1):23–25, 2001.

[129] Frank E Harrell. *Regression modeling strategies: with applications to linear models, logistic regression, and survival analysis*. Springer Science & Business Media, 2013.

[130] Michael H Kutner, Chris Nachtsheim, and John Neter. *Applied linear regression models*. McGraw-Hill/Irwin, 2004.

[131] Henry Theil. Economic forecasts and policy. 1958.

[132] Caleb Serafy, Bing Shi, and Ankur Srivastava. Geometric approach to chip-scale tsv shield placement for the reduction of tsv coupling in 3d-ics. In *Proceedings of the 23rd ACM international conference on Great lakes symposium on VLSI*, GLSVLSI '13, pages 275–280, New York, NY, USA, 2013. ACM.

[133] Caleb Serafy and Ankur Srivastava. Coupling-aware Force Driven Placement of TSVs and Shields in 3D-IC Layouts. In *International Symposium on Physical Design*. ACM, 2014.

[134] Moongon Jung, Taigon Song, Yang Wan, Yarui Peng, and Sung Kyu Lim. On enhancing power benefits in 3d ics: Block folding and bonding styles perspective. In *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE*, pages 1–6, June 2014.

[135] Terry J Dishongh, Jason T Cassezza, and Kevin S Rhodes. Microfluidic cooling of integrated circuits, January 26 2010. US Patent 7,652,372.