BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Dissertation

CROSS-LAYER DESIGN OF THERMALLY-AWARE 2.5D SYSTEMS

by

YENAI MA

B.S., University of Alberta, Canada, 2014
M.S., Boston University, 2019

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2020

Approved by

First Reader

Ajay J. Joshi, PhD
Associate Professor of Electrical and Computer Engineering

Second Reader

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

Third Reader

Milos A. Popovic, PhD
Associate Professor of Electrical and Computer Engineering

Fourth Reader

Wenchao Li, PhD
Assistant Professor of Electrical and Computer Engineering
Assistant Professor of Systems Engineering

Acknowledgments

First of all, I would like to express my sincere gratitude to my PhD advisor, Prof. Ajay Joshi, for his dedicated support and guidance throughout my PhD study. Prof. Joshi continuously provided encouragement and was always willing and enthusiastic to assist in any way he could on both research and my future career.

I would particularly like to thank Prof. Ayse Coskun for her valuable advice on thermal and EDA related aspects. Discussions with her have always been enlightening and productive. From her, I learned to always keep an eye on the big picture. In addition, I would like to thank the rest of my thesis committee members, Prof. Milos Popovic and Prof. Wenchao Li, and my defense chair Prof. Michel Kinsy, for their precious time, generous support, and insightful feedback.

Many thanks to my collaborators and co-authors for productive collaborations and all the helpful discussions: Prof. Andrew B. Kahng, Dr. Vaishnav Srinivas, Anjun Gu, and John Recchio at University of California San Diego, Prof. Jose L. Abellan at Catholic University of Murcia, Prof. David Kaeli, Dr. Amir K. Ziabari, Yifan Sun, Dana Schaa, and Rafael Ubal at Northeastern University, Prof. John Kim at Korea Advanced Institute of Science & Technology, Prof. Jonathan Klamkin and Warren Jin at University of California Santa Barbara, Dr. Tiansheng Zhang, Furkan Eris, Saiful Mojumder, Aditya Narayan, and Leila Delshadtehrani at Boston University.

I also want to thank all the members of the ICSG research group, the PEACLab research group, and the CAAD research group. They are great collaborators, lab-mates, and friends, and they made my PhD experience productive and joyful.

Finally, I want to thank my parents and my parents-in-law for their unconditional love and support, expecially my father-in-law. Without his visionary encouragement, I would not start my PhD journey. From the bottom of my heart, I want to say a special thank you to my husband, Dr. Yanchen Lu, who has been a constant source

of support and encouragement during the past ten years, especially the last six years of my PhD life. We accomplished many life milestones while pursuing PhD degrees: we engaged, we married, and we have a lovely baby. Thank you to Dr. Yanchen Lu, for making me always optimistic and happy, even during the sleepless nights before deadlines.

CROSS-LAYER DESIGN OF THERMALLY-AWARE 2.5D SYSTEMS

YENAI MA

Boston University, College of Engineering, 2020

Major Professor: Ajay J. Joshi, PhD
Associate Professor of Electrical and Computer Engineering

ABSTRACT

Over the past decade, CMOS technology scaling has slowed down. To sustain the historic performance improvement predicted by Moore's Law, in the mid-2000s the computing industry moved to using manycore systems and exploiting parallelism. The on-chip power densities of manycore systems, however, continued to increase after the breakdown of Dennard's Scaling. This leads to the 'dark silicon' problem, whereby not all cores can operate at the highest frequency or can be turned on simultaneously due to thermal constraints. As a result, we have not been able to take full advantage of the parallelism in manycore systems. One of the 'More than Moore' approaches that is being explored to address this problem is integration of diverse functional components onto a substrate using 2.5D integration technology. 2.5D integration provides opportunities to exploit chiplet placement flexibility to address the dark silicon problem and mitigate the thermal stress of today's high-performance systems. These opportunities can be leveraged to improve the overall performance of the manycore heterogeneous computing systems.

Broadly, this thesis aims at designing thermally-aware 2.5D systems. More specif-

ically, to address the dark silicon problem of manycore systems, we first propose a single-layer thermally-aware chiplet organization methodology for homogeneous 2.5D systems. The key idea is to strategically insert spacing between the chiplets of a 2.5D manycore system to lower the operating temperature, and thus reclaim dark silicon by allowing more active cores and/or higher operating frequency under a temperature threshold. We investigate manufacturing cost and thermal behavior of 2.5D systems, then formulate and solve an optimization problem that jointly maximizes performance and minimizes manufacturing cost. We then enhance our methodology by incorporating a cross-layer co-optimization approach. We jointly maximize performance and minimize manufacturing cost and operating temperature across logical, physical, and circuit layers. We propose a novel *gas-station* link design that enables pipelining in passive interposers. We then extend our thermally-aware optimization methodology for network routing and chiplet placement of heterogeneous 2.5D systems, which consist of central processing unit (CPU) chiplets, graphics processing unit (GPU) chiplets, accelerator chiplets, and/or memory stacks. We jointly minimize the total wirelength and the system temperature. Our enhanced methodology increases the thermal design power budget and thereby improves thermal-constraint performance of the system.

Contents

List of Tables

List of Figures

List of Abbreviations

2.5D	.....	2.5 Dimensional
AP	.....	Acceptance Probability
Arb	.....	Arbitrary placement of chiplets
AWS	.....	Amazon Web Service
BEOL	.....	Back-End-Of-Line
CPU	.....	Central Processing Unit
DVFS	.....	Dynamic Voltage and Frequency Scaling
EMIB	.....	Embedded Multi-die Interconnect Bridge
FEOL	.....	Front-End-Of-Line
GPU	.....	Graphics Processing Unit
GS	.....	Gas Stations
HBM	.....	High-Bandwidth Memory
HMC	.....	Hybrid Memory Cube
HPC	.....	High Performance Computing
HTC	.....	Hard Temperature Constraint
Mat	.....	Matrix-style chiplet placement
MILP	.....	Mixed Integer-Linear Program
noGS	.....	no Gas Stations
PNR	.....	Place aNd Route
ROI	.....	Region Of Interest
RX	.....	Receiver
SA	.....	Simulated Annealing
SCC	.....	Single-chip Cloud Computer
SiP	.....	System-in-Package
STC	.....	Soft Temperature Constraint
TDP	.....	Thermal Design Power
TSV	.....	Through-silicon vias
TX	.....	Transmitter

Chapter 1

Introduction

1.1 Problem Statement

Moore's Law (Schaller, 1997), which has dominated the computing industry since the 1960s, is approaching the end. With voltage scaling no longer in line with the transistor size, the power density is increasing quickly as the transistor size shrinks. This is known as the breakdown of Dennard's Scaling (Dennard et al., 1974). As a result, the 'free-lunch' performance improvement from scaling has stopped since 2004 (as shown in Figure 1·1).

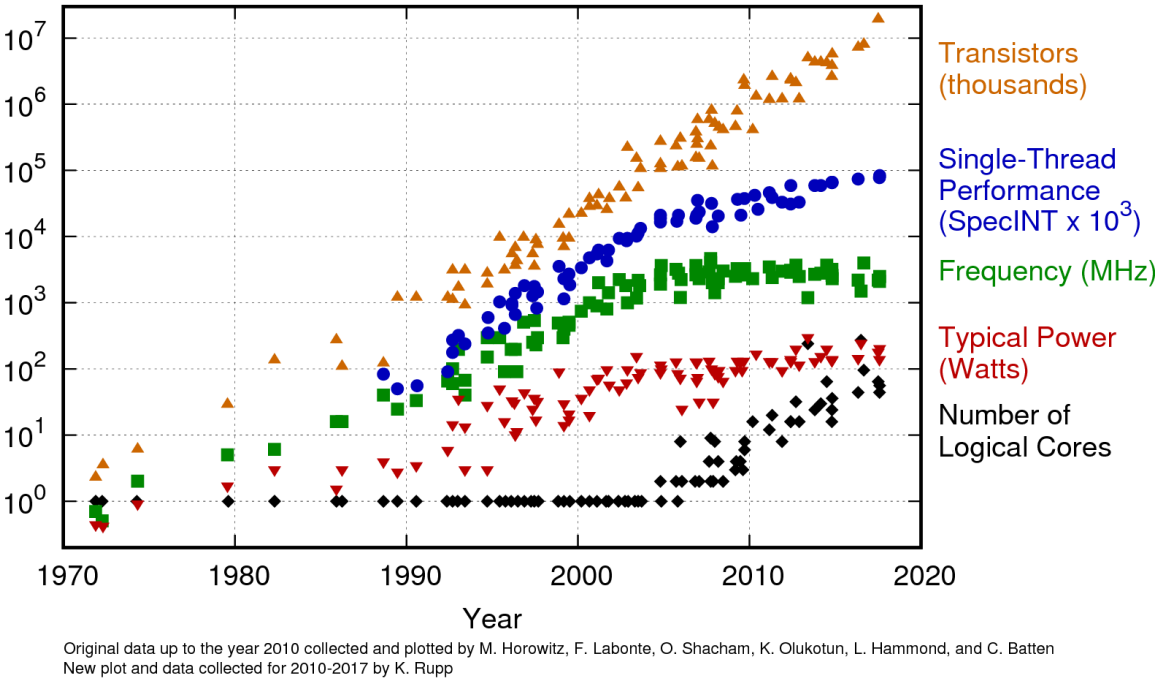

Figure 1·1: 42 years of microprocessor trend data (Rupp, 2018)

To continue the performance improvement predicted by Moore's Law despite the speed limit, the computing industry has been exploring 'More-than-Moore' approaches (Waldrop, 2016), (ITRS, 2015a). A common 'More-than-Moore' approach is to pack many cores on a single die and use parallelism to improve performance, as shown in Figure 1·1. However, although we have enough transistors to support many cores on a chip, we cannot use all of them because of the 'dark silicon' problem (Esmaeilzadeh et al., 2011). Dark silicon is the phenomenon that not all cores can be operated at the highest frequency or even turned ON simultaneously due to thermal design power (TDP) constraint. This leads to inactive regions on the chip and limits the performance of manycore systems. To address the dark silicon problem, researchers have proposed a variety of solutions through hardware level to system management level such as dynamic voltage and frequency scaling (DVFS) (Muthukaruppan et al., 2013), (Swaminathan et al., 2013), (Allred et al., 2012), designing customized hardware (Venkatesh et al., 2011), (Goulding-Hotta et al., 2011), near-threshold computing (Dreslinski et al., 2010), (Silvano et al., 2014), approximate computing (Han and Orshansky, 2013), (Kulkarni et al., 2011), power budgeting (Pagani et al., 2014), and computational sprinting (Raghavan et al., 2012). While some of these dark silicon solutions may apply to manycore systems, they do not focus on the challenge specific to manycore systems, which is to run a larger number of cores persistently.

Another path that has been taken to sustain the historical performance improvement involves integrating diverse functional components into a package (ITRS, 2015a). This transformation focuses on the overall system performance and cost rather than on individual components to push the system-level scaling, and heterogeneous integration is the backbone of this approach (Iyer, 2016). Thus, die-stacking technologies, such as 2.5D and 3D integration (Loh et al., 2007), (Kannan et al., 2015), (Stow et al., 2016), have emerged to support heterogeneous integration. 2.5D

integration places multiple dies side-by-side on a silicon interposer, while 3D integration stacks dies vertically. These die-stacking technologies offer both high bandwidth and reduced latency (Kannan et al., 2015), which could be utilized to handle the growing data traffic requirements of today's applications (ITRS, 2015a).

This new trend of heterogeneous systems poses many opportunities and challenges. As more functional components are packed in the same package, thermal dissipation becomes more critical. Therefore, although 3D integration has been proven to be a popular option in memory design (Waldrop, 2016), it is not a good candidate for high-performance computing because of the high power density resulting from the vertical stacking of dies (Loh et al., 2007). 2.5D integration is often less prone to the thermal challenges aggravated by 3D stacking (Stow et al., 2016), but still requires good heat dissipation capability. Although there are already interposer-based commercial products in the market, such as Xilinx Virtex 7 (Xilinx, 2016), AMD Fiji (Macri, 2015), Nvidia Tesla (Nvidia, 2016), and Intel Foveros (Intel, 2018), they typically place the chiplets next to each other on an interposer to embrace the benefits of low communication latency due to short inter-chiplet links and low manufacturing cost resulting from small interposer sizes. The design and optimization of 2.5D systems, including chiplet placement, inter-chiplet network architecture, design of inter-chiplet links and microbump assignment, need to be thoroughly explored to maximize the benefits of 2.5D integration (ITRS, 2015b). Especially, the opportunities of leveraging 2.5D integration technology to maximize system heat dissipation and lower operating temperature have not been discussed and utilized in prior works.

In 2.5D system design, a top-down or a bottom-up approach is typically used. However, both of them lead to sub-optimal solutions. Consider the following two examples that highlight the need for a cross-layer approach. (1) If we adopt a top-down performance-centric approach, an architecture-level analysis of network topolo-

gies indicates that high-radix, low-diameter networks provide the best overall system performance for inter-chiplet networks. However, in the physical layer, such networks usually require long wires, which would limit the network performance, and hence, the overall system performance. In the circuit layer, such long wires require active (rather than passive) interposer to house repeaters and/or pipelines to maintain high performance. Since active interposers are 10× more expensive than passive interposers (Parès, 2013), the system cost becomes expensive and so the top-down approach does not provide a desirable solution. (2) A bottom-up, cost-centric approach prefers to use passive interposers, which can only support repeaterless links in the circuit layer, thus degrading link performance and limiting maximum link length between chiplets in the physical layer. Consequently, in the logical layer, we have to adopt low-radix, high-diameter inter-chiplet networks, which result in a low-performance system. Therefore, a cross-layer optimization methodology is needed for designing 2.5D systems.

1.2 Thesis Contribution

At a broader level, this thesis aims to tackle the challenges in designing thermally-aware 2.5D systems. As a first step, we focus on homogeneous manycore systems, and propose a thermally-aware chiplet organization methodology to address the dark silicon problem. The main idea is to separate a single-chip manycore system into multiple chiplets and strategically insert spacing in between by leveraging the placement flexibility of 2.5D integration technology. We optimize the chiplet organization to jointly maximize performance and minimize manufacturing cost. Then we extend our thermally-aware chiplet organization methodology to optimize performance, manufacturing cost and peak operating temperature across logical, physical, and circuit layers. Our methodology jointly considers network topology, physical chiplet placement, and

inter-chiplet interconnect design and routing. We propose a novel inter-chiplet link design, named *gas-station* links, to enable pipelining in a cost-effective passive interposer instead of using expensive active interposer. For heterogeneous systems, we leverage 2.5D integration technology and extend our thermally-aware chiplet organization methodology to systems consisting of CPUs, GPUs, memory stacks, and/or accelerators. We formulate a multi-objective optimization framework to floorplan the heterogeneous components in a thermally-aware fashion and optimize the routing of the inter-chiplet interconnects between these components of the heterogeneous 2.5D system. The main contributions of my PhD research are as follows.

- **Leveraging Thermally-Aware Chiplet Organization in Homogeneous 2.5D Systems to Reclaim Dark Silicon:** We are the first to propose a thermally-aware chiplet organization methodology to address the dark silicon problem in homogeneous manycore systems. The key idea is to divide a large monolithic chip into multiple smaller chiplets and intelligently place these chiplets on a passive silicon interposer in a thermally-aware fashion. We strategically insert spacing between the chiplets of a 2.5D manycore system to lower the peak operating temperature and thus reclaim dark silicon by allowing the system to operate with a larger number of active cores and/or at a higher operating frequency without violating the thermal constraints. We investigate manufacturing cost and thermal behavior of chiplet-based 2.5D systems, formulate and solve an optimization problem that jointly maximizes performance and minimizes manufacturing cost of the 2.5D manycore systems. We design a multi-start greedy approach to find (near-)optimal solutions efficiently. Our analysis demonstrates that by using our proposed technique, an optimized 2.5D manycore system improves performance by 41% and 16% on average and by up to 87% and 39% for temperature thresholds of 85 ^oC and 105 ^oC, respectively,

compared to a traditional single-chip system at the same manufacturing cost. When maintaining the same performance as an equivalent single-chip system, our thermally-aware chiplet organization approach is able to reduce the 2.5D system manufacturing cost by 36%.

- **Cross-Layer Co-Optimization of Network Design and Chiplet Placement in Homogeneous 2.5D Systems:** We generalize our thermally-aware chiplet organization methodology to explore the tradeoffs across logical, physical, and circuit layers and form a cross-layer co-optimization methodology. The outcome of our methodology includes the design choice of network topology, chiplet placement, inter-chiplet link design and routing. Our cross-layer methodology jointly optimizes performance, manufacturing cost, and operating temperature of 2.5D systems. We use a soft constraint for peak temperature in the optimization problem to achieve better overall performance gain or cost reduction by allowing a small amount of thermal violation, while still ensuring thermal safety and routability. In order to maintain cost-effective and high-performance communication between chiplets in 2.5D systems, we propose a novel *gas-station* link which enables pipelining between chiplets in a passive interposer. We develop a simulated annealing algorithm to search the high-dimensional placement solution space, which supports arbitrary placements that consider non-matrix and asymmetric chiplet organizations. Our cross-layer methodology achieves better performance-cost tradeoffs of 2.5D systems and yields better solutions in optimizing inter-chiplet network and 2.5D system designs than prior methods. Compared to single-chip systems, 2.5D systems designed using our new approach achieve 88% higher performance at the same manufacturing cost or 29% lower cost with the same performance. Compared to the closest state-of-the-art (Coskun et al., 2018), our new approach achieves

40-68% (49% on average) iso-cost performance improvement and 30-38% (32% on average) iso-performance cost reduction.

- **Inter-Chiplet Network Design in Heterogeneous 2.5D Systems:** We propose a methodology for efficient routing of inter-chiplet wires and thermally-aware placement of chiplets in heterogeneous 2.5D systems, which integrate various components such as CPUs, GPUs, memory stacks, and/or accelerators on a silicon interposer. Our methodology jointly minimizes the total wirelength and the system temperature with strategic insertion of spacing between chiplets. We develop an SA-based approach to optimize the routing of inter-chiplet wires and thermally-aware chiplet placement for heterogeneous 2.5D systems. We enhance the traditional floorplanning algorithm for monolithic chips to support 2.5D systems. We use a flexible data structure to represent chiplet placement with strategically inserted spacing, which is not supported in traditional floorplan data structures. Our methodology increases the TDP without using any advanced and costly active cooling methods. This increase in TDP envelope allows higher power budget, which can be used to improve performance.

1.3 Organization

The rest of this thesis start with a review of the background and related work on die-stacking technologies, an overview of 2.5D systems, the dark silicon problem, cross-layer methodology, and thermally-aware floorplanning in Chapter 2. Chapter 3 introduces our work on single-layer optimization methodology in homogeneous 2.5D systems. Chapter 4 presents our work on cross-layer co-optimization methodology in homogeneous 2.5D systems. Then in Chapter 5 we show how we extended our cross-layer optimization methodology to heterogeneous 2.5D systems. Chapter 6 discusses future directions and concludes the thesis.

Chapter 2

Background and Related Work

In this chapter we provide an overview of background and related work on die-stacking technologies, 2.5D systems, the dark silicon problem, cross-layer methodology, and thermally-aware floorplanning.

2.1 Die-stacking Technologies

Die-stacking technologies, such as 2.5D integration and 3D integration, have emerged as a popular "More than Moore" approach to continue the computing performance improvement (Loh et al., 2007), (Kannan et al., 2015), (Stow et al., 2016). The multi-die systems using either 2.5D integration or 3D integration technology are viewed as cost-effective alternatives to single-chip systems (also called 2D systems), as breaking down a chip into multiple chiplets alleviates the manufacturing yield drop suffered in a large 2D chip. These technologies also enable the design of System-in-Package (SiP) that consists of multiple heterogeneous functional chiplets (CPU, GPU, memory, etc.) fabricated using different technologies and processes to further push the system-level improvement of performance and cost (HIR, 2019), (ITRS, 2015a). 3D integration stacks chiplets vertically on top of each other to form a system and uses through-silicon vias (TSVs) to communicate between chiplets. It reduces system footprint, communication distance, and increases memory bandwidth (Kannan et al., 2015). However, the vertical stacking of chiplets exacerbates the thermal challenges (Loh et al., 2007). Therefore, 3D integration has been a popular option in memory de-

sign (Waldrop, 2016), such as Hybrid Memory Cube (HMC) (Jeddeloh and Keeth, 2012) and High-Bandwidth Memory (HBM) (Tran et al., 2016), but it is rarely used for thermally-stressed high-power systems. 2.5D integration places the chiplets side by side on a silicon interposer. The chiplets communicate with each other through high-density fine-grained microbumps and interconnects in the interposer. 2.5D integration provides additional routing resources through the interposer, and thus supports high-density die-to-die communication (Kannan et al., 2015). Compared to 3D integration, 2.5D integration requires larger X-Y size while 3D systems are more compact. 2.5D integration technology is more mature and cost-effective while 3D integration technology often requires redesigning the chiplets to account for the TSV overhead and alignment (Radojcic, 2017). Moreover, 2.5D integration is less prone to the thermal challenges observed in 3D systems (Stow et al., 2016).

2.2 Overview of 2.5D Systems

2.5D integration is a promising technology that enables the integration of homogeneous or heterogeneous sets of chiplets onto a carrier. The carrier provides additional wiring resources that can be leveraged to increase the communication bandwidth between the chiplets and improve system performance (Jerger et al., 2014). Furthermore, 2.5D integration is more cost effective than building large 2D chips and is more thermally efficient than 3D-stacked systems (Stow et al., 2016). Currently, 2.5D integration technology is being widely explored by both academia (Jerger et al., 2014), (Kannan et al., 2015), (Grani et al., 2016), (Stow et al., 2016), (Stow et al., 2017), (Karim et al., 2013) and industry (Xilinx, 2016), (Chaware et al., 2012), (Macri, 2015), (Urino et al., 2014), (Nvidia, 2016), (Intel, 2018).

There are multiple options for 2.5D integration technology. Embedded Multi-die Interconnect Bridge (EMIB) (Intel, 2019), (Hot Chips, 2017) and silicon inter-

poser (Xilinx, 2016) are two commonly used carrier options. EMIB is a novel integration method, which embeds small pieces of silicon interconnect bridge chips in the organic package substrate to connect the edges of adjacent chiplets for die-to-die communication. Silicon interposer technology uses a relatively large interposer to house all chiplets. It is more mature and has already been used in commercial products (Xilinx, 2016), (Macri, 2015). Both EMIB and silicon interposer can provide high density die-to-bridge and die-to-interposer connections, respectively, and correspondingly, high-density die-to-die connections (Hot Chips, 2017). EMIB-based approach requires less silicon area than silicon interposer-based approach and thus has lower silicon cost (Hot Chips, 2017). However, EMIB has limited die-to-die connections per layer (Ramalingam, 2016), and also has higher complexity in the manufacturing of organic substrates (Mahajan et al., 2016). Furthermore, EMIB can only hook up adjacent chiplets and requires multi-hop communication for logically connected chiplets that are physically placed far apart.

Interposer-based integration, including active interposers and passive interposers, provides more flexibility in chiplet placement, network design and interconnect routing. Thus, it has better thermal dissipation capability as it does not require chiplets to be placed close to each other. An active interposer is effectively a large carrier chip containing transistors to house other chiplets. It is expensive as it requires front-end-of-line (FEOL) process and suffers from yield loss when the area is large. A passive interposer is transistor-free, so it can be fabricated using back-end-of-line (BEOL) process and inherently has high yield (Parès, 2013). Thus, a passive interposer is much cheaper than an active interposer ($500 per wafer without yield loss for passive interposer vs. $5000 per wafer with yield loss for active interposer (Parès, 2013)). In this thesis, we focus on passive interposer based 2.5D integration, which provides both cost effectiveness and placement flexibility.

Figure 2·1 shows the cross-section view of a passive interposer based 2.5D system. A 2.5D-integrated system consists of three main layers: an organic substrate, a silicon interposer, and a chiplet layer. Fine-pitch microbumps connect the chiplets and the silicon interposer. Through-silicon vias (TSVs) connect the top and the bottom of the interposer, and C4 bumps connect the interposer and the organic substrate. Epoxy resin is often used to underfill the connection layers (C4 bumps layer and microbumps layer) and the empty spaces between chiplets (Zhang and Wong, 2004).

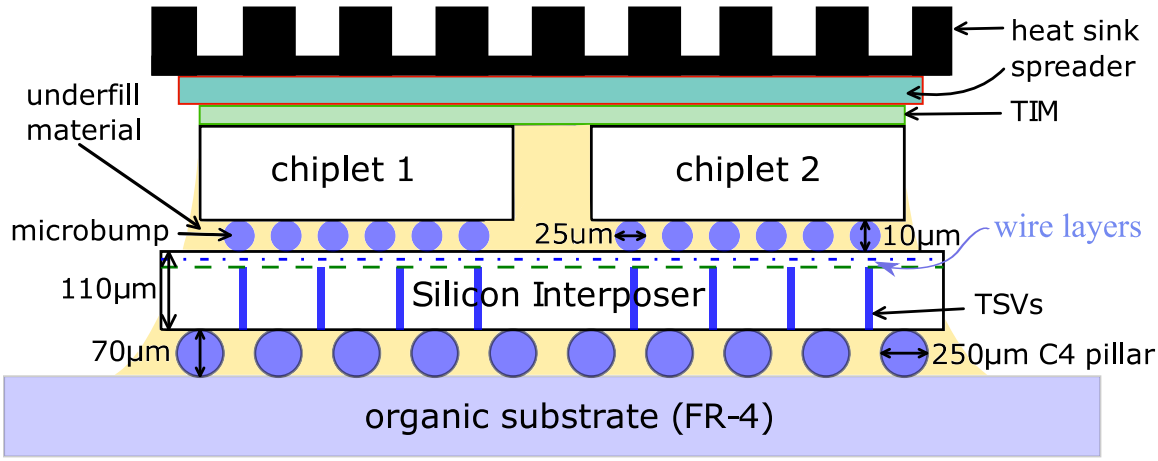

Figure 2·1: Cross-sectional view of a 2.5D integrated system.

2.3 Dark Silicon Problem

In manycore systems, dark silicon is the phenomenon where not all cores can operate at the highest frequency or can be turned ON simultaneously due to thermal design power (TDP) constraints. Over the past few years, a number of solutions have been proposed to alleviate the dark silicon problem. The proposed solutions include the use of specialized cores (Venkatesh et al., 2011), (Goulding-Hotta et al., 2011), DVFS (Muthukaruppan et al., 2013), (Yan et al., 2012), near-threshold computing (Dreslinski et al., 2010), (Silvano et al., 2014), approximate computing (Han and Orshansky, 2013), (Kulkarni et al., 2011), power budgeting (Pagani et al., 2014), and

computational sprinting (Raghavan et al., 2012). A specialized core is application-specific and enables efficient execution of that specific application with a smaller number of transistors. However, a specialized core cannot execute other types of applications efficiently. Applying DVFS degrades system performance, while near-threshold computing and approximate computing trade off accuracy and reliability for energy efficiency. Power budgeting enables operating at a thermally-safe power instead of a constant TDP to achieve a higher total performance. Computational sprinting (where the system runs with a larger number of cores in short bursts) incorporates phase-change materials for higher thermal capacitance, and thus allows violation of the thermal power budget for a short time. Power budgeting and computational sprinting, however, require a 'cooling down' period after the performance boost. Hence, these works cannot harness the full potential of manycore systems persistently.

Our work (Eris et al., 2018) leverages the placement flexibility and cost effectiveness of 2.5D systems to tackle the dark silicon problem. We strategically place chiplets in a thermally-aware fashion to facilitate heat dissipation, and thus raise the thermally-safe power budget without additional cooling cost to improve performance persistently.

2.4 Cross-layer methodology

2.5D integration of smaller chiplets on a large interposer has been demonstrated to achieve a higher compute throughput per watt (or volume) than a single large die (Stow et al., 2017), (Knickerbocker et al., 2012). Several related studies have explored the design and optimization of 2.5D systems, with primary focus being placed on individual design layers: logical, physical, and circuit.

At the logical layer, Jerger *et al.* (Jerger et al., 2014) present a hybrid network

topology between the cores and memory. They account for different coherence and memory traffic characteristics across applications, and design a hybrid network-on-chip (NoC) that has low latency and high throughput. In their follow-up work, Kannan *et al.* (Kannan et al., 2015) evaluate the impact of different network topologies on 2.5D systems, and demonstrate that disintegration of a large 2D chip into multiple chiplets improves manufacturing yield and lowers costs. However, their work overlooks the microbump overhead. Ahmed *et al.* (Ahmed et al., 2017) identify that interposer's routing resources are highly under-utilized due to the high interconnect pitch in 2.5D systems. To maximize performance, they propose a hierarchical mesh network for inter-chiplet communication. Akgun *et al.* (Akgun et al., 2016) perform a design space exploration of different memory-to-core network topologies and routing algorithms. However, a static placement of chiplets in their work limits a complete cross-layer exploration that leaves much of the performance benefits in 2.5D systems untapped. While these works aim to maximize the system performance under different traffic conditions, they do not account for the thermal impact and a complete manufacturing cost model in the NoC design and optimization. In addition, these works do not consider different chiplet placement and link routing options.

At the physical layer, there have been several optimization-based approaches aimed at providing routing and placement solutions for 2.5D systems. Placing chiplets closer to each other results in lower manufacturing cost and higher performance (reduced wirelength), but higher temperature. Therefore, finding a thermally-aware placement and routing solution that maximizes performance and/or minimizes cost is essential in 2.5D systems. Osmolovskyi *et al.* (Osmolovskyi et al., 2018) optimize the chiplet placement to reduce the interconnect length using pruning techniques. Ravishankar *et al.* (Ravishankar et al., 2018) determine the quality of different placement options in a 2D grid using a stochastic model and implement a placer for 2.5D

FPGAs. Seemuth *et al.* (Seemuth et al., 2015) consider the increased design solution space in 2.5D systems due to flexible I/Os in their chiplet placement problem. They present a method for die placement and pin assignment using simulated annealing to minimize the total wirelength. Much of the focus of routing in 2.5D systems has been placed on minimizing IR drops and total wirelength in inter-chiplet links (Fang et al., 2015) and minimizing the number of metal layers (Liu et al., 2014). None of these physical layer optimization solutions consider thermal effects.

Prior research at the circuit layer of 2.5D systems generally focuses on link optimization techniques to improve the network and system throughput. Karim *et al.* (Karim et al., 2013) evaluate the power efficiency of electrical links with and without electrostatic discharge (ESD) capacitance. Stow *et al.* (Stow et al., 2017) evaluate both repeater and repeaterless links to explore the benefits of active and passive interposers respectively. There have also been efforts on using emerging technologies like wireless links (Shamim et al., 2017) and silicon-photonic links for communication in 2.5D systems (Grani et al., 2017), (Kim et al., 2017), (Narayan et al., 2019).

A common drawback among these previous works is that their design and optimization only focus on a single design layer. In contrast, in our work (Coskun et al., 2018), (Coskun et al., 2020) we optimize the cost, performance and temperature by jointly considering the logical, physical and circuit layers of the inter-chiplet network. We evaluate various logical topologies and their feasibilities at the physical and circuit layer. At the physical layer, we design an overlap-free and thermally-safe routing and placement solution that results in the lowest cost and operating temperature. The circuit layer provides us with multiple circuit design options for inter-chiplet links. Our cross-layer methodology, thus, presents a rich solution space to evaluate a variety of network options at different design layers for 2.5D systems, thus enabling accurate and complete modeling of such systems.

2.5 Thermally-Aware Floorplanning

In addition to the traditional design objectives, such as area and wirelength, many floorplanning works consider the thermal aspect. A number of previous approaches have introduced thermally-aware floorplanning methods to reduce hot spots while optimizing area (Hung et al., 2005), to reduce peak temperature inside a microprocessor (Sankaranarayanan et al., 2005), and to reduce peak temperature and thermal gradients of 3D ICs (Frantz et al., 2012). Healy *et al.* (Healy et al., 2006) present a multiobjective microarchitectural floorplanning algorithm for 2D and 3D systems to achieve both high performance and thermal reliability. Cong *et al.* (Cong et al., 2004) propose a thermal-driven 3D floorplanning algorithm. All of these works consider placement of components to reduce temperature, but they do not focus on placement on a 2.5D interposer. In addition, these works are limited to compact placement, which cannot be applied to 2.5D systems to leverage the placement flexibility with a larger solution space. In contrast, we offer a thermally-aware chiplet placement approach that strategically adjusts spacing among chiplets on 2.5D systems to reduce the peak temperature.

Chapter 3

Single-Layer Optimization Methodology in Homogeneous 2.5D Systems

In this chapter, we discuss our optimization methodology for single-layer thermally-aware chiplet organization. We use a homogeneous 256-core manycore system as the target system. All the chiplets in our system have same architecture and size, as discussed in Section 3.1. We investigate manufacturing cost model of 2.5D systems in Section 3.2, perform a detailed design space exploration of chiplet thermal behavior in Section 3.3, formulate and solve an optimization problem in Section 3.4, demonstrate our simulation framework in Section 3.5, and discuss the results in Section 3.6.

3.1 Target System

We use a 256-core homogeneous system as our example manycore system. The core architecture of the 256-core system is based on the IA-32 core from Intel Single-chip Cloud Computer (SCC) (Howard et al., 2011), with size and power scaled to 22 nm technology (Zhang et al., 2014). Each core has a 16 KB I/D L1 cache and a 256 KB private L2 cache. The area of each core (including L1 cache) is 0.93 mm^2, and the area of each L2 cache is 0.35 mm^2. We assume each L2 cache is placed next to the corresponding core, and each core together with its L2 cache is square shaped, with an area of 1.28 mm^2 (1.13 mm $\times$ 1.13 mm) (Zhang et al., 2014). The total size of the 256-core chip is 18 mm $\times$ 18 mm.

We split the 256-core single chip into chiplets and form a 2.5D system as described

in Figure 2·1 and Section 2.2. The interposer is passive and designed using 65 nm technology. The dimensions of the 2.5D system (shown in Table 3.1) are based on the prototypes from CEA-Leti (Charbonnier et al., 2012) and Xilinx (Chaware et al., 2012). Our evaluation uses the conventional 2D single-chip system as a baseline, where the 256-core chip is placed directly on top of an organic substrate using C4 bumps for connection.

Table 3.1: Dimensions of 2.5D-integrated system.

Layers	Thickness	Materials	
Heat Sink	6.9 mm		
Spreader	1 mm		
Interface Material	20 μm		
CMOS Chiplet Layer	150 μm	Silicon, Epoxy	
Microbump Layer	10 μm	Copper, Epoxy	
Silicon Interposer	110 μm	Silicon, Copper (TSV)	
C4 Layer	70 μm	Copper, Epoxy	
Organic Substrate	200 μm	FR-4	
Component	Diameter	Height	Pitch
Microbumps	25 μm	10 μm	50 μm
TSVs	10 μm	100 μm	50 μm
C4 bumps	250 μm	70 μm	600 μm

We use an electrical mesh network (single-cycle routers and single-cycle links) for the example 256-core system. Intra-chiplet communication is through on-chiplet interconnects, while inter-chiplet communication is through links in the interposer. We use DSENT (Sun et al., 2012) to calculate power of on-chip links and routers, and HSpice (HSPICE, 2009) to compute power of inter-chiplet links based on a 2.5D interconnect model (Karim et al., 2013). We size up the drivers to ensure single-cycle propagation delay in the inter-chiplet links. The electrical mesh in the 2.5D system consumes upto 8.4 W, based on real benchmarks activities obtained from Sniper (Carlson et al., 2011). An electrical mesh network with the same micro architecture consumes 3.9 W in case of a single-chip system. Essentially, we trade off power

to match the performance of the network in the 2.5D system with that in a single-chip system. This power increase, however, has negligible impact on the thermal profile of the whole system.

3.2 Manufacturing Cost Model

The cost benefit of 2.5D systems has already been discussed in prior work (Stow et al., 2016), (Kannan et al., 2015), where a 20% to 30% reduction in cost can be achieved by replacing a single chip with a 4-chiplet 2.5D system. Smaller chiplets utilize more wafer area around the edge and achieve higher yield (Kannan et al., 2015), thus, leading to lower cost per unit area. Though an extra interposer is needed to integrate these small chiplets, the cost is rather low in case of a passive interposer (typically \$500 per 300 mm diameter wafer (Pares, 2013)) because it can be manufactured using older process technologies (Chaware et al., 2012), and with high yield (as much as 98%) (Tran et al., 2016).

To estimate the cost of our 2.5D systems, we adopt the manufacturing cost model proposed by Stow et al. (Stow et al., 2016), which takes into account the cost and yield of CMOS chiplets, microbump bonding, and interposer, assuming known good dies[1]. All notations are listed in Table 3.2. Equation (3.1) computes the number of CMOS dies that can be cut out from a wafer. Equation (3.2) computes the number of interposer dies per wafer. Equation (3.3) calculates the yield of CMOS chiplet. Equation (3.4) and Equation (3.5) compute the cost of a CMOS die and an interposer die, respectively. Equation (3.6) adds up all the components to get the overall cost of a 2.5D system.

[1]We do not explicitly model the testing cost. We assume the testing costs of a single-chip system and a 2.5D system are similar because a 2.5D system costs less in per-chiplet testings but has an additional cost associated with testing the 2.5D system as a whole.

$$N_{CMOS} = \frac{\pi \times (\phi_{wafer}/2)^2}{A_{CMOS}} - \frac{\pi \times \phi_{wafer}}{\sqrt{2 \times A_{CMOS}}} \tag{3.1}$$

$$N_{int} = \frac{\pi \times (\phi_{wafer_{int}}/2)^2}{A_{int}} - \frac{\pi \times \phi_{wafer_{int}}}{\sqrt{2 \times A_{int}}} \tag{3.2}$$

$$Y_{CMOS} = (1 + A_{CMOS} D_0/\alpha)^{-\alpha} \tag{3.3}$$

$$C_{CMOS} = C_{wafer}/N_{CMOS}/Y_{CMOS} \tag{3.4}$$

$$C_{int} = C_{wafer_{int}}/N_{int}/Y_{int} \tag{3.5}$$

$$C_{2.5D} = \frac{C_{int} + \Sigma_{i=1}^{n}(C_{CMOS} + C_{bond})}{Y_{bond}^{n-1}} \tag{3.6}$$

Table 3.2: Notation used in Equations (3.1) through (3.12)

Notation	Definition	Assumed Value
$\phi_{wafer}, \phi_{wafer_{int}}$	Diameter of CMOS and interposer wafer	300 mm
N_{CMOS}, N_{int}	CMOS and interposer dies per wafer	Eq. (3.1)
D_0	Defect density	$0.25/mm^2$ (Stow et al., 2016)
α	Defect clustering parameter	3 (Stow et al., 2016)
Y_{int}	Yield of an interposer	98% (Tran et al., 2016)
Y_{CMOS}	Yield of a CMOS chiplet	from Eq. (3.3)
C_{wafer}	CMOS wafer cost	\$5000 (Pares, 2013)
$C_{wafer_{int}}$	Interposer wafer cost	\$500 (Pares, 2013)
$C_{int}, C_{CMOS}, C_{2D}$	Chiplet, interposer, and 2D chip cost	from Eq. (3.4)
Y_{bond}	Chiplet bonding yield	99% (Stow et al., 2016)
$C_{2.5D}$	Cost of the 2.5D system	from Eq. (3.6)
l_g	Guard band along each interposer edge	1 mm
w_{2D}, h_{2D}	Width and height of the baseline 2D chip	18 mm
w_{int}, h_{int}	Width and height of the interposer (in mm)	from Eq. (3.11)
w_c, h_c	Width and height of the chiplets	from Eq. (3.10)

Notation	Definition
A_{CMOS}, A_{int}	CMOS, interposer die area
C_{bond}	Bonding cost of a chiplet (Farrens and MicroTec, 2010)
r	Number of chiplets in a row or column
n	Number of chiplets $n = r \times r$, $n \in \{4, 16\}$
F	Frequency set $\{1000,\ 800,\ 533,\ 400,\ 320\ MHz\}$
V	Corresponding voltage set $\{0.9,\ 0.87,\ 0.71,\ 0.63,\ 0.63\ V\}$
f	Operating frequency $f \in F$
p	Active core count $p \in \{32,\ 64,\ 96,\ 128,\ 160,\ 192,\ 224,\ 256\}$
$IPS_{2.5D}, IPS_{2D}$	Instructions per second (IPS) of 2.5D system and 2D system
s_1, s_2, s_3	Chiplet spacings (Figure 3·5(a)). $s_1 = s_2 = 0$ for 4-chiplet case
$T_{peak}, T_{threshold}$	Peak operating temperature and Temperature threshold for safety

Figure 3·1 shows the manufacturing cost of the 2.5D systems with various (square-shaped) interposer sizes normalized to an equivalent 18 mm × 18 mm single-chip system for a range of defect densities (Stow et al., 2016). As the interposer size increases, the cost is higher since there are fewer interposers that can be cut out from a wafer. The 2.5D system with a minimal interposer size has a cost saving ranging from 30% to 42%, compared to the cost of the single-chip system at the same defect density. With higher defect density, the 2D system costs more due to lower yield. Thus, there are higher cost saving from splitting chiplets for a larger defect density. With a larger chiplet count, the CMOS chiplet yield is higher but the bonding yield is lower. The 16-chiplet case costs more at a larger interposer size due to lower bonding yield. While at a smaller interposer size, the costs of both chiplet counts are close, because the lower bonding yield of the 16-chiplet case is compensated by the higher CMOS yield of smaller chiplets.

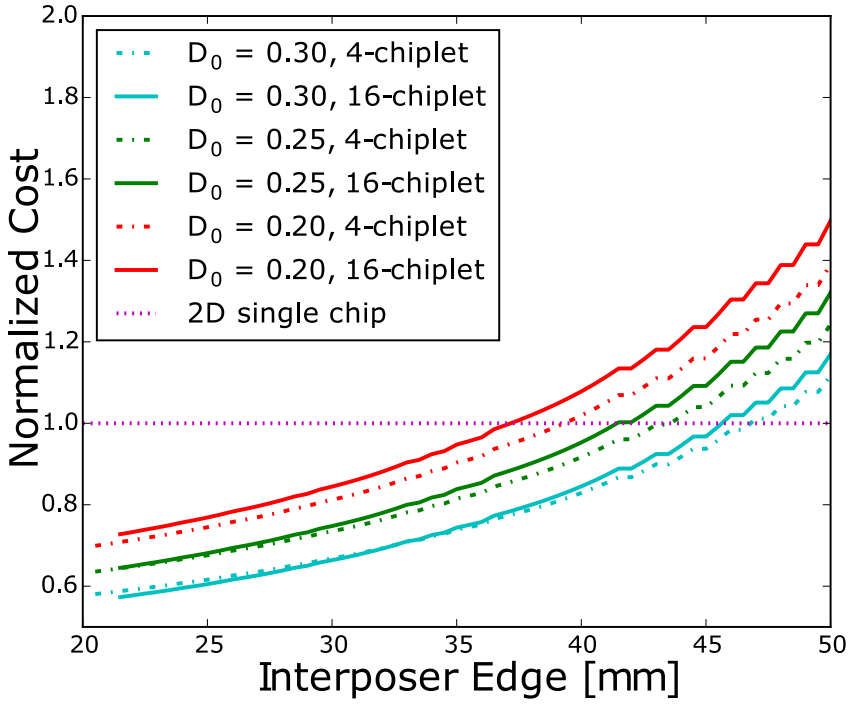

Figure 3·1: Impact of defect densities on 2.5D system cost normalized to the single-chip system costs at the same defect densities.

3.3 Thermal Behavior of 2.5D Systems

To understand the thermal behavior of a 2.5D system, we first analyze the impact of chiplet count on the operating temperature of the 2.5D system. In this study, we divide a single chip of 18 mm × 18 mm into $r \times r$ identical chiplets ($r \times r = n$, and r varies from 2 to 10) and place them onto an interposer in a matrix fashion with **uniform** spacing between adjacent chiplets. For each value of r, we vary the interposer edge length from 20 mm to 50 mm in steps of 1 mm and calculate the corresponding spacing between chiplets. For example, if an interposer has an edge length of L mm, the spacing between the adjacent chiplets is $(L - 18 - 2 \times l_g)/(r - 1)$ mm (where $l_g = 1$ mm is the guard-band spacing along each interposer edge), and an individual chiplet edge is $18/r$ mm. For a given interposer size, as the chiplet count increases, the spacing between the chiplets decreases. We assign synthetic power densities from 0.5 W/mm^2 to 2.0 W/mm^2 to the chiplets and perform thermal simulations via HotSpot (Zhang et al., 2015) to get a better understanding of the thermal trends in 2.5D systems.

Figure 3·2 shows the impact of chiplet counts, interposer sizes, and power densities on peak temperature of 2.5D systems. In general, as expected, for the same chiplet count and interposer size, the peak temperature increases with power density. For the same chiplet count and power density, as the interposer size increases, the peak temperature decreases due to the increased spacing among chiplets. For the same interposer size and power density, the peak temperature decreases with increasing chiplet count. However, there are some exceptions that the 2×2 case has lower peak temperature than others for small interposer sizes, and the 4×4 case has lower peak temperature than the 5×5 case. This is because 'even' chiplet counts avoid placing a chiplet at the interposer center where it is harder to dissipate heat than at the interposer edge. As the chiplet count increases, the individual chiplet size decreases,

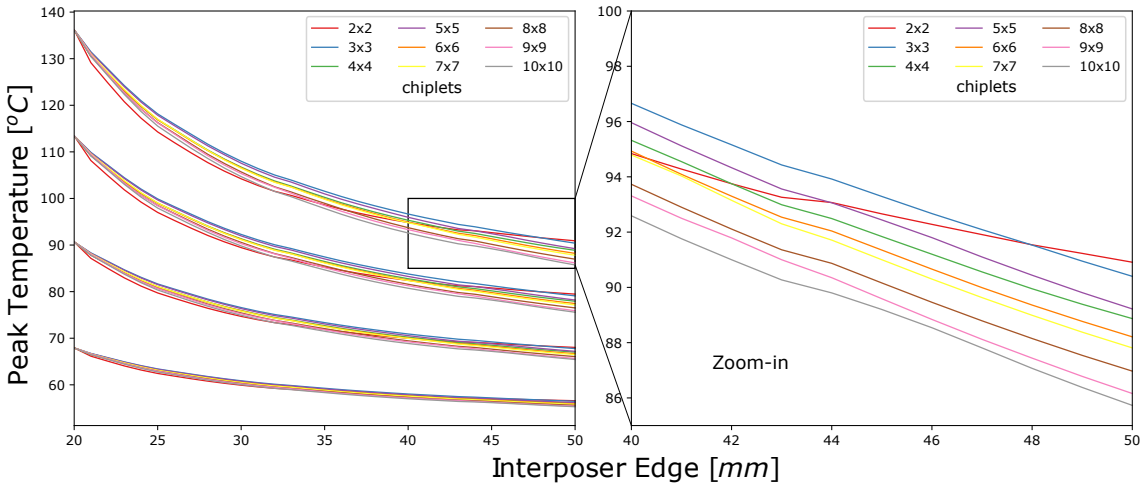

Figure 3·2: Impact of chiplet counts, interposer sizes, and power densities on peak temperature of 2.5D systems with **uniform** spacing between chiplets.

resulting in less power dissipated by the center chiplet. Hence, we do not observe such exceptions for chiplet counts greater than 5×5.

Although a single chip with the **same power profile** and the **same area** as our 2.5D system would achieve a similar thermal profile, the single-chip solution is not the best choice from a cost perspective. For example, based on Equations (3.1)-(3.6) and parameters in Table 3.2, increasing the single chip size from 20 $mm \times 20$ mm to 40 $mm \times 40$ mm results in 27× higher cost because of drastically lower yield. Alternatively, an equivalent 2.5D system with four smaller chiplets and a 40 $mm \times$ 40 mm passive silicon interposer has 27% lower cost (where the interposer cost is 30% of the 2.5D system) than a 20 $mm \times 20$ mm single chip.

From the cost perspective, as chiplet count increases in a 2.5D system, the time for the serial bonding process increases and the overall bonding yield drops, which increases the cost. Due to the limited thermal advantages of increasing chiplet count beyond 4×4 and the bonding yield consideration, we only consider 2.5D systems with 2×2 and 4×4 chiplets in the following sections.

Our work reduces operating temperature without requiring a more powerful heatsink or introducing advanced cooling technology such as liquid cooling. We conduct an experiment to analyze the impact of heatsink. We use the 16-chiplet 2.5D system with 2 W/mm^2 synthetic power density as an example. The interposer size varies from 20 $mm\times$ 20 mm to 50 $mm\times$ 50 mm and the chiplets are distributed in a matrix fashion with uniform spacing. The default heat transfer coefficient value is 122 W/m^2K (Whitelaw, 1997), denoted as 1$\times$ in Figure 3·3. We vary the heat transfer coefficient of the heatsink from 0.25$\times$ to 4$\times$ the default value. To be noted here, 0.25$\times$ the default heat transfer coefficient value is within the range of laptop heatsink with single small fan (the typical range is 25-100 W/m^2K (Long, 2013)). The 1$\times$ default value is in the range of optimized heatsink with multiple fans for desktops (the typical range is 50-150 W/m^2K (Long, 2013)). The 2$\times$ default value is hardly to achieve with forced air cooling and the 4$\times$ default value is the lower bound of liquid cooling. We show them here just to understand the trend.

As shown in Figure 3·3, when we increase the heat transfer coefficient (i.e., a more powerful heatsink or a more advanced cooling technology), the peak temperature reduces. At a small interposer size where the chiplets are more compact, the temperature reduction resulting from heatsink is large, while at a large interposer size where the chiplets are more distributed and far apart, the temperature reduction is relatively small. Our methodology reduces peak temperature significantly with a poor heatsink, and the benefits decrease when the heatsink gets more powerful and when the cooling technology is upgraded to liquid cooling. For a realistic heatsink (from 0.25$\times$ to 1$\times$ default value), our thermally-aware chiplet placement methodology can achieve higher temperature reduction than replacing a poor heatsink with a better one. This also indicates that our methodology facilitates heat dissipation greatly with a large interposer, thus, reducing the need of using a powerful heatsink.

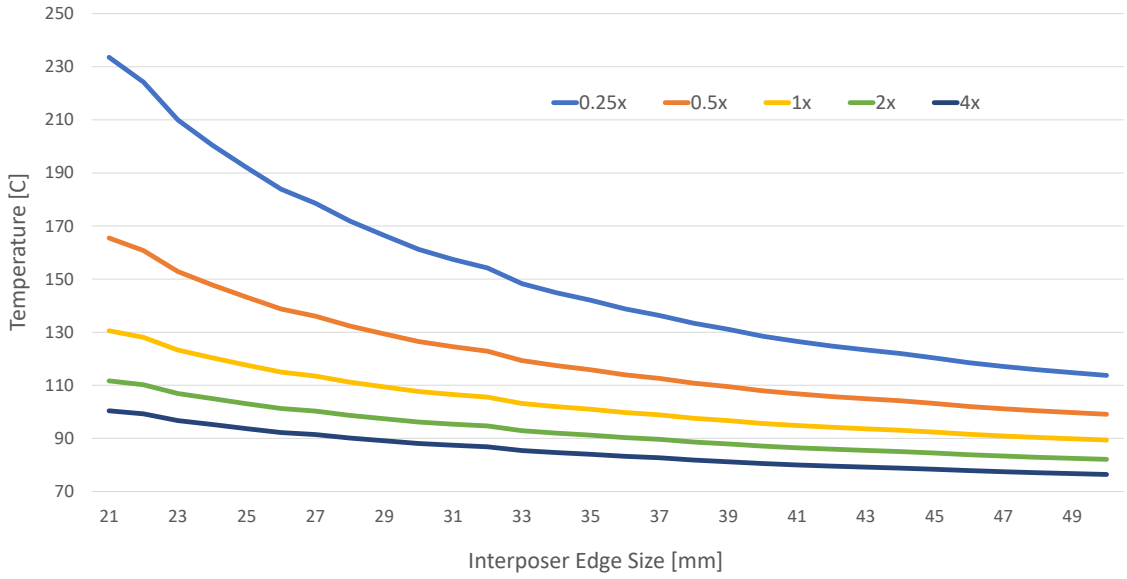

Figure 3·3: Impact of different heat transfer coefficients of the heatsink (normalized to 122 W/m^2K) on peak temperature of 16-chiplet 2.5D systems with **uniform** spacing between chiplets.

3.4 Optimization of Chiplet Organizations

To determine the optimal thermally-aware chiplet organization (including chiplet count, chiplet placement, active core count, and operating frequency), we formulate an objective function that maximizes system performance while minimizing system cost, as shown in Equation (3.7). In Equation (3.7), 2.5D system performance (in terms of instructions per second (IPS)) and cost are normalized to the baseline single-chip system, and the user-specified weight factors α and β have no units. The objective function is subject to a peak temperature constraint [Equation (3.8)], an interposer size constraint [Equation (3.9)], and inter-related spacing constraints for chiplet placement [Equations (3.10), (3.11) and (3.12)]. The cost of the 2.5D system is calculated using Equations (3.1) to (3.6). All notations used in Equations (3.7) to (3.12) are listed in Table 3.2. Widths, heights, and spacings are in mm.

Minimize:

$$\alpha \times \frac{IPS_{2D}}{IPS_{2.5D}} + \beta \times \frac{C_{2.5D}}{C_{2D}} \tag{3.7}$$

Subject to:

$$T_{peak} <= T_{threshold} \tag{3.8}$$

$$w_{int} <= 50, \quad h_{int} <= 50 \tag{3.9}$$

$$w_c = \frac{w_{2D}}{r}, h_c = \frac{h_{2D}}{r} \tag{3.10}$$

$$\begin{aligned} w_{int} &= w_c \times r + 2 \times s_1 + s_3 + 2 \times l_g, \\ h_{int} &= h_c \times r + 2 \times s_1 + s_3 + 2 \times l_g \end{aligned} \tag{3.11}$$

$$2 \times s_1 + s_3 - 2 \times s_2 > 0 \tag{3.12}$$

In Equation (3.8) we assume that the default temperature threshold, $T_{threshold}$, for safe operation is 85 ^oC. Equation (3.9) limits the interposer size to be no larger than 50 $mm\times$ 50 mm. We make this choice so that the interposer size is within the exposure field size of 2X JetStep Wafer Stepper (Cochet et al., 2014) to avoid extra stitching cost. In our design, the chiplets are distributed on the silicon interposer, and we assume the number of chiplets in a row is equal to the number of chiplets in a column. We consider all chiplet organizations on an interposer that are axially and diagonally symmetric. We use the Mintemp (Zhang et al., 2014) workload allocation policy for our analysis, which minimizes operating temperature by assigning threads starting from outer rows or columns and then moving to inner rows or columns of the whole system in a chessboard manner. Equation (3.10) calculates the chiplet width and height in terms of the width and height of the 2D system and chiplet count.

Equation (3.11) calculates the interposer width and height as a function of chiplet spacings (s_1, s_2, and s_3 in Figure 3·4, which vary **independently**). Equation (3.12) ensures there is no overlap between center chiplets.

To determine the minimum value of the objective function, we can use an exhaustive search approach, which takes $180k$ CPU hours (a calendar month with 250 computers running in parallel) to run thermal simulations for the whole design space. Cost-wise, it requires a one-time cost of $1080 on Amazon Web Service (AWS) (Amazon, 2018) at $0.0059 per core hour rate. The simulation time is long because there are over $680k$ chiplet organizations ($17k$ chiplet placement options with 0.5 mm granularity, five voltage/frequency levels, and eight different active core counts) for each benchmark, and each organization takes up to 2 $mins$ for a thermal simulation. Note that it takes $1.5k$ CPU hours in total to determine performance for all the 40 (f, p) pairs when using an architectural simulator to run the benchmarks listed in Section 3.5.1, which is insignificant compared to thermal simulation time.

To speed up the process of finding a solution to our optimization problem, we de-

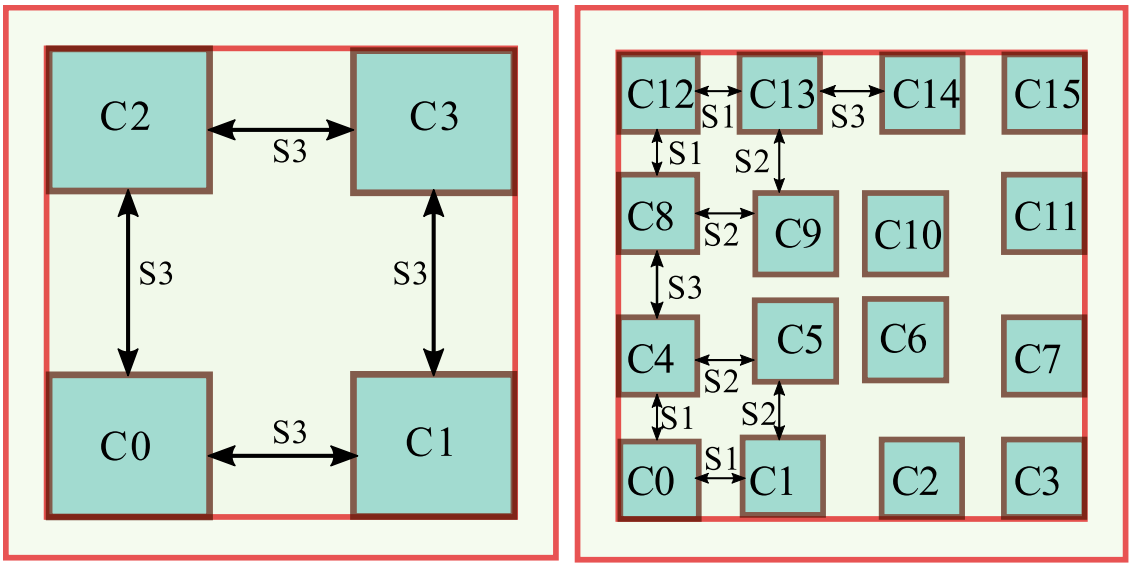

Figure 3·4: Chiplet count and placement options. We vary the chiplet spacings **independently** to find the optimal chiplet placement.

sign a multi-start greedy approach to reduce the number of thermal simulations (see Pseudocode). Our approach has three steps. In the first step, we calculate the performance of the 256-core system for all 40 (f, p) pairs using Sniper (Carlson et al., 2011), and the cost $(C_{2.5D})$ of both 4-chiplet and 16-chiplet cases for discretized interposer sizes from 20 mm to 50 mm with 0.5 mm granularity using Equations (3.1) to (3.6). In the second step, we compute the objective function value for each $(f, p, C_{2.5D})$ combination using user-specified weights α and β, and sort these $(f, p, C_{2.5D})$ combinations in ascending order of objective function values. In the third step, we go through the list of $(f, p, C_{2.5D})$ combinations in the sorted order to find a chiplet organization that meets the temperature threshold. Here, for each $(f, p, C_{2.5D})$ combination, we use m starting points (for each starting point, spacing values s_1, s_2 and s_3 are randomly picked), and we greedily explore the design space from these starting points. Each starting point $(S_{current})$ has six neighboring points (obtained by varying one of s_1, s_2 or s_3 by ± 0.5 mm). We randomly[2] pick one neighbor $(S_{neighbor})$ and evaluate the peak temperature of $S_{current}$ and this $S_{neighbor}$. If the neighbor has a peak temperature lower than the temperature threshold, it is a chiplet placement solution for the current $(f, p, C_{2.5D})$ combination. We then stop the process and pick this organization as our solution. If $S_{neighbor}$ has a lower peak temperature than $S_{current}$ (but higher than the temperature threshold), we make $S_{neighbor}$ the next $S_{current}$ and repeat the substeps mentioned earlier to check if it has a neighbor with lower peak temperature. If $S_{neighbor}$ has higher temperature than $S_{current}$, we pick another neighbor of $S_{current}$ and evaluate its peak temperature. If all neighbors of $S_{current}$ have higher peak temperature than $S_{current}$, we move on to the next random starting point. If there is no feasible solution among all the m starting points, then we go to the next $(f, p, C_{2.5D})$ combination. If none of the $(f, p, C_{2.5D})$ combinations lead to a feasible solution, it

[2]We randomly pick the neighbor placement because out of the six neighbors, the neighbor that has the lowest peak temperature may not necessarily lead to a local minimum. We also avoid any biases resulting from evaluating neighbors in a fixed order.

Pseudocode: Multi-Start Greedy Approach

```
1) calculate cost and performance of 2.5D system for all (f, p, C_2.5D) combinations
2) input obj. func. weights (α,β)
   sort (f, p, C_2.5D) combinations based on obj. func. from low to high
3) foreach (f, p, C_2.5D) combination in the sorted order do
     generate random start points of (s_1, s_2, s_3)
     foreach start point (S_current) do
       evaluate peak temperature T of S_current
       repeat
         generate a random neighbor placement (S_neighbor)
         evaluate peak temperature T' of S_neighbor
         if T' < T_threshold then
           output S_neighbor and (f, p, C_2.5D) combination and exit
         if T' < T then
           update minimum peak temperature T ← T'
           update current placement S_current ← S_neighbor
       until T < peak temperature of all the neighbor placements
     end for
   end for
```

means that the manycore system is unable to run at any (f, p) pair within the given temperature threshold.

We validate our multi-start greedy algorithm by comparing with the exhaustive search approach. The greedy algorithm with ten starting points (there is a tradeoff between accuracy and speed for different number of starting points) achieves the same result as the exhaustive search approach 99% of the time. Using the multi-start greedy approach, we can reduce the thermal simulation time from $180k$ to $0.45k$ CPU hours (400× speedup), and speed up the total simulation time (Sniper and Hotspot simulations) by 100× compared to the exhaustive search approach.

3.5 Evaluation Methodology

Our evaluation framework is shown in Figure 3·5. We use Sniper (Carlson et al., 2011) for performance evaluation and McPAT (Li et al., 2009) to compute power based on the performance statistics from Sniper. We perform thermal simulation using HotSpot-6.0 (Zhang et al., 2015), based on the power traces from McPAT and floorplans of the 2.5D system. There is a closed loop between the chiplet organizer, the floorplan generator, and HotSpot. The chiplet organizer is implemented using the multi-start greedy algorithm (with ten starting points) as discussed in Section 3.4. The details of our evaluation infrastructure are in the subsections below.

3.5.1 Performance Evaluation

We use Sniper (Carlson et al., 2011) for performance evaluation. We use multi-threaded benchmarks from PARSEC (`blackscholes`, `swaptions`, `streamcluster`, `canneal`) (Bienia et al., 2008), SPLASH-2 (`cholesky`, `lu.cont`) (Woo et al., 1995), HPCCG1 (`hpccg`) (Heroux, 2007), and UHPC(`shock`) (Campbell et al., 2012) suites that cover workloads of various performance and power profiles. We use different

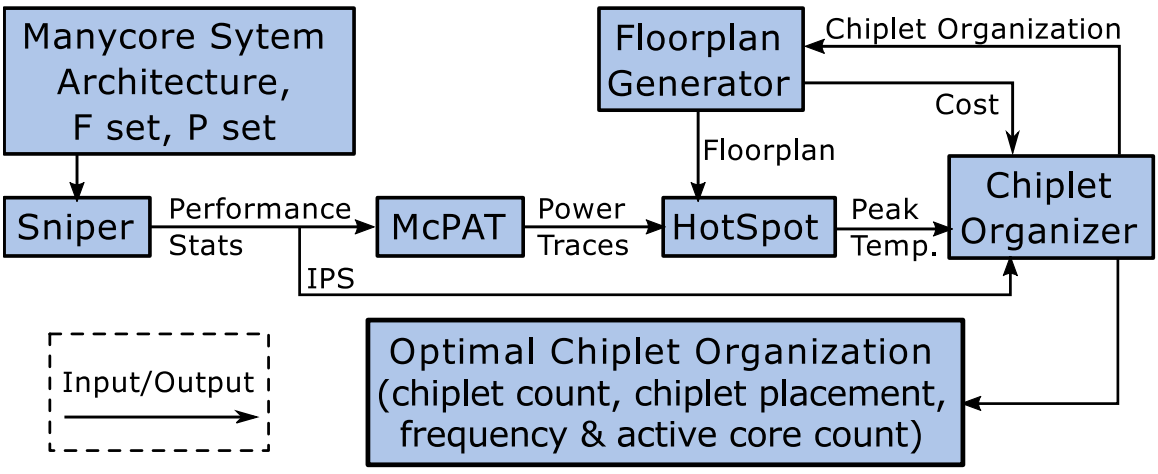

Figure 3·5: Evaluation framework.

frequency/voltage levels and different numbers of active cores (see Table 3.2) while evaluating the 256-core system. For each combination of voltage/frequency level and active core count (40 combinations in total) for each benchmark, we simulate 10 billion instructions in the parallel region or the full Region of Interest (ROI) if it finishes earlier. We collect performance statistics for each core every 1 ms.

3.5.2 Power Calculation

We use McPAT (Li et al., 2009) to calculate the power consumption of each core based on the performance stats from Sniper. We calibrate the McPAT output with the measured power dissipation data of Intel SCC (Howard et al., 2011), scaled to 22 nm. We assume that the idle cores enter sleep mode and consume negligible power (close to 0 W).

3.5.3 Thermal Simulation

We use HotSpot-6.0 (Zhang et al., 2015) for our thermal simulations, which can model a layer composed of heterogeneous materials in a 3D structure (Meng et al., 2012). We model 2.5D systems based on industry prototypes (Charbonnier et al., 2012), (Chaware et al., 2012) (see Table 3.1). We generate detailed floorplan files specifying material properties of all blocks in each layer. For the interface material, the spreader, and the heat sink, we use the default conventions in HotSpot, assuming spreader edge size is 2× interposer's edge, and heat sink edge size is 2× spreader edge. We adjust the convective resistance of heat sink to keep heat transfer coefficient of 122 W/m^2K (Whitelaw, 1997) consistent. We set ambient temperature to 45 ^oC.

We implement a temperature-dependent leakage power model in our thermal simulations. We extract a linear leakage model from published power and temperature data of Intel 22 nm processors (Wong, 2012). We calibrate core power from the McPAT output at 60 ^oC with measured power from Intel SCC (Howard et al., 2011) and

assume 30% of power comes from leakage at this temperature (Wong, 2012). We use HotSpot to get an initial system temperature profile, then adjust the leakage power of each core based on its temperature, and then re-run HotSpot to update the thermal profile until the temperature converges.

3.6 Evaluation Results

3.6.1 Peak Temperature Reduction using 2.5D Integration

We first study the impact of spacing between chiplets on the peak temperature (for different chiplet counts) with all cores active at 1 GHz for various benchmarks (see Figure 3·6). The 0 mm spacing case refers to the single-chip system. For the 2.5D integration cases, we organize the chiplets in a matrix fashion with a **uniform** spacing (from 0.5 mm to 10 mm with a granularity of 0.5 mm) between adjacent chiplets, given the 50 $mm\times$ 50 mm upper limit of the interposer size. As discussed in Section 3.2, the 64-chiplet and 256-chiplet cases are not viable due to low overall bonding yield. We present them here to show the overall thermal trends.

The reported power values are the total power consumption under the single-chip case. These power values, which are unrealistic for 2D systems, can be viable for 2.5D systems from a thermal perspective. Even at these large power consumption values, a 2.5D system can operate below a typical temperature threshold of 85 ^oC. The challenge then will be the design of a power delivery network that can provide the current required for this large power consumption[3].

In general, for all 2.5D integration cases, the peak temperature decreases as chiplet spacing increases. High-power benchmarks need larger chiplet spacing to stay below the 85 ^oC threshold. For example, high-power benchmarks (`shock`, `blackscholes`,

[3]Based on expert opinion (Friedman, 2016), there are no fundamental limits in designing power delivery circuits for high-power chips (e.g., 500 W), but a number of engineering challenges would need to be addressed.

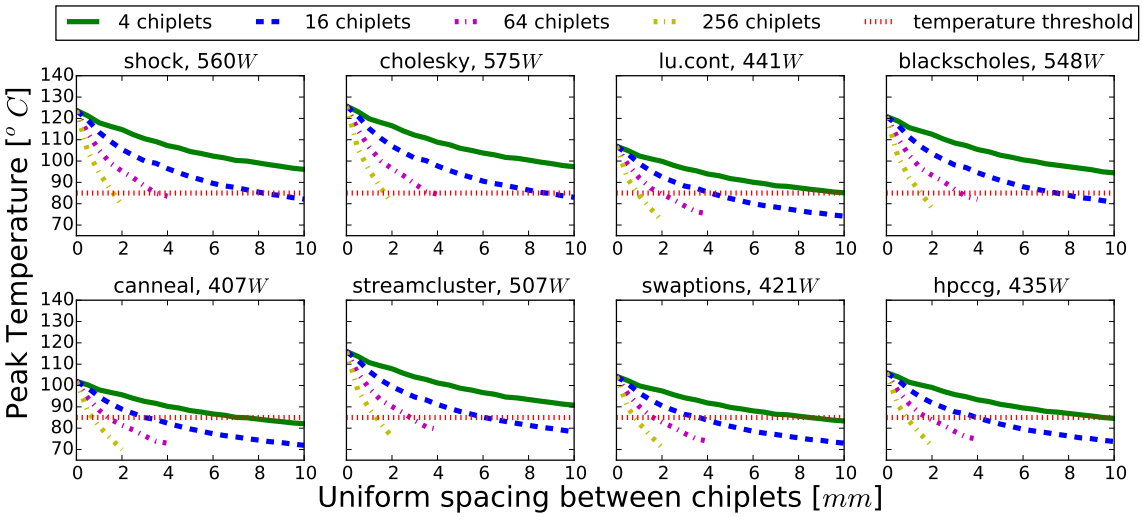

Figure 3·6: Peak temperature of a 256-core system with all cores active at 1 GHz for single-chip case (0 mm) and 2.5D integration cases for various chiplet counts and spacings (with chiplets placed in a matrix fashion).

and `cholesky`) need a 16-chiplet system with 10 mm spacing to meet the 85 ^oC constraint, while low-power benchmarks (`canneal` and `swaptions`) can easily meet the same constraint with 16 chiplets and 4 mm spacing or with 4 chiplets and 8 mm spacing. This analysis shows that even a naive chiplet organization can lower peak temperature significantly and provide opportunities to improve performance.

3.6.2 Balancing Performance and Cost of 2.5D Systems

In this subsection, we optimize the chiplet organization by considering **non-uniform** spacing between chiplets. Figure 3·7 shows the normalized maximum IPS and cost of 2.5D systems. The maximum IPS, in general, remains unchanged as the interposer size increases, until the interposer size is large enough to find a chiplet placement that can operate the system at a higher performance level within the temperature threshold. The IPS curves have steps because we use discretized frequencies and active core counts. Since the cost of 2.5D systems only depends on the chiplet count

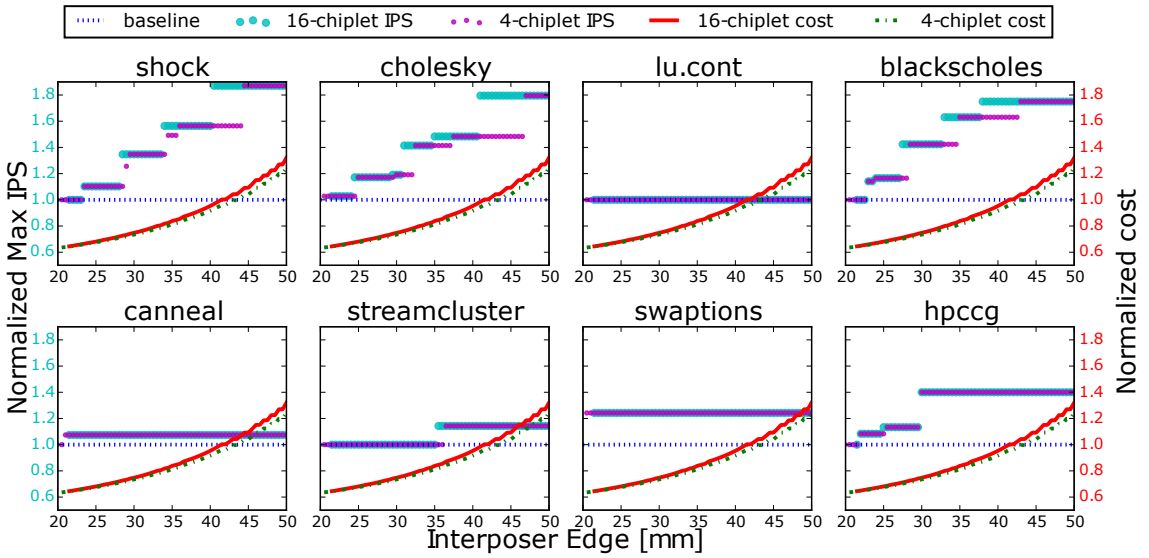

Figure 3·7: Maximum IPS and cost of 2.5D systems (normalized to maximum IPS and cost of a single-chip system) under 85 ^oC for various interposer sizes and benchmarks.

and the size of interposer, the cost curves are the same across all benchmarks. With the minimum interposer size, the system cost decreases by 36% without performance loss. This reduction in cost is due to the higher yield of the smaller CMOS chiplets compared to the single-chip baseline.

At the same cost as the baseline, a thermally-aware 2.5D system with 16 chiplets can improve the performance by 41% on average across 8 benchmarks and by up to 87%. For the high-power benchmarks `shock`, `cholesky` and `blackscholes`, our approach achieves 87%, 80% and 75% performance improvement, respectively. As for the remaining benchmarks, our approach has 40% improvement for `hpccg`, 24% for `swaptions`, and 14% for `streamcluster`; however, there is only 7% improvement for `canneal` and no performance gain for `lu.cont` when using 2.5D integration technology. The performance improvements for these benchmarks are limited because they do not need all cores active to maximize performance. For example, to achieve maximum performance, `canneal` needs 192 active cores, which is thermally feasible

at small interposer sizes, while for `lu.cont` the maximum performance is achievable with 96 active cores even in conventional single-chip system under the temperature threshold. Although 2.5D systems do not bring performance benefits for `lu.cont`, our proposed thermally-aware chiplet organization can still provide lower operating temperature, which improves transistor lifetime and reliability.

Figure 3·8 shows the minimum objective function (Equation (3.7)) values of three different choices for α and β across different interposer sizes and different benchmarks. When $\alpha = 0$ and $\beta = 1$, the curves are the same as normalized minimum cost curves. When $\alpha = 1$ and $\beta = 0$, the curves are the same as inversed normalized maximum performance. When $\alpha = 0.5$ and $\beta = 0.5$, the objective function value is the weighted sum of $\frac{IPS_{2D}}{IPS_{2.5D}}$ and $\frac{C_{2.5D}}{C_{2D}}$. For a given pair of α and β, the optimal chiplet organization occurs at the minimum point on the objective function curve. For example, `cholesky` has the optimal organization at the interposer size of 31 mm, running at 1 GHz with 192 active cores. The optimal chiplet organization, however, varies across benchmarks.

To choose the final chiplet organization, a designer would need to choose appropriate α and β values. Figure 3·9 shows examples of optimal chiplet organization and the workload allocation for $\alpha = 1$ and $\beta = 0$ under an 85 ^oC constraint. For `cholesky`, our technique improves performance by 80% by increasing frequency from 533 MHz to 1 GHz, while the cost is similar compared to the baseline. For `hpccg`, our 2.5D system achieves 40% higher performance by increasing active core count from 160 to 256 and lowers cost by 28%. For `canneal`, the performance benefit is 7% because it saturates with 192 active cores; however, our approach reduces the cost by 36%. These results demonstrate that our thermally-aware chiplet organization technique can reclaim dark silicon by having more active cores and/or operate the cores at a higher frequency without violating the temperature threshold.

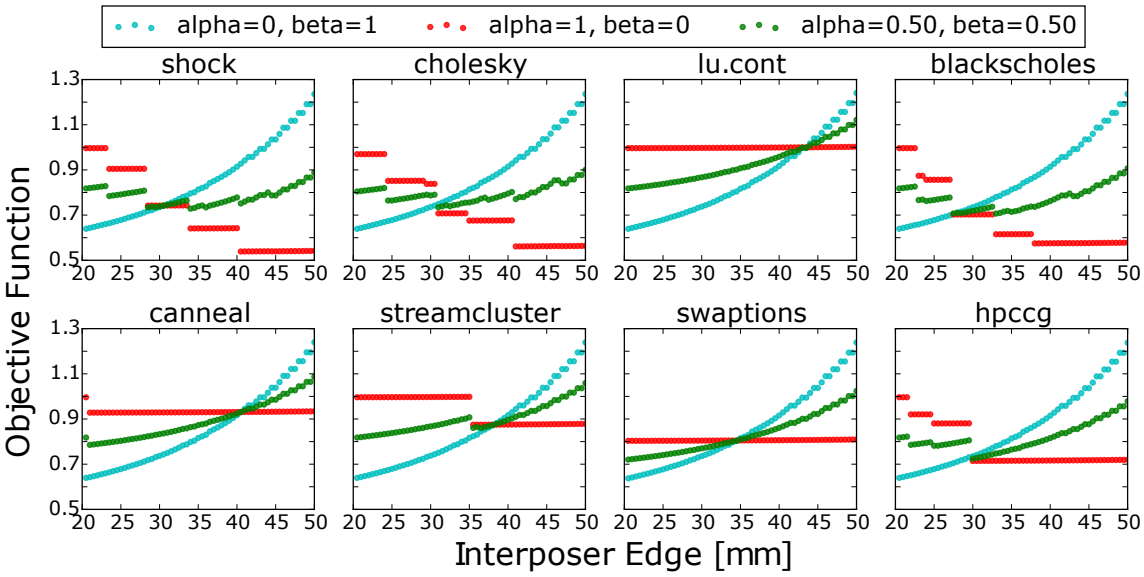

Figure 3·8: Minimum objective function (from Equation (3.7)) value for different (α, β) pairs across different interposer sizes for different benchmarks.

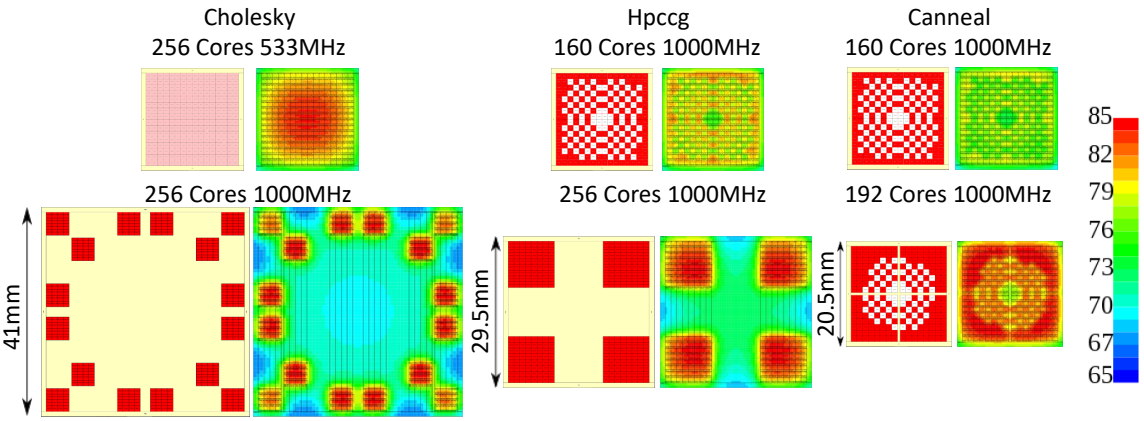

Figure 3·9: Choice of chiplet organizations that maximizes the performance under 85 ^oC for single-chip baseline (top) and 2.5D systems (bottom).

We analyze the sensitivity of our proposed approach to different temperature thresholds ranging from 75 ^oC to 105 ^oC. The performance of the baseline single-chip system is lower at a lower temperature threshold, so there is more room for performance improvement. For the temperature thresholds of 75 ^oC, 85 ^oC, 95 ^oC, and

105 ^oC, our thermally-aware chiplet organization approach improves the performance by 41%, 41%, 27%, and 16%, respectively, on average across all 8 benchmarks.

3.7 Summary

This chapter has proposed a thermally-aware chiplet organization methodology to reclaim dark silicon in homogeneous 2.5D manycore systems. The high-level idea is to split a manycore system across multiple chiplets in the 2.5D system and then strategically insert spacing between the chiplets to reduce the operating temperature of the overall system, thus allowing more cores to operate at a higher frequency under the same safe peak temperature threshold. We have used a multi-start greedy approach to determine the optimal chiplet organization that jointly maximizes performance and minimizes cost. Experimental results show that for a 256-core system, compared to a single-chip design, our thermally-aware 2.5D integration approach improves performance by 41% (16%) on average and up to 87% (39%) without increasing the cost while staying below a peak temperature threshold of 85 ^oC (105 ^oC), or reduces system cost by 36%, without performance loss, at all temperature thresholds.

Chapter 4

Cross-Layer Co-Optimization Methodology in Homogeneous 2.5D Systems

While single-layer optimization approach leads to a better 2.5D system design, to take full advantage of 2.5D integration technology we need to optimize the 2.5D system across all layers. To this end, we have developed a cross-layer co-optimization methodology. The ultimate goal of our cross-layer co-optimization methodology is to jointly maximize performance, minimize manufacturing cost, and minimize peak operating temperature. Our methodology encompasses a wide design space across logical, physical and circuit layers, and integrates multiple simulation tools and analytical models that evaluate aspects of system performance, manufacturing cost, interconnect performance, temperature, and routing.

In this chapter, Section 4.1 first introduces the cross-layer co-optimization problem formulation and the methodology we use to solve it. Figure 4·1 shows our cross-layer methodology and provides an outline of upcoming subsections. Section 4.2 describes the optimization knobs in the design space across the logical, physical and circuit layers. These knobs form the basis for modeling the 2.5D network and chiplet placement, and enable cross-layer optimization. Section 4.3 presents the tools and evaluation framework that models the 2.5D system and evaluates the system metrics of performance, power, temperature and cost. We present five tools that work within the framework to evaluate these system metrics: (1) System Performance Oracle that

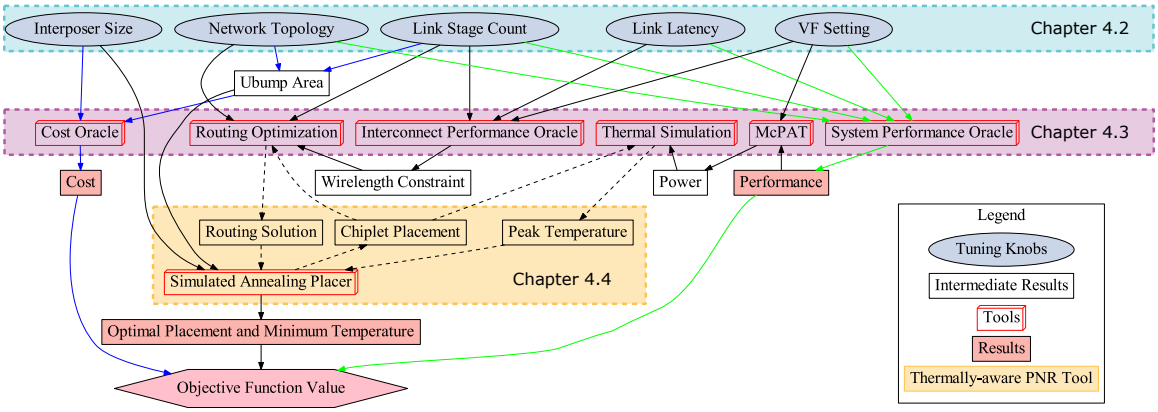

Figure 4·1: Cross-layer co-optimization methodology.

uses Sniper (Carlson et al., 2011) and McPAT (Li et al., 2009); (2) Cost Oracle that computes the manufacturing cost of the 2.5D system; (3) Interconnect Performance Oracle that uses HSPICE (Meta-software, 1996) simulations to evaluate the interconnect circuit timing; (4) Thermal Analysis Tool that uses HotSpot (Zhang et al., 2015) to evaluate the temperature; and (5) Routing Optimizer that uses an MILP to solve for the optimal routing solution and the corresponding maximum wirelength. Section 4.4 demonstrates the thermally-aware place and route (PNR) tool that is based on simulated annealing and interactively uses the oracles described in Section 4.3 to explore the chiplet placement solution space to minimize operating temperature and meet routing constraints.

4.1 Optimization Problem Formulation and Methodology

Our objective is to jointly maximize performance, minimize manufacturing cost, and minimize peak operating temperature. While minimizing temperature for longer system lifetime, we also maintain the peak temperature below a threshold to avoid failures. We explore various network topologies, link options (stage count and latency), interposer sizes, frequency and voltage settings, and chiplet placements to find an optimal solution that is routable and thermally-safe. Ensuring that timing is met

across the inter-chiplet links is crucial for the design, and the placement and routing have a dramatic impact on closing timing. The temperature threshold is relatively negotiable, as there is usually some headroom between the threshold and the actual temperature that causes rapid failures. Exceeding the temperature threshold (85 ^oC in our case) by a few degrees would not immediately burn the system, and the impact on system lifetime could be alleviated by applying reliability management techniques that stress different parts of a chip over time. Thus, in the objective function we apply a soft constraint for peak temperature instead of a hard constraint. We use the notations listed in Table 4.1 to formulate our optimization problem as follows:

Minimize:

$$\alpha \times (\frac{1}{IPS})_{norm} + \beta \times Cost_{norm} + \gamma \times T_{norm} + \eta \times g(T, T_{th}) \tag{4.1}$$

Subject to:

$$g(T, T_{th}) = \frac{1}{10}(max(T - T_{th}, 0))^2 \tag{4.2}$$

$$L \ \leq \ L_{th} \tag{4.3}$$

$$w_{int} \ \leq \ 50 \tag{4.4}$$

$$max(|X_i - X_j|, |Y_i - Y_j|) \geq \frac{w_{2D}}{4} + 2 \times w_{ubump} + w_{gap}, \forall i, j, i \neq j \tag{4.5}$$

Equation (4.1) is the cross-layer objective function, which jointly maximizes performance (IPS) while minimizing manufacturing cost ($Cost$) and peak operating temperature (T). We normalize each term using Min-Max Scaling ($X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$) to reduce the impact of imbalanced ranges and values of raw data. α, β, and γ are user-specified weights having no units, and we set the sum of α, β, and γ to 1. The

Table 4.1: Notations used in the cross-layer co-optimization methodology.

Notation	Meaning
α, β, γ	Coefficients for the cross-layer objective function.
η	Penalty function weight.
IPS	Instructions per nanosecond as a performance metric.
$Cost$	Manufacturing cost of the 2.5D system.
T	Peak operating temperature of the 2.5D system.
T_{th}	Peak temperature threshold of 85 ^oC.
L	Maximum wirelength in the routing solution.
L_{th}	Maximum wirelength threshold to meet transmission timing.
w_{int}	Interposer edge width.
w_{2D}	Width of the 2D chip: 18 mm.
w_{ubump}	microbump stretch-out width from original chiplets. Stretch-out width corresponds to the necessary increase of chiplet's dimensions to accommodate the microbumps needed for the off-chiplet communication.
w_{gap}	Minimum gap width between two adjacent chiplets.
X_i, Y_i	Left bottom x- and y-coordinates for chiplet i.

last term $g(T, T_{th})$ is the penalty function for peak temperature, and η is the penalty weight. It is important to pick an appropriate value for η for a soft-temperature-constrained problem. If η is too small, the optimization problem has no thermal constraint, but if η is too large, the optimization problem effectively becomes a hard-temperature-constrained problem. In our case, we explore a range of η from 0.001 to 1 and pick η to be 0.01, which gives a good balance between not having any constraint and having a hard temperature constraint. Equation (4.2) describes the penalty function. The penalty term is zero when T meets the threshold T_{th}, and positive otherwise. We use a quadratic function instead of a linear function to suppress the penalty for a small violation and highlight the penalty for a large violation. Equation (4.3) is the routing constraint, where the wirelength must be shorter than the reachable length for a given voltage-frequency setting and target latency (see Figure 4·5). Equation (4.4) constrains the interposer size to be no larger than 50 $mm \times 50$ mm, which is within the exposure field size of 2X JetStep Wafer Stepper (Cochet et al., 2014)

and avoids extra stitching cost. Equation (4.5) ensures there is no overlap between chiplets.

To solve the optimization problem, we integrate simulation tools and analytic models discussed in Section 4.3. We first generate a complete table of all the combinations of network topologies, inter-chiplet link stage counts and latencies, voltage-frequency settings, and interposer sizes (see Section 4.2). We precompute system performance, power, allowable inter-chiplet link length, and manufacturing cost for each entry in the table. We normalize the performance as well as the cost, and compute the weighted sum of the first two terms in the objective function ($\alpha \times (1/IPS)_{norm} + \beta \times Cost_{norm}$), and denote it as $Obj2$, where 2 indicates the number of terms. We then sort the table entries based on the values of $Obj2$ in ascending order. To get the temperature term for each table entry, we build a thermally-aware PNR tool to determine the chiplet placement that minimizes the system operating temperature while meeting the routability requirement (see Section 4.4). For our design-time optimization, we assign the worst-case power, which is the highest core power among 256 cores of high-power application `cholesky`, to all the cores while determining the optimal chiplet placement using our thermally-aware PNR tool. Then, we run real applications on top of the optimal chiplet placement to get the actual application temperature. Our thermally-aware PNR tool iterates chiplet placement, and interactively evaluates peak operating temperature and maximum inter-chiplet wirelength of each placement. Each temperature simulation takes approximately 30 seconds and each routing optimization takes a few seconds to 10 minutes. For manageable simulation time, for each table entry we limit the number of placement iterations to 1000, while determining the minimum peak temperature.

To speed up the simulation, we progressively reduce the number of table entries for which we need to complete the thermally-aware PNR process, which determines

the minimum peak temperature and the corresponding chiplet placement for each table entry. Once the process completes for a table entry, all the terms (performance, cost, temperature, and penalty) in the objective function for that table entry become available. We add up the four terms to get the objective function value of the entry, and denote it as $Obj4$, where 4 indicates the number of terms. We keep track of the minimum of the available $Obj4$ values using $Obj4_{min}$. For the entries whose $Obj2$ value is greater than $Obj4_{min}$, there is no need to run the thermally-aware PNR tool, since the tool cannot find a solution whose $Obj4$ value is less than $Obj4_{min}$. We start the thermally-aware PNR process with the entries in the sorted order based on $Obj2$ values, progressively removing the entries that have no chance to be optimal, and stop when all the remaining entries have available temperature and $Obj4$ values. Using this technique of progressively reducing solution space, we achieve 6× speedup for the performance-focused case ($(\alpha, \beta, \gamma) = (0.8, 0.1, 0.1)$), 7.8× speedup for the cost-focused case ($(\alpha, \beta, \gamma) = (0.1, 0.8, 0.1)$), and 1.5× speedup for the case that jointly focuses on performance, cost, and temperature ($(\alpha, \beta, \gamma) = (0.333, 0.333, 0.333)$). For the temperature-focused case ($(\alpha, \beta, \gamma) = (0.1, 0.1, 0.8)$), we only achieve 1.02× speedup because the temperature term dominates, and thus, we can barely rule out any of the table entries using the $Obj2$ and $Obj4_{min}$ comparison. In this paper, our experiments are based on the performance-focused case.

4.2 Cross-layer Optimization Knobs

4.2.1 Logical Layer

One of the main questions in 2.5D logical design is how to connect multiple chiplets using the interposer. In the logical layer, we explore two types of network topologies for 2.5D systems. In Figure 4·2, we show the logical views of network topologies. These views only illustrate the logical connections and not the actual chiplet placement.

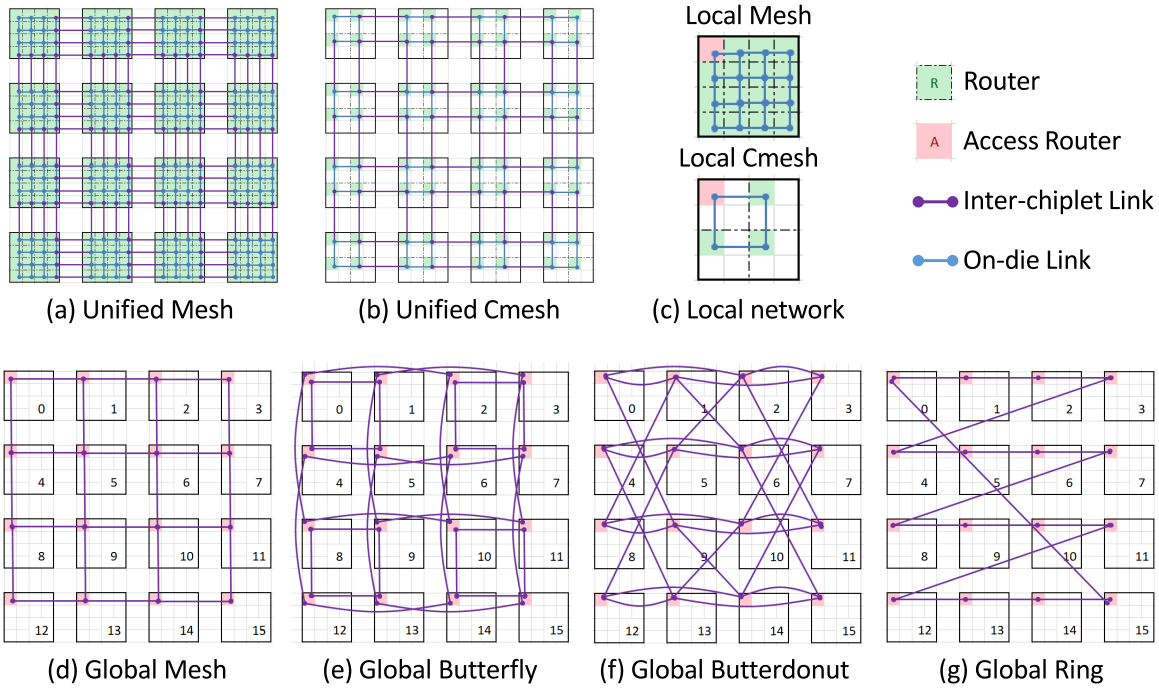

Figure 4·2: Logical view of network topologies. (a)-(b) are unified networks, (c)-(g) are used to form hierarchical networks.

The first type is a unified network, which directly maps a NoC topology designed for a 2D system onto a 2.5D system to preserve the same logical connections and routing paths. We explore Unified-Mesh (U-M), where each core has a router, and Unified-Cmesh (U-CM), where four cores share a router, as shown in Figure 4·2(a)-(b). Unlike single-chip NoCs, the source and the destination of a logical channel in 2.5D systems may not reside on the same chiplet. The inter-chiplet link has to travel through the silicon interposer, which may not always meet the single-cycle latency due to long physical wires. In our evaluation, we consider inter-chiplet links with latencies varying from single cycle to five cycles.

The second type is a hierarchical network, which breaks down the overall network into two levels: one level has multiple disjoint local networks and the other level has a global network. In 2.5D systems, each chiplet has an on-chip local network and an access router. The global network hooks up all the access routers using inter-

chiplet links embedded in the interposer. Intra-chiplet packets travel through the local network, while inter-chiplet packets first travel through the local network to the access router of the source chiplet, then use the global network to reach the access router of the destination chiplet, and finally use the local network of the destination chiplet to reach the destination. The local network and the global network can be designed independently. For local networks, we explore Mesh (M) and Cmesh (CM) topologies (Figure 4·2(c)); while for global networks, we explore Mesh (M), Butterfly (BF), Butterdonut (BD) (Kannan et al., 2015) and Ring (R) topologies, (see Figure 4·2(d)-(g)). We use *G-X-L-Y* notation to denote a hierarchical network, where X and Y correspond to the global and local network topologies, respectively.

4.2.2 Physical Layer

Physical design of 2.5D systems determines the chiplet placement and a routing solution, subject to the chosen network topology. The placement of chiplets not only impacts the system temperature profile, but also affects the inter-chiplet link lengths. The routing solution affects the microbump assignment and circuit choice of inter-chiplet links. In our approach, we explicitly evaluate the area overhead of microbumps and the inter-chiplet link transceivers that are placed along the peripheral regions of the chiplets.

Microbumps connect chiplets and the interposer. Inter-chiplet signals first exit the source chiplet through microbumps, travel along the wires in the interposer, and then pass through microbumps again to reach the destination chiplet. Microbumps are typically placed along the periphery of the chiplet, for the purpose of signal escaping (Radojcic, 2017). The microbump area overhead is determined by the number of inter-chiplet channels, channel bandwidth, and microbump pitch. We list the microbump area overhead for various network topologies in Table 4.2, where we use a 128-*bit* wide bus for each channel, 45 μm microbump pitch, and 4.5 $mm \times 4.5$ mm

Table 4.2: microbump count, stretch-out width of microbump region (w_{ubump}), and microbump area (A_{ubump}) overhead per chiplet for different network topologies designed using repeaterless links, 2-stage and 3-stage *gas-station* links.

		Unified Mesh	*Unified Cmesh*	*Global Mesh*	*Global Butterfly*	*Global Butterdonut*	*Global Ring*	*Global Clos*
#bidirectional inter-chiplet channels		16	8	4	4	4	2	32
repeaterless links	**#microbumps**	4916	2458	1229	1229	1229	615	9831
	w_{ubump} (mm)	0.54	0.27	0.135	0.135	0.135	0.09	0.945
	A_{ubump} Overhead (%)	53.8	25.4	12.4	12.4	12.4	8.2	101.6
2-stage *gas station*	**#microbumps**	9831	4916	2458	2458	2458	1229	19661
	w_{ubump} (mm)	0.945	0.54	0.27	0.27	0.27	0.135	1.665
	A_{ubump} Overhead (%)	101.6	53.8	25.4	25.4	25.4	12.4	202.8
3-stage *gas station*	**#microbumps**	14746	7373	3687	3687	3687	1844	29492
	w_{ubump} (mm)	1.305	0.72	0.405	0.405	0.405	0.225	2.25
	A_{ubump} Overhead (%)	149.6	74.2	39.2	39.2	39.2	21.0	300.0

chiplet size, and assume 20% additional microbumps are reserved for power delivery and signal shielding (Radojcic, 2017). Here, w_{ubump} indicates the stretch-out width from the chiplet edge to accommodate the microbumps, as shown in Figure 4·3. In Table 4.2, we also include Global Clos topology (Joshi et al., 2009), which is a commonly used low-diameter-high-radix network. However, the area overhead is too high to make Clos a feasible inter-chiplet network option.

Inter-chiplet links can be routed on either a passive interposer or an active interposer. An active interposer enables better link bandwidth and latency because re-

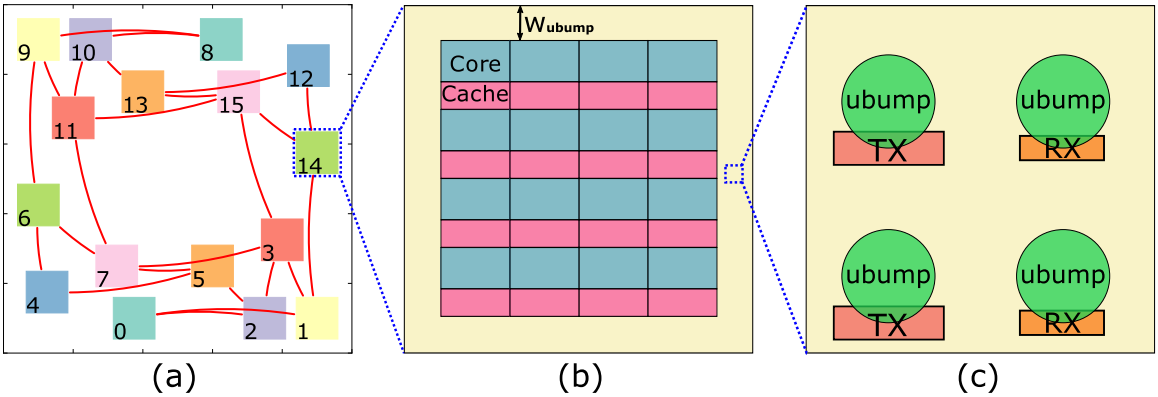

Figure 4·3: Illustration of (a) chiplet placement on an interposer with logical connections, (b) a chiplet with microbump overhead, and (c) microbumps with TX/RX regions (not drawn to scale).

peaters and flip-flops (for pipelining) can be inserted in the interposer (Parès, 2013). However, an active interposer is expensive due to the use of a FEOL (front-end-of-line) process and associated yield loss. A passive interposer is a cost-effective alternative. The passive interposer is transistor-free, can be fabricated in a BEOL (back-end-of-line) process, and inherently has high yield (Parès, 2013). We conducted a study of the performance benefit of an active interposer over a passive interposer. We observed 2× to 3× latency improvement for the same link length, or 50% longer maximum allowed link length for the same throughput, but these benefits come at a 10× cost overhead ($500 per wafer for passive interposer vs. $5000 per wafer for active interposer (Parès, 2013)). Due to this cost overhead, we focus on the passive interposer in our present study. Active interposers, however, are currently being considered for 2.5D systems (Jerger et al., 2014), (Kannan et al., 2015). Our methodology can be easily extended to active interposers, and we leave this as future work.

4.2.3 Circuit Layer

In the circuit layer, we explore multiple circuit designs for inter-chiplet links. Due to the high cost of an active interposer, we do not consider repeatered links. A link on a passive interposer is naturally repeaterless and non-pipelined because active components such as repeaters or pipelines cannot be placed in a passive interposer. Such a link has limited performance, especially in 2.5D systems, where inter-chiplet links may need to reach a few *cm*. Essentially, a passive interposer cannot always ensure single-cycle communication latency due to signal degradation and rise-/fall-time constraints. Hence, we explore a range of repeaterless inter-chiplet link (Path 1 in Figure 4·4) latencies from single cycle to five cycles, which corresponds to a variety of inter-chiplet link lengths (see Figure 4·5). This provides sufficient flexibility in chiplet placement. In addition, we propose a novel '*gas-station*' link design (Coskun et al., 2018), which enables pipelining in a passive interposer, to overcome the performance

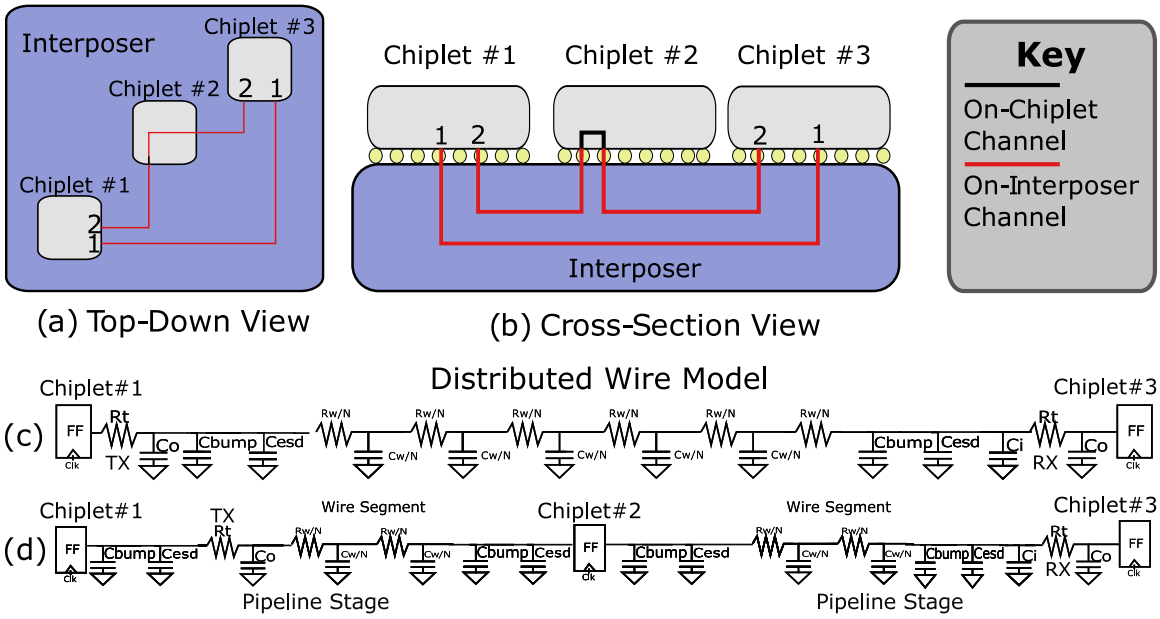

Figure 4·4: Illustration of (a) top-down view and (b) cross-section view of inter-chiplet link implementation, and distributed wire models for (c) repeaterless link (Path 1 in (a)-(b)) and (d) *gas-station* link (Path 2 in (a)-(b)).

loss. Unlike repeated links or pipelined links, which place repeaters or flip-flops in an active interposer, our '*gas-station*' link leverages flip-flops placed on other chiplets along the way to 'refuel' a passive link. As shown in Figure 4·4, Chiplet #2 is a *gas station* for Path 2 from Chiplet #1 to Chiplet #3, where signals first enter Chiplet #2 through microbumps, get repeated or retimed, and then return to the passive interposer through microbumps. Here we trade off microbump area overhead computed in Table 4.2 for performance. It is important to note the differences between an inter-chiplet repeaterless pipelined link and a *gas-station* link (Coskun et al., 2018). A repeaterless pipelined link requires an active interposer to house flip-flops and these flip-flops are designed using the active interposer's technology node. A *gas-station* link only needs a passive interposer and inserts active elements in the intermediate chiplets. Thus, the active elements are designed using the chiplets' technology node (22 nm in our case). In our analysis, we set t_{rise}/t_{cycle} upper bound to be 0.5 and

Table 4.3: Technology node parameters.

Technology Node	**22 nm**	**65 nm**
Wire Thickness	300 nm	1.5 μm
Dielectric Height	300 nm	0.9 μm (Karim et al., 2013)
Wire Width	200 nm	1 μm (Radojcic, 2017)
C_{bump}	4.5 fF	4.5 fF (Karim et al., 2013)
C_{esd}	50 fF	50 fF (Karim et al., 2013)
C_{g_t} (Gate Cap)	1.08 $fF/\mu m$	1.05 $fF/\mu m$
C_{d_t} (Drain Cap)	$1.5 \times C_g$	$1.5 \times C_g$
R_t (Inverter resistance)	450 $\Omega \cdot \mu m$	170 $\Omega \cdot \mu m$
Driver NMOS Sizing	22 $nm \times 100$	65 $nm \times 100$
Wire Pitch	0.4 μm	2 μm (Radojcic, 2017)
Flip-Flop Energy per Bit	14 fJ/bit (Chen et al., 2015)	28 fJ/bit (Knudsen, 2008)
Flip-Flop $t_{c-q} + t_{setup}$	49 ps (Chen et al., 2015)	70.9 ps (Consoli et al., 2012)

ensure full voltage swing at all nodes in the inter-chiplet link to account for non-idealities such as supply noise and jitter. We also explore t_{rise}/t_{cycle} of 0.8, which allows signals to go longer distances without repeaters. Relaxing the clock period or allowing for multi-cycle bit-periods permits us to use longer inter-chiplet links.

Figure 4·4(c) and (d) show the distributed circuit models in a passive interposer for repeaterless link and *gas-station* link, respectively. We model wire parasitics using a distributed, multi-segment π model. We use 22 nm technology parameters for intra-chiplet components (drivers, receivers, repeaters, and flip-flops) and 65 nm parameters for the inter-chiplet wires. Table 4.3 shows technology parameters used in our experiments. We calculate capacitance and resistance based on the model in Wong *et al.* (Wong et al., 2000), and we calibrate our stage and path delay estimates based on extraction from layout and Synopsys PrimeTime timing reports. Figure 4·5 shows maximum reachable wirelengths that meet both the propagation time constraint and the rise-time constraint for various frequencies and cycles. For a given rise time constraint, as the inter-chiplet link latency constraint increases, the distance that a signal can travel in a single cycle increases. In a single cycle, a signal can travel more

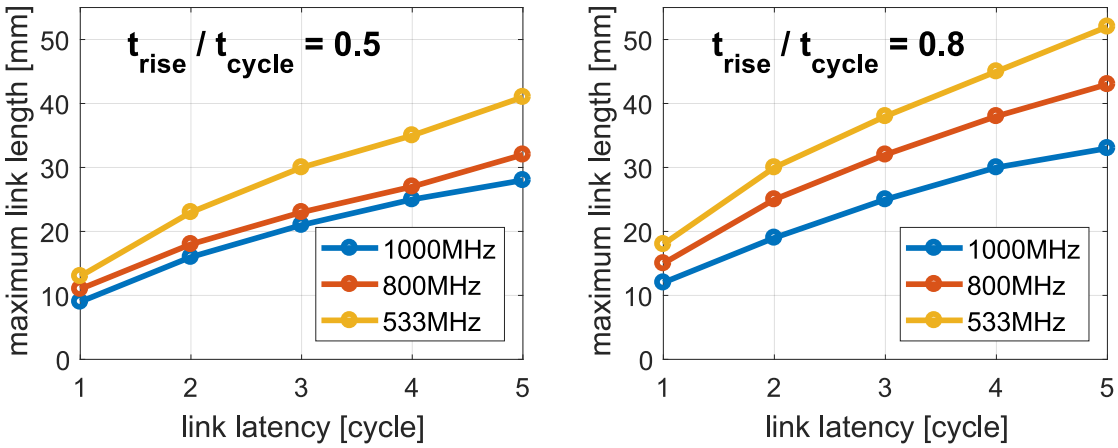

Figure 4·5: Maximum reachable inter-chiplet link length w.r.t. clock cycles for various frequencies and rise-time constraints.

than 10 *mm* owing to the relaxed rise time constraint as well as low interconnect RC parasitics (i.e., due to using an older technology node for the interposer).

4.3 Evaluation Framework

4.3.1 System Performance Oracle

We construct a manycore system performance oracle (evaluation block) that tells us the manycore system performance and core power for a given choice of network topology, voltage-frequency setting, link type, and link latency. We use Sniper (Carlson et al., 2011) to precompute system performance. Our target system has 256 homogeneous cores, whose architecture is based on the IA-32 core from the Intel SCC (Howard et al., 2011), with size and power scaled to 22 *nm* technology (Zhang et al., 2014). We divide the 256-core system into 16 identical chiplets.[1] In Sniper, we implement the unified and hierarchical network models described in Section 4.2.1. For inter-chiplet links, we use either passive links or *gas-station* links (see Section 4.2.2). We vary link latency from one to five cycles for passive links and ex-

[1]Our methodology is applicable to any system with even number of chiplets, each with aspect ratio of 1.

plore 2-stage and 3-stage pipelines for *gas-station* links. We explore three voltage-frequency settings: (0.9 V, 1 GHz), (0.89 V, 800 MHz), and (0.71 V, 533 MHz). We use multi-threaded benchmarks that cover high-power applications (`cholesky` from SPLASH-2 suite (Woo et al., 1995)), medium-power applications (`streamcluster` and `blackscholes` from PARSEC suite (Bienia et al., 2008)), and low-power applications (`lu.cont` from SPLASH-2 suite). We fast-forward the sequential initialization region and simulate 10 billion instructions in the parallel region with all cores active to collect performance statistics. Then, we feed the performance results to McPAT (Li et al., 2009) to compute the core power. We calibrate the McPAT power output with the measured power dissipation data of Intel SCC (Howard et al., 2011), scaled to 22 nm.

4.3.2 Cost Oracle

We construct a cost oracle that computes the manufacturing cost of 2.5D systems for a given choice of network topology, chiplet size and count, link type and stage count, and interposer size. We adopt the 2.5D manufacturing cost model published by Stow (Stow et al., 2017), which takes into account the cost and yield of CMOS chiplets, microbump bonding, and the interposer. The model assumes known-good-dies. We enhance the cost model to account for the impact of microbump overhead on the dies per wafer count and yield. Notations are listed in Table 4.4.

$$A_{chiplet} = (\frac{w_{2D}}{4})^2 \tag{4.6}$$

$$A_{ubump} = (\frac{w_{2D}}{4} + 2 \times w_{ubump})^2 - A_{chiplet} \tag{4.7}$$

$$N_{int} = \frac{\pi \times (\phi_{wafer_{int}}/2)^2}{A_{int}} - \frac{\pi \times \phi_{wafer_{int}}}{\sqrt{2 \times A_{int}}} \tag{4.8}$$

$$N_{chiplet} = \frac{\pi \times (\phi_{wafer}/2)^2}{A_{chiplet} + A_{ubump}} - \frac{\pi \times \phi_{wafer}}{\sqrt{2 \times (A_{chiplet} + A_{ubump})}} \tag{4.9}$$

Table 4.4: Notations used in the cost oracle.

Notation	Meaning
A_{int}	Area of interposer.
$A_{chiplet}$	Chiplet area without microbump overhead.
A_{ubump}	Area of microbump region in a chiplet.
A_{TXRX}	Critical transceiver area in microbump region.
ϕ_{wafer}	Diameter of CMOS wafer: 300 mm.
$\phi_{wafer_{int}}$	Diameter of interposer wafer: 300 mm.
N_{int}	Number of interposer dies per wafer.
$N_{chiplet}$	Number of CMOS dies per wafer.
D_0	Defect density: $0.25/cm^2$ (Stow et al., 2016).
ϵ	Defect clustering parameter: 3 (Stow et al., 2016).
$Y_{chiplet}$	Yield of a CMOS chiplet.
Y_{int}	Yield of an interposer: 98% (Tran et al., 2016).
Y_{bond}	Chiplet bonding yield: 99% (Stow et al., 2016).
C_{wafer}	Cost of CMOS wafer.
$C_{wafer_{int}}$	Cost of passive interposer wafer.
$C_{chiplet}$	Cost of a chiplet.
C_{int}	Cost of an interposer.
C_{bond}	Cost of chiplet bonding.
$C_{2.5D}$	Manufacturing cost of a 2.5D system.

$$Y_{chiplet} = (1 + (A_{chiplet} + A_{TXRX}) \times D_0/\epsilon)^{-\epsilon} \tag{4.10}$$

$$C_{int} = C_{wafer_{int}}/N_{int}/Y_{int} \tag{4.11}$$

$$C_{chiplet} = C_{wafer}/N_{chiplet}/Y_{chiplet} \tag{4.12}$$

$$C_{2.5D} = \frac{C_{int} + \Sigma_1^{16}(C_{chiplet} + C_{bond})}{Y_{bond}^{15}} \tag{4.13}$$

Equation (4.6) computes the equivalent functional area of chiplets generated by dividing a 2D chip. Equation (4.7) evaluates the microbump area overhead, where w_{ubump} is the stretch-out width from original chiplet. Equations (4.8) and (4.9) determine the number of interposer dies and the number of CMOS dies, respectively, that can be cut from a wafer (Stow et al., 2017). Here the first term counts the number of dies purely based on the wafer area and the die area, and the second subtraction

term compensates for incomplete dies along the wafer periphery. In Equation (4.9), we take into account the microbump area overhead A_{ubump}. Equation (4.10) is the negative binomial yield model, where D_0 is the defect density and $\epsilon = 3$ indicates moderate defect clustering (Stow et al., 2017). Unlike the center area of chiplets that has high transistor density, the microbump regions have very limited active regions that contain inter-chiplet link transmitters (TXs) and receivers (RXs). Only the defects occurring in the active regions would cause a failure, while the rest of the passive region is non-critical. Hence, our yield calculation (Equation (4.10)) uses only the critical active area. The yield of a passive interposer is as high as 98% (Tran et al., 2016) because it does not have any active components. Equations (4.11) and (4.12) calculate the per-die cost of the interposer and the chiplets, respectively. Equation (4.13) estimates the overall manufacturing cost of the 2.5D system by adding up the costs of the chiplets, the interposer, and bonding.

Figure 4·6 shows the manufacturing cost of 2.5D systems with respect to interposer sizes from 20 mm to 50 mm for two different microbump stretch-out widths, which correspond to the minimum value (for *G-R-L-M/CM* topology without *gas stations*) and maximum value (for *U-M* topology with 3-stage *gas-station* links) in our experiments. The 2.5D system costs are normalized to the cost of 2D system. The 2.5D system cost increases with the interposer size. The cost model in our prior work (Eris et al., 2018) did not consider microbump overhead and thus, the 2.5D system cost was independent of w_{ubump}. The cost model in our latest work (Coskun et al., 2018) overestimated the yield drop due to microbump regions and thus, overestimated the overall cost. This error of this cost model (Coskun et al., 2018) is trivial with a small w_{ubump}, but with a large w_{ubump}, the error is not negligible (up to 10% of the 2D system cost in our example). With a small w_{ubump}, the predicted cost of a 2.5D system using our enhanced model is cheaper than the cost of a 2D system, when

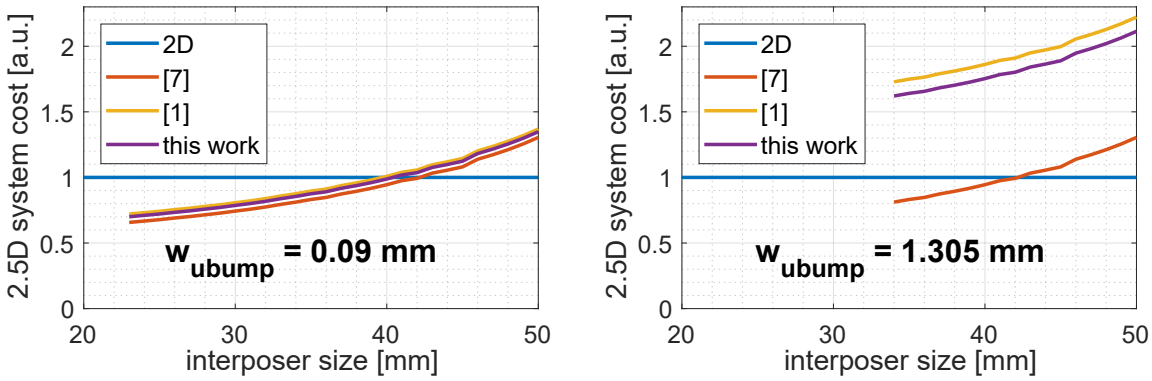

Figure 4·6: Comparison between the cost of a 2D system, and the cost of a 2.5D system estimated using prior cost models (Eris et al., 2018), (Coskun et al., 2018) and our enhanced cost model for interposer sizes from 20 mm to 50 mm and microbump stretch-out widths (w_{ubump}) of 0.09 mm and 1.305 mm, which correspond to the lower and upper limits of w_{ubump} in our analysis, respectively.

the interposer is smaller than 40 $mm \times 40$ mm. With a large w_{ubump}, the predicted cost of a 2.5D system using our enhanced model is always higher than that of a 2D system. This eliminates some network topologies, such as Clos, that require large w_{ubump}.

4.3.3 Interconnect Performance Oracle

We build an interconnect performance oracle that analyzes the maximum reachable length of an inter-chiplet link for a given operating voltage and frequency, rise-time constraint, and propagation time constraint in the unit of cycles. We use HSPICE (HSPICE, 2009) to simulate the link models discussed in Section 4.2.3. The TX circuit is designed using up to six (the exact number depends on the wirelength) cascaded inverters with standard fan-out of 4, and the RX circuit consists of two cascaded inverters of the minimum size. We estimate the TX and RX area using the physical layout of the standard inverter cell in NanGate 45 nm Open Cell Library (Knudsen, 2008), and scale it down to 22 nm technology. The area of TX and RX logic (A_{TXRX})

takes up less than 1% of the microbump area. The interposer wire resistance is $14.66\times 10^{-3}\ \Omega/\mu m$ and the capacitance is $114.72 \times 10^{-3}\ fF/\mu m$, for the wire dimensions provided in Table 4.3 for 65 nm technology. Since the inter-chiplet link latency is wire dominated, we set a sizing upper limit of $100\times$ the minimum size for the last inverter in the set of cascaded inverters of TX in 22 nm technology since the drivers are placed in chiplets instead of the interposer. We do not increase the size beyond $100\times$ because we do not observe latency improvement. For the workloads that we have considered, the inter-chiplet link power is up to 22 W, which is insignificant compared to the total average system power of 508 W. Hence, inter-chiplet link power has negligible influence on chiplet placement.[2]

4.3.4 Thermal Simulation

We use HotSpot (Zhang et al., 2015) to simulate thermal profiles for given chiplet placement choices and core power values. We use an extension of HotSpot (Meng et al., 2012) that provides detailed heterogeneous 3D modeling features. To model our 2.5D system, we stack several layers of different thickness and heterogeneous materials on top of each other and model each layer with a separate floorplan on a 64×64 grid. Our 2.5D system model follows the properties (such as layer thickness, materials, dimensions of bumps and TSVs) of real systems (Chaware et al., 2012), (Charbonnier et al., 2012). We use the HotSpot default conventions for the thermal interface material properties, the ambient temperature of 45 ^oC, and the sizing of the spreader and the heatsink such that the spreader edge size is $2\times$ the interposer edge size and the heatsink edge size is $2\times$ the spreader edge size. To keep the heat transfer coefficient consistent across all simulations, we adjust the convective resistance of the heatsink.

[2]If link power were to increase substantially, this would affect the system temperature, which in turn would affect the chiplet placement.

We implement a linear model of temperature-dependent leakage power based on published data of Intel 22 nm processors (Wong, 2012). We assume 30% of power is due to leakage at 60 ^oC (Zhang et al., 2014). We update the core power to include the leakage power based on initial temperature obtained from HotSpot and iterate the thermal simulation. In all of our studies, the leakage-dependent temperature quickly converges after two iterations.

Figure 4·7 shows the temperature of the best chiplet placement for each interposer size, while running `cholesky` benchmark with *Mesh* network using single-cycle links without *gas stations*. As the interposer size increases, the peak temperature decreases due to the increasing flexibility of chiplet placement. Although the main direction of heat dissipation is vertical through the heatsink on top of the system and the lateral heat transfer is relatively weak, the effect of lateral heat flow is sufficient to motivate thermally-aware chiplet placement (Zhang et al., 2017). The temperature benefit shown in Figure 4·7 comes at the cost of a larger interposer. The cost of the interposer has been accounted in our cost model and the user can adjust the cost weight in the objective function for different design needs.

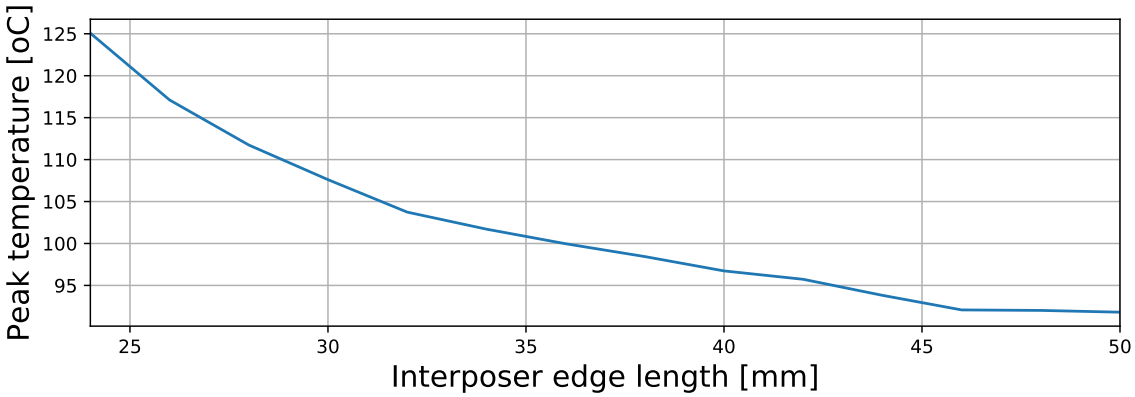

Figure 4·7: Temperature of best chiplet placement for each interposer size, running `cholesky` with *Mesh* network using single-cycle link without *gas stations*.

4.3.5 Routing Optimization

We build an MILP to solve for the optimal routing solution and the corresponding maximum wirelength given the logical network topology, chiplet placement, link stage count, and microbump resources. The MILP objective is a weighted function of the maximum length of a route on the interposer and the total routing area overhead. We group the microbumps along the chiplet periphery into pin clumps to limit the problem size and the MILP runtime. We use 4 pin clumps per chiplet in our experiments. We frame the delivery of required number of wires between chiplets as multi-commodity flow, and formulate the MILP to find optimal routing solutions that encompass the finite availability of microbumps in each pin clump.

Table 4.5 describes the notations used in the MILP. We use ILOG CPLEX v12.5.1 to implement and run the MILP. The number of variables and constraints in the MILP instance are both bounded by $O(|C|^2 \cdot |P|^2 \cdot |N|)$. For our 16-chiplet design, $|N|$ is 48 for Mesh/Cmesh, 56 for Butterdonut, 64 for Butterfly and 32 for Ring networks. The outputs of our MILP implementation are the optimal value of the objective function and the values of the variables f^n_{ihjk}, which describe the routing solution and microbump assignment to pin clumps.

Based on the inputs to the routing optimization step (see Table 4.6), we precompute d_{ihjk}, the routing distance (assuming Manhattan routing) from pin clump h on chiplet i to pin clump k on chiplet j, using Equation (4.14). Equation (4.15) is the objective function for the MILP that includes the maximum length L, and the total length of the routes. In all reported experiments, we set $\theta = 1$ and $\varphi = 0$. Equation (4.16) ensures that the flow variable f^n_{ihjk} is a non-negative number. Equation (4.17) is the flow constraint governing the flow variables f^n_{ihjk}. It guarantees the sum of all flows for a net n, over all pin clumps from chiplet s_n to chiplet t_n, meets the R_{ij} requirement. It also makes sure that net flow is 0 for all other (non-source, non-sink)

Table 4.5: Notations used in routing optimization.

Notation	Meaning
C	Set of chiplets.
P	Set of pin clumps.
N	Set of nets.
c, i, j	Index of a chiplet $\in C$.
p, h, k	Index of a pin clump $\in P$.
n	A net $\in N$.
s_n	Source chiplet of net n.
t_n	Sink chiplet of net n.
x_p, y_p	x- and y-offsets from left bottom of the chiplet for pin clump p.
d_{ihjk}	Distance from pin clump h on chiplet i to pin clump k on chiplet j. Note that $d_{ihjk} = d_{jkih}$.
P_{ih}^{max}	Pin capacity for a pin clump h on chiplet i.
R_{ij}	Input requirement on the wire count between chiplet i and chiplet j.
f_{ihjk}^{n}	Flow variable. Number of wires from pin clump h of chiplet i to pin clump k of chiplet j that belong to net n.
λ_{ihjk}^{n}	Binary indicator for a route between pin clump h on chiplet i to pin clump k on chiplet j belonging to net n.
S_{max}	Maximum permissible segment count allowed for any route; a segment is defined as a route between chiplets. For the case where no *gas stations* are permitted, $S_{max} = 1$. Permitted values of S_{max} include 1, 2 or 3.
θ, φ	Coefficients for the objective function of routing optimization.

chiplets for the given net. $\sum_{h\in P, j\in C, k\in P} f_{ihjk}^{n}$ is the outgoing flow of chiplet i, while $\sum_{h\in P, j\in C, k\in P} f_{jkih}^{n}$ is the incoming flow of chiplet i. Equation (4.18) assures that there is no input flow (for net n) for any pin clump in the source chiplet s_n from any other chiplet's pin clump. Similarly, Equation (4.19) ascertains that there is no output flow (for net n) for any pin clump in the sink chiplet t_n to any other chiplet's pin clump. Equation (4.20) maintains that the sum of input and output flows from a given pin clump is always less than or equal to the capacity of the pin clump. This insures that all routes have available pins. Equation (4.21) defines λ_{ihjk}^{n} as a boolean value based on f_{ihjk}^{n}. This helps identify the maximum route length L, as shown in Equation

Table 4.6: Inputs to routing optimization.

Input	Properties
Chiplets	$\|C\|$ Chiplet instances, at $\{X_c, Y_c\}$ left bottom, $c \in C$. The locations provided for the chiplets are assumed to be legal.
Pin Clumps	$\|P\|$ Pin clump instances of pin capacity P_{ih}^{max} each. Each pin clump p has a predetermined location $\{x_p, y_p\}$ relative to the left bottom of the chiplet.
Required Connections	R_{ij} between every pair of chiplets $\{i, j\}$ indicating the number of wires that need to go between the pair of chiplets. If $R_{ij} > 0$ then a net n exists between chiplet i and chiplet j with source $s_n = i$ and sink $t_n = j$.
Routing Rules	Maximum number of segments, S_{max} equal to 1, 2 or 3. $S_{max} \leq 3$ to limit impact on latency.

(4.22). Equation (4.23) constrains the maximum number of segments (S_{max}) to be either 1, 2 or 3. A segment is defined as a portion of the net connecting two chiplets. If $S_{max} = 1$, then the net connects s_n and t_n directly, and no *gas stations* are permitted, while if $S_{max} = 2$ or $S_{max} = 3$, then *gas stations* are permitted, where the net connects s_n and t_n through 1 or 2 other chiplets respectively, i.e. *gas station* hops.

$$d_{ihjk} = |X_i + x_h - X_j - x_k| + |Y_i + y_h - Y_j - y_k| \tag{4.14}$$

Minimize:

$$\theta \cdot L + \varphi \cdot \sum_{i \in C, h \in P, j \in C, k \in P, n \in N} d_{ihjk} \cdot f_{ihjk}^{n} \tag{4.15}$$

Subject to:

$$f_{ihjk}^{n} \geq 0, \qquad \forall i \in C,\, h \in P,\, j \in C,\, k \in P, n \in N \tag{4.16}$$

$$\sum_{h\in P,j\in C,k\in P} f^n_{ihjk} - \sum_{h\in P,j\in C,k\in P} f^n_{jkih} = \begin{cases} R_{s_nt_n}, & \text{if } i = s_n, \forall n \in N \\ -R_{s_nt_n}, & \text{if } i = t_n, \forall n \in N \\ 0, \forall i \neq s_n || t_n, \forall n \in N \end{cases} \tag{4.17}$$

$$f^n_{jks_nh} = 0, \qquad \forall n \in N, \forall h \in P, \forall j \in C, \forall k \in P \tag{4.18}$$

$$f^n_{t_nhjk} = 0, \qquad \forall n \in N, \forall h \in P, \forall j \in C, \forall k \in P \tag{4.19}$$

$$\sum_{j\in C,k\in P,n\in N} f^n_{ihjk} + \sum_{j\in C,k\in P,n\in N} f^n_{jkih} \leq P^{max}_{ih}, \quad \forall i \in C, h \in P \tag{4.20}$$

$$\lambda^n_{ihjk} = \begin{cases} 1 & \text{if } f^n_{ihjk} > 0, \forall i \in C, h \in P, j \in C, k \in P, n \in N \\ 0 & \text{otherwise}, \forall i \in C, h \in P, j \in C, k \in P, n \in N \end{cases} \tag{4.21}$$

$$L \geq d_{ihjk} \cdot \lambda^n_{ihjk}, \forall i \in C,\ h \in P,\ j \in C,\ k \in P,\ n \in N \tag{4.22}$$

$$\sum_{i\in C,h\in P,j\in C,k\in P} f^n_{ihjk} \leq \begin{cases} R_{s_nt_n}, \text{ if } S_{max} = 1 \\ 2 \cdot R_{s_nt_n} - \sum_{h\in P,k\in P} f^n_{s_nht_nk}, \text{ if } S_{max} = 2 \\ 3 \cdot R_{s_nt_n} - 2 \cdot \sum_{h\in P,k\in P} f^n_{s_nht_nk} - \\ \sum_{i\in C|i\neq s_n||t_n} min(\sum_{h\in P,k\in P} f^n_{s_nhik}, \\ \sum_{h\in P,k\in P} f^n_{ikt_nh}), \text{if } S_{max} = 3 \end{cases} \tag{4.23}$$

4.4 Thermally-Aware Placement Algorithm

Our thermally-aware PNR tool supports arbitrary chiplet placements that consider non-matrix and asymmetric chiplet organization styles while searching for the optimal placement for each table entry. Including arbitrary placements, the solution space explodes to quadrillions (10^{15}) of placement options with 1 mm granularity. It is impractical to exhaustively search such a vast space. In addition, the solution space is non-convex. Approaches like gradient descent or greedy search (Eris et al., 2018) can easily get trapped in a local minima. Therefore, we use simulated annealing to explore chiplet placement and find the optimal placement solution that gives lowest peak temperature while meeting the maximum wirelength. Simulated annealing is a probabilistic technique to approximate the global optimum. We introduce the key components of our algorithm below.

4.4.1 Placement Description

Prior works (Eris et al., 2018), (Coskun et al., 2018) only consider 4×4 matrix-style chiplet placement, which covers a small portion of the overall solution space and the chiplets have limited freedom to move. For example, the corner chiplets cannot move, the edge chiplets can only slide along the periphery of the interposer, and the center chiplets can only slide along the interposer diagonal. Thus, the previous approach of matrix-style chiplet placement cannot cover the cases where the four chiplets along an edge of the interposer do not align or the cases where the first row does not always have four chiplets. In addition, the previous assumption of 4-fold rotational symmetry does not allow us to ever find the optimal placement for some topologies. For Butterdonut and Butterfly networks, because of the 4-fold rotational symmetry, the maximum wirelength cannot be shortened with chiplet movement due to the connection between a chiplet and its reflection in any one of the remaining quadrants. Therefore, we

enhance our cross-layer co-optimization methodology to support arbitrary placement and relax our symmetry assumption to 2-fold rotational symmetry. We use x- and y-coordinates to specify the locations of the first eight chiplets, and the coordinates of the remaining eight chiplets are based on the rotational image of the first eight. We assume 1 mm granularity for placement, such that the coordinates of the center of each chiplet has to be positive integer numbers. The chiplets cannot overlap with each other and there is a 1 mm guardband along the interposer periphery. The minimum gap between two chiplets is 0.1 mm (Chaware et al., 2012), (Murayama et al., 2013).

4.4.2 Neighbor Placement

A neighbor placement is the placement obtained by either moving a chiplet by the minimum step size in any of the 8 directions (N, S, E, W, NE, NW, SE, SW) or swapping a pair of chiplets from a current placement. Without swapping, it is likely to have a 'sliding tile puzzle' issue, i.e., that a chiplet cannot move in some directions because other chiplets block the way, especially when the interposer size is small.

4.4.3 Acceptance Probability

The decision of whether a neighbor placement is accepted or not depends on the *delta* calculated using Equation (4.24). Here T_{curr}, L_{curr}, T_{nei}, L_{nei} are the peak temperature of current placement, the longest wirelength of current placement, the peak temperature of neighbor placement, and the longest wirelength of neighbor placement, respectively. When both the current placement and the neighbor placement meet the wirelength constraint, we emphasize the temperature difference when calculating *delta*. Similarly, when either the neighbor or the current placement violates the wirelength constraint, we emphasize the wirelength difference while calculating *delta* as there is no point in considering temperature because we do not have a viable solution. We compute the acceptance probability AP using Equation (4.25), where K

is the annealing temperature. Here K decays from 1 to 0.01 with a factor of 0.8 every v iterations, where v is proportional to the interposer edge width w_{int}. We accept the neighbor placement if AP is greater than a random number between 0 and 1. In the case that a neighbor placement is better ($delta > 0$), AP evaluates to greater than 1 and we are forced to accept the neighbor placement. In the case that a neighbor placement is worse ($delta < 0$ and $0 < AP < 1$), there is still a nonzero probability of accepting the worse neighbor placement to avoid being trapped in a local minima. The worse a neighbor placement is, the lower is the probability of accepting it. As the annealing temperature K decays, the solution converges since the probability of accepting a worse neighbor placement decreases.

$$delta = \begin{cases} 0.9 \times (T_{curr} - T_{nei}) + 0.1 \times (L_{curr} - L_{nei}), \\ \qquad if\ L_{curr} \leq L_{th}\ and\ L_{nei} \leq L_{th} \\ 0.1 \times (T_{curr} - T_{nei}) + 0.9 \times (L_{curr} - L_{nei}), \\ \qquad if\ L_{curr} > L_{th}\ or\ L_{nei} > L_{th} \end{cases} \tag{4.24}$$

$$AP = e^{\frac{delta}{K}},\ accept\ if\ AP > rand(0,1) \tag{4.25}$$

4.4.4 Multi-Start and Multi-Phase Techniques

As a probabilistic algorithm, simulated annealing approximates the global minimum but provides no guarantee to find it. It is also challenging to find a good enough solution due to the astronomical non-convex solution space (up to quadrillions of placement options) and the limited simulation time (up to a thousand moves). In order to improve the solution quality of simulated annealing, we adopt multi-start and multi-phase techniques. For multi-start, we repeat the thermally-aware PNR process ten times for each table entry and pick the placement solution which has the lowest peak temperature and meets the routing constraint. Given the probabilistic

nature of the simulated annealing algorithm, the multi-start technique is helpful in reducing the chance of getting a poor solution. We can run the multiple starts of the multi-start technique in parallel, so as not to increase the time required to arrive at the solution. For multi-phase, we map an existing placement solution of a smaller interposer to a larger interposer (while keeping all the other tuning knobs the same) and use it as the initial starting placement to find the placement solution for the larger interposer. This improves the quality of the final placement solution for a table entry without increasing the simulation time or the electricity bill. The multi-phase step size must be a multiple of 2 mm since we assume 1 mm placement granularity. A smaller step size yields better solution quality, but requires longer actual simulation time. In our case, we set the multi-phase step size to 4 mm, which provides a good balance between the simulation time and the solution quality.

4.5 Evaluation Results

In this section, we first provide the maximum performance and the optimal chiplet placement for various networks. We compare the maximum performance using our new approach against the prior work (Coskun et al., 2018), with and without *gas stations*. Next, we present the iso-cost performance improvement, the iso-performance cost reduction using our new approach, and the Pareto Frontier curve of performance and cost. We then show the thermal maps for high-power, medium-power, and low-power applications on their respective optimal chiplet placement solution. In addition, we evaluate the running of medium-power and low-power applications on the optimal chiplet solution for a high-power application. Lastly, we conduct a sensitivity analysis to show the optimal combinations of performance, cost and peak temperature with respect to different temperature thresholds and different choices of constraints.

4.5.1 Optimal Chiplet Placement Analyses

Figure 4·8 shows the maximum performance, the corresponding cost and the corresponding peak operating temperature for various networks and link designs running the high-power `cholesky` benchmark for three different approaches. Here the focus is on performance. The first approach corresponds to our prior work (Coskun et al., 2018) that only considers matrix-style chiplet placement (*Mat*) and a hard temperature constraint (*HTC*) of 85 ^oC, with and without *gas stations*. We use *Mat-HTC-GS* and *Mat-HTC-noGS* to denote these cases. The second approach uses the same *HTC* of 85 ^oC but allows arbitrary placement of chiplets (*Arb*). We use *Arb-HTC-GS* and *Arb-HTC-noGS* to denote these cases. The third approach uses a soft temperature constraint (*STC*) of 85 ^oC and arbitrary placement, as described in Section 4.4. We

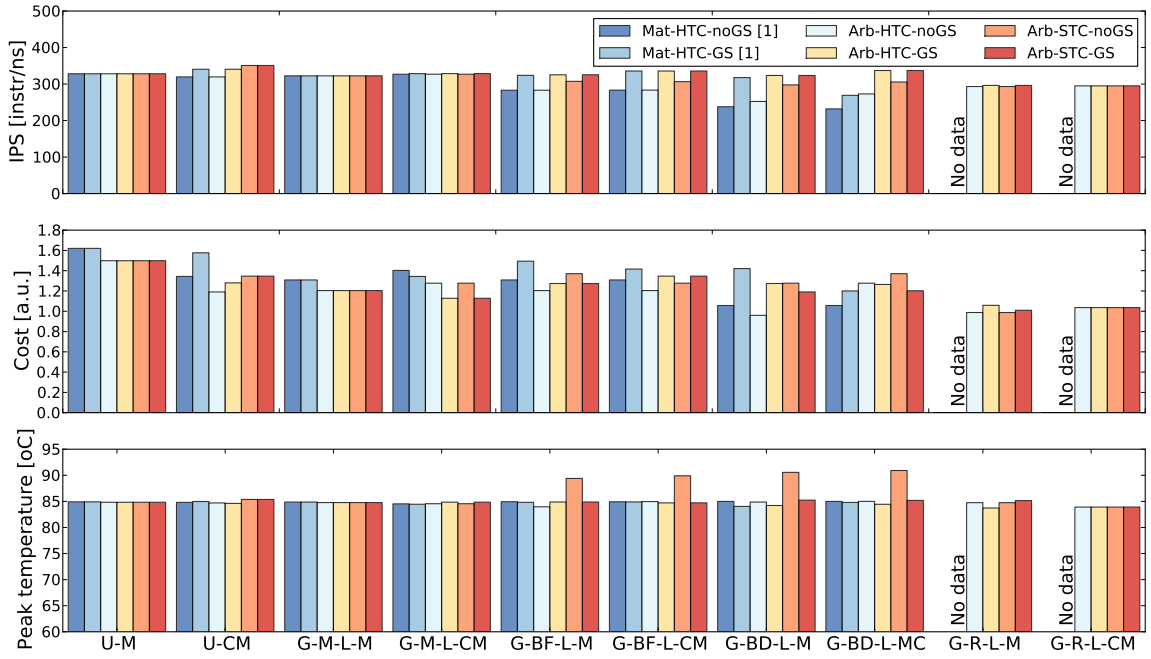

Figure 4·8: Maximum performance, the corresponding cost and the corresponding peak temperature for various networks with and without *gas-station* links when running `cholesky` benchmark. Here the optimization goal is to maximize performance; the cost values are normalized to the cost of a 2D system.

use *Arb-STC-GS* and *Arb-STC-noGS* to denote these cases.

For the mesh-like networks (*G-M-L-M*, *G-M-L-CM*, *U-M*, and *U-CM*), our *Arb-HTC* approach does not improve the performance over the previous *Mat-HTC* approach (Coskun et al., 2018). This is because the previous approach already achieves the maximum performance for *G-M-L-M*, *G-M-L-CM*, and *U-M*, while for *U-CM*, there is not much room for improvement with arbitrary placement since the optimal placement also follows a matrix style. However, we achieve a 8-19% (11% on average) reduction in cost. The *Arb-STC* approach achieves the highest performance (10% improvement) with *U-CM* network at a manufacturing cost which is equal to the *Mat-HTC-noGS* case, while exceeding the temperature threshold by less than 0.5 ^oC. For the remaining three mesh-style networks, the *Arb-STC* approach does not improve performance but it does reduce cost in some cases. Even when using our thermally-aware PNR tool with the option of arbitrary placement, the optimal chiplet placements are matrix style. Since these four mesh-like networks have similar optimal placement patterns, we just show the logical connection and thermal map of *U-CM* network in Figure 4·9(a).

For Butterfly networks, the *Arb-STC-GS* approach achieves the same maximum performance as achieved using *Mat-HTC-GS* approach (Coskun et al., 2018) and reduces the cost by 5% (see Figure 4·8). The optimal placement for Butterfly network is shown in Figure 4·9(b). Note in the top subfigure, we only show the logical connections instead of actual routing path of *gas-station* links. For Butterdonut networks, the *Arb-STC-GS* approach improves the performance by 25% without increasing the cost (see Figure 4·8). Figure 4·9(c) shows the optimal placement for Butterdonut network. The Ring networks (*G-R-L-M/CM*) are not included in the prior work (Coskun et al., 2018), thus we do not show the comparison. The chiplets are distributed along the periphery of the interposer in the optimal placement for the Ring topology (see

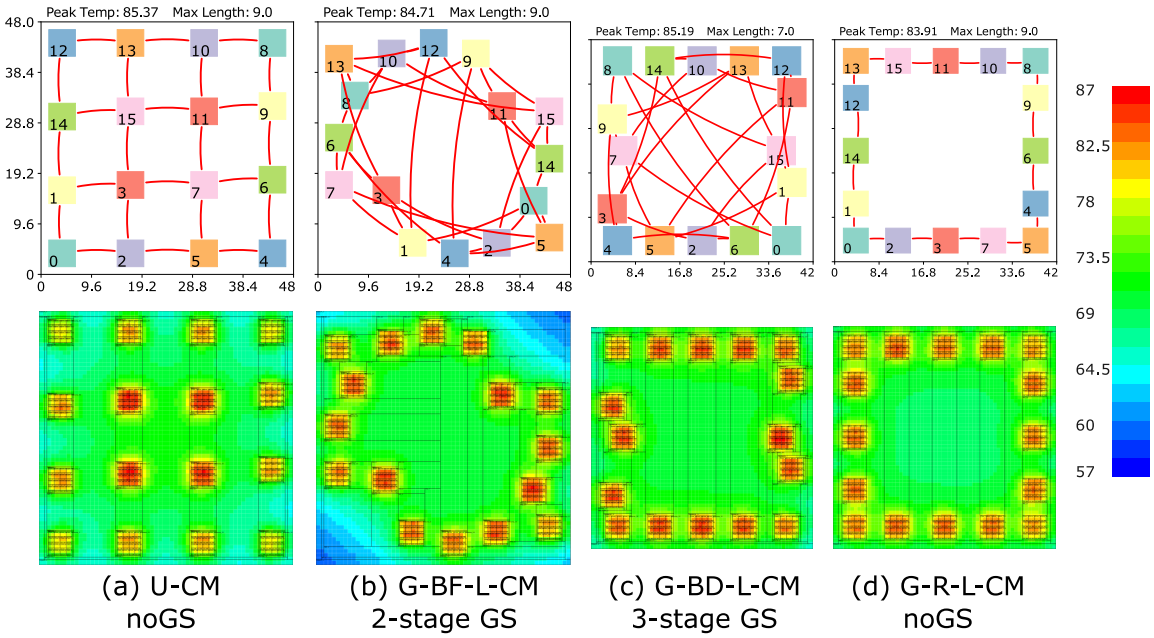

Figure 4·9: Optimal chiplet placement for maximum performance and corresponding thermal maps when running the `cholesky` benchmark in 2.5D systems with different network topologies. The figures are scaled to the interposer sizes.

Figure 4·9(d)), which is good for heat dissipation. Thus, the performance of the Ring topology saturates at a relatively small interposer size, and we observe lower cost and temperature than those of other networks (see Figure 4·8).

4.5.2 Iso-cost and Iso-performance Analyses

Figure 4·10 shows the iso-cost performance for various networks running `cholesky` benchmark, while not exceeding the cost of a 2D system. In general, our *Arb-HTC* approach improves the iso-cost performance by 13-37% (20% on average), and our *Arb-STC* approach improves the iso-cost performance by 40-68% (49% on average), compared to our prior *Mat-HTC* approach (Coskun et al., 2018). The previous work (Coskun et al., 2018) shows that the *U-M* network cannot be implemented feasibly due to the large microbump area overhead and the incorrectly estimated yield drop. Using our more accurate cost model, it is actually feasible to implement the *U-M*

network within the cost budget.

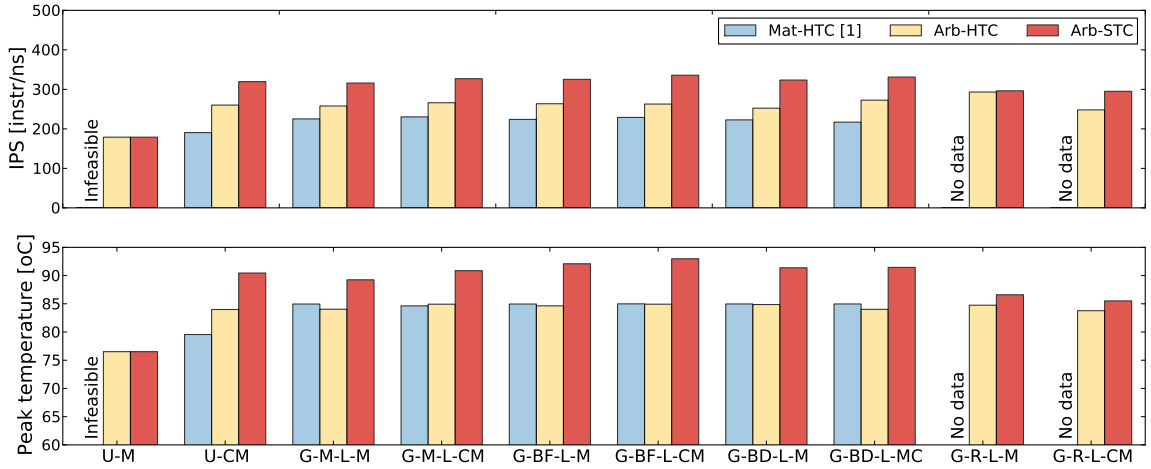

Figure 4·10: Iso-cost performance and the corresponding peak temperature when running `cholesky` benchmark for various networks, while not exceeding the cost budget of a 2D system.

Figure 4·11 shows the iso-performance cost and the corresponding peak temperature for each network. Here, for each network, we match the performance of the 2.5D system designed using our proposed approach with the corresponding maximum performance of the 2.5D system designed using prior *Mat-HTC* approach (Coskun et al., 2018) when running `cholesky` benchmark. The cost values are normalized to the cost of a 2D system. Under the same hard temperature constraint as the prior work (Coskun et al., 2018), our *Arb-HTC* approach reduces manufacturing cost by 5-20% (14% on average) without lowering the performance. Using the *Arb-STC* approach, we can push the iso-performance cost saving to 30-38% (32% on average) with up to 91 ^oC overall system peak temperature.

Finding Pareto Frontiers is a widely used method in engineering fields. For a given system, the Pareto Frontier is the set of choices that are Pareto efficient, i.e., no individual criterion can be better without making at least one individual criterion worse. It is helpful to let designers focus on the tradeoffs within the constrained set of parameters, rather than considering the full ranges of parameters (Costa and

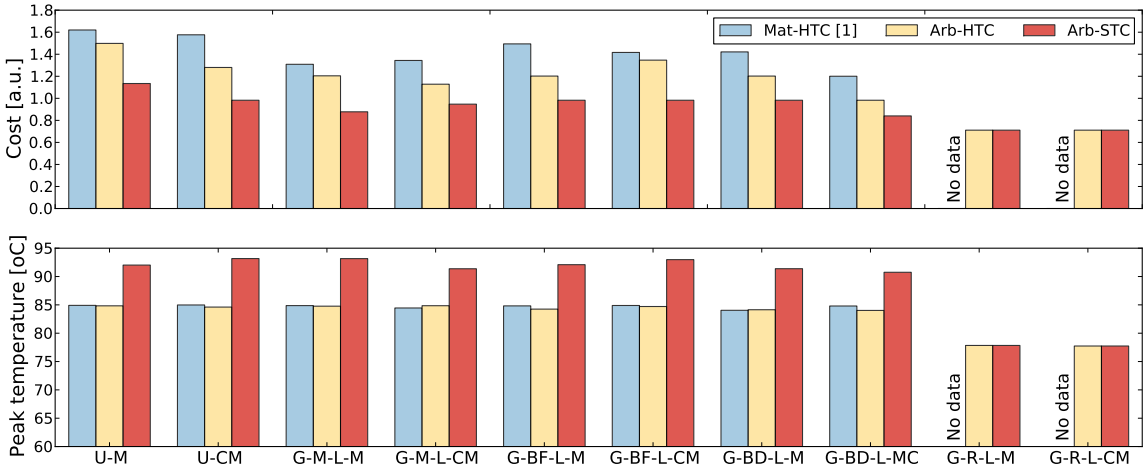

Figure 4·11: Iso-performance cost and the corresponding peak temperature for each network. Here the performance is equal to the maximum performance achieved using *Mat-HC-GS* (Coskun et al., 2018) when running `cholesky` benchmark. The cost values are normalized to the cost of a 2D system.

Lourenço, 2015). Figure 4·12 shows the Pareto Frontier Curve of normalized performance ($1/IPS$) and normalized cost using *Mat-HTC* approach (Coskun et al., 2018), *Arb-HTC* approach, and *Arb-STC* approach. Our arbitrary placement pushes the Pareto Frontier curve towards higher performance and lower cost, and the soft temperature constraint approach pushes the frontier further.

4.5.3 Analyses of Different Types of Applications

Figure 4·13 shows the thermal maps of 2.5D systems designed for high-power (`cholesky`), medium-power (`streamcluster`), and low-power (`lu.cont`) applications using *Mat-HTC* (Coskun et al., 2018), *Arb-HTC* and *Arb-STC* approaches. For comparison, we choose the same optimization objective as in the prior work (Coskun et al., 2018), which focuses on performance ($(\alpha, \beta, \gamma) = (0.999, 0.001, 0)$). With the *Arb-HTC* approach, we can achieve the same performance as using the prior *Mat-HTC* approach (Coskun et al., 2018) and reduce the manufacturing cost by 19%, 14%, and 3% for high-power, medium-power, and low-power applications, respectively. The

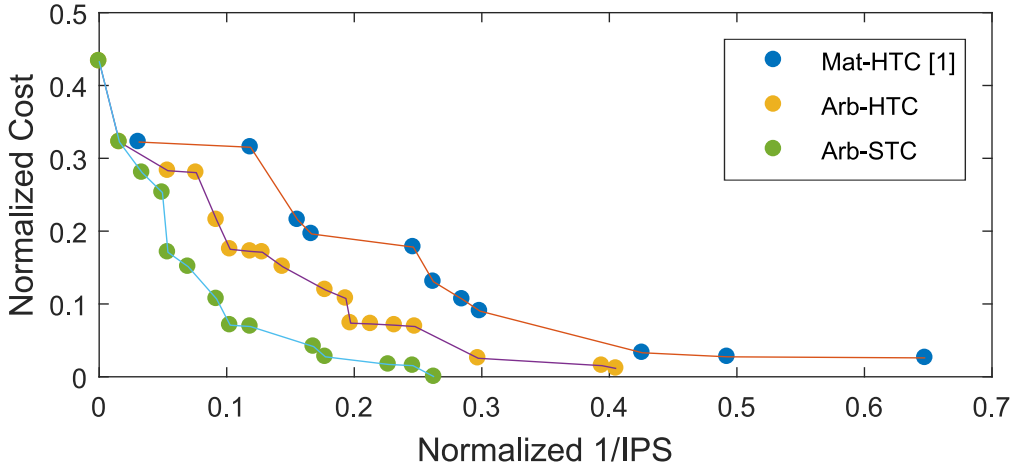

Figure 4·12: Pareto Frontier Curve of normalized performance ($1/IPS$) and normalized cost using *Mat-HTC* approach (Coskun et al., 2018), *Arb-HTC* approach, and *Arb-STC* approach.

equivalent performance is achieved at a smaller interposer size where the chiplets are pushed to the periphery of the interposer to ease the heat dissipation. For high-power and medium-power applications, 2-stage *gas-station* links are used, which provides flexibility in chiplet placement to form a ring shape for mesh-like networks, while for low-power application, such a ring-shape placement is not feasible as we need to provide routability of single-cycle links.

Using *Arb-STC* approach, for high-power application, we can achieve the maximum possible performance (3% higher than both *Mat-HTC* approach (Coskun et al., 2018) and *Arb-HTC* approach) and 15% lower cost. The improvement is achieved by violating the temperature threshold by 0.5 ^oC and using single-cycle inter-chiplet links without *gas stations*, which constrains distance between chiplets and forms a matrix-style placement. For medium-power application, we get identical network choices and placement solutions using *Arb-STC* and *Arb-HTC* approaches. For low-power application, our *Arb-STC* approach achieves the maximum possible performance while violating the temperature threshold by 1.4 ^oC. This improvement also comes with

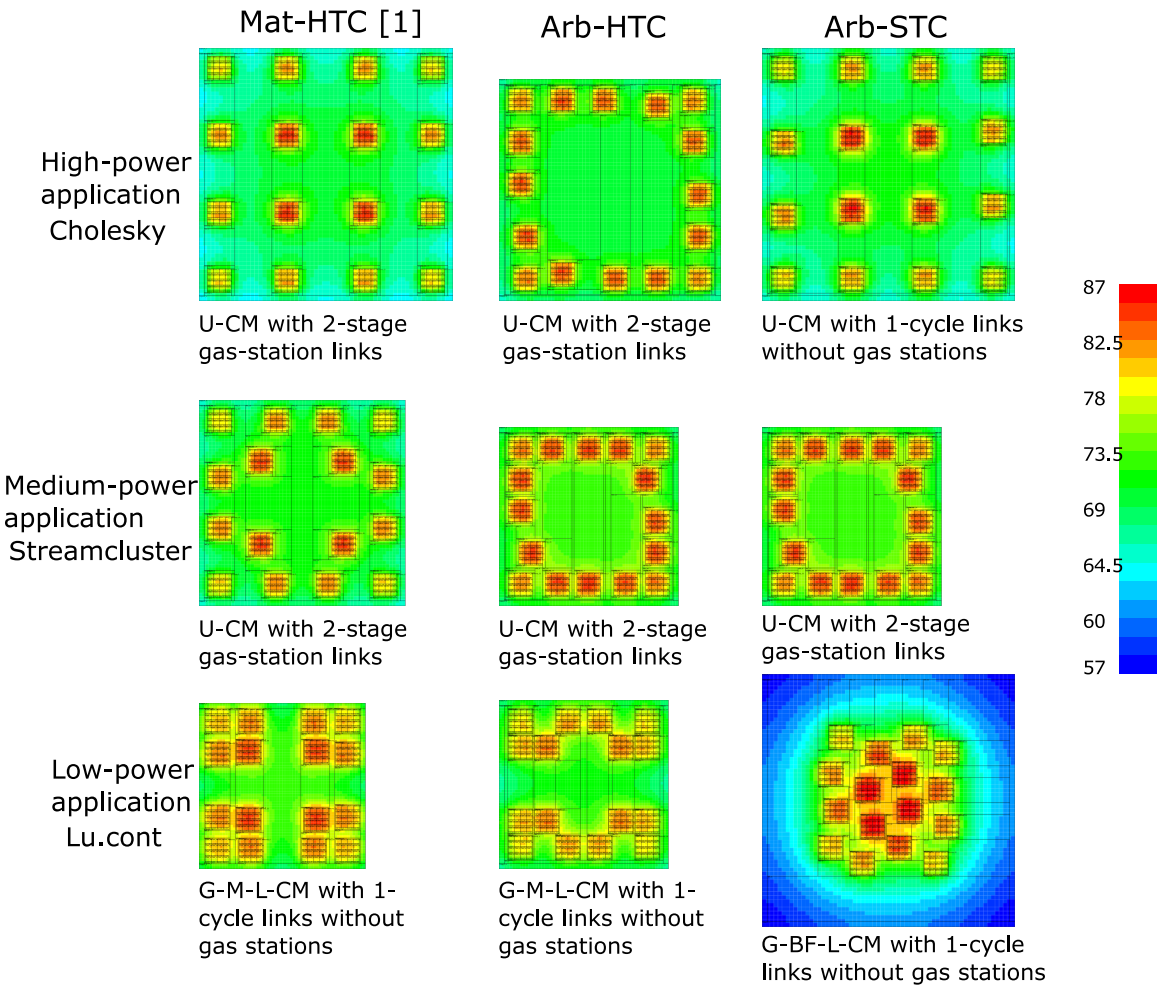

Figure 4·13: Thermal maps of 2.5D systems designed for high-power, medium-power, and low-power applications using *Mat-HTC* (Coskun et al., 2018), *Arb-HTC*, *Arb-STC* approaches. The figures are scaled to the interposer sizes.

40% cost overhead, but in this example, cost is not our concern. The chiplets cluster in the center of the interposer to meet single-cycle latency constraint for a butterfly topology, and leave large empty space on the edges of the interposer to help heat dissipation.

It should be noted that the results we show in Figure 4·13 assume that we know what application will be running at the design time, and we optimize for each application. For unknown target applications or a mix of known and unknown appli-

cations, we optimize for the worst-case (highest power application) scenario at the design time, and run the target application on the optimized organization (including network topology, interposer size, chiplet placement, and inter-chiplet link design). For example, if a system is expected to run high-power (`cholesky`), medium-power (`streamcluster`), and low-power (`lu.cont`) applications, we design and optimize the system using the high-power application. When running medium-power application on the system optimized for high-power application, we observe the same performance, 23% higher cost, and 6 ^oC lower temperature compared to that of a system custom designed for medium-power application. When running low-power application on the system designed for high-power application, we observe 5% lower performance, 5% higher cost, and 12 ^oC lower temperature compared to that of a system custom designed for low-power application.

4.5.4 Analyses of Cross-layer Co-optimization Benefits

To understand the benefits of co-optimizing across multiple design layers simultaneously, we conduct a comparison of cross-layer optimization meethodology against single-layer or two-layer methods while running the `Blackscholes` benchmark. We compare multiple cases in Table 4.7. The baseline is the optimal solution of our cross-layer optimization methodology. We use three letters to represent the choices at each of the logical, physical, and circuit layers, for the remaining nine cases to show the contribution of each layer, and results using single-layer or two-layer optimization methods. Here O means cross-layer optimal choice, W means worst choice, F means prefixed choice, B means best choice. We report performance improvement, cost increase, and temperature for each case. To better compare the different cases, we use the **Performance/Unit Cost** metric. For example, the OOW case corresponds to use of the same design choices as the optimal cross-layer solution at the logical and physical layers, and use of the worst possible choice at the circuit layer. This case

Table 4.7: Comparison of cross-layer optimization solution against other cases that optimize at single layer or two layers. Here O means cross-layer optimal choice, W means worst choice, F means prefixed choice, B means best choice.

	Logical Layer	**Physical Layer**	**Circuit Layer**	**Performance Improvement**	**Cost Increase**	**Temperature** $[^oC]$	**Perf/Unit Cost**
Cross-layer	O	O	O	0%	0%	86	3.10
Contribution of each layer	O	O	W	4%	-8%	99.9	3.50
	O	W	O	0%	-22%	108.0	3.97
	W	O	O	-20%	56%	84.2	1.59
Single-layer	F	F	B	-39%	-34%	100.9	2.88
	F	B	F	4%	11%	102.5	2.92
	B	F	F	-16%	-36%	103.4	4.09
Two-layer	F	B	B	-9%	-4%	85.8	2.94
	B	F	B	-35%	-34%	100	3.09
	B	B	F	2%	3%	86.2	3.06

shows the contribution of the circuit layer in our cross-layer co-optimization methodology. A bad choice in the circuit layer results in slightly better performance and cost (4% higher performance and 8% lower cost), but at a infeasibly high operating temperature of 99.9 ^oC. Similarly, a bad choice in the physical layer (Case OWO) leads to 22% lower cost but the temperature is as high as 108 ^oC. A bad choice in the logical layer (Case WOO) does not stress the peak operating temperature, but degrades performance by 20% with 56% higher cost. The cases FFB, FBF, and BFF are optimizing single layer while fixing the other two layers. The cases FBB, BFB, and BBF optimize two layers simultaneously while fixing the remaining layer. For example, in the FFB case, we fix the design choices at the logical and physical layers, and only optimize the circuit layer. For the cases of FFB, FBF, BFF, and BFB, we get either higher performance at higher cost or lower performance at lower cost, but the temperature becomes infeasibly high. For the cases of FBB and BBF, the temperature is safe, while performance and cost offset each other. In terms of the **Performance/Unit Cost** metric, our cross-layer co-optimization approach performs better than all cases except OOW, OWO and BFF, but these cases have high infeasible temperature.

4.5.5 Sensitivity Analysis

We conduct a sensitivity analysis (see Figure 4·14) to show the optimal combinations of performance, cost and peak temperature, and the corresponding objective function values with respect to different temperature thresholds from 75 ^oC to 95 ^oC and different temperature constraint choices (including hard temperature constraint, soft temperature constraint with linear and square penalty functions, and no temperature constraint). We choose the weights to be $((\alpha, \beta, \gamma) = (0.8, 0.1, 0.1))$ as an example for a performance-focused objective function.

With no temperature constraint, we can always achieve the maximum performance and the lowest cost, at a temperature of 93.2 ^oC. Thus, with a temperature threshold of 94 ^oC or higher, the optimal performance, cost, and temperature combinations with different constraint choices are the same. With a hard temperature constraint, any case that exceeds the temperature threshold is considered as infeasible, thus, the peak temperature is close to, but below the temperature threshold. As the temperature threshold increases, there are more feasible design choices and

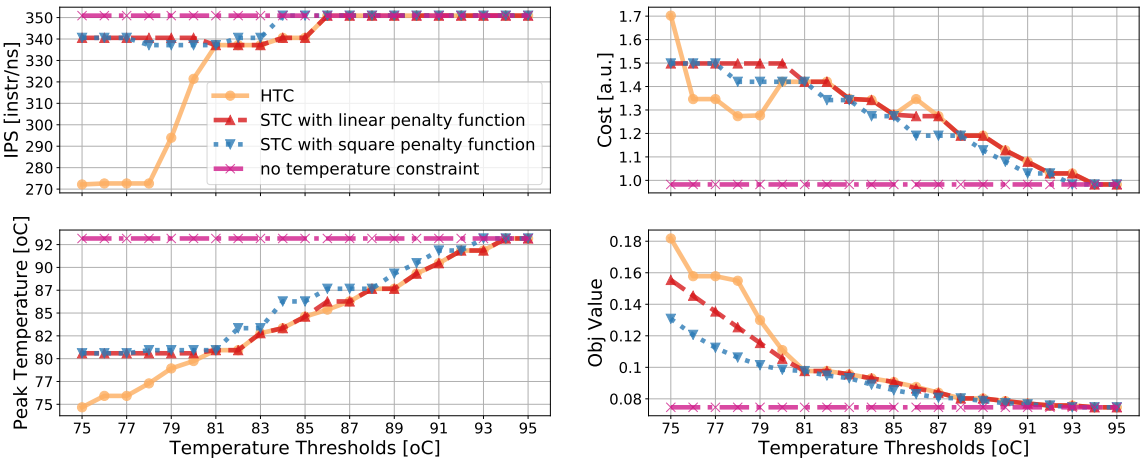

Figure 4·14: Sensitivity analysis comparing hard temperature constraint, soft temperature constraints with linear function and square function, and no temperature constraint of various temperature thresholds from 75-95 ^oC.

the objective function value decreases. A soft temperature constraint allows violating the temperature threshold and translates the violation into a penalty in the objective function. The soft temperature constraint approach provides more choices and thus is guaranteed to have a solution that better or equal to that obtained using hard temperature constraint approach. For the soft temperature constraint approach with a linear penalty function, we are allowed to violate the temperature threshold only slightly to find a solution that has higher performance and/or lower cost than the hard temperature constraint approach. A square penalty function suppresses the penalty for a small violation and highlights the penalty for a large violation of the temperature threshold. Thus, with a soft temperature constraint approach with the square penalty function, we can achieve higher performance and lower cost compared to the case with the linear penalty function. For example, with a temperature threshold of 80 ^oC, the result with the hard temperature constraint has lowest performance. With the soft temperature constraint with the linear penalty function, we violate the temperature threshold by 0.59 ^oC and achieve 6% higher performance but at 5% higher cost compared to the hard temperature constraint approach. With the soft temperature constraint with the square penalty function, we violate the temperature threshold by 0.93 ^oC and achieve 5% higher performance at the same cost compared to the hard temperature constraint approach.

4.6 Summary

In this chapter we have presented a cross-layer co-optimization methodology for network design and chiplet placement in homogeneous 2.5D systems. Our methodology optimizes network topology design, inter-chiplet link design, and chiplet placement across logical, physical, and circuit layers to jointly improve performance, lower manufacturing cost, and reduce operating temperature. We have searched a vast design

space with arbitrary chiplet placement and applied a soft temperature constraint to improve the overall benefits. Our methodology shifts the performance-cost Pareto tradeoff curve for homogeneous 2.5D systems substantially. Experimental results show that our approach improves thermal constrained performance by 88% at the same manufacturing cost and reduces the cost by 29% at the same performance in comparison to 2D systems. Compared to the our prior work (Coskun et al., 2018), our optimization methodology with a soft temperature constraint and arbitrary placement achieves 40-68% (49% on average) higher iso-cost performance and 30-38% (32% on average) lower iso-performance cost.

Chapter 5

Cross-layer Optimization Methodology in Heterogeneous 2.5D Systems

In this chapter, we present a methodology for physical design of inter-chiplet networks for heterogeneous 2.5D systems. Our goal is to find an inter-chiplet routing network solution that jointly minimizes the peak operating temperature and the total inter-chiplet wirelength, given a logical inter-chiplet network topology. We use HotSpot (Zhang et al., 2015) to evaluate the operating temperature for a given chiplet placement and chiplet power profile. To minimize the total inter-chiplet wirelength, we use Mixed Integer-Linear Program (MILP) to build a routing optimizer to solve for the optimal routing solution and the corresponding wirelength value. We develop an SA-based thermally-aware placer to optimize the placement solution and interactively evaluate the temperature and the wirelength using the HotSpot and MILP tools. Our work targets general heterogeneous 2.5D systems. Our methodology suggests optimal chiplet placement and routing solution regardless of the chiplet count and chiplet types, as long as the chiplet dimensions and power profiles are provided. Due to the lack of a simulation tool for architectural performance analysis of such generic heterogeneous systems, our work only focuses on the physical layer and circuit layer.

5.1 Thermal Evaluation

Our thermal simulation takes the chiplet placement from the thermally-aware placer and uses the 2.5D system configuration (including chiplet widths and heights, power profiles, system layer descriptions, and material properties) to evaluate the operating temperature. We use an extension (Meng et al., 2012) of HotSpot that provides detailed heterogeneous 3D modeling features, which supports heterogeneous materials in each modeling layer. To model our 2.5D system, we stack six modeling layers on top of each other. From the bottom up, the layers are organic substrate, C4 bump layer, silicon interposer, microbump layer, chiplet layer, and thermal interface material (TIM). We use a separate floorplan for each layer to describe the placement and materials. Our 2.5D system model follows the properties (such as layer thickness, materials, dimensions of bumps, and TSVs) of real systems (Chaware et al., 2012), (Charbonnier et al., 2012). The thickness and the material properties of each layer are listed in Table 5.1.

We use a realistic air-forced pin fin heatsink as the cooling technique. Following the HotSpot default conventions, we set the ambient temperature to 45 ^oC, set the

Table 5.1: Thermal modeling of 2.5D systems (Chaware et al., 2012), (Charbonnier et al., 2012).

Layer	**Thickness** $[\mu m]$	**Material**	**Resistivity** $[m \cdot K/W]$	**Specific Heat Capacity** $[10^6\ J/m^3K]$
TIM Layer	20	TIM	0.25	4
Chip Layer	150	Silicon	0.01	1.75
		Underfill	0.625	2.32
Microbump Layer	10	Microbump	0.0025	3.49
		Underfill	0.625	2.32
Interposer Layer	110	Silicon	0.01	1.75
C4 Bump Layer	70	C4 bump	0.0025	3.49
		Underfill	0.625	2.32
Substrate Layer	200	FR-4	3.33	1.06

grid model resolution to 64×64, and set the heat spreader edge size to be $2\times$ the interposer edge size and the heatsink edge size to be $2\times$ the spreader edge size. To keep the heat transfer coefficient consistent across all simulations, we adjust the convective resistance of the heatsink. The runtime for each HotSpot simulation is 23 seconds on average.

5.2 Routing Optimization

The objective of our routing tool is to find a routing solution that minimizes the total wirelength of the inter-chiplet network. The inputs of the tool are the chiplet placement from the thermally-aware placer, the estimated microbump resources for inter-chiplet communication, and the inter-chiplet connectivity and bandwidth requirements of the 2.5D system. We formulate the MILP and build an MILP solver using IBM ILOG CPLEX v12.8 Python API. We group the microbumps along the chiplet periphery into pin clumps to limit the problem size and the MILP runtime. In our experiments, we use 4 pin clumps per chiplet, where each pin clump accounts for the microbumps on an edge of the chiplet (Coskun et al., 2018), (Coskun et al., 2020). We frame the delivery of required number of wires between chiplets as multi-commodity flow, and formulate the MILP to find optimal routing solutions that encompass the finite availability of microbumps in each pin clump as follows (the notations are listed in Table 5.2).

Minimize:

$$\sum_{i\in C,l\in P,j\in C,k\in P,n\in N} d_{iljk}\cdot f^n_{iljk} \tag{5.1}$$

Subject to:

$$d_{iljk} = |X_i + x_l - X_j - x_k| + |Y_i + y_l - Y_j - y_k| \tag{5.2}$$

$$f^n_{iljk} \geq 0, \qquad \forall i \in C,\, l \in P,\, j \in C,\, k \in P, n \in N \tag{5.3}$$

$$\sum_{l\in P,j\in C,k\in P} f^n_{iljk} - \sum_{l\in P,j\in C,k\in P} f^n_{jkil} = \begin{cases} R_{s_nt_n}, & \text{if } i = s_n, \forall n \in N \\ -R_{s_nt_n}, & \text{if } i = t_n, \forall n \in N \\ 0, \forall i \neq s_n || t_n, \forall n \in N \end{cases} \tag{5.4}$$

$$f^n_{jks_nl} = 0, \qquad \forall n \in N, \forall l \in P, \forall j \in C, \forall k \in P \tag{5.5}$$

$$f^n_{t_nljk} = 0, \qquad \forall n \in N, \forall l \in P, \forall j \in C, \forall k \in P \tag{5.6}$$

$$\sum_{j\in C,k\in P,n\in N} f^n_{iljk} + \sum_{j\in C,k\in P,n\in N} f^n_{jkil} \leq P^{max}_{il}, \quad \forall i \in C, l \in P \tag{5.7}$$

$$\sum_{i\in C,l\in P,j\in C,k\in P} f^n_{iljk} \leq R_{s_nt_n} \tag{5.8}$$

Table 5.2: Notations.

Notation	Meaning
C, P, N	Set of chiplets, set of pin clumps, and set of nets, respectively.
c, i, j	Index of a chiplet $\in C$.
p, l, k	Index of a pin clump $\in P$.
n	A net $\in N$.
d_{iljk}	Distance from pin clump l on chiplet i to pin clump k on chiplet j. Note that $d_{iljk} = d_{jkil}$.
f^n_{iljk}	Flow variable. Number of wires from pin clump l of chiplet i to pin clump k of chiplet j that belong to net n.
X_c, Y_c	Center x- and y-coordinates for chiplet c.
x_p, y_p	x- and y-offsets from center point of the chiplet for pin clump p.
s_n, t_n	Source chiplet and sink chiplet of net n, respectively.
R_{ij}	Input requirement on the wire count between chiplet i and chiplet j.
P^{max}_{il}	Microbump capacity for a pin clump l on chiplet i.
w_i, h_i	Width and height of chiplet i.
w_{gap}	Minimum spacing between two chiplets: 100 μm (Xilinx, 2016).
w_{int}	Edge length of interposer, $w_{int} \leq 50$ mm (Coskun et al., 2018).

Equation (5.1) is the objective function for the MILP, which sums up the total length of the routes. Here, f^n_{iljk} is the flow variable, which indicates the number of wires from pin clump l of chiplet i to pin clump k of chiplet j that belong to net n, and d_{iljk} is the distance of these wires. The route distance d_{iljk} is calculated using Equation (5.2), based on the coordinates of pin clump l of chiplet i and pin clump k of chiplet j, assuming Manhattan distance. Equation (5.3) ensures that the flow variable f^n_{iljk} is non-negative. Equation (5.4) guarantees the sum of all flows for a net n, over all pin clumps from source chiplet s_n to sink chiplet t_n, meets the bandwidth requirement, and also assures that the net flow (total outgoing flows f^n_{iljk} minus total incoming flows f^n_{jkil}) is 0 for all other (non-source, non-sink) chiplets for the given net. Equation (5.5) makes sure that there is no input flow (for net n) for any pin clump in the source chiplet s_n from any other chiplet's pin clump. Similarly, Equation (5.6) ascertains that there is no output flow (for net n) for any pin clump in the sink chiplet t_n to any other chiplet's pin clump. Equation (5.7) insures that all routes

have available pins. Equation (5.8) constrains the sum of all flows for a net n within the bandwidth requirement between the source and sink chiplets of the net.

In addition to the repeaterless non-pipelined inter-chiplet links, we also consider *gas-station* links (Coskun et al., 2018), which use transistors on an intermediate chiplet to 'refuel' the signals and thus enable pipelining in passive interposers. To formulate 2-stage *gas-station* links, we replace Equation (5.8) with Equation (5.9), where the net connects s_n and t_n through at most one other chiplet.

$$\sum_{i\in C, l\in P, j\in C, k\in P} f^n_{iljk} \leq 2 \cdot R_{s_n t_n} - \sum_{l\in P, k\in P} f^n_{s_n l t_n k} \tag{5.9}$$

Based on our formulation, both the number of variables and constraints in the MILP are bounded by $O(|C|^2 \cdot |P|^2 \cdot |N|)$. The average runtime for each routing optimization is 5 s in our simulation, given our case studies have up to 8 chiplets, 4 pin clumps per chiplet, and up to 8 channels.

5.3 Thermally-Aware Placement Algorithm

Simulated annealing (SA) is a widely used technique to solve floorplanning problems (Murata et al., 1996), (Lin and Chang, 2001), (Chen and Chang, 2006). It is a probabilistic based approach to approximate global minimum, which emulates the physical process of heating a material and then slowly lowering the temperature to decrease defects. Unlike deterministic approaches such as gradient descent or greedy search, SA accepts worse moves at a non-zero probability to avoid being trapped at a local minima. The probability of accepting a worse move decreases during the annealing process, and thus, SA algorithm converges eventually.

We develop an SA-based algorithm to determine the thermally-aware chiplet placement for heterogeneous 2.5D systems with the provided inter-chiplet connectivity at the logical level. Our methodology faces two main challenges. First, we strategi-

cally insert spacing between chiplets to improve heat dissipation. So we cannot use the state-of-the-art floorplan representations, such as Sequence Pair (Murata et al., 1996), TCG (Lin and Chang, 2001), O-tree (Guo et al., 1999), and B*-tree (Chen and Chang, 2006), as these representations assume compact placement. Second, the thermal evaluation and routing optimization processes for each chiplet placement take approximately 30 seconds. For manageable simulation time, our methodology has to find a satisfactory solution with limited steps. We present the details of the key components of our algorithm in the subsections below, including placement description, initial placement, neighbor placement, SA cost function, and acceptance probability.

5.3.1 Placement description

To represent unrestricted placements, we use x and y coordinates of the center points of chiplets, together with the widths and heights of the chiplets. To avoid an infinite solution space, we divide the interposer into a discrete grid, and we assume that the center of a chiplet can only be placed on the intersection nodes of the grid. We assume 1 mm granularity for the grid to place the centers of chiplets (the widths and heights of the chiplets can be any value), which provides good balance between the solution space (increases with finer granularity) and the solution quality (decreases with finer granularity). A valid chiplet placement has no overlap between any pair of chiplets and ensures 0.1 mm minimum gap between chiplets (Chaware et al., 2012) (Equation (5.10)). It is also necessary for a chiplet to be completely on the interposer (Equation (5.11)).

$$max\{(X_i - \frac{w_i}{2}) - (X_j + \frac{w_j}{2}), (X_j - \frac{w_j}{2}) - (X_i + \frac{w_i}{2}), (Y_i - \frac{h_i}{2}) - (Y_j + \frac{h_j}{2}), (Y_j - \frac{h_j}{2}) - (Y_i + \frac{h_i}{2})\} \geq w_{gap}, \qquad \forall i \in C, \forall j \in C, i \neq j \quad (5.10)$$

$$\frac{w_i}{2} \leq X_i \leq w_{int} - \frac{w_i}{2}, \quad \frac{h_i}{2} \leq Y_i \leq w_{int} - \frac{h_i}{2}, \qquad \forall i \in C \quad (5.11)$$

5.3.2 Initial placement

Theoretically, the initial placement does not matter in an SA process, as long as the process can run long enough to cover a substantial portion of the solution space. However, we want to find a satisfactory solution in a limited amount of time. Thus, a good initial placement is critical as it can help the SA process use the limited steps more efficiently, and explore the placements that are closer to the optimal choice.

In our methodology, we implement the floorplanning method developed by Chen *et al.* (Chen and Chang, 2006) to generate an initial placement. This method uses B*-tree data structure, which is known to be the most efficient and flexible floorplan representation, and uses fast-SA algorithm, which efficiently searches for a solution of modern fixed-outline floorplanning problem for both area reduction and wirelength minimization. We use the compact chiplet placement solution from the B*-tree and fast-SA based method as the initial placement for our methodology.

5.3.3 Neighbor placement

To find a neighbor placement, we perturb the current chiplet placement with rotate, move, and jump operations to get a new valid placement. For a rotate operation, we randomly pick a chiplet and rotate it by 90 degree. For a move operation, we randomly pick a chiplet and move it by a minimum step size (1 mm in our case) in up, down, left or right directions. With only the rotate and move operations, the relative positions of the chiplets are unlikely to change. Thus, the SA process may run into the 'sliding tile puzzle' issue where a chiplet cannot move in certain directions because other chiplets block the way. To resolve this 'sliding tile puzzle' issue, we introduce the jump operation. With a jump operation, a randomly picked chiplet can jump to any valid empty location on the interposer. A valid neighbor placement should have no overlap between chiplets and should be completely on the interposer.

5.3.4 SA cost function

The goal of our approach is to find an inter-chiplet routing solution while minimizing the operating temperature and the total wirelength for heterogeneous 2.5D systems with a given network connectivity. Equation (5.12) shows our SA cost function. The temperature (T) and wirelength (W) are normalized using Min-Max Scaling to alleviate the impact of imbalanced values and ranges of raw data. α and $(1-\alpha)$ are the weights of the temperature and wirelength terms, respectively. Here, α is picked by our algorithm rather than by users because we are seeking a thermally-feasible solution that also minimizes wirelength, rather than a solution with optimized wirelength but infeasibly high temperature that could immediately burn the system. So we dynamically adjust α to be aware of the temperature level, as shown in Equation (5.13). At a higher temperature, our algorithm prioritizes lowering the temperature (effectively choosing an α value of greater than 0.5), which is critical to maintain safe operation. When the temperature is below 85^oC, the algorithm focuses purely on minimizing the wirelength (effectively choosing an α value of less than 0.5), as there is no point to trade off wirelength for lower temperature. In our experiments, the value of α ranges from 0.1 to 0.9.

$$Cost = \alpha \times \frac{T - T_{min}}{T_{max} - T_{min}} + (1-\alpha) \times \frac{W - W_{min}}{W_{max} - W_{min}} \tag{5.12}$$

$$\alpha = \begin{cases} min\{0.1 + \frac{T-45}{100}, 0.9\}, & if\ T > 85^oC \\ 0, & if\ T \leq 85^oC \end{cases} \tag{5.13}$$

5.3.5 Acceptance probability

The decision of whether a neighbor placement is accepted or not depends on the Acceptance Probability (AP). We compute the AP using Equation (5.14), where the

cost of current and neighbor placements are computed using Equation (5.12), and K is the annealing temperature, which decays from 1 to 0.01 with a factor of 0.95. We accept the neighbor placement if AP is greater than a random number between 0 and 1. In the case that a neighbor placement is better or equal ($Cost_{neighbor} \leq Cost_{current}$), then AP value becomes greater than or equal to 1 and our algorithm accepts the neighbor placement solution. In the case that a neighbor placement is worse ($Cost_{neighbor} > Cost_{current}$), there is still a nonzero probability of accepting the worse neighbor placement to avoid getting trapped in a local minima. The worse a neighbor placement is the lower is the probability of accepting it. As the annealing temperature K decays, the solution converges because the probability of accepting a worse neighbor placement decreases.

$$AP = e^{(Cost_{current} - Cost_{neighbor})/K} \tag{5.14}$$

5.4 Evaluation Results

In this section, we discuss the results of applying our approach to both existing and conceptual heterogeneous 2.5D systems. The logical network topologies of the heterogeneous 2.5D systems we evaluated are shown in Figure 5·1. Here (a) is a conceptual 2.5D multi-GPU system, consisting of a CPU chiplet, 2 GPU chiplets, and 3 HBMs. The CPU and GPU chiplets are connected to each other, and each HBM serves as the dedicated memory to either a CPU chiplet or a GPU chiplet. (b) is a conceptual 2.5D CPU-DRAM system described by Kannan *et al.* (Kannan et al., 2015), which has 4 CPU chiplets and 4 3D-DRAM chiplets. The CPU chiplets are connected using mesh topology and each 3D-DRAM chiplet connects to one of the CPU chiplets. (c) is the Huawei Ascend 910 system (Huawei, 2019), which consists of a Virtuvian processor chiplet, a Nimbus chiplet, and 4 HBMs. The processor chiplet

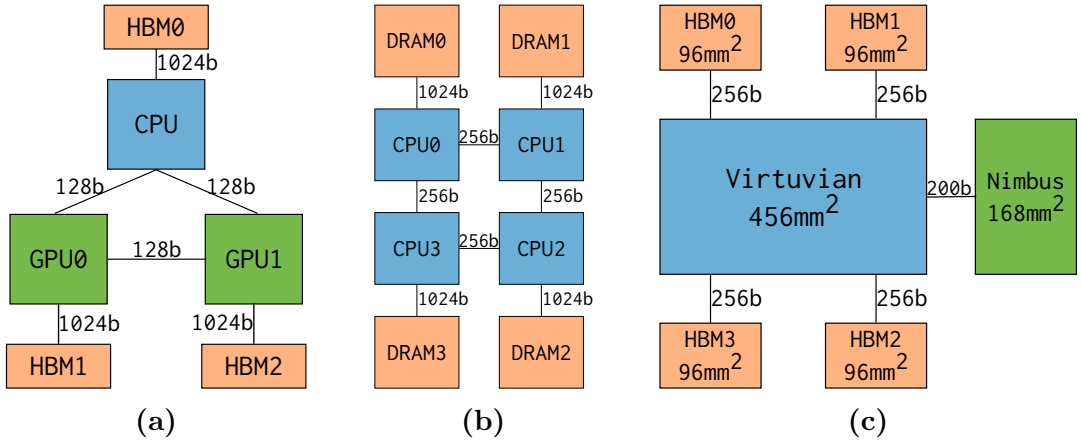

Figure 5·1: Logical network topologies for heterogeneous 2.5D examples: (a) a conceptual Multi-GPU System, (b) CPU-DRAM System (Kannan et al., 2015), and (c) Huawei Ascend 910 System (Huawei, 2019). Numbers shown next to the inter-chiplet links refers to the bit widths.

Table 5.3: Chiplet dimensions and powers in 2.5D examples.

	Multi-GPU System			CPU-DRAM System		Ascend 910 System		
Chiplet	CPU	GPU	HBM	CPU	DRAM	Virtuvian	Nimbus	HBM
Widths $[mm]$	12	18.2	7.75	8.25	8.75	31.4	10.5	7.75
Height $[mm]$	12	18.2	11.87	9	8.75	14.5	16	11.87
Power $[W]$	105	295	20	150	20	256	14	20

connects to all other chiplets using a star topology through a silicon interposer. We use publicly available data for the dimensions and power consumption of the chiplets (see Table 5.3).[1] Our evaluation uses 45 $mm \times 45$ mm interposers unless otherwise specified. This interposer size is the minimum required for the 3 systems we evaluated. Of course, for smaller systems this interposer size will be smaller. Since SA is a probabilistic approach, we run the algorithm 5 times and pick the best solution.

[1]In case the data is not publically unavailable, we apply standard technology scaling rules. Our our approach methodology is independent of the area and power values.

5.4.1 Case Study 1: Multi-GPU System

Figure 5·2 shows the thermal maps of a conceptual Multi-GPU System. The placement in Figure 5·2(a) is obtained by using the B*-tree and fast-SA approach, which minimizes wirelength and area, but does not account for temperature. This system operates at 95.31 ^oC with a total wirelength (sum of all inter-chiplet link lengths) of $88,059$ mm. Figure 5·2(b) is the output from our methodology that uses a physical network with repeaterless non-pipelined inter-chiplet links. This layout has a peak temperature of 91.25 ^oC with $96,906$ mm total wirelength as it pushes the high-power CPU and GPU chiplets to the corners. Figure 5·3 shows the tradeoffs between wirelength and temperature as our algorithm determines a solution. Figure 5·2(c) is our placement solution using *gas-station* links. The temperature of the system is 91.52 ^oC but the total wirelength reduces to $51,010$ mm. This is achieved by placing the HBMs in the middle of the CPU and GPU chiplets, where the HBM chiplets provide 'gas-stations' for connections between CPU and GPU chiplets.

Impact of interposer sizes: We use 45 $mm \times 45$ mm interposer in this case study as we can fit all chiplets in that area. When we increase the interposer size

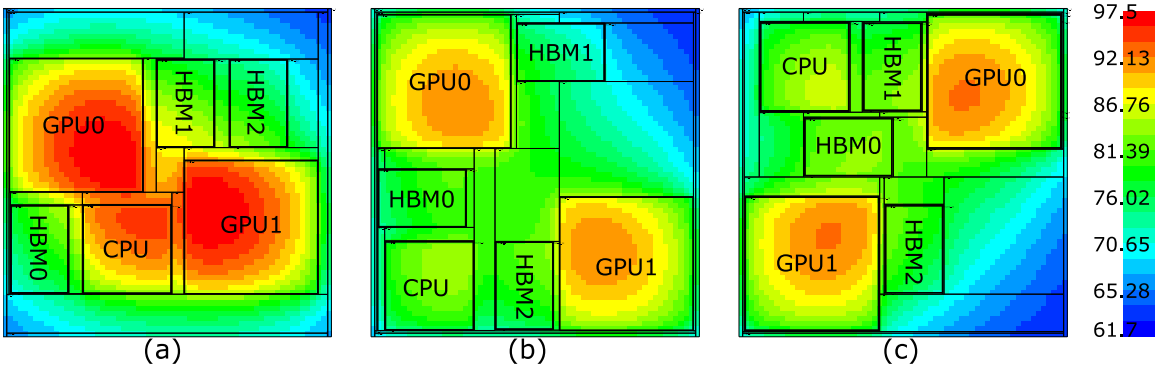

Figure 5·2: Thermal maps of a conceptual Multi-GPU System: (a) a placement solution using B*-tree and fast-SA approach, (b) our thermally-aware placement solution using repeaterless non-pipelined inter-chiplet links, and (c) our placement solution using *gas-station* links.

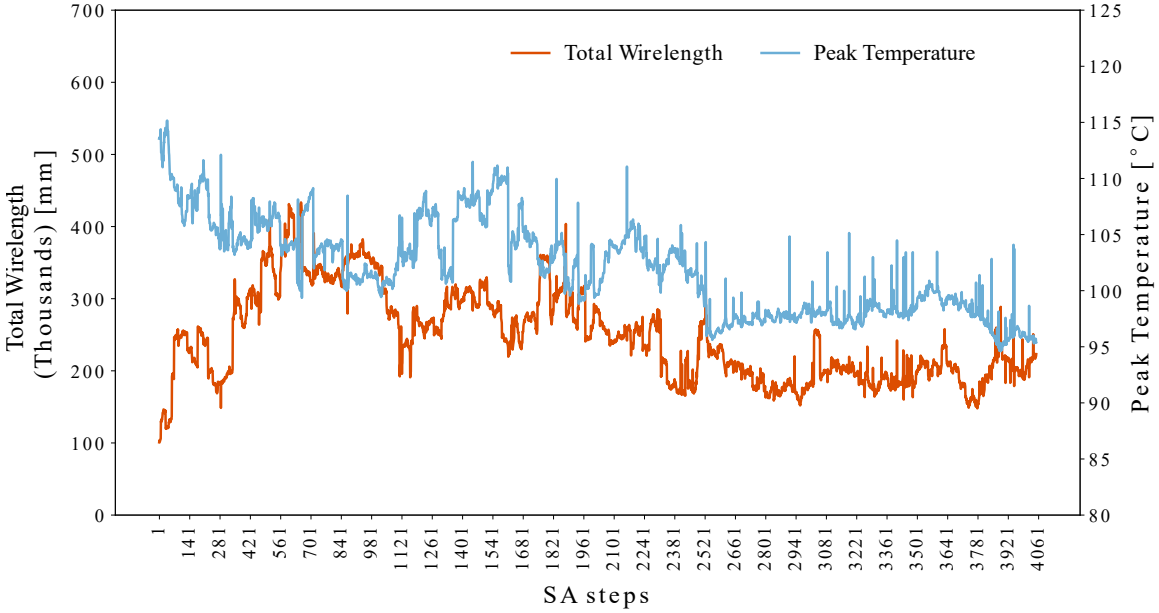

Figure 5·3: Wirelength and temperature at each SA step of our simulated annealing based algorithm for the Multi-GPU case study.

to 50 $mm \times 50$ mm and apply our methodology, we achieve lower temperature but longer wirelength. Compared to the 45 $mm \times 45$ mm interposer, the 50 $mm \times 50$ mm interposer has 2.51 ^oC lower temperature at 5% higher wirelength for the non-pipelined link case and 2.38 ^oC lower temperature at 17% higher wirelength for the *gas-station* link case. However, this tradeoff comes at a 33% higher interposer cost.[2]

5.4.2 Case Study 2: CPU-DRAM System

Figure 5·4 shows the thermal maps of the CPU-DRAM System, where (a) is the original placement (Kannan et al., 2015), (b) is the placement solution using B*-tree and fast-SA approach, (c) and (d) are our thermally-aware placement solutions using repeaterless non-pipelined inter-chiplet link and using *gas-station* links, respectively. The original placement (a) is optimal from the routing perspective (total wirelength

[2]The increase in wirelength could lower performance, but that can be recovered as we are reducing temperature which enables operating at higher voltage and frequency.

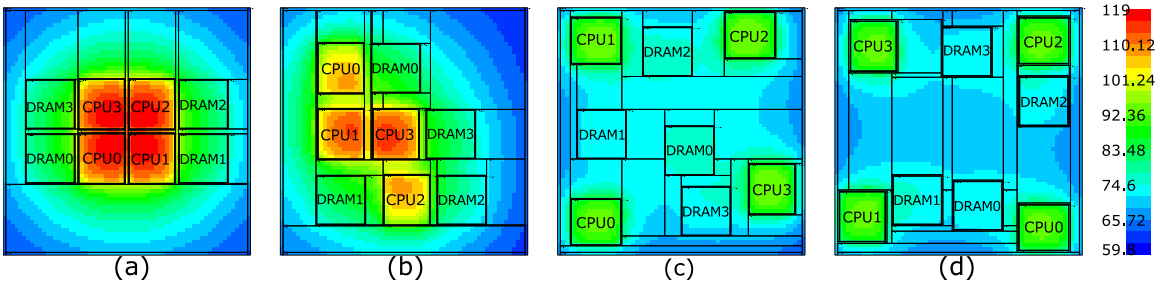

Figure 5·4: Thermal maps of the CPU-DRAM System (Kannan et al., 2015): (a) the original placement, (b) a placement solution using B*-tree and fast-SA approach, (c) our thermally-aware placement solution using repeaterless non-pipelined inter-chiplet link, and (d) using *gas-station* links.

of $67,686$ mm according to our evaluation). However, our HotSpot simulations show that the system operates at 115.94 ^oC, which is thermally infeasible. The placement in (b) is also relatively compact (the total wirelength is $100,864$ mm), therefore, the peak temperature is 113.54 ^oC, which is also thermally infeasible. Our thermally-aware placement solutions in (c) and (d) successfully reduce the peak temperature to 94.89 ^oC and 93.89 ^oC, respectively. It is achieved by pushing the high-power CPU chiplets to the corners of the interposer. The total wirelengths for solutions in (c) and (d) are $216,064$ mm and $138,956$ mm, respectively. *It should be noted here, we are not trading off the 2× to 3× longer wirelength (compared to the original solution (a)) for a lower temperature, the longer wirelength is the price we have to pay to turn a thermally-infeasible design to a thermally-feasible design.* Figure 5·5 shows wirelength and temperature at each SA steps for the case in Figure 5·4(c). After uphill climbing stages, our approach converges at a low temperature with small wirelength overhead.

Impact on TDP: We complete a TDP analysis to highlight the benefit of our thermally-aware physical network design.[3] We vary the CPUs' power in this

[3]We did not do a TDP analysis for case studies 1 and 3. For case study 1, we could vary either CPU power or GPU power, and still operate the system under the same temperature constraint. However, different combinations of CPU and GPU powers lead to different TDP envelopes. For case study 3, we achieve similar placement solution as the commercial product, and there is no change in the TDP envelope.

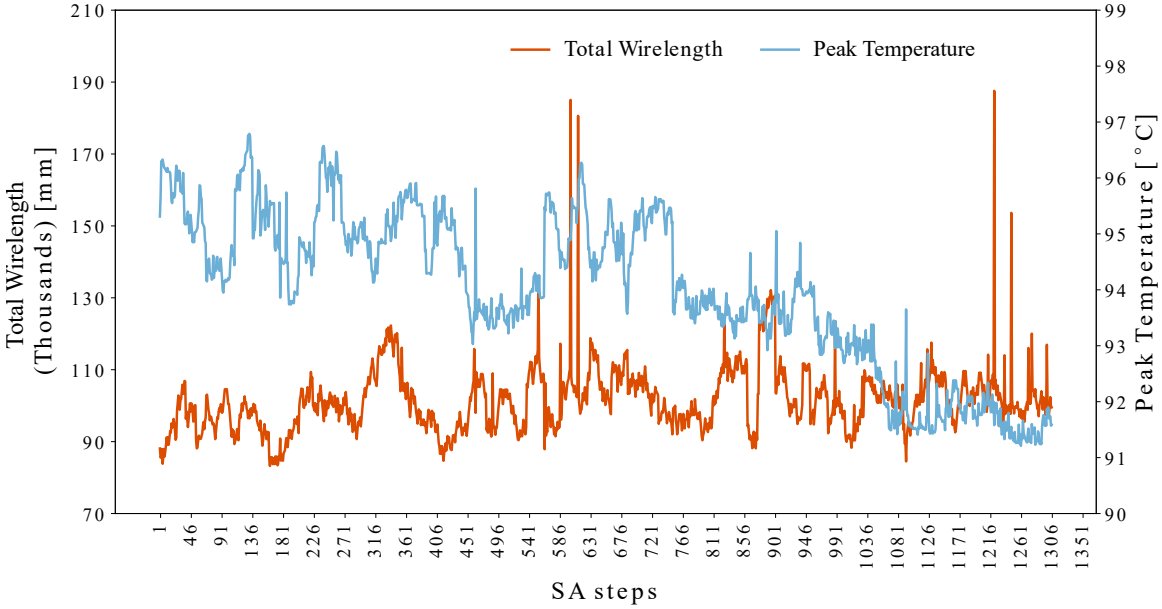

Figure 5·5: Wirelength and temperature at each SA step of our simulated annealing based algorithm for the CPU-DRAM case study.

case study to determine the TDP envelopes (the maximum power of all the chiplets without violating 85 ^oC temperature constraint) of the original CPU-DRAM System (Figure 5·4(a)) (Kannan et al., 2015) and our placement solution (Figure 5·4(c)). The original system shown in (a) can tolerate 400 W, and the system using our approach shown in (c) increases the TDP to 550 W. The TDP increase is achieved by pushing the high-power chiplets away from each other to avoid heat aggregation, which needs longer inter-chiplet links. The power of inter-chiplet network is negligible from prior studies (Coskun et al., 2018), (Coskun et al., 2020). Based on our evaluation using PARSEC, SPLASH2 and UHPC benchmarks, increasing the inter-chiplet link latency from 1 cycle to 2 cycles results in 5% to 18% (11% on average) performance loss, and increasing the latency from 1 cycle to 3 cycles results in 18% to 39% (25% on average) performance loss. However, the increase in TDP envelope can be leveraged to improve performance (e.g., increasing the operating frequency by 30%) without increasing cooling cost.

5.4.3 Case Study 3: Huawei Ascend 910 System

Figure 5·6 shows the thermal maps of the existing Huawei Ascend 910 System (Huawei, 2019). The original layout of Ascend 910 System (Figure 5·6(a)) already achieves minimum wirelength and is thermally-safe when running at the nominal frequency. According to our simulations, the peak temperature of Ascend 910 System is 75.48 ^oC which is below the typical acceptable threshold of 85 ^oC, and the total wirelength is $16,426$ mm. Figure 5·6(b) is a placement solution using B*-tree and fast-SA approach, which focus on reducing wirelength and area. The total wirelength of (b) is $23,794$ mm and the temperature is 75.13 ^oC. We use it as the initial placement in our approach. Figure 5·6(c) is the solution using our approach for the system (it yields the same placement solution with or without *gas-station* links). The solution has $16,597$ mm total wirelength and 75.47 ^oC temperature. Our placement solution is comparable to the actual solution of the commercial chip. This example indicates that for a system already operating at a safe temperature, our methodology focuses on minimizing the wirelength. We show the wirelength and temperature at each SA step in Figure 5·7. The temperature does not change. The wirelength increases in

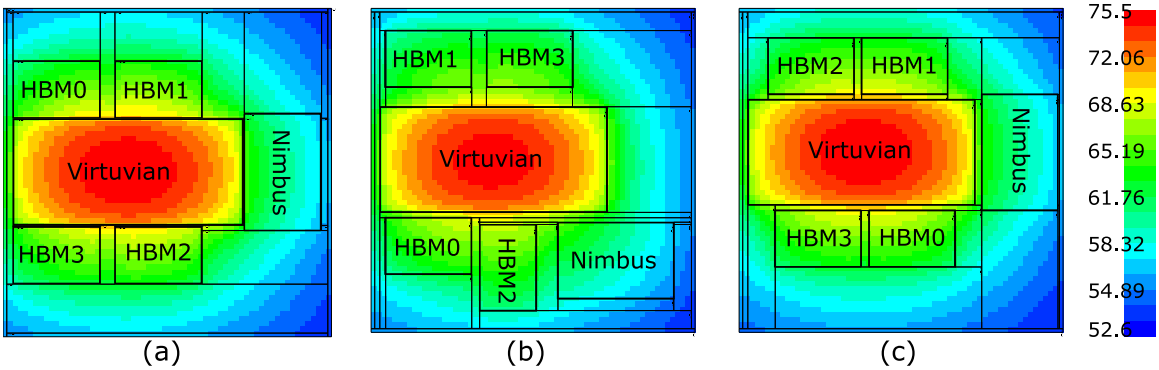

Figure 5·6: Thermal maps of the existing Huawei Ascend 910 System (Huawei, 2019): (a) the exact placement layout, (b) a placement solution using B*-tree and fast-SA approach, and (c) our thermally-aware placement solution.

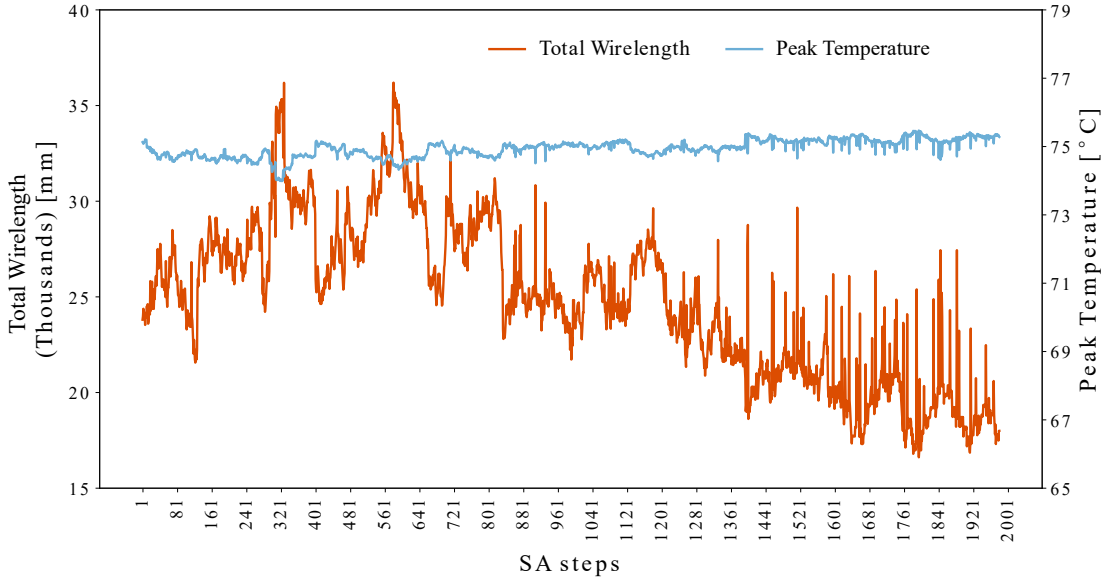

Figure 5·7: Wirelength and temperature at each SA step of our simulated annealing based algorithm for the Ascend 910 case study.

the beginning when the annealing temperature K is large enough to accept a worse neighbor. Approaching to the end of the process, a worse neighbor is unlikely to be accepted and the wirelength converges.

5.4.4 Discussion on Scalability

The case studies we have shown are relatively small 2.5D system examples with up to 8 chiplets. Our methodology also supports heterogeneous 2.5D systems with a large number of chiplets, but requires longer simulation time. The simulation bottleneck of our approach is the external evaluation of thermal profile and routing optimization. The thermal evaluation time is independent of the chiplet count, as we use a fixed grid size (64×64) for the systems. The time spent on routing optimization scales with $O(|C|^2 \cdot |P|^2 \cdot |N|)$, where $|C|$, $|P|$, $|N|$ are the number of chiplets, the number of pin clumps per chiplet, and the number of inter-chiplet channels, respectively. As part of our future work, we will explore the use of machine learning techniques to accelerate the thermal analysis and routing optimization.

5.5 Summary

In this chapter, we have presented an inter-chiplet network design methodology for heterogeneous 2.5D systems. The goal of our methodology is to find the physical design solution for an inter-chiplet network by jointly minimizing the operating temperature of the overall system and total inter-chiplet network wirelength. Our methodology strategically inserts spacing between chiplets to improve heat dissipation, and thus increases the thermal design power of the overall system. We develop a simulated annealing based approach, which searches for a thermally-aware chiplet placement and optimizes the routing of inter-chiplet wires for heterogeneous 2.5D systems. We demonstrate the usage of our methodology by applying it to three heterogeneous 2.5D systems.

Chapter 6

Conclusion and Future Work

6.1 Summary of Major Contributions

2.5D integration is a promising technique for designing homogeneous and heterogeneous computing systems. For homogeneous systems, it breaks down a large monolithic chip into smaller chiplets to increase yield and lower manufacturing cost. For heterogeneous systems, it integrates diversified functional units, which are designed using appropriate technologies and processes, in a package to push the system-level scaling of performance and cost. It provides additional routing resources for high-bandwidth communication between chiplets. 2.5D integration is gaining more popularity in the semiconductor industry, and already there are multiple commercial products designed using this technology, such as Xilinx Virtex 7 (Xilinx, 2016), AMD Fiji (Macri, 2015), Nvidia Tesla (Nvidia, 2016), and Intel Foveros (Intel, 2018). However, these existing products typically place the chiplets adjacent to each other on an interposer to shorten the inter-chiplet link lengths for lower communication latency and reduce the interposer sizes for lower manufacturing cost. The full potential of 2.5D integration technology has not been fully exploited, especially in the thermal aspect. There is a great opportunity to leverage both the cost-effectiveness and the placement flexibility of 2.5D integration to design thermally-aware 2.5D systems.

Our proposed thermally-aware 2.5D systems jointly maximize performance, minimize manufacturing cost, and minimize operating temperature, by selecting the best combination of network topology in the logical layer, chiplet placement in the physi-

cal layer, and inter-chiplet interconnect design in the circuit layer. Our optimization methodology improves heat dissipation capability, and in turn reduces operating temperature, increases TDP budget, and improves thermally-constrained performance of the system. The major contributions of the thesis are summarized below.

In this thesis, we first propose a single-layer optimization methodology for thermally-aware chiplet organization for homogeneous 2.5D systems. The main idea is to break down a large monolithic chip into several smaller chiplets and strategically place them on a silicon interposer in a thermally-aware manner. Our thermally-aware chiplet organization methodology reduces the peak operating temperature of 2.5D systems, and thus reclaims dark silicon by allowing more active cores running at a higher frequency persistently without violating the thermal constraints. We investigate the manufacturing cost model of 2.5D systems, and analyze the thermal behavior of 2.5D systems. We formulate an optimization problem of chiplet organization to jointly maximize performance and minimize manufacturing cost. To solve the optimization problem, we design a multi-start greedy approach to find (near-)optimal solutions efficiently. Our results show that our proposed methodology improves performance by 41% and 16% on average and by up to 87% and 39% for temperature thresholds of 85 ^oC and 105 ^oC, respectively, compared to a traditional single-chip system at the same manufacturing cost. While maintaining the same performance as an equivalent single-chip system, our approach is able to reduce the 2.5D system manufacturing cost by 36%.

Second, we enhance our single-layer methodology to consider the challenges and opportunities across network topologies in the logical layer, chiplet placement and routing in the physical layer, and inter-chiplet link design in the circuit layer. We explore the tradeoffs across these layers and form a cross-layer co-optimization methodology. Our upgraded methodology jointly maximizes performance, minimizes man-

ufacturing cost, and minimizes operating temperature of 2.5D systems. We apply a soft constraint for peak temperature in the optimization problem to achieve higher overall performance and/or lower manufacturing cost by accepting a small amount of thermal violation, while still ensuring thermal safety and routability. We propose a novel *gas-station* link design which enables pipelining between chiplets in a passive interposer to maintain low-cost and high-performance communication between chiplets in 2.5D systems. We develop a simulated annealing algorithm to search the high-dimensional placement solution space, which supports arbitrary placements that consider non-matrix and asymmetric chiplet organizations. Our cross-layer co-optimization methodology achieves better performance-cost tradeoffs of 2.5D systems and yields better solutions in optimizing inter-chiplet network and 2.5D system designs than prior methods. Compared to single-chip systems, 2.5D systems designed using our new approach achieve 88% higher performance at the same manufacturing cost, or 29% lower cost with the same performance. Compared to the closest state-of-the-art, our new approach achieves 40-68% (49% on average) iso-cost performance improvement and 30-38% (32% on average) iso-performance cost reduction.

Third, we extend our cross-layer optimization methodology for homogeneous 2.5D systems and apply it to heterogeneous 2.5D systems, which integrate various components such as CPUs, GPUs, memory stacks, and/or accelerators on a silicon interposer. We apply our extended methodology and develop an EDA tool to account for thermally-aware chiplet placement and efficient routing of inter-chiplet wires in heterogeneous 2.5D systems. Our methodology jointly minimizes the total wirelength and the system temperature with strategically inserted spacing between chiplets. We develop an SA-based approach to optimize the routing of inter-chiplet wires and thermally-aware chiplet placement for heterogeneous 2.5D systems. We enhance the traditional floorplanning algorithm for monolithic chips to support 2.5D systems.

We use a flexible data structure to represent chiplet placement with strategically inserted spacing, which is not supported in traditional floorplan data structures. Our methodology increases the TDP envelope without using any advanced and costly active cooling methods. This increase in TDP envelope allows higher power budget, which can be used to improve performance.

6.2 Future Research Directions

6.2.1 Using Machine Learning Techniques to Speed up Evaluations

The solution space increases exponentially when we generalize our chiplet placement modeling from symmetric matrix-style (approximately $17k$ placement options with 0.5 mm granularity) to arbitrary placement (up to quadrillions (10^{15}) placement options with 1 mm granularity), and from homogeneous systems (where chiplets are identical) to heterogeneous systems (where chiplets are distinct). It is impractical to exhaustively search the entire space to determine the global optimal solution. Therefore, we adopt simulated annealing, a probabilistic approach to approximate the global minimum. The solution quality is related to the number of steps the SA algorithm explores. However, our simulation framework has to depend on external tools for thermal evaluation and routing optimization, which takes at least 30 s for each step. For manageable simulation time, we stop simulation after a few thousand steps.

One possible future research direction is to adopt machine learning technique to speed up the thermal evaluation and routing optimization, which are the bottlenecks of simulation time. Thus, we can speed up the simulation process significantly with the same number of steps, or we can allow more steps to improve the solution quality within the same time budget.

6.2.2 Extending Our Methodology for Active Interposer

In this thesis, we develop our cross-layer optimization methodology to maximize performance and minimize manufacturing cost and operating temperature. From a cost perspective, our work focus on passive interposer, which is much cheaper than active interposer. Active interposer is also a popular 2.5D integration option (Kannan et al., 2015), (Jerger et al., 2014), especially in the case that performance is the major goal and cost is not a critical concern.

One potential future direction is to extend our methodology to active interposers. Active interposer can house active components such as repeaters and flip-flops to enable repeated pipelined links, which provides higher link performance and lower latency since the flip-flops do not have to be placed in the *gas stations* of the chiplets. The wire routing can then potentially avoid the detours and be more flexible. Another advantage of active interposer over passive interposer is that it can house routers as well. The story of routing would be completely changed, as shown in Figure 6·1. For passive interposers, the routers have to be placed in the chiplets. We have to use either repeaterless non-pipelined link to connect them through the interposer or stop by intermediate *gas stations* on other chiplets. Thus, it is likely to end up

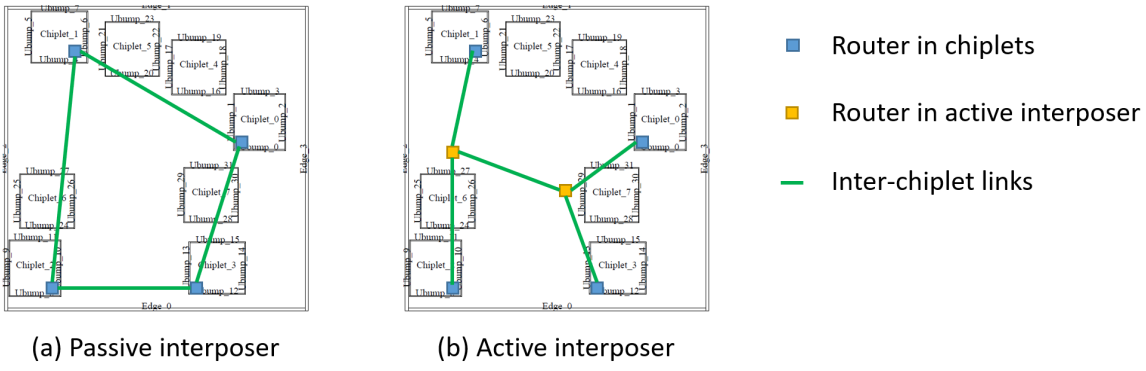

Figure 6·1: Router Placement in (a) passive interposer and (b) active interposer.

with long distance inter-chiplet links, which either requires longer latency or higher energy consumption. For active interposers, the routers can be placed anywhere in the interposer, as shown in Figure 6·1(b). Therefore, appropriate choice of router placement could potentially reduce the link length and latency, and in turn improve system performance.

6.2.3 Using Photonic Links to Provide High-bandwidth Low-latency Communication

The silicon photonic link is another promising candidate for inter-chiplet communication in 2.5D systems, which provides high bandwidth and low energy consumption. It is believed that future high-performance systems will dramatically benefit from integrating silicon photonic links, as electrical interconnects do not scale with system requirements at certain thresholds of bandwidth, power, distance, latency, and cost (Abellán et al., 2016), (Arakawa et al., 2013), (Glick, 2013), (Krishnamoorthy et al., 2015), (Batten et al., 2009), (Joshi et al., 2009), (Ziabari et al., 2015), (Beamer et al., 2009). Ayar Labs and Intel have already demonstrated early progress using photonic links to replace traditional electrical interconnects in a package under DARPAs Photonics in the Package for Extreme Scalability (PIPES) program (Leibson, 2020),(Labs, 2020).

One potential future work direction is to incorporate silicon photonic links into our cross-layer optimization methodology to assist design automation with focus on 2.5D systems and photonic links. It would be beneficial to both the photonic device designers and system designers. To bridge the gap, a thorough investigation of performance and cost tradeoffs of using silicon-photonic interposer, a complete design space exploration of 2.5D silicon photonic network design including architecture, laser placement and waveguide layout, and a comprehensive optimization methodology that focuses on both thermally-aware and temperature-gradient-aware 2.5D systems are required.

References

Abellán, J. L., Coskun, A. K., Gu, A., Jin, W., Joshi, A., Kahng, A. B., Klamkin, J., Morales, C., Recchio, J., Srinivas, V., et al. (2016). Adaptive tuning of photonic devices in a photonic NoC through dynamic workload allocation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(5):801–814.

Ahmed, M. M., Shamim, M. S., Mansoor, N., Mamun, S. A., and Ganguly, A. (2017). Increasing interposer utilization: A scalable, energy efficient and high bandwidth multicore-multichip integration solution. In *2017 Eighth International Green and Sustainable Computing Conference (IGSC)*, pages 1–6. IEEE.

Akgun, I., Zhan, J., Wang, Y., and Xie, Y. (2016). Scalable memory fabric for silicon interposer-based multi-core systems. In *2016 IEEE 34th International Conference on Computer Design (ICCD)*, pages 33–40. IEEE.

Allred, J., Roy, S., and Chakraborty, K. (2012). Designing for dark silicon: a methodological perspective on energy efficient systems. In *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*, pages 255–260.

Amazon (2018). Amazon EC2 reserved instances pricing. `https://aws.amazon.com/ec2/pricing/reserved-instances/pricing/`.

Arakawa, Y., Nakamura, T., Urino, Y., and Fujita, T. (2013). Silicon photonics for next generation system integration platform. *IEEE Communications Magazine*, 51(3):72–77.

Batten, C., Joshi, A., Orcutt, J., Khilo, A., Moss, B., Holzwarth, C. W., Popovic, M. A., Li, H., Smith, H. I., Hoyt, J. L., et al. (2009). Building many-core processor-to-dram networks with monolithic cmos silicon photonics. *IEEE Micro*, 29(4):8–21.

Beamer, S., Sun, C., Kwon, Y.-j., Joshi, A., Batten, C., Stojanovic, V., and Asanovi, K. (2009). Re-architecting dram with monolithically integrated silicon photonics. In *Proceedings of the 37th International Symposium on Computer Architecture*, pages 129–140.

Bienia, C., Kumar, S., Singh, J. P., and Li, K. (2008). The PARSEC benchmark suite: Characterization and architectural implications. In *Proceedings of the 17th*

international conference on Parallel architectures and compilation techniques, pages 72–81.

Campbell, D., Bader, D., Brandt, S., Cook, D., Gokhale, M., Hornung, R., Keasler, J., LeBlanc, P., Marin, G., Mulvaney, B., et al. (2012). Ubiquitous high performance computing: Challenge problems specification. *Georgia Tech. Res. Inst., Atlanta, GA, USA, Tech. Rep. HR0011-10-C-0145*.

Carlson, T. E., Heirman, W., and Eeckhout, L. (2011). Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–12.

Charbonnier, J., Assous, M., Bally, J.-P., Miyairi, K., Sunohara, M., Cuchet, R., Feldis, H., Bouzaida, N., Bernard-Henriques, N., Hida, R., et al. (2012). High density 3D silicon interposer technology development and electrical characterization for high end applications. In *2012 4th Electronic System-Integration Technology Conference*, pages 1–7. IEEE.

Chaware, R., Nagarajan, K., and Ramalingam, S. (2012). Assembly and reliability challenges in 3D integration of 28nm FPGA die on a large high density 65nm passive interposer. In *2012 IEEE 62nd Electronic Components and Technology Conference*, pages 279–283. IEEE.

Chen, G., Anders, M. A., Kaul, H., Satpathy, S. K., Mathew, S. K., Hsu, S. K., Agarwal, A., Krishnamurthy, R. K., De, V., and Borkar, S. (2015). A 340 mV-to-0.9 V 20.2 Tb/s source-synchronous hybrid packet/circuit-switched 16× 16 network-on-chip in 22 nm tri-gate CMOS. *IEEE Journal of Solid-State Circuits*, 50(1):59–67.

Chen, T.-C. and Chang, Y.-W. (2006). Modern floorplanning based on B/sup*/-tree and fast simulated annealing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(4):637–650.

Cochet, K. R. P., McCleary, R., Rogoff, R., and Roy, R. (2014). Lithography challenges for 2.5 D interposer manufacturing. In *2014 IEEE 64th Electronic Components and Technology Conference (ECTC)*, pages 523–527. IEEE.

Cong, J., Wei, J., and Zhang, Y. (2004). A thermal-driven floorplanning algorithm for 3D ICs. In *IEEE/ACM International Conference on Computer Aided Design, 2004. ICCAD-2004.*, pages 306–313. IEEE.

Consoli, E., Alioto, M., Palumbo, G., and Rabaey, J. (2012). Conditional push-pull pulsed latches with 726fJ· ps energy-delay product in 65nm CMOS. In *2012 IEEE International Solid-State Circuits Conference*, pages 482–484. IEEE.

Coskun, A., Eris, F., Joshi, A., Kahng, A. B., Ma, Y., Narayan, A., and Srinivas, V. (2020). Cross-layer co-optimization of network design and chiplet placement in 2.5 d systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

Coskun, A., Eris, F., Joshi, A., Kahng, A. B., Ma, Y., and Srinivas, V. (2018). A cross-layer methodology for design and optimization of networks in 2.5D systems. In *Proceedings of International Conference on Computer-Aided Design (ICCAD), San Diego, CA*, page 101.

Costa, N. R. and Lourenço, J. A. (2015). Exploring pareto frontiers in the response surface methodology. In *Transactions on Engineering Technologies*, pages 399–412. Springer.

Dennard, R. H., Gaensslen, F. H., Rideout, V. L., Bassous, E., and LeBlanc, A. R. (1974). Design of ion-implanted MOSFET's with very small physical dimensions. *IEEE Journal of Solid-State Circuits*, 9(5):256–268.

Dreslinski, R. G., Wieckowski, M., Blaauw, D., Sylvester, D., and Mudge, T. (2010). Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits. *Proceedings of the IEEE*, 98(2):253–266.

Eris, F., Joshi, A., Kahng, A. B., Ma, Y., Mojumder, S., and Zhang, T. (2018). Leveraging thermally-aware chiplet organization in 2.5D systems to reclaim dark silicon. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1441–1446. IEEE.

Esmaeilzadeh, H., Blem, E., Amant, R. S., Sankaralingam, K., and Burger, D. (2011). Dark silicon and the end of multicore scaling. In *2011 38th Annual international symposium on computer architecture (ISCA)*, pages 365–376. IEEE.

Fang, E. J.-W., Shih, T. C.-J., and Huang, D. S.-Y. (2015). IR to routing challenge and solution for interposer-based design. In *The 20th Asia and South Pacific Design Automation Conference*, pages 226–230. IEEE.

Farrens, S. and MicroTec, S. (2010). Wafer and die bonding technologies for 3D integration. *Equipment for Electronic Products Manufacturing*, 10.

Frantz, F., Labrak, L., and O'Connor, I. (2012). 3D IC floorplanning: Automating optimization settings and exploring new thermal-aware management techniques. *Microelectronics Journal*, 43(6):423–432.

Friedman, E. (Dec. 2016). Personal Communication.

Glick, M. (2013). Optical interconnects in next generation data centers: An end to end view. In *Optical Interconnects for Future Data Center Networks*, pages 31–46. Springer.

Goulding-Hotta, N., Sampson, J., Venkatesh, G., Garcia, S., Auricchio, J., Huang, P.-C., Arora, M., Nath, S., Bhatt, V., Babb, J., et al. (2011). The greendroid mobile application processor: An architecture for silicon's dark future. *IEEE Micro*, 31(2):86–95.

Grani, P., Proietti, R., Akella, V., and Ben Yoo, S. (2016). Photonic interconnects for interposer-based 2.5D/3D integrated systems on a chip. In *Proceedings of the Second International Symposium on Memory Systems*, pages 377–386.

Grani, P., Proietti, R., Akella, V., and Yoo, S. B. (2017). Design and evaluation of AWGR-based photonic noc architectures for 2.5D integrated high performance computing systems. In *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 289–300. IEEE.

Guo, P.-N., Cheng, C.-K., and Yoshimura, T. (1999). An O-tree representation of non-slicing floorplan and its applications. In *Proceedings 1999 Design Automation Conference (Cat. No. 99CH36361)*, pages 268–273. IEEE.

Han, J. and Orshansky, M. (2013). Approximate computing: An emerging paradigm for energy-efficient design. In *2013 18th IEEE European Test Symposium (ETS)*, pages 1–6. IEEE.

Healy, M., Vittes, M., Ekpanyapong, M., Ballapuram, C. S., Lim, S. K., Lee, H.-H. S., and Loh, G. H. (2006). Multiobjective microarchitectural floorplanning for 2-D and 3-D ICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(1):38–52.

Heroux, M. (2007). HPCCG MicroApp.

HIR (2019). Heterogeneous integration roadmap 2019 edition. `https://eps.ieee.org/technology/heterogeneous-integration-roadmap/2019-edition.html`.

Hot Chips (2017). Hot chips 2017: Intel deep dives into EMIB. `https://www.tomshardware.com/news/intel-emib-interconnect-fpga-chiplet,35316.html`.

Howard, J., Dighe, S., Vangal, S. R., Ruhl, G., Borkar, N., Jain, S., Erraguntla, V., Konow, M., Riepen, M., Gries, M., et al. (2011). A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling. *IEEE Journal of Solid-State Circuits*, 46(1):173–183.

HSPICE (2009). Reference manual: Commands and control options.

Huawei (2019). Huawei ascend 910 provides a nvidia ai training alternative. `https://www.servethehome.com/huawei-ascend-910-provides-a-nvidia-ai-training-alternative/`.

Hung, W.-L., Xie, Y., Vijaykrishnan, N., Addo-Quaye, C., Theocharides, T., and Irwin, M. J. (2005). Thermal-aware floorplanning using genetic algorithms. In *Sixth international symposium on quality electronic design (isqed'05)*, pages 634–639. IEEE.

Intel (2018). Intel introduces foveros: 3D die stacking for more than just memory. `https://arstechnica.com/gadgets/2018/12/intel-introduces-foveros-3d-die-stacking-for-more-than-just-memory/`.

Intel (2019). Intel EMIB White Paper. `https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/wp/wp-01251-enabling-nextgen-with-3d-system-in-package.pdf`.

ITRS (2015a). Heterogeneous Integration Chapter in ITRS 2.0. `http://www.itrs2.net/itrs-reports.html`.

ITRS (2015b). System Integration Chapter in ITRS 2.0. `http://www.itrs2.net/itrs-reports.html`.

Iyer, S. S. (2016). Heterogeneous integration for performance and scaling. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 6(7):973–982.

Jeddeloh, J. and Keeth, B. (2012). Hybrid memory cube new dram architecture increases density and performance. In *2012 symposium on VLSI technology (VLSIT)*, pages 87–88. IEEE.

Jerger, N. E., Kannan, A., Li, Z., and Loh, G. H. (2014). NoC architectures for silicon interposer systems: Why pay for more wires when you can get them (from your interposer) for free? In *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 458–470. IEEE.

Joshi, A., Batten, C., Kwon, Y.-J., Beamer, S., Shamim, I., Asanovic, K., and Stojanovic, V. (2009). Silicon-photonic clos networks for global on-chip communication. In *2009 3rd ACM/IEEE International Symposium on Networks-on-Chip*, pages 124–133. IEEE.

Kannan, A., Jerger, N. E., and Loh, G. H. (2015). Enabling interposer-based disintegration of multi-core processors. In *2015 48th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 546–558. IEEE.

Karim, M. A., Franzon, P. D., and Kumar, A. (2013). Power comparison of 2D, 3D and 2.5D interconnect solutions and power optimization of interposer interconnects. In *2013 IEEE 63rd Electronic Components and Technology Conference*, pages 860–866. IEEE.

Kim, D.-W., Au, K., Luo, H. Y. L. X., Ye, Y. L., Bhattacharya, S., and Lo, G. Q. (2017). 2.5 D silicon optical interposer for 400 Gbps electronic-photonic integrated circuit platform packaging. In *2017 IEEE 19th Electronics Packaging Technology Conference (EPTC)*, pages 1–4. IEEE.

Knickerbocker, J., Andry, P., Colgan, E., Dang, B., Dickson, T., Gu, X., Haymes, C., Jahnes, C., Liu, Y., Maria, J., et al. (2012). 2.5D and 3D technology challenges and test vehicle demonstrations. In *2012 IEEE 62nd Electronic Components and Technology Conference*, pages 1068–1076. IEEE.

Knudsen, J. (2008). NanGate 45nm open cell library. *CDNLive, EMEA*.

Krishnamoorthy, A., Schwetman, H., Zheng, X., and Ho, R. (2015). Energy-efficient photonics in future high-connectivity computing systems. *Journal Of Lightwave Technology*, 33(4):889–900.

Kulkarni, P., Gupta, P., and Ercegovac, M. (2011). Trading accuracy for power with an underdesigned multiplier architecture. In *2011 24th Internatioal Conference on VLSI Design*, pages 346–351. IEEE.

Labs, A. (2020). Ayar labs, darpa and intel replace electronic i/o with efficient optical signaling. `https://insidehpc.com/2020/03/ayar-labs-darpa-and-intel-replace-electronic-i-o-with-efficient-optical-signaling/`.

Leibson, S. (2020). Ayar labs and Intel demo FPGA with optical transceivers in DARPA PIPES project: 2 Tbps now, ¿100 Tbps is the goal. `https://blogs.intel.com/psg/ayar-labs-and-intel-demo-fpga-with-optical-transceivers-in-darpa-pipes-project-2-tbps-now-100-tbps-is-the-goal/`.

Li, S., Ahn, J. H., Strong, R. D., Brockman, J. B., Tullsen, D. M., and Jouppi, N. P. (2009). McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 469–480.

Lin, J.-M. and Chang, Y.-W. (2001). TCG: A transitive closure graph-based representation for non-slicing floorplans. In *Proceedings of the 38th annual Design Automation Conference*, pages 764–769.

Liu, W.-H., Chien, T.-K., and Wang, T.-C. (2014). Metal layer planning for silicon interposers with consideration of routability and manufacturing cost. In *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6. IEEE.

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

Long, B. (2013). Estimating heat transfer coefficients. https://altasimtechnologies.com/electronic-cooling/estimating-heat-transfer-coefficients/.

Macri, J. (2015). AMD's next generation GPU and high bandwidth memory architecture: FURY. In *2015 IEEE Hot Chips 27 Symposium (HCS)*, pages 1–26. IEEE.

Mahajan, R., Sankman, R., Patel, N., Kim, D.-W., Aygun, K., Qian, Z., Mekonnen, Y., Salama, I., Sharan, S., Iyengar, D., et al. (2016). Embedded multi-die interconnect bridge (EMIB)–a high density, high bandwidth packaging interconnect. In *2016 IEEE 66th Electronic Components and Technology Conference (ECTC)*, pages 557–565. IEEE.

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

Meta-software (1996). HSPICE users manual. Inc.: Campbell, CA.

Murata, H., Fujiyoshi, K., Nakatake, S., and Kajitani, Y. (1996). VLSI module placement based on rectangle-packing by the sequence-pair. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(12):1518–1524.

Murayama, K., Aizawa, M., Hara, K., Sunohara, M., Miyairi, K., Mori, K., Charbonnier, J., Assous, M., Bally, J.-P., Simon, G., et al. (2013). Warpage control of silicon interposer for 2.5 D package application. In *2013 IEEE 63rd Electronic Components and Technology Conference*, pages 879–884. IEEE.

Muthukaruppan, T. S., Pricopi, M., Venkataramani, V., Mitra, T., and Vishin, S. (2013). Hierarchical power management for asymmetric multi-core in dark silicon era. In *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–9. IEEE.

Narayan, A., Thonnart, Y., Vivet, P., Tortolero, C. F., and Coskun, A. K. (2019). Waves: Wavelength selection for power-efficient 2.5 d-integrated photonic nocs. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 516–521. IEEE.

Nvidia (2016). Nvidia: NVIDIA Tesla P100. https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

Osmolovskyi, S., Knechtel, J., Markov, I. L., and Lienig, J. (2018). Optimal die placement for interposer-based 3D ICs. In *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 513–520. IEEE.

Pagani, S., Khdr, H., Munawar, W., Chen, J.-J., Shafique, M., Li, M., and Henkel, J. (2014). TSP: Thermal safe power: Efficient power budgeting for many-core systems in dark silicon. In *Proceedings of the 2014 International Conference on Hardware/Software Codesign and System Synthesis*, pages 1–10.

Parès, G. (2013). 3D interposer for silicon photonics. *LETI Innovations Days*.

Pares, G. (2013). 3D technology for photonics silicon interposer. In *Green IT workshop-Leti Days*.

Radojcic, R. (2017). *More-than-Moore 2.5D and 3D SiP Integration*. Springer.

Raghavan, A., Luo, Y., Chandawalla, A., Papaefthymiou, M., Pipe, K. P., Wenisch, T. F., and Martin, M. M. (2012). Computational sprinting. In *IEEE international symposium on high-performance comp architecture*, pages 1–12. IEEE.

Ramalingam, S. (2016). HBM package integration: Technology trends, challenges and applications. In *2016 IEEE Hot Chips 28 Symposium (HCS)*, pages 1–17. IEEE.

Ravishankar, C., Gaitonde, D., and Bauer, T. (2018). Placement strategies for 2.5D FPGA fabric architectures. In *2018 28th International Conference on Field Programmable Logic and Applications (FPL)*, pages 16–164. IEEE.

Sankaranarayanan, K., Velusamy, S., Stan, M., Skadron, K., et al. (2005). A case for thermal-aware floorplanning at the microarchitectural level. *Journal of Instruction-Level Parallelism*, 7(1):8–16.

Schaller, R. R. (1997). Moore's law: past, present and future. *IEEE spectrum*, 34(6):52–59.

Seemuth, D. P., Davoodi, A., and Morrow, K. (2015). Automatic die placement and flexible I/O assignment in 2.5D IC design. In *Sixteenth International Symposium on Quality Electronic Design*, pages 524–527. IEEE.

Shamim, M. S., Mansoor, N., Narde, R. S., Kothandapani, V., Ganguly, A., and Venkataraman, J. (2017). A wireless interconnection framework for seamless inter and intra-chip communication in multichip systems. *IEEE Transactions on Computers*, 66(3):389–402.

Silvano, C., Palermo, G., Xydis, S., and Stamelakos, I. (2014). Voltage island management in near threshold manycore architectures to mitigate dark silicon. In *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6. IEEE.

Stow, D., Akgun, I., Barnes, R., Gu, P., and Xie, Y. (2016). Cost analysis and cost-driven IP reuse methodology for SoC design based on 2.5D/3D integration. In *Proceedings of the 35th International Conference on Computer-Aided Design*, pages 1–6.

Stow, D., Xie, Y., Siddiqua, T., and Loh, G. H. (2017). Cost-effective design of scalable high-performance systems using active and passive interposers. In *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 728–735. IEEE.

Sun, C., Chen, C.-H. O., Kurian, G., Wei, L., Miller, J., Agarwal, A., Peh, L.-S., and Stojanovic, V. (2012). DSENT-a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling. In *2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip*, pages 201–210. IEEE.

Swaminathan, K., Kultursay, E., Saripalli, V., Narayanan, V., Kandemir, M. T., and Datta, S. (2013). Steep-slope devices: From dark to dim silicon. *IEEE Micro*, 33(5):50–59.

Tran, K. et al. (2016). High-bandwidth memory white paper: Start your HBM/2.5D design today. *Amkor Technology Inc., Tech. Rep.* `https://c44f5d406df450f4a66b-1b94a87d576253d9446df0a9ca62e142.ssl.cf2.rackcdn.com/2017/12/Start_Your_HBM_25D_Design_Today_WhitePaper_0416.pdf`.

Urino, Y., Usuki, T., Fujikata, J., Ishizaka, M., Yamada, K., Horikawa, T., Nakamura, T., and Arakawa, Y. (2014). High-density and wide-bandwidth optical interconnects with silicon optical interposers. *Photonics Research*, 2(3):A1–A7.

Venkatesh, G., Sampson, J., Goulding-Hotta, N., Venkata, S. K., Taylor, M. B., and Swanson, S. (2011). Qscores: Trading dark silicon for scalable energy efficiency with quasi-specific cores. In *2011 44th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 163–174. IEEE.

Waldrop, M. M. (2016). More than moore. *Nature*, 530(7589):144–148.

Whitelaw, J. H. (1997). Convective heat transfer. *International Encyclopedia of Heat and Mass Transfer*, page 237.

Wong, H. (2012). A comparison of Intels 32nm and 22nm core i5 CPUs: Power, voltage, temperature, and frequency. `http://blog.stuffedcow.net/2012/10/intel32nm-22nm-core-i5-comparison/`.

Wong, S.-C., Lee, G.-Y., and Ma, D.-J. (2000). Modeling of interconnect capacitance, delay, and crosstalk in vlsi. *IEEE Transactions on semiconductor manufacturing*, 13(1):108–111.

Woo, S. C., Ohara, M., Torrie, E., Singh, J. P., and Gupta, A. (1995). The SPLASH-2 programs: characterization and methodological considerations. In *Proceedings of the 22nd annual international symposium on Computer architecture*, pages 24–36.

Xilinx (2016). FPGA VC707 evaluation kit. Virtex-7, Xilinx.

Yan, G., Li, Y., Han, Y., Li, X., Guo, M., and Liang, X. (2012). Agileregulator: A hybrid voltage regulator scheme redeeming dark silicon for power efficiency in a multicore architecture. In *IEEE International Symposium on High-Performance Comp Architecture*, pages 1–12. IEEE.

Zhang, R., Stan, M. R., and Skadron, K. (2015). Hotspot 6.0: Validation, acceleration and extension. *University of Virginia, Tech. Rep.* `https://www.cs.virginia.edu/~skadron/Papers/HotSpot60_TR.pdf`.

Zhang, T., Abellán, J. L., Joshi, A., and Coskun, A. K. (2014). Thermal management of manycore systems with silicon-photonic networks. In *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6. IEEE.

Zhang, Y., Sarvey, T. E., and Bakir, M. S. (2017). Thermal evaluation of 2.5-D integration using bridge-chip technology: Challenges and opportunities. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 7(7):1101–1110.

Zhang, Z. and Wong, C. (2004). Recent advances in flip-chip underfill: materials, process, and reliability. *IEEE transactions on advanced packaging*, 27(3):515–524.

Ziabari, A. K. K., Abellán, J. L., Ubal, R., Chen, C., Joshi, A., and Kaeli, D. (2015). Leveraging silicon-photonic noc for designing scalable gpus. In *Proceedings of the 29th ACM on International Conference on Supercomputing*, pages 273–282.

CURRICULUM VITAE

Yenai Ma

Education

Ph.D., Boston University, 09/2020
Computer Engineering Department
Advisor: Professor Ajay Joshi
Dissertation Title: "Design Thermally-Aware 2.5D Systems"

B.S., University of Alberta, Canada, 06/2014
Department of Electrical and Computer Engineering

Research Projects

Inter-Chiplet Network Design in Heterogeneous 2.5D Systems, 05/2019 to 05/2020
The goal of the project is to develop a thermally-aware chiplet placement and network routing methodology for heterogeneous 2.5D systems. We developed a simulated annealing based algorithm to search for placement solution. We optimize for performance, cost, and operating temperature.

Cross-Layer Co-Optimization of Network Design and Chiplet Placement in 2.5D Systems, 09/2017 to 05/2019
The goal is to optimize network, link type choices, and floorplan for 2.5D systems in three layers (logical, physical, and circuit) designs to maximize performance and minimize cost and operating temperature. We propose novel 'gas-station links to enable pipelined inter-chiplet links in passive interposer. Our methodology supports arbitrary chiplet placement instead of matrix-styled placement, improve cost model, and apply soft temperature constraint to achieve a better performance-cost tradeoff.

Reclaiming Dark Silicon using Thermally-Aware Chiplet Organization in 2.5D Systems, 01/2016 to 09/2017
The goal of the project is to address the dark silicon problem in manycore systems. We investigate manufacturing cost and thermal behavior of 2.5D systems, and strategically place the chiplets in a thermally-aware fashion to reduce operating temperature and thus allowing more active cores running at a higher frequency to improve per-

formance. We optimize the chiplet organization that jointly maximizes performance and minimizes manufacturing cost.

Cross-layer Floorplan Optimization for Silicon Photonic NoCs, 01/2015 to 09/2015
The goal of the project is to optimize the floorplan of silicon photonic network (PNoC), which is more scalable and power efficient compared to electrical NoCs. We address cross-layer effects that span optical and electrical boundaries, chip thermal profiles, or effects of job scheduling policies. We optimize the PNoC to minimize power and area.

Asymmetric NoC Architecture for GPU Systems, 09/2014 to 01/2015
The goal of the project is to improve network energy efficiency for GPU systems. We analyze the memory access pattern of GPU systems and use it to tailor the network design. We design asymmetric networks for L1-to-L2 and L2-to-L1 traffic to provide energy-efficient communication while maintaining performance.

Teaching Experience

EC311: Introduction to Logic Design, Spring 2016 - Fall 2016

Publications

(*Authorship in these publications is in alphabetical order; $^{+}$Y. MA is the primary author.)

1. *$^{+}$ Ayse K. Coskun, Furkan Eris, Ajay Joshi, Andrew B. Kahng, **Yenai Ma**, Aditya Narayan and Vaishnav Srinivas. "Cross-Layer Co-Optimization of Network Design and Chiplet Placement in 2.5 D Systems." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2020).

2. * Ayse K. Coskun, Furkan Eris, Ajay Joshi, Andrew B. Kahng, **Yenai Ma**, and Vaishnav Srinivas. "A cross-layer methodology for design and optimization of networks in 2.5 d systems." In *Proceedings of the International Conference on Computer-Aided Design*, pp. 1-8. 2018.

3. *$^{+}$ Furkan Eris, Ajay Joshi, Andrew B. Kahng, **Yenai Ma**, Saiful Mojumder, and Tiansheng Zhang. "Leveraging thermally-aware chiplet organization in 2.5 D systems to reclaim dark silicon." In IEEE *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1441-1446, 2018.

4. Amir Kavyan Ziabari, Yifan Sun, **Yenai Ma**, Dana Schaa, Jos L. Abelln, Rafael Ubal, John Kim, Ajay Joshi, and David Kaeli. "UMH: A hardware-based unified

memory hierarchy for systems with multiple discrete GPUs." ACM *Transactions on Architecture and Code Optimization (TACO)* 13, no. 4: 1-25, 2016.

5. * Ayse K. Coskun, Anjun Gu, Warren Jin, Ajay Joshi, Andrew B. Kahng, Jonathan Klamkin, **Yenai Ma**, John Recchio, Vaishnav Srinivas, and Tiansheng Zhang. "Cross-layer Floorplan Optimization for Silicon Photonic NoCs In Many-core Systems". In *Proc. Design, Automation and Test in Europe (DATE)*, pp. 1309-1314, 2016.

6. Ziabari, Amir Kavyan, Jos L. Abelln, **Yenai Ma**, Ajay Joshi, and David Kaeli. "Asymmetric NoC architectures for GPU systems." In *Proceedings of the 9th International Symposium on Networks-on-Chip*, pp. 1-8. 2015.