A Cross-Layer Methodology for Design and Optimization of Networks in 2.5D Systems

Ayse Coskun[1], Furkan Eris[1], Ajay Joshi[1], Andrew B. Kahng[2,3], Yenai Ma[1], and Vaishnav Srinivas[2]

[1]ECE Department, Boston University, Boston, MA, USA; [2]ECE and [3]CSE Departments, UC San Diego, La Jolla, CA, USA
acoskun@bu.edu,fe@bu.edu,joshi@bu.edu,yenai@bu.edu,abk@eng.ucsd.edu,vaishnav@ucsd.edu

ABSTRACT

2.5D integration technology is gaining popularity in the design of homogeneous and heterogeneous many-core computing systems. 2.5D network design, both inter- and intra-chiplet, impacts overall system performance as well as its manufacturing cost and thermal feasibility. This paper introduces a cross-layer methodology for designing networks in 2.5D systems. We optimize the network design and chiplet placement jointly across logical, physical, and circuit layers to achieve an energy-efficient network, while maximizing system performance, minimizing manufacturing cost, and adhering to thermal constraints. In the logical layer, our co-optimization considers eight different network topologies. In the physical layer, we consider routing, microbump assignment, and microbump pitch constraints to account for the extra costs associated with microbump utilization in the inter-chiplet communication. In the circuit layer, we consider both passive and active links with five different link types, including a *gas station* link design. Using our cross-layer methodology results in more accurate determination of (superior) inter-chiplet network and 2.5D system designs compared to prior methods. Compared to 2D systems, our approach achieves 29% better performance with the same manufacturing cost, or 25% lower cost with the same performance.

ACM Reference Format:
Ayse Coskun[1], Furkan Eris[1], Ajay Joshi[1], Andrew B. Kahng[2,3], Yenai Ma[1], and Vaishnav Srinivas[2]. 2018. A Cross-Layer Methodology for Design and Optimization of Networks in 2.5D Systems. In *IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN (ICCAD '18), November 5–8, 2018, San Diego, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3240765.3240768

1 INTRODUCTION

The need to sustain the historical performance and cost scaling in computing systems has led to a growing interest in 2.5D systems [1, 8, 9, 15, 16, 29]. In 2.5D design, multiple *chiplets* are placed on a silicon interposer, and the chiplets communicate using links integrated into the interposer. 2.5D integration technology provides multiple potential benefits compared to 2D systems, including greater system performance within thermal constraints [12], heterogeneous integration of multiple technologies [1, 6], and reduced overall system cost [16]. However, 2.5D integration technology also opens up a number of design challenges, ranging from circuit and physical challenges (design and routing of inter-chiplet links, placement and floorplanning of chiplets on the interposer, microbump assignment, etc.) to architectural and system-level challenges (design of the inter-chiplet network architecture, partitioning a system into 2.5D-integrated heterogeneous functional components, etc.). Many other technical and business challenges, including design for thermomechanical stress, test strategy, and supply chain structure, are identified by Radojcic [26].

In this paper, we perform a *cross-layer co-optimization* of 2.5D inter-chiplet network design and chiplet placement. Our co-optimization methodology focuses on network topologies, link circuit options, and microbump pitch- and interconnect RC-aware routing of links. It maximizes performance and/or minimizes cost at the system level, while satisfying system power and thermal constraints. The need for such a cross-layer methodology as ours can be easily seen by considering the following. If we adopt a top-down approach, an architecture analysis of network topologies tells us that high-radix, low-diameter networks should be used for inter-chiplet networks, as they provide the best overall system performance (in instructions per cycle). However, in the physical layer, realization of high-radix, low-diameter networks requires long wires, which can limit the network performance and, hence, the overall system performance. Using repeaters on long wires to improve performance would necessitate active (rather than passive) interposer technology. Since active interposers are 10× more expensive than passive interposers [25], the system cost equation changes and the top-down intuition is flawed. On the other hand, a bottom-up, cost-centric perspective prompts the use of a passive interposer, which can only support repeaterless links in the circuit layer, thus limiting link performance and maximum link length. This leads to the adoption of low-radix, high-diameter networks at the inter-chiplet level, which lowers overall system performance.

Our work fills a significant gap in the literature on inter-chiplet network design and floorplan/placement optimization of 2.5D systems. No prior work has simultaneously considered thermal behavior of chiplets, multiple potential network topologies, multiple inter-chiplet link options, and physical design constraints associated with routing these links. Thus, previous approaches can incorrectly evaluate cost, performance, power and thermal feasibility, as well as other important parameters of 2.5D system solutions. Consequently, there is a risk of identifying suboptimal inter-chiplet network and 2.5D system floorplan solutions, which can lead to inefficient architectural decisions. For example, in our recent work [12], we describe a methodology to place chiplets (connected by a mesh) that results in thermally-safe, high-performance, and low-cost 2.5D systems. However, in the logical layer, we only consider a *Unified-Mesh*[1] network topology. In the physical layer, there is no accounting of the area overhead associated with the microbumps

[1] We classify networks either as *Unified* when we have single-level logical topology or as *Global-Local* when we have two-level logical topology with *Global* as the inter-chiplet and *Local* as the intra-chiplet level logical topology. A *Unified* network logically treats all cores as if they are on the same die and we connect them as such, while in a *Global-Local* network we have a hierarchy of connections.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICCAD '18, November 5–8, 2018, San Diego, CA, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5950-4/18/11…$15.00
https://doi.org/10.1145/3240765.3240768

required for the inter-chiplet links. In the circuit layer, we consider only one type of link. As elaborated below, the present paper shows that a careful accounting of microbump overhead, along with consideration of multiple network topology options and link design options, leads to a solution that can achieve 16% higher performance at comparable cost, and/or 18% lower cost at comparable performance, with respect to our prior best solutions.

The main contributions of this paper are as follows:

- We develop a cross-layer co-optimization methodology that optimizes inter-chiplet network design jointly with chiplet placement across logical, physical, and circuit layers. Our methodology optimizes a given 2.5D system for performance, cost, and wirelength, while ensuring that it is thermally safe. The outcome of the co-optimization comprises placement of chiplets on the interposer, logical topology of the inter-chiplet network, and circuit design and routing of the links that form the network.
- Our co-optimization considers a rich solution space. (i) In the logical layer, we consider a variety of *Global*, *Local*, and *Unified* network topologies. (ii) In the physical layer, we incorporate well-calibrated microbump overhead models into our area and cost models. We further consider the finite density of microbumps per unit die area, and assess achievable physical wiring distances (hence, achievable link latencies). (iii) In the circuit layer, we explore repeaterless non-pipelined, repeaterless pipelined, repeatered pipelined, and repeatered non-pipelined types of inter-chiplet links. We further consider a *gas station* link design to enable pipelining in passive inter-chiplet links.
- Our heuristic-based cross-layer co-optimization has several novel elements. (i) For a given chiplet placement and network topology, we perform routing and microbump assignment using a flow-based mixed integer-linear program (MILP) to minimize the maximum link latency. (ii) We use workload- and network throughput-aware thermal simulation outputs from HotSpot [32] to assess the thermal feasibility of placement and network topology solutions. (iii) We apply simulated annealing to search over our high-dimensional system solution space.

2 RELATED WORK

Related work on design and optimization of networks in 2.5D systems can be categorized based on the design layer: logical, physical, and circuit. Unlike our present work, previous approaches are generally limited in scope to a single layer of design.

In the logical layer, Kannan et al. [16] have evaluated various logical topologies for 2.5D systems, but their work does not consider microbump area overheads, different inter-chiplet link options, or physical implementation of the 2.5D layout. Ahmed et al. [2] propose a hierarchical mesh network for inter-chiplet communication. Both Kannan et al. [16] and Ahmed et al. [2] assume a “minimally active” interposer, which could be unrealistic from a cost perspective (see Section 3). Akgun et al. [3] evaluate three specialized memory-to-core network topologies, yet the evaluation is limited to the logical layer with a static placement of chiplets, and implications of design choices on physical and circuit layers are not explored. None of these works take thermal effects into account or perform a physical design optimization of the 2.5D inter-chiplet network.

In the physical layer, Funke et al. [14] have proposed various algorithms that exhaustively search for optimal placement and routing solutions for up to six chiplets. The recent work of Osmonolovskyi et al. [24] handles up to 11-chiplet design complexity using pruning methods. Minz et al. [22] and Fang et al. [13] focus on routing of inter-chiplet links on an interposer. Liu et al. [20] aim to reduce the number of metal layers in the interposer. These works do not consider thermal effects while finding placement and routing solutions. In our prior work [12], we propose a thermally-aware chiplet placement solution. However, our prior work does not perform routing and only computes a placement solution.

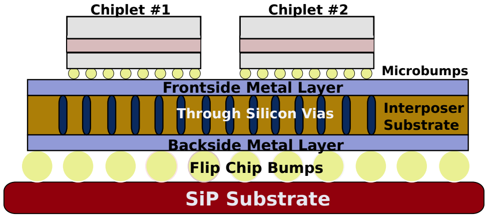


Figure 1: The cross-section of a 2.5D system.

In the circuit layer, research has generally focused on per-link analyses and optimizations, without considering overheads or trade-offs with respect to network or system throughput. The works of Stow et al. [28] and Karim et al. [17] explore both repeaterless and repeatered electrical links, while Shamim et al. [27] and Grani et al. [15] respectively consider wireless and photonic links. Ehrett et al. [11] analyze the power and delay overhead of microbumps and conclude that microbump overheads are small. However, they overlook electrostatic discharge (ESD) capacitance, which leads to underestimation network of power and latency.

In contrast to these previous works, our methodology jointly considers logical, physical, and circuit design of the inter-chiplet network. We evaluate a variety of logical topologies, while being aware of the network design feasibility in both the circuit layer and the physical layer. In the circuit layer, we evaluate various link design options. In the physical layer, we develop a thermally-aware placement and routing solution. Our cross-layer methodology, thus, obtains 2.5D system solutions that, having more complete and accurate modeling foundations, come closer to defining the true envelope of 2.5D system performance and cost under power and thermal constraints.

3 INTER-CHIPLET NETWORK DESIGN

A cross-layer inter-chiplet network design methodology must comprehend a vast design space that spans the logical, physical and circuit design layers. In this section, we describe the design space for each of these three layers, along with key parameters of interest.

3.1 2.5D System Architecture

Our studies use a 256-core homogeneous (i.e., all cores are of the same type) system. To enable comparisons against the previous literature, we specifically adopt the core design used in our prior work [12]. Cores have the following architectural specifications:

- 16KB I/D L1 Cache
- 256KB Private L2 Cache
- $0.93mm^2$ Core + L1 Area
- $0.35mm^2$ L2 Area
- $1.28mm^2$ ($1.13mm \times 1.13mm$) Total Area [33]
- $18mm \times 18mm$ Total 256-core Chip Area

Each core, together with its L1 and L2 caches, has a square layout. Following our prior work [12], we study chiplet-based integration of 16 identical chiplets on an interposer, where each chiplet contains 16 cores. Figure 1 illustrates the cross-section view of a 2.5D system. We assume that the $22nm$ chiplets are placed on an interposer that is designed in 65nm process technology. Microbumps connect the chiplets to the interposer substrate. The system is placed on a System-in-Package (SiP) substrate, with C4 ("flip chip bumps") connecting the interposer to the SiP substrate. We enable direct comparison of this work with our prior work [12] by designing a *Unified-Mesh* network using our cross-layer methodology. It should be noted that our broader conclusions are agnostic of the specific core count, core architecture, and technology nodes for the chiplets and the interposer.

3.2 Logical Layer

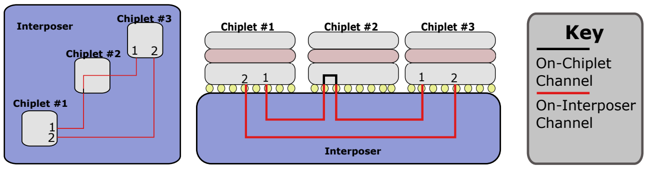
In the logical layer, we explore several different network topologies [35]. We limit the intra-chiplet network to *Local-Mesh* and *Local-Cmesh* topologies. For the inter-chiplet network, we design and evaluate *Global-Butterfly*, *Global-Butterdonut* [16], and *Global-Mesh* topologies. For the *Unified* networks, we evaluate *Unified-Mesh* and *Unified-Cmesh*.

3.3 Physical Layer

Physical design of the inter-chiplet network consists of placement of the chiplets, along with a routing solution connecting the chiplets[2] that is consistent with the chosen network topology (see Section 3.2). The placement of chiplets affects the temperature map and the length of the links among chiplets, while the routing solution in turn affects the microbump assignment and circuit choices for the link. Further, we explicitly account for the area overhead of microbumps and the associated inter-chiplet drivers and receivers placed along peripheral regions of the chiplets.

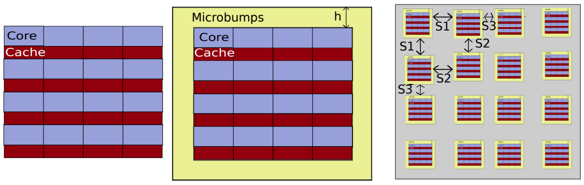
Inter-chiplet links can be routed on a passive or an active interposer. Microbumps and ESD protection are required at the beginning and the end of links that go through interposers, and this design constraint adds capacitance [17]. Passive interposers cost less due to their lower manufacturing cost and higher yield [25]. Active interposers allow for repeaters and/or flip-flops (for pipelining) on the interposer. This enables better link bandwidth and latency at the expense of higher manufacturing cost [25]. We conduct a preliminary study of the performance benefit of an active interposer. We observe 2× to 3× latency improvements for the same link length and 50% longer links for the same throughput, but this comes at a 10× cost overhead ($500 per wafer for passive interposer vs. $5000 per wafer for active interposer [25]). Given this cost overhead, we rule out active interposers as a realistic option in the near term, and do not consider this option in our present study.

Passive interposers limit the bandwidth of the signal by degrading rise/fall times. Hence, we use a *gas station* link, where we can "refuel" a passive link using repeaters and/or flip-flops that are inside other chiplets along the way from the source chiplet to the sink chiplet. Figure 2 shows two implementation schemes for a chiplet-to-chiplet link. Figure 2(a) shows the top view of the paths for the two links connecting Chiplet #1 to Chiplet #3, which are far (e.g., $> 10mm$) from each other. Figure 2(b) shows a cross-sectional view of the two paths between Chiplet #1 and Chiplet #3. Path 1 uses Chiplet #2 as a *gas station*, while Path 2 is a direct connection without any *gas station*. It is important to note the differences between an inter-chiplet repeaterless pipelined link and a *gas station* link. (i) Pipelining repeaterless links requires an active interposer, while for *gas station* links we can use a passive interposer. (ii) Active elements required for repeaterless pipelined links are designed using the active interposer's technology node, while active elements required for *gas station* links are designed using the chiplet's technology node. (iii) Using *gas station* links requires additional microbumps, and in turn, has an area overhead.



(a) Top View (b) Cross-Section View

Figure 2: Possible link implementation schemes including *Gas Station*, which is shown as Path 1.



(a) Chiplet Without Microbump Area (b) Chiplet With Microbump Area (c) Chiplet Arrangement

Figure 3: Illustration of the extra microbump area required per chiplet.

	Unified Mesh	*Unified Cmesh*	*Global Mesh*	*Global Butterfly*	*Global Butterdonut*	*Global Clos*
#microbumps	1024	512	256	256	256	2048
h (*mm*)	0.585	0.315	0.18	0.18	0.18	1.125
Chiplet Size (*mm*)	5.67	5.13	4.86	4.86	4.86	6.75
Microbump Area Overhead (%)	58.76	29.96	16.64	16.64	16.64	125.0

Table 1: Microbump area overhead for network topologies with shielding overhead included.

When considering 2.5D inter-chiplet links, recent works have overlooked the microbump overhead while assessing 2.5D integration benefits. Generally, the number of required microbumps will change according to the network topology. An increase in the number of inter-chiplet links increases the number of required microbumps. Further, additional microbumps (20% according to Radojcic [26]) must be reserved for power delivery and signal shielding purposes. Figure 3 shows the chiplet without and with the extra area required for microbumps. Table 1 presents the overhead due to microbumps for different network topologies designed using repeaterless non-pipelined links. The calculations are for the 256-core system divided across 16 chiplets, with each chiplet having an area of $4.5mm \times 4.5mm$, and a microbump pitch of $45um$. Here, h indicates the width of the extra space along the chiplet periphery required for the microbumps used for the inter-chiplet links [26]. The use of *gas station* link design will further increase microbump count. We do not list the microbump area overhead associated with use of *gas station* links since this depends upon the placement solution as well as the network type.

3.4 Circuit Layer

There are multiple circuit design options for inter-chiplet links. For passive interposers, the link on the interposer itself is repeaterless, but with the inclusion of *gas stations,* the link can use repeaters and/or flip-flops (for pipelining) in intermediate chiplets to regenerate and retime the signal. We limit t_{rise}/t_{cycle} to less than 0.5, to ensure full voltage swing at all nodes in the presence of non-idealities such as supply noise and jitter. We also explore t_{rise}/t_{cycle} of 0.8 that allows us 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.

[2] We aim to minimize the maximum physical link distance, which is our proxy for link latency.

Technology Node	$22nm$	$65nm$
Wire Thickness	$300nm$	$1.5\mu m$
Dielectric Height	$300nm$	$0.9\mu m$ [17]
Wire Width	$200nm$	$1\mu m$ [26]
C_{bump}	$4.5fF$	$4.5fF$ [17]
C_{esd}	$50fF$	$50fF$ [17]
C_{g_t} (Gate Cap)	$1.08fF/\mu m$	$1.05fF/\mu m$
C_{d_t} (Drain Cap)	1.5 × Cg	1.5 × Cg
R_t (Inverter resistance)	$450\Omega \cdot \mu m$	$170\Omega \cdot \mu m$
Wire Pitch	$0.4\mu m$	$2\mu m$ [26]
Flip-Flop Energy per Bit	$14fJ/bit$ [10]	$28fJ/bit$ [18]
Flip-Flop $t_{c-q} + t_{setup}$	$49ps$ [10]	$45ps$ [18]

Table 2: Technology node parameters.

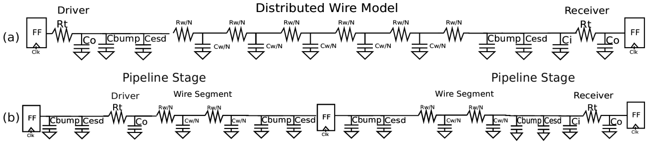


Figure 4: Distributed inter-chiplet link models: (a) repeaterless link and (b) *gas station* link, in a passive interposer.

Figure 4 shows distributed circuit models for link types; (a) repeaterless link in passive interposer, and (b) *gas station* link in passive interposer. We model wire parasitics using a distributed, multi-segment π model. We use $22nm$ technology parameters for intra-chiplet components (drivers, receivers, repeaters, and flip-flops of the links), while we use $65nm$ parameters for the inter-chiplet components of the links. Table 2 shows technology parameter values used in our experiments. We calculate capacitance and resistance based on the model in Wong et al. [30], and we calibrate our stage and path delay estimates based on extraction from layout and Synopsys PrimeTime timing reports.

4 CROSS-LAYER CO-OPTIMIZATION

In this section, we describe how we optimize the network design across the layers described in Section 3, using a cross-layer approach. We show our evaluation framework in Figure 5. We first construct *oracles* for system performance, cost, and interconnect performance. Each of these oracles gives us an element (performance, cost, and latency) of the co-optimization function. Our method for finding a placement solution of chiplets uses a simulated annealing algorithm. We build a search and sort engine that places the oracles and the placement algorithm in a loop to search for a solution across the logical, physical and circuit layers. Table 4 shows the notations we use in the various steps of our cross-layer co-optimization methodology. The placement algorithm uses HotSpot to determine the thermal profile and an MILP is used to find the optimal routing solution. Thus, we determine the feasibility of each placement using HotSpot simulations and the MILP solution.

4.1 System Performance Oracle

We build a system performance oracle that tells us the overall system performance and total core power for a given network topology, voltage-frequency setting, and link latency. To create the oracle, we use Sniper [7] to precompute system performance for a variety of network topologies, voltage-frequency settings, and link latencies. Our system architecture is the 256-core architecture described in Section 3.1. Eight memory controllers are placed next to the top and bottom rows of cores. We implement the inter-chiplet and intra-chiplet network models discussed in Section 3.2 using either passive links or *gas station* links (see Section 3.3). For passive links without *gas stations*, we vary inter-chiplet latency values from 1 to 5 cycles, and for *gas station* links we consider 2- or 3-stage pipelined links. We apply three voltage-frequency settings, ($0.9V$, $1000MHz$), ($0.89V$, $800MHz$) and ($0.71V$, $533MHz$). We fast-forward sequential initialization regions and simulate up to 10 billion instructions in the region of interest using Sniper, with all 256 cores active, to collect performance statistics for five benchmarks. This takes 1.7k CPU hours. We use McPAT [19] to convert the performance results to power traces needed for generating the thermal profile.

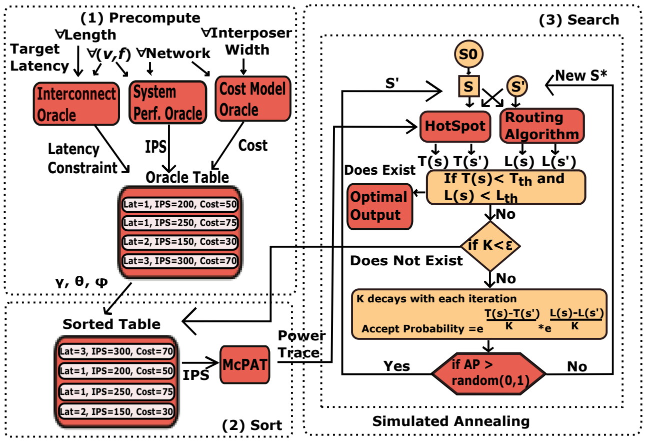


Figure 5: Cross-layer co-optimization flow.

4.2 Cost Oracle

We build a cost oracle that tells us the manufacturing cost of 2.5D systems for a given interposer size, network topology and *gas station* stage count. We adopt the 2.5D cost model proposed by Stow et al. [28], which takes the cost and yield of chiplets, interposer, and microbump bonding into account, assuming known good dies. We compute the cost of various interposer sizes from $20mm \times 20mm$ to $50mm \times 50mm$. To estimate the chiplet cost, we compute the number of microbumps required for different network topologies and *gas station* stages, determine the corresponding chiplet area overhead, and from all these we then calculate the manufacturing cost.[3]

4.3 Interconnect Performance Oracle

We construct an interconnect performance oracle that tells us the maximum length a signal can travel for a given voltage-frequency setting, rise time constraint, and number of cycles. The link models discussed in Section 3.4 are simulated in HSPICE [21]. For the wire dimensions in the $65nm$ interposer, i.e., $1\mu m$ wire width, $2\mu m$ wire pitch, and $1.5\mu m$ wire height, the wire resistance is $14.666 \times 10^{-3} \Omega/\mu m$ and the wire capacitance is $114.726 \times 10^{-3} fF/\mu m$. We use a maximum driver size of 100× the minimum size because the wire latency is largely wire dominated and increasing the driver size beyond 100× does not give latency improvements. We then use these values with our MILP placement solutions to check for placement feasibility. In Table 3, we provide the maximum link lengths we are capable of driving for different cycle numbers, voltage-frequency settings, and rise time constraints. We use power values from HSPICE, along with utilization values from the system performance oracle, to find the total power of the network.

[3] For details and justifications related to the comparison between the manufacturing cost of 2.5D systems and 2D systems, we refer the readers to our prior work [12].

$(v(V), f(MHz))$	$t_{rise}/t_{cycle} = 0.5$			$t_{rise}/t_{cycle} = 0.8$		
	(0.9, 1000)	(0.89, 800)	(0.71, 533)	(0.9, 1000)	(0.89, 800)	(0.71, 533)
1 Cycle	9	11	13	12	15	18
2 Cycles	16	18	23	19	25	30
3 Cycles	21	23	30	25	32	38
4 Cycles	25	27	35	30	38	45
5 Cycles	28	32	41	33	43	52
6+ Cycles	>32	>36	>45	>37	>48	>57

Table 3: Maximum link lengths (in *mm*) for a given network latency (in cycles), voltage-frequency setting, and rise time constraint.

4.4 Placement Optimization

We use simulated annealing to find a placement that meets the thermal constraint and the maximum link length constraint as evaluated by HotSpot and the routing MILP (with the maximum values provided by the interconnect performance oracle), respectively. We assume a symmetric layout similar to that used in our prior work [12]. As shown in Figure 3(c), we use {s1, s2, s3} as the spacings between chiplets. Simulated annealing searches the solution space in the manner shown in Figure 5. The placement optimization also estimates the microbump area overhead based on the routing solution, link type, and network choice.

4.4.1 Thermal Analysis.

We model the 2.5D system in HotSpot using the heterogeneous detailed 3D modeling features [34]. In our thermal model, we use the 2.5D system properties (layer thickness, materials, dimensions of bumps and TSVs, etc.) given in recent work [9, 23]. We use a method similar to our prior work [12], and model each layer of material with separate floorplans on a 64 × 64 grid with ambient temperature at 45°C with default HotSpot sizing convention of the heat sink and spreader. We model leakage as a linear model and assume it to be 30% of the total power at 60°C [33] and rerun HotSpot until temperature convergence is achieved.

4.4.2 Routing Optimization.

We build an MILP that takes the placement of chiplets and the logical network connections as input, and provides the optimal routing solution, including microbump assignment, as an output. The routing optimization is performed internally in the placement optimization as seen in Figure 5. The objective of the MILP is a weighted function of the maximum length of a route on the interposer and the total routing area overhead. We frame the delivery of required numbers of wires between chiplets as multi-commodity flow, and formulate an MILP to find optimal routing solutions that comprehend the finite availability of microbumps in regions of the chiplet periphery.

Table 4 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 the number of constraints in the MILP instance are both bounded by $O(|C|^2 \cdot |P|^2 \cdot |N|)$. 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 5), 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 (1). Equation (2) 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 $\alpha = 1$ and $\beta = 0$. Equation (3) imposes an upper bound on L, ensuring that the solution has routes satisfying the input maximum-length constraint D_{max}. Equation (4) ensures that the flow variable f^n_{ihjk} is a non-negative number. Equation (5) is the flow constraint governing the flow variables f^n_{ihjk}. It ensures 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 ensures that net flow is 0 for all other (non-source, non-sink) chiplets for the given net. Equation (6) ensures that there is no input flow (for net n) for the source chiplet of net n.

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_c	Left bottom x-coordinate for chiplet c.
Y_c	Left bottom y-coordinate for chiplet c.
x_p	x-offset from left bottom (within chiplet instance) for pin clump p.
y_p	y-offset from left bottom (within chiplet instance) for pin clump p.
d_{ihjk}	Distance from pin clump h on chiplet i to pin clump k on chiplet j.
λ^n_{ihjk}	Binary indicator for a route between pin clump h on chiplet i to pin clump k on chiplet j belonging to net n.
R_{ij}	Input requirement on the number of wires between chiplet i and chiplet j.
P^{max}_{ih}	Pin capacity for a pin clump h on chiplet i.
f^n_{ihjk}	Flow variable. Number of wires from pin clump h of chiplet i to pin clump k of chiplet j that belong to net n.
D_{max}	Maximum permissible length for any route.
S_{max}	Maximum permissible number of segments 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.
α, β	Coefficients for the objective function.
Gas Station	The MILP treats a *gas station* as a chiplet other than the source (s_n) or sink (t_n) that is used to route wires of net n.
NW	Set of logical networks: {*Unified-Mesh*, *Unified- Cmesh*, *Global-Mesh-Local-Mesh*, *Global-Mesh-Local-Cmesh*, *Global-Butterfly-Local-Mesh*, *Global-Butterfly-Local-Cmesh*, *Global-Butterdonut-Local-Mesh*, *Global- Butterdonut-Local-Cmesh*}.
(V, F)	Set of voltage-frequency settings: $\{(0.9V, 1000MHz), (0.89V, 800MHz), (0.71V, 533MHz)\}$.
l_{wire}	Wirelength $\in \{1 - 40mm\}$.
Nw	A network $\in NW$.
(v, f)	A voltage-frequency setting $\in (V, F)$.
w_{int}	An interposer width $\in \{20 - 50mm\}$.
w_{2D}	Width of the 2D chip: $18mm$.
w_g	Width of the guardband along the interposer periphery: $1mm$.
$s1$, $s2$, $s3$	Spacing between chiplets.
L	Maximum route length among all routes in the routing solution for a given $s1$, $s2$, $s3$, Nw.
L_{th}	Maximum route length threshold given a (v, f) and τ_{target}.
τ_{target}	Target link latency value.
T	Peak temperature in the system for a given $s1$, $s2$, $s3$, Nw and (v, f).
T_{th}	Peak temperature threshold set at $85°C$.
IPS	Instructions per second (IPS) for a given (v, f) and Nw.
IPS_0	Instructions per second (IPS) of *Global-Butterdonut-Local-Cmesh* topology baseline.
A_{bump}	Microbump area overhead for a given network and *gas station* stage count.
$Cost$	Manufacturing cost of 2.5D systems for a given w_{int}, Nw, and A_{bump}.
$Cost_0$	Cost of *Global-Butterdonut-Local-Cmesh* topology baseline.
τ	Latency.
τ_0	Latency of *Global-Butterdonut-Local-Cmesh* topology baseline.
γ, θ, ϕ	Coefficients for the cross-layer objective function.
K	Annealing factor.
ϵ	Annealing threshold.
AP	Acceptance probability.

Table 4: Notations used in the various steps of our cross-layer co-optimization methodology.

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^{max}_{ih} 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 length of a route, D_{max}. Maximum number of segments, S_{max} equal to 1, 2 or 3. $S_{max} \leq 3$ to limit impact on latency.

Table 5: Inputs to the routing optimization.

Similarly, Equation (7) ensures that there is no output flow from the sink chiplet of net *n*. Equation (8) ensures 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 ensures that all routes have available pins. Equation (9) defines λ^n_{ihjk} as a boolean value based on f^n_{ihjk}. This helps identify the maximum route length *L*, as shown in Equation (10). Equation (11) constrains the maximum number of segments (S_{max}) to be either 1, 2 or 3. If $S_{max} = 1$, no *gas stations* are permitted, while if $S_{max} = 2$ or $S_{max} = 3$, then *gas stations* are permitted, allowing for 1 or 2 *gas station* hops, respectively.

$$d_{ihjk} = \left|X_i + x_h - X_j - x_k\right| + \left|Y_i + y_h - Y_j - y_k\right| \tag{1}$$

We solve:

Minimize:

$$\alpha \cdot L + \beta \cdot \sum_{i \in C, h \in P, j \in C, k \in P, n \in N} d_{ihjk} \cdot f^n_{ihjk} \tag{2}$$

Subject to:

$$L \leq D_{max} \tag{3}$$

$$f^n_{ihjk} \geq 0,\ \forall i \in C, h \in P, j \in C, k \in P, n \in N \tag{4}$$

$$\sum_{h \in P, j \in C, k \in P} f^n_{ihjk} - \sum_{h \in P, j \in C, k \in P} f^n_{jkih} = \begin{cases} R_{s_n t_n}, & \text{if } i = s_n, \forall n \in N \\ -R_{s_n t_n}, & \text{if } i = t_n, \forall n \in N \\ 0 \text{ otherwise}, \forall n \in N \end{cases} \tag{5}$$

$$\sum_{h \in P, j \in C, k \in P} f^n_{jks_n h} = 0,\ \forall n \in N \tag{6}$$

$$\sum_{h \in P, j \in C, k \in P} f^n_{t_n hjk} = 0,\ \forall n \in N \tag{7}$$

$$\sum_{j \in C, k \in P, n \in N} f^n_{ihjk} + \sum_{j \in C, k \in P, n \in N} f^n_{jkih} \leq P^{max}_{ih},\ \forall i \in C, h \in P \tag{8}$$

$$\lambda^n_{ihjk} = \begin{cases} 1 \text{ if } f^n_{ihjk} > 0, \forall i \in C, h \in P, j \in C, k \in P, n \in N \\ 0 \text{ otherwise}, \forall i \in C, h \in P, j \in C, k \in P, n \in N \end{cases} \tag{9}$$

$$L \geq d_{ihjk} \cdot \lambda^n_{ihjk}, \forall i \in C, h \in P, j \in C, k \in P, n \in N \tag{10}$$

$$\sum_{i \in C, h \in P, j \in C, k \in P} f^n_{ihjk} \leq \begin{cases} R_{s_n t_n}, \text{ if } S_{max} = 1 \\ 2 \cdot R_{s_n t_n} - \sum_{h \in P, k \in P} f^n_{s_n h t_n k}, \text{ if } S_{max} = 2 \\ 3 \cdot R_{s_n t_n} - 2 \cdot \sum_{h \in P, k \in P} f^n_{s_n h t_n k} - \\ \sum_{i \in C | i \neq s_n || t_n} min(\sum_{h \in P, k \in P} f^n_{s_n hik}, \\ \sum_{h \in P, k \in P} f^n_{ikt_n h}), \text{ if } S_{max} = 3 \end{cases} \tag{11}$$

4.5 Cross-Layer Co-Optimization Flow

To design the inter-chiplet network in the 2.5D system, we formulate a cross-layer co-optimization problem to maximize performance while minimizing manufacturing cost and latency, as shown in Equations (12) - (17). Equation (12) is the objective function, where (γ, θ, and ϕ) are the weight factors of performance, cost, and latency of our target 2.5D system. We normalize the performance, cost, and latency to the baseline 2.5D system described in Kannan et al. [16], where *Global-Butterdonut-Local-Cmesh* network with a 4-stage pipelined link is used for communication, and the chiplets are separated with minimal spacing of 0.5*mm*. The objective function is subject to a peak temperature constraint of 85°C (Equation (13)), a maximum wirelength constraint for a given link type and target latency (Equation (14)), and a maximum interposer size constraint of 50*mm* × 50*mm* (Equation (15)). Equation (16) computes the interposer size based on spacing variables {s1, s2, s3} as defined in Figure 3(c), with a fixed guardband of 1*mm* along the periphery of the interposer. Equation (17) makes sure there is no overlap between the center chiplets. {s1, s2, s3} > 0 guarantees that there is no overlap between periphery chiplets.

Minimize:

$$\gamma \times \frac{IPS_0}{IPS((v, f), Nw)} + \theta \times \frac{Cost(w_{int}, A_{bump}, Nw)}{Cost_0} + \phi \times \frac{\tau}{\tau_0} \tag{12}$$

Subject to:

$$T((v, f), Nw, s1, s2, s3) \leq T_{th} \tag{13}$$

$$L(Nw, s1, s2, s3) \leq L_{th}((v, f), \tau_{target}) \tag{14}$$

$$w_{int} \leq 50 \tag{15}$$

$$w_{int} = w_{2D} + 2 \times s1 + s3 + 2 \times w_g \tag{16}$$

$$2 \times s1 + s3 - 2 \times s2 > 0 \tag{17}$$

Our flow to solve the cross-layer co-optimization problem is shown in Figure 5. The co-optimization flow has the following three steps:

Precompute. We use the system performance, cost, and interconnect performance oracles to precompute a table of all possible 8800 combinations of the system performance, cost and maximum interconnect length.

Sort. For a given set of co-optimization function coefficients (γ, θ, and ϕ) in Equation (12), we compute the objective function values for each entry in the table of 8800 combinations and sort the table entries from low to high objective function values. We normalize all three components (system performance, cost, and interconnect latency) to *Global-Butterdonut-Local-Mesh* [16].

Search. For each entry in the sorted table, we use simulated annealing to search for a valid chiplet placement, {s1, s2, s3} that meets both the temperature (Equation (13)) and wirelength (Equation (14)) constraints. The search space for each entry cannot be rapidly traversed using exhaustive search due to large simulation times in HotSpot. In our prior work [12] we had used greedy search to search for thermally valid solutions. Given the dual constrained nature of the problem in the current work, we choose simulated annealing over greedy search. For all interposer sizes and chiplet sizes, the total solution space has more than 17000 combinations of {s1, s2, s3}. We would like to note that between the 17000 combinations of {s1, s2, s3} and the 8800 combinations of the oracles, there is a many-to-many mapping. In other words, each of the 8800 combinations can have one or more combinations of {s1, s2, s3} that give the same minimum value for the objective function. The same one-to-many mapping exists in the reverse direction. We set an initial annealing factor *K* to 1, a stopping factor to 0.01, and a decay factor to 0.9. The annealing factor decays every *i* iterations, where *i* is set proportional to the interposer size. A neighbor placement (denoted as S') of current {s1, s2, s3} (denoted as *S*) is randomly generated by varying one of the {s1, s2, s3} by ±0.5*mm*. We evaluate the probability of accepting a neighbor placement by comparing peak temperature and maximum wirelength of the neighbor and the current placement using the function $e^{\frac{T(S)-T(S')}{K}} \times e^{\frac{L(S)-L(S')}{K}}$. We accept the neighbor placement if the probability is greater than a random number between 0 and 1. If the neighbor placement is a better solution with lower peak temperature and/or lower maximum wirelength, the probability function is greater than 1 to force the acceptance. If the neighbor placement is worse than the current placement, there is still a nonzero probability of accepting the neighbor placement to avoid being trapped in a local minimum. As the annealing factor *K* decays, the probability of accepting a worse neighbor goes down. During the search, if there is a placement that meets both peak temperature and maximum wirelength constraints, we stop the search and output this placement as our solution. If there is no valid placement after finishing simulated annealing, we move down to the next entry in the sorted table.

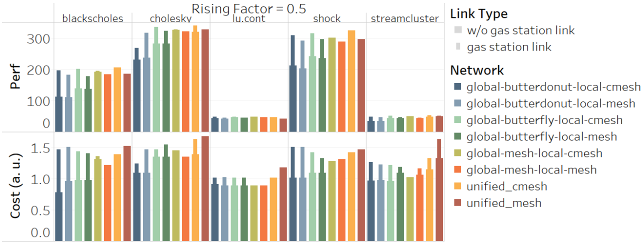


Figure 6: Maximum performance and corresponding cost for t_{rise}/t_{cycle} = 0.5.

With our simulated annealing parameters, the algorithm explores between 1000 to 2200 moves, depending on the design space for a given interposer size. Among the moves, 30% to 45% of the moves are accepted. There is almost no acceptance of a neighbor placement in the last few hundreds of moves, and thus, our simulated annealing algorithm converges.

5 EVALUATION RESULTS

In this section, we discuss the results of application of our proposed cross-layer co-optimization methodology. We run multi-threaded workloads from SPLASH-2 (*cholesky, lu.cont*) [31], PARSEC (*blackscholes, streamcluster*) [4], and UHPC (*shock*) [5] to get a variety of power and performance profiles. For each benchmark, we determine the chiplet placement solution, network routing solution, link type, voltage-frequency setting and network topology. In Figure 6, we show the maximum achievable performance and the corresponding cost of all networks across the five benchmarks for t_{rise}/t_{cycle} = 0.5. We show results with and without *gas stations*.

If we do not use *gas station* links, *Unified-Mesh* outperforms other networks when running *cholesky* and *streamcluster* by 1% to 39%. *Unified-Cmesh* outperforms all other networks for the remaining benchmarks by <1% to 85%. The higher performance of *Unified-Mesh/Cmesh* is because they have shorter inter-chiplet links and so they easily achieve single-cycle latency even without *gas stations*. The latency penalty of long links in *Global-Butterfly-Local-Mesh/Cmesh* and *Global-Butterdonut-Local-Mesh/Cmesh* leads to lower performance. On average *Unified-Cmesh* network has the best performance among all networks. It has more inter-chiplet channels compared to *Global* networks that results in less contention in the inter-chiplet links, and at the same time it has lower hop count than *Unified-Mesh* that results in lower latency. The higher performance of *Unified-Mesh/Cmesh* comes at a cost. *Unified-Mesh* network is the most expensive and has a manufacturing cost that is 6% to 90% higher than other networks.

With *gas stations*, we can pipeline longer links to improve network throughput. As a result, *Global-Butterfly-Local-Mesh/Cmesh* and *Global-Butterdonut-Local-Mesh/Cmesh* networks can achieve better performance with *gas stations*. Across all benchmarks we see *Unified-Cmesh* outperforms all other networks by <1% to 21%. However, *Unified-Mesh* has 1% to 60% higher manufacturing cost compared to all other networks for all benchmarks, except *shock*. For *shock*, *Global-Butterdonut-Local-Mesh/Cmesh* has the highest cost, which is 1% to 20% higher than all remaining networks.

To better understand the design space, we also evaluate maximum performance and corresponding cost for networks with and without *gas station* links when t_{rise}/t_{cycle} is 0.8. With this t_{rise}/t_{cycle}, longer inter-chiplet link lengths without *gas stations* are feasible. The relaxed length constraint also reduces the microbump and pipeline stage count, which reduces the cost. For t_{rise}/t_{cycle} of 0.8, without *gas stations*, *Unified-Cmesh* outperforms other networks by <1% to 47%. *Unified-Mesh* has the highest cost and it is 4% to 52% greater than that of other networks. With *gas stations*, the performance of *Unified-Cmesh* is <1% to 11% greater than other networks. *Unified-Mesh* has the highest cost for all benchmarks except *blackscholes* and *shock*, and it is 8% to 60% higher than the cost of the remaining networks. For *blackscholes*, *Global-Butterfly-Local-Cmesh* has the highest cost. Here the cost is 2% to 18% higher than the remaining networks. For *shock*, *Global-Butterdonut-Local-Cmesh* has the highest cost and it is 2% to 20% higher than the remaining networks.

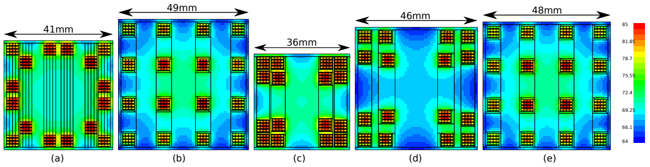


Figure 7: Floorplan examples for *cholesky* benchmark.

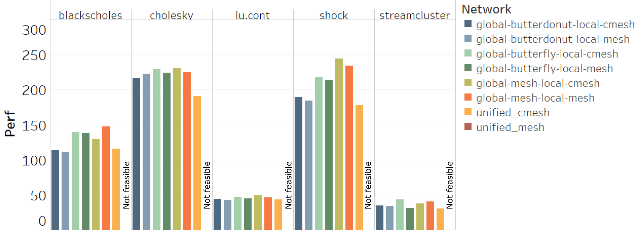


Figure 8: Network designs up to 35th cost percentile.

We now highlight differences between outcomes of our previous approach [12] and our present approach. Figure 7(a) shows the placement solution for the *cholesky* benchmark using our previous approach [12]. That work had predicted a performance boost of 80% with cost comparable to a 2D baseline, while optimizing performance. To make a fair comparison, we apply our cross-layer co-optimization algorithm, running the same benchmark and using the same *Unified-Mesh* network. Figure 7(b) shows the placement solution from our cross-layer co-optimization. Cost is almost 1.7× higher than that predicted previously [12], while achieving the same (80%) improvement over the 2D baseline system. Figure 7(c) shows the system organization when using our cross-layer co-optimization such that the cost does not exceed the cost of the optimal system organization in Figure 7(a) [12]. Here, we obtain substantially muted performance benefits: rather than 80% performance boost, we achieve a performance boost of 25% compared to the 2D baseline system. Figure 7(d) shows the solution when considering different network topology options while using the cross-layer co-optimization approach to minimize manufacturing cost at equal or higher performance than that of the solution in Figure 7(a). The cost of the solution shown in Figure 7(d) is 1.4× higher than that of the solution in Figure 7(a), but it is 20% lower compared to the solution in Figure 7(b). This 20% cost improvement is achieved due to the choice of *Global-Mesh-Local-Cmesh* in place of *Unified-Mesh*. Finally, in Figure 7(e), we show the solution using our cross-layer co-optimization methodology when using all possible design knobs. With *Unified-Cmesh*, (0.9*V*, 1000*MHz*) voltage-frequency setting, and 48*mm* interposer width, (i) we obtain 90% performance improvement compared to the 2D system, which is 60% better than the performance improvement determined by our prior work; and (ii) we obtain this performance improvement at 16% lower cost compared to our prior work.

Figure 8 provides insights regarding the maximum performance possible in a low-cost regime. We sort the 8800 table entries mentioned earlier by manufacturing cost from low to high. We then pick the first 35% of the table entries and identify the placement and routing solution for each network that gives highest performance. With low cost budgets, we see that the higher-performance configurations are dominated by *Global-Mesh-Local-Mesh/Cmesh* networks. *Global-Mesh-Local-Cmesh* performs the best in *cholesky*, *lu.cont*, and *shock*, with 1% to 42% better performance than other networks. *Global-Mesh-Local-Mesh* performs 7% to 50% better than other networks for *blackscholes*, while *Global-Butterfly-Local-Cmesh* gives between 1% to 29% better performance than other networks for *streamcluster*. This is expected, as mesh-like networks have shorter links and can achieve relatively high performance without having to utilize expensive *gas station* links. Further, in the low-cost regime, we see that *Unified-Mesh* is not feasible to implement due to the large number of links, which need a large number of microbumps and consequently have a high cost. Since our prior work [12] only considers *Unified-Mesh* topology, this result shows that it is not a viable solution for low-cost budgets. When we include solutions with up to the 65^{th} cost percentile, we see that *Global-Butterdonut-Local-Mesh/Cmesh* and *Global-Butterfly-Local-Mesh/Cmesh* topologies begin to catch up in performance with *Global-Mesh-Local-Mesh/Cmesh* networks. This is because we can utilize *gas station* links for the *Global-Butterdonut-Local-Mesh/Cmesh* and *Global-Butterfly-Local-Mesh/Cmesh* networks. *Global-Mesh-Local-Mesh/Cmesh* networks do not benefit as much from the relaxed cost constraint.

Finally, we discuss the power of the inter-chiplet network. We see that the highest inter-chiplet network utilization is seen when we run *shock* on a *Unified-Cmesh* network. While running *shock* on *Unified-Cmesh*, inter-chiplet network power is at most 2% of the overall system power.[4] Theoretically, in very highly threaded applications of the future, we could get much higher network utilizations and then the power of the inter-chiplet network would become a concern.

6 CONCLUSION AND FUTURE WORK

In this paper, we have introduced a cross-layer co-optimization methodology for inter-chiplet network design and chiplet placement in 2.5D systems. We have jointly considered network design in the logical, physical, and circuit layers to determine the optimal network choices, link choices, chiplet placements, and link routes to achieve a multi-objective co-optimization goal. We have also proposed to use a *gas station* link design to enable pipelined inter-chiplet links when using a passive cost-effective interposer. Our optimization has leveraged well-calibrated models of prior work. We have demonstrated that, compared to 2D systems, our optimized 2.5D systems can achieve 29% better performance with the same manufacturing cost, or 25% lower cost with the same performance.

Throughout this work, we have focused on running a single parallel application at a time and have shown the co-optimization outcomes for a variety of benchmarks. Based on these results, a 2.5D system can be further optimized in an application-aware manner (e.g., based on specific applications or worst-/average-case results). Interesting open problems include co-optimization with multi-application scenarios, allocation of threads in a network-aware manner, co-optimization with heterogeneous chiplets, and exploration of active interposer. Also, while we have designed our system for the worst-case link latencies under a global latency constraint, future work involves designing networks with variable link latencies.

ACKNOWLEDGMENT

This work was supported by NSF grants CCF-1149549, CCF-1564302, and CCF-1716352.

REFERENCES

[1] DARPA CHIPS. http://www.darpa.mil/news-events/2016-07-19

[2] M. M. Ahmed et al., “Increasing Interposer Utilization: A Scalable, Energy Efficient and High Bandwidth Multicore-multichip Integration Solution”, *Proc. IGSC*, 2017, pp. 1–6.

[3] I. Akgun et al., “Scalable Memory Fabric for Silicon Interposer-based Multi-core Systems”, *Proc. ICCD*, 2016, pp. 33–40.

[4] C. Bienia et al., “The PARSEC Benchmark Suite: Characterization and Architectural Implications”, *Proc. PACT*, 2008, pp. 72–81.

[5] D. Campbell et al., “Ubiquitous High Performance Computing: Challenge Problems Specification”, Georgia Tech. Res. Inst., Atlanta, GA, USA, Tech. Rep. HR0011-10-C-0145.

[6] J. -A. Carballo et al., “ITRS 2.0: Toward a Re-framing of the Semiconductor Technology Roadmap”, *Proc. ICCD*, 2014, pp. 139–146.

[7] T. E. Carlson et al., “Sniper: Exploring the Level of Abstraction for Scalable and Accurate Parallel Multi-core Simulation”, *Proc. SC*, 2011, pp. 1–12.

[8] J. Charbonnier et al., “High Density 3D Silicon Interposer Technology Development and Electrical Characterization for High End Applications”, *Proc. ESTC*, 2012, pp. 1–7.

[9] R. Chaware et al., “Assembly and Reliability Challenges in 3D Integration of 28nm FPGA Die on a Large High Density 65nm Passive Interposer”, *Proc. ECTC*, 2012, pp. 279–283.

[10] G. Chen et al., “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 JSSC* 50(1) (2015), pp. 59–67.

[11] P. Ehrett et al., “Analysis of Microbump Overheads for 2.5 D Disintegrated Design”, UMich. Ann Arbor Tech. Rep. CSE-TR-002-17.

[12] F. Eris et al., “Leveraging Thermally-Aware Chiplet Organization in 2.5D Systems to Reclaim Dark Silicon”, *Proc. DATE*, 2018.

[13] E. J. Fang et al., “IR to Routing Challenge and Solution for Interposer-based Design”, *Proc. ASP-DAC*, 2015, pp. 226–230.

[14] J. Funke et al., “An Exact Algorithm for Wirelength Optimal Placements in VLSI Design”, *Integration, the VLSI Journal* 52 (2016), pp. 355–366.

[15] P. Grani et al., “Photonic Interconnects for Interposer-based 2.5 D/3D Integrated Systems on a Chip”, *Proc. MEMSYS*, 2016, pp. 377–386.

[16] A. Kannan et al., “Enabling Interposer-based Disintegration of Multi-core Processors”, *Proc. MICRO*, 2015, pp. 546–558.

[17] M. A. Karim et al., “Power Comparison of 2D, 3D and 2.5 D Interconnect Solutions and Power Optimization of Iinterposer Interconnects”, *Proc. ECTC*, 2013, pp. 860–866.

[18] J. Knudsen, “Nangate 45nm Open Cell Library”, *CDNLive, EMEA* (2008).

[19] S. Li et al., “McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore architectures”, *Proc. MICRO*, 2009, pp. 469–480.

[20] W. Liu et al., “Metal Layer Planning for Silicon Interposers with Consideration of Routability and Manufacturing Cost”, *Proc. DATE*, 2014, p. 359.

[21] HSPICE User Guide, *Synopsys Inc.*, 2017.

[22] J. Minz and S. K. Lim, “Block-level 3-D Global Routing with an Application to 3-D Packaging”, *IEEE TCAD* 25(10) (2006), pp. 2248–2257.

[23] K. Murayama et al., “Warpage Control of Silicon Interposer for 2.5 D Package Application”, *Proc. ECTC*, 2013, pp. 879–884.

[24] S. Osmolovskyi et al., “Optimal Die Placement for Interposer-based 3D ICs”, *Proc. DAC*, 2018, pp. 513–520.

[25] G Parès, “3D Interposer for Silicon Photonics”, *LETI Innovations Days*, 2013.

[26] R. Radojcic, *More-than-Moore 2.5 D and 3D SiP Integration*, Springer, 2017.

[27] Md. S. Shamim et al., “A Wireless Interconnection Framework for Seamless Inter and Intra-chip Communication in Multichip Systems”, *IEEE Trans. Comput.* 66(3) (2017), pp. 389–402.

[28] D. Stow et al., “Cost-effective Design of Scalable High-performance Systems Using Active and Passive Interposers”, *Proc. ICCAD*, 2017, pp. 728–735.

[29] Xilinx Virtex 7, FPGA VC707 Evaluation Kit.

[30] S. Wong et al., “Modeling of Interconnect Capacitance, Delay, and Crosstalk in VLSI”, *IEEE Trans. Semiconductor Manufacturing* 13(1) (2000), pp. 108–111.

[31] S. C. Woo et al., “The SPLASH-2 Programs: Characterization and Methodological Considerations”, *ACM SIGARCH Computer Architecture News* 23 (1995), pp. 24–36.

[32] R. Zhang et al., “HotSpot 6.0: Validation, Acceleration and Extension”, University of Virginia, Tech. Rep. CS-2015-04.

[33] T. Zhang et al., “Thermal Management of Manycore Systems with Silicon-photonic Networks”, *Proc. DATE*, 2014, pp. 1–6.

[34] J. Meng et al., “Optimizing Energy Efficiency of 3-D Multicore Systems with Stacked Dram under Power and Thermal Constraints”, *Proc. DAC*, 2012, pp. 648-655.

[35] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*, Elsevier, 2004.

[4] If we include the power of the intra-chiplet networks (which have more links/routers), the contribution of the overall network to the total system power will be larger.