Cross-Layer Co-Optimization of Network Design and Chiplet Placement in 2.5-D Systems

Ayse Coskun, *Senior Member, IEEE*, Furkan Eris, *Student Member, IEEE*, Ajay Joshi, *Senior Member, IEEE*, Andrew B. Kahng, *Fellow, IEEE*, Yenai Ma, *Student Member, IEEE*, Aditya Narayan, *Student Member, IEEE*, and Vaishnav Srinivas, *Member, IEEE*

***Abstract*—2.5-D integration technology is gaining attention and popularity in manycore computing system design. 2.5-D systems integrate homogeneous or heterogeneous chiplets in a flexible and cost-effective way. The design choices of 2.5-D systems impact overall system performance, manufacturing cost, and thermal feasibility. This article proposes a cross-layer co-optimization methodology for 2.5-D systems. We jointly optimize the network topology and chiplet placement across logical, physical, and circuit layers to improve system performance, reduce manufacturing cost, and lower operating temperature, while ensuring thermal safety and routability. We also propose a novel *gas-station* link, which enables pipelined interchiplet links in passive interposers. Our cross-layer methodology achieves better performance-cost tradeoffs of 2.5-D systems and yields better solutions in optimizing interchiplet network and 2.5-D system designs than prior methods. Compared to single-chip systems, 2.5-D 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.**

***Index Terms*—2.5-D integration, cross-layer optimization, manycore systems, networks, place and route, thermal.**

I. Introduction

CMOS technology scaling has been slowing down over the past decade. It is getting increasingly difficult to continue technology scaling; hence, the industry has started to seek alternative solutions in the "more than Moore" direction. Instead of putting more transistors in a monolithic chip, one approach is to pack multiple dies in a package [2]–[4]. This approach enables flexible integration of homogeneous or heterogeneous dies, and speeds up the design and manufacturing of semiconductor systems. Therefore, die-stacking technologies like 2.5-D and 3-D integration have gained traction.

These multidie systems are cost-effective alternatives to single-chip systems (also called 2-D systems), as breaking down a chip into multiple chiplets alleviates the manufacturing yield drop suffered in a large 2-D chip. 3-D integration stacks chiplets vertically to increase memory bandwidth and reduce system footprint [5], but aggravates thermal challenges [6]. 2.5-D integration places multiple chiplets on a silicon interposer, which can be either passive or active. The chiplets communicate with each other through high-density fine-grained μbumps and interconnects in the interposer. Both 2.5-D and 3-D integration technologies enable designing high-bandwidth, low-latency networks, which could be utilized to handle the growing data traffic requirements of today's applications [2]–[4]. Compared to 2-D systems, 2.5-D systems have better thermally safe system performance [7], enable integration of heterogeneous technologies [4], and have lower cost [5]. Compared to 3-D systems, 2.5-D systems have better thermal dissipation capability, provide additional routing resources, and are more cost effective [5], [8].

Therefore, 2.5-D systems are gaining attention and popularity as competitive candidates to sustain the performance and cost scaling in computing systems [4], [5], [9]–[12]. There are already commercial 2.5-D products in the market, such as Xilinx Virtex 7 [12], AMD Fiji [13], Nvidia Tesla [14], and Intel Foveros [15]. These existing products typically place the chiplets adjacent to each other on an interposer to embrace the benefits of low communication latency due to short interchiplet links and low manufacturing cost resulting from small interposer sizes. However, the design and optimization of 2.5-D systems, including chiplet placement, interchiplet network architecture, design of interchiplet links, and μbump assignment, need to be thoroughly explored to maximize the benefits of 2.5-D integration [3].

In this article, we perform a cross-layer co-optimization of 2.5-D interchiplet network design and chiplet placement across logical, physical, and circuit layers. Our methodology jointly optimizes network topologies, link circuit and routing options, μbump assignment, and chiplet placement. Consider the following two cases that highlight the need for such a cross-layer approach.

1) If we adopt a top-down approach, an architecture-level analysis of network topologies indicates that high-radix, low-diameter networks provide the best overall system performance (in instructions per cycle) for interchiplet

Manuscript received June 27, 2019; revised October 29, 2019; accepted January 2, 2020. Date of publication January 28, 2020; date of current version November 20, 2020. This work was supported by the National Science Foundation under Grant CCF-1149549, Grant CCF-1564302, and Grant CCF-1716352. This article was recommended by Associate Editor C. Zhuo. *(Corresponding author: Yenai Ma.)*

Ayse Coskun, Furkan Eris, Ajay Joshi, Yenai Ma, and Aditya Narayan are with the Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215 USA (e-mail: yenai@bu.edu).

Andrew B. Kahng is with the Department of Computer Science and Engineering, University of California at San Diego, La Jolla, CA 92093 USA, and also with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: abk@ucsd.edu).

Vaishnav Srinivas is with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA.

Digital Object Identifier 10.1109/TCAD.2020.2970019

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 repeaters and/or need to be pipelined to achieve high performance, which necessitate active (rather than passive) interposer technology. Since active interposers are $10\times$ more expensive than passive interposers [16], 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. This leads to the adoption of low-radix, high-diameter interchiplet networks, which lowers overall system performance. Our cross-layer methodology comprehends the logical layer, physical layer, and circuit layer together, leading to a better system solution compared to using solely top-down or bottom-up approaches as in previous works.

Our cross-layer methodology fills a significant gap in the literature on 2.5-D system optimization by, including interchiplet network design and chiplet placement together. Cross-layer co-optimization allows for simultaneous consideration of thermal behavior of chiplets, multiple potential network topologies, and multiple interchiplet link options, including their circuit designs, physical design constraints, and routing costs. Previous works have explored limited tradeoffs among cost, power, thermal feasibility, and performance of 2.5-D systems due to the lack of such a cross-layer co-optimization methodology. For example, our prior work [7] describes a chiplet placement method that results in high-performance, low-cost, and thermally safe 2.5-D systems. However, that method lacks a true cross-layer co-optimization as it considers only a unified-mesh (U-M) network topology in the logical layer, determines the physical design of interchiplet links without accounting for the μbump overhead in the physical layer, and uses only a repeaterless link in the circuit layer. Our latest work [1] improves on our prior work [7] by jointly accounting for network topologies, μbump overhead, and interchiplet circuit designs across the three layers, but it covers a limited set of chiplet placement options.

As shown in the rest of this article, our proposed cross-layer co-optimization methodology achieves better performance-cost tradeoffs of 2.5-D systems. Our methodology explores a rich solution space. Specifically, in the logical layer, we consider a variety of network topologies, including Mesh, Concentrated-Mesh (Cmesh), Butterfly, Butterdonut [5], and Ring. In the physical layer, we search for the chiplet placement that minimizes operating temperature and meets the routing constraints. In the circuit layer, we explore interchiplet link designs. We co-optimize network topology, chiplet placement and routing, as well as interchiplet link design and provide a solution that achieves 88% iso-cost performance improvement and 29% iso-performance cost reduction compared to a single-chip design. Compared to our prior work [1], we achieve 40%–68% (49% on average) iso-cost performance improvement and 30%–38% (32% on average) iso-performance cost savings. The main contributions of this article are as follows.

1) We develop a cross-layer co-optimization methodology that jointly optimizes 2.5-D systems across logical, physical, and circuit layers. The outcome of our methodology includes network topology, chiplet placement, interchiplet link design, and routing.
2) Our methodology maximizes performance, minimizes manufacturing cost, and minimizes operating temperature. 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.
3) We develop a simulated annealing algorithm to search the high-dimensional placement solution space. Our placer supports arbitrary placements that consider nonmatrix and asymmetric chiplet organizations. We enhance a 2.5-D cost model [17] to incorporate a comprehensive μbump overhead analysis on chiplet area and yield. We use *gas-station* link design [1] to enable pipelining in a passive interposer.

II. Background

2.5-D 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 communication bandwidth between the chiplets and improve system performance [18]. Furthermore, 2.5-D integration is more cost effective than large 2-D chips and is more thermally efficient than 3-D systems [17]. Currently, 2.5-D integration technology is being widely explored by both academia [5], [9], [18], and industry [11]–[15].

Embedded multidie interconnect bridge (EMIB) [19] and interposer [12] are two commonly used carrier options for 2.5-D integration technology. EMIB is a novel integration method, which embeds small pieces of silicon interconnect bridges in the organic package substrate to connect the edges of adjacent chiplets for die-to-die communication. Silicon interposer technology uses a relatively large silicon interposer to house all chiplets. It is more mature and has been used in commercial products [12], [13]. Both EMIB and interposer can provide high density die-to-bridge and die-to-interposer connections, respectively, and correspondingly, high-density die-to-die connections [19]. EMIB-based approach requires less silicon area than silicon interposer-based approach and thus has lower silicon cost [19]. However, the number of die-to-die connections per layer of EMIB is limited by bridge interface length [20], and EMIB increases organic substrate manufacturing complexity [21]. Furthermore, EMIB can only hook up adjacent chiplets. When two chiplets that are far apart are logically connected, they cannot have direct links and need multihop communication using EMIB technology. Interposer-based integration provides more flexibility in chiplet placement, network design, and interconnect routing, and thus, has better thermal dissipation capability as it does not require chiplets to be placed close to each other. Therefore, we focus on interposer-based 2.5-D integration in this article.

A 2.5-D-integrated system consists of three main layers: 1) an organic substrate; 2) a silicon interposer; and 3) a chiplet layer. μbumps 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

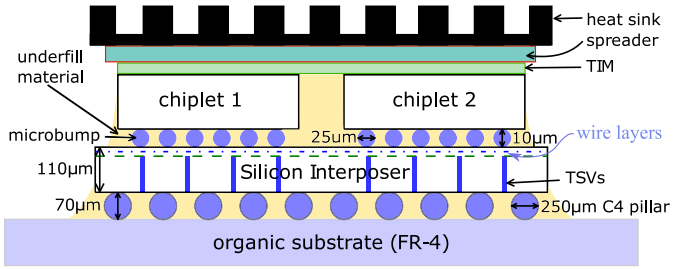

Fig. 1. Cross-section view of a 2.5-D system.

the organic substrate. Epoxy resin is often used to underfill the connection layers (C4 bumps layer and μbumps layer) and the empty spaces between chiplets. Fig. 1 shows the cross section view of a 2.5-D system in this article.

III. Related Work

2.5-D 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 [17]. Several related studies have explored the design and optimization of 2.5-D systems, with primary focus being placed on individual design layers: logical, physical, and circuit.

At the logical layer, Jerger *et al.* [18] presented 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.* [5] evaluated the impact of different network topologies on 2.5-D systems, and demonstrate that disintegration of a large 2-D chip into multiple chiplets improves manufacturing yield and lowers costs. However, their work overlooks the μbump overhead. Ahmed *et al.* [22] identified that interposer's routing resources are highly underutilized due to the high interconnect pitch in 2.5-D systems. To maximize performance, they propose a hierarchical mesh network for interchiplet communication. Akgun *et al.* [23] performed 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 2.5-D 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 2.5-D 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 2.5-D systems. Osmolovskyi *et al.* [24] optimized the chiplet placement to reduce the interconnect length using pruning techniques. Ravishankar *et al.* [25] determined the quality of different placement options in a 2-D grid using a stochastic model and implement a placer for 2.5-D FPGAs. Seemuth *et al.* [26] considered the increased design solution space in 2.5-D 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 2.5-D systems has been placed on minimizing IR drops and total wirelength in interchiplet links [27] and minimizing the number of metal layers [28]. None of these physical layer optimization solutions consider thermal effects.

Prior research at the circuit layer of 2.5-D systems generally focuses on link optimization techniques to improve the network and system throughput. Karim *et al.* [29] evaluated the power efficiency of electrical links with and without electrostatic discharge (ESD) capacitance. Stow *et al.* [17] evaluated 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 [30] and silicon-photonic links for communication in 2.5-D systems [31].

A common drawback among these previous works is that their design and optimization only focus on a single design layer. In contrast, we optimize the cost, performance, and temperature by jointly considering the logical, physical, and circuit layers of the interchiplet 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 interchiplet links. Our cross-layer methodology, thus, presents a rich solution space to evaluate a variety of network options at different design layers for 2.5-D systems, thus, enabling accurate and complete modeling of such systems.

IV. Cross-Layer Co-Optimization of Network Design and Chiplet Placement in 2.5-D Systems

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 comprehends 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 section, Section IV-A first introduces the cross-layer co-optimization problem formulation and the methodology we use to solve it. Fig. 2 shows our cross-layer methodology and provides an outline of upcoming sections. Section IV-B describes the optimization knobs in the design space across the logical, physical, and circuit layers. These knobs form the basis for modeling the 2.5-D network and chiplet placement, and enable cross-layer optimization. Section IV-C presents the tools and evaluation framework that models the 2.5-D 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 uses Sniper [32] and McPAT [33]; 2) cost oracle that computes the manufacturing cost of the 2.5-D system; 3) interconnect performance oracle that uses HSPICE [34] simulations to evaluate the interconnect circuit timing; 4) thermal analysis tool that uses HotSpot [35] to evaluate the temperature; and 5) routing optimizer that uses an

TABLE I
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: $18mm$.
w_{ubump}	μbump stretch-out width from original chiplets. Stretch-out width corresponds to the necessary increase of chiplet's dimensions to accommodate the μbumps 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.

MILP to solve for the optimal routing solution and the corresponding maximum wirelength. Section IV-D demonstrates the thermally aware place and route (PNR) tool that is based on simulated annealing and interactively uses the oracles described in Section IV-C to explore the chiplet placement solution space to minimize operating temperature and meet routing constraints.

A. 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 interchiplet 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 °C 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 I to formulate our optimization problem as follows:

$$\text{Minimize: } \alpha \times \left(\frac{1}{\text{IPS}}\right)_{\text{norm}} + \beta \times \text{Cost}_{\text{norm}} + \gamma \times T_{\text{norm}} + \eta \times g(T, T_{\text{th}}) \quad (1)$$

$$\text{Subject to: } g(T, T_{\text{th}}) = \frac{1}{10}(\max(T - T_{\text{th}}, 0))^2 \quad (2)$$

$$L \leq L_{\text{th}} \quad (3)$$

$$w_{\text{int}} \leq 50 \quad (4)$$

$$\max\big(|X_i - X_j|, |Y_i - Y_j|\big) \geq \frac{w_{2D}}{4} + 2 \times w_{\text{ubump}} + w_{\text{gap}} \ \forall i, j, i \neq j. \quad (5)$$

Equation (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 ($X_{\text{norm}} = [(X - X_{\min})/(X_{\max} - X_{\min})]$) 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 last term $g(T, T_{\text{th}})$ 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 (HTC). Equation (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 (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 Fig. 6). Equation (4) constrains the interposer size to be no larger than 50 mm × 50 mm, which is within the exposure field size of 2X JetStep Wafer Stepper and avoids extra stitching cost. Equation (5) ensures there is no overlap between chiplets.

To solve the optimization problem, we integrate simulation tools and analytic models discussed in Section IV-C. We first generate a complete table of all the combinations of network topologies, interchiplet link stage counts and latencies, voltage-frequency settings, and interposer sizes (see Section IV-B). We precompute system performance, power, allowable interchiplet 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 ($\alpha \times (1/\text{IPS})_{\text{norm}} + \beta \times \text{Cost}_{\text{norm}}$), and denote it as Obj2, where two 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 IV-D). For our design-time optimization, we assign the worst-case power, which is the highest core power among 256 cores of the 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 interchiplet wirelength of each placement. Each temperature simulation takes approximately 30 s and each routing optimization takes a few seconds to 10 min. 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 four indicates the number of terms. We keep track of the minimum

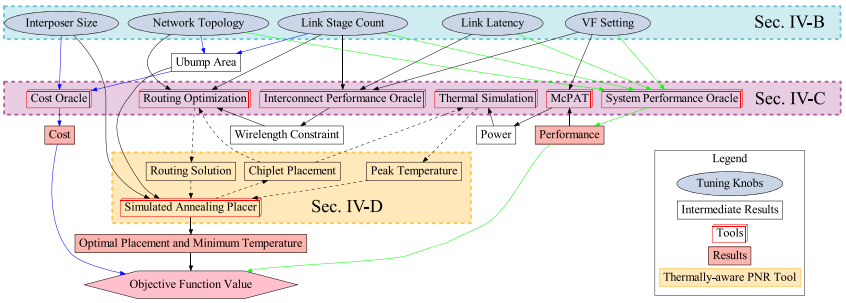

Fig. 2. Cross-layer co-optimization methodology.

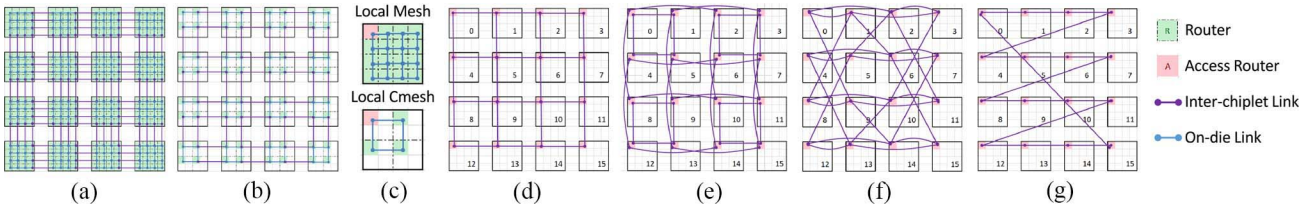

Fig. 3. Logical view of network topologies. (a) and (b) Unified networks and (c)–(g) are used to form hierarchical networks.

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 article, our experiments are based on the performance-focused case.

B. Cross-Layer Optimization Knobs

1) Logical Layer: One of the main questions in 2.5-D logical design is how to connect multiple chiplets using the interposer. In the logical layer, we explore two types of network topologies for 2.5-D systems. In Fig. 3, we show the logical views of network topologies. These views only illustrate the logical connections and not the actual chiplet placement. The first type is a unified network, which directly maps an NoC topology designed for a 2-D system onto a 2.5-D system to preserve the same logical connections and routing paths. We explore U-M, where each core has a router, and unified-Cmesh (U-CM), where four cores share a router, as shown in Fig. 3(a) and (b). Unlike single-chip NoCs, the source and the destination of a logical channel in 2.5-D systems may not reside on the same chiplet. The interchiplet 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 interchiplet 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 2.5-D systems, each chiplet has an on-chip local network and an access router. The global network hooks up all the access routers using interchiplet links embedded in the interposer. Intrachiplet packets travel through the local network, while interchiplet 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 (Fig. 3(c)); while for global networks, we explore Mesh (M), Butterfly (BF), Butterdonut (BD) [5], and Ring (R) topologies, (see Fig. 3(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.

2) Physical Layer: The physical design of 2.5-D 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 interchiplet link lengths. The routing solution affects the μbump assignment and circuit choice of interchiplet links. In our approach, we explicitly evaluate the area overhead of μbumps and the interchiplet link transceivers that are placed along the peripheral regions of the chiplets.

μbumps connect chiplets and the interposer. Interchiplet signals first exit the source chiplet through μbumps, travel along the wires in the interposer, and then pass through μbumps again to reach the destination chiplet. μbumps are typically placed along the periphery of the chiplet, for the purpose of signal escaping [36]. The μbump area overhead is determined by the number of interchiplet channels, channel bandwidth, and μbump pitch. We list the μbump area overhead for various network topologies in Table II, where we use a 128-bit wide bus for each channel, 45μm μbump

TABLE II
μBUMP COUNT, STRETCH-OUT WIDTH OF μBUMP REGION (w_{ubump}), AND μBUMP 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	**#μbumps**	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*	**#μbumps**	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*	**#μbumps**	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

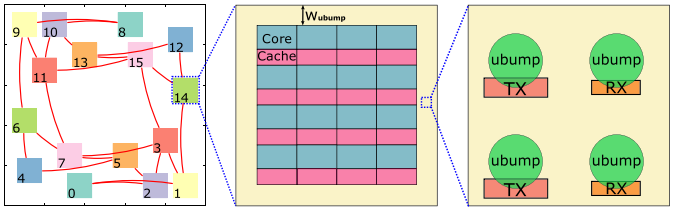

Fig. 4. Illustration of (a) chiplet placement on an interposer with logical connections, (b) chiplet with μbump overhead, and (c) μbumps with TX/RX regions (not drawn to scale).

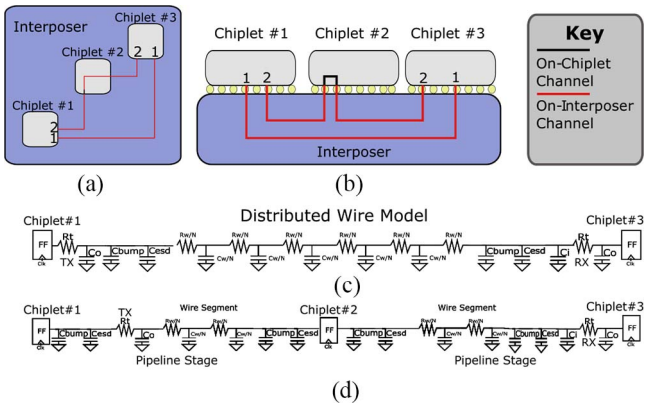

Fig. 5. Illustration of (a) top-down view and (b) cross section view of interchiplet link implementation, and distributed wire models for (c) repeaterless link [Path 1 in (a) and (b)] and (d) *gas-station* link [Path 2 in (a) and (b)].

pitch, and 4.5 mm $\times$ 4.5 mm chiplet size, and assume 20% additional μbumps are reserved for power delivery and signal shielding [36]. Here, w_{ubump} indicates the stretch-out width from the chiplet edge to accommodate the μbumps, as shown in Fig. 4. In Table II, we also include global Clos topology, which is a commonly used low-diameter-high-radix network. However, the area overhead is too high to make Clos a feasible interchiplet network option.

Interchiplet links can be routed on either a passive interposer or an active interposer. An active interposer enables better link bandwidth and latency because repeaters and flip-flops (for pipelining) can be inserted in the interposer [16]. However, an active interposer is expensive due to the front-end-of-line (FEOL) process and yield loss. A passive interposer is a cost-effective alternative. The passive interposer is transistor-free, can be fabricated with back-end-of-line (BEOL) process, and inherently has high yield [16]. We conducted a study of the performance benefit of an active interposer over a passive interposer. We observed 2$\times$ to 3$\times$ 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$\times$ cost overhead ($500 per wafer for passive interposer versus $5000 per wafer for active interposer [16]). Due to this cost overhead, we focus on the passive interposer in this article. Active interposers, however, are currently being considered for 2.5-D systems [5], [18]. Our methodology can be easily extended to active interposers, and we leave this as future work.

3) Circuit Layer: In the circuit layer, we explore multiple circuit designs for interchiplet 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 nonpipelined. Such a link has limited performance, especially, in 2.5-D systems, where interchiplet links could 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 interchiplet link (Path 1 in Fig. 5) latencies from single cycle to five cycles, which corresponds to a variety of interchiplet link lengths (see Fig. 6). This provides sufficient flexibility in chiplet placement. In addition, we use a novel "gas-station" link design [1], which enables pipelining in a passive interposer, to overcome the performance loss. Our gas-station link leverages flip-flops placed on other chiplets along the way to "refuel" a passive link. As shown in Fig. 5, Chiplet #2 is a *gas station* for Path 2 from Chiplet #1 to Chiplet #3, where signals first enter Chiplet #2 through μbumps, get repeated or retimed, and then return to the passive interposer through μbumps. Here, we tradeoff μbump area overhead computed in Table II for performance. It is important to note the differences between an interchiplet repeaterless pipelined link and a *gas-station* link [1]. 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 $t_{\text{rise}}/t_{\text{cycle}}$ upper bound to be 0.5 and ensure full voltage swing at all nodes in the interchiplet link to account for nonidealities such as supply noise and jitter. We also explore $t_{\text{rise}}/t_{\text{cycle}}$ of 0.8, which allows signals to go longer distances without repeaters. Relaxing the clock period or allowing for multicycle bit-periods permits us to use longer interchiplet links.

Fig. 5(c) and (d) shows the distributed circuit models in a passive interposer for repeaterless link and *gas-station* link, respectively. We model wire parasitics using a distributed,

TABLE III
TECHNOLOGY NODE PARAMETERS

Technology Node	22nm	65nm
Wire Thickness	$300nm$	$1.5\mu m$
Dielectric Height	$300nm$	$0.9\mu m$ [35]
Wire Width	$200nm$	$1\mu m$ [45]
C_{bump}	$4.5fF$	$4.5fF$ [35]
C_{esd}	$50fF$	$50fF$ [35]
C_{g_t} (Gate Cap)	$1.08fF/\mu m$	$1.05fF/\mu m$
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	$22nm \times 100$	$65nm \times 100$
Wire Pitch	$0.4\mu m$	$2\mu m$ [45]
Flip-Flop Energy per Bit	$14fJ/bit$ [47]	$28fJ/bit$ [48]
Flip-Flop $t_{c-q} + t_{setup}$	$49ps$ [47]	$70.9ps$ [49]

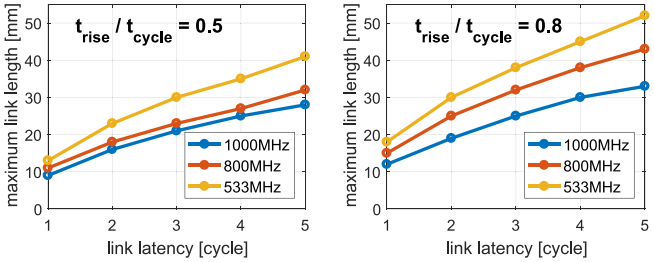

Fig. 6. Maximum reachable interchiplet link length with respect to clock cycles for various frequencies and rise-time constraints.

multisegment π model. We use 22-nm technology parameters for intrachiplet components (drivers, receivers, repeaters, and flip-flops) and 65 nm parameters for the interchiplet wires. Table III shows technology parameters used in our experiments. We calculate capacitance and resistance based on the model in Wong *et al.* [40], and we calibrate our stage and path delay estimates based on extraction from layout and Synopsys PrimeTime timing reports. Fig. 6 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 interchiplet 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 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).

C. Evaluation Framework

1) System Performance Oracle: We construct a manycore system performance oracle 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 [32] to precompute system performance. Our target system has 256 homogeneous cores, whose architecture is based on the IA-32 core from the Intel single-chip cloud computer (SCC) [41], with size and power scaled to 22-nm technology [42]. We divide the 256-core system into 16 identical chiplets.[1] In Sniper, we implement the unified and hierarchical network models described in Section IV-B1. For interchiplet links, we use either passive links or *gas-station* links (see Section IV-B2). We vary link latency from one to five cycles for passive links and explore 2-stage and 3-stage pipelines for *gas-station* links. We explore three

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

TABLE IV
NOTATIONS USED IN THE COST ORACLE

Notation	Meaning
A_{int}	Area of interposer.
$A_{chiplet}$	Chiplet area without μbump overhead.
A_{ubump}	Area of μbump region in a chiplet.
A_{TXRX}	Critical transceiver area in μbump region.
ϕ_{wafer}	Diameter of CMOS wafer: $300mm$.
$\phi_{wafer_{int}}$	Diameter of interposer wafer: $300mm$.
N_{int}	Number of interposer dies per wafer.
$N_{chiplet}$	Number of CMOS dies per wafer.
D_0	Defect density: $0.25/cm^2$ [9].
ϵ	Defect clustering parameter: 3 [9].
$Y_{chiplet}$	Yield of a CMOS chiplet.
Y_{int}	Yield of an interposer: 98% [54].
Y_{bond}	Chiplet bonding yield: 99% [9].
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.

voltage-frequency settings: (0.9 V, 1 GHz), (0.89 V, 800 MHz), and (0.71 V, 533 MHz). We use multithreaded benchmarks that cover the high-power applications (*Cholesky* from SPLASH-2 suite [43]), medium-power applications (*Streamcluster* and *Blackscholes* from PARSEC suite [16]), and low-power applications (*Lu.cont* from SPLASH-2 suite). We fast-forward the sequential initialization region and simulate ten billion instructions in the parallel region with all cores active to collect performance statistics. Then, we feed the performance results to McPAT [33] to compute the core power. We calibrate the McPAT power output with the measured power dissipation data of Intel SCC [41], scaled to 22 nm.

2) Cost Oracle: We construct a cost oracle that computes the manufacturing cost of 2.5-D systems for a given choice of network topology, chiplet size and count, link type and stage count, and interposer size. We adopt the 2.5-D manufacturing cost model published by Stow [17], which takes into account the cost and yield of CMOS chiplets, μbump bonding, and the interposer. The model assumes known-good-dies. We enhance the cost model to account for the impact of μbump overhead on the dies per wafer count and yield

$$A_{\text{chiplet}} = \left(\frac{w_{2D}}{4}\right)^2 \tag{6}$$

$$A_{\text{ubump}} = \left(\frac{w_{2D}}{4} + 2 \times w_{\text{ubump}}\right)^2 - A_{\text{chiplet}} \tag{7}$$

$$N_{\text{int}} = \frac{\pi \times \left(\phi_{\text{wafer}_{\text{int}}}/2\right)^2}{A_{\text{int}}} - \frac{\pi \times \phi_{\text{wafer}_{\text{int}}}}{\sqrt{2 \times A_{\text{int}}}} \tag{8}$$

$$N_{\text{chiplet}} = \frac{\pi \times (\phi_{\text{wafer}}/2)^2}{A_{\text{chiplet}} + A_{\text{ubump}}} - \frac{\pi \times \phi_{\text{wafer}}}{\sqrt{2 \times \left(A_{\text{chiplet}} + A_{\text{ubump}}\right)}} \tag{9}$$

$$Y_{\text{chiplet}} = \left(1 + \left(A_{\text{chiplet}} + A_{\text{TXRX}}\right) \times D_0/\epsilon\right)^{-\epsilon} \tag{10}$$

$$C_{\text{int}} = C_{\text{wafer}_{\text{int}}}/N_{\text{int}}/Y_{\text{int}} \tag{11}$$

$$C_{\text{chiplet}} = C_{\text{wafer}}/N_{\text{chiplet}}/Y_{\text{chiplet}} \tag{12}$$

$$C_{2.5D} = \frac{C_{\text{int}} + \Sigma_1^{16}\left(C_{\text{chiplet}} + C_{\text{bond}}\right)}{Y_{\text{bond}}^{15}}. \tag{13}$$

Equation (6) (see Table IV for all notations) computes the equivalent functional area of chiplets generated by dividing a 2-D chip. Equation (7) evaluates the μbump area overhead. Equations (8) and (9) determine the number of interposer

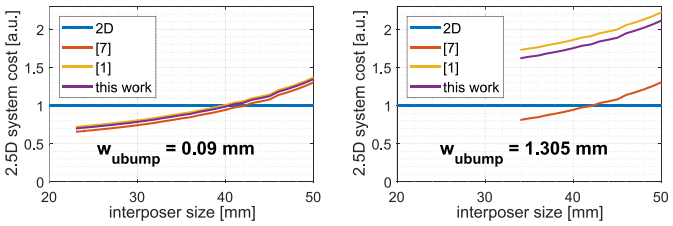

Fig. 7. Comparison between the cost of a 2-D system, and the cost of a 2.5-D system estimated using prior cost models [1], [7] and our enhanced cost model for interposer sizes from 20 mm to 50 mm and μbump stretch-out widths (w_{ubump}) of 0.09 mm and 1.305 mm, which correspond to the lower and upper limits of w_{ubump} in our analysis, respectively.

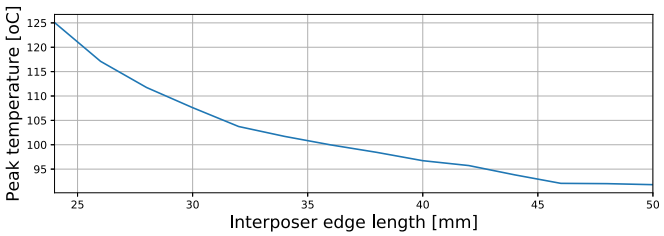

Fig. 8. Temperature of best chiplet placement for each interposer size, running *Cholesky* with *Mesh* network using single-cycle link without *gas stations*.

dies and the number of CMOS dies, respectively, that can be cut from a wafer [17]. 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 (9), we take into account the μbump area overhead A_{ubump}. Equation (10) is the negative binomial yield model, where D_0 is the defect density and $\epsilon = 3$ indicates moderate defect clustering [17]. Unlike the center area of chiplets that has high transistor density, the μbump regions have very limited active regions that contain interchiplet 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 noncritical. Hence, our yield calculation (Equation (10)) uses only the critical active area. The yield of a passive interposer is as high as 98% [44] because it does not have any active components. Equations (11) and (12) calculate the per-die cost of the interposer and the chiplets, respectively. Equation (13) estimates the overall manufacturing cost of the 2.5-D system by adding up the costs of the chiplets, the interposer, and bonding.

Fig. 7 shows the manufacturing cost of 2.5-D systems with respect to interposer sizes from 20 mm to 50 mm for two different μbump 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 2.5-D system costs are normalized to the cost of the 2-D system. The 2.5-D system cost increases with the interposer size. The cost model in our prior work [7] did not consider μbump overhead and thus, the 2.5-D system cost is independent of w_{ubump}. The cost model in our latest work [1] overestimated the yield drop due to μbump regions and thus, overestimated the overall cost. This error of this cost model [1] is trivial with a small w_{ubump}, but with a large w_{ubump}, the error is not negligible (up to 10% of the 2-D system cost in our example). With a small w_{ubump}, the predicted cost of a 2.5-D system using our enhanced model is cheaper than the cost of a 2-D system, when the interposer is smaller than 40 mm $\times$ 40 mm. With a large w_{ubump}, the predicted cost of a 2.5-D system using our enhanced model is always higher than that of a 2-D system. This eliminates some network topologies, such as Clos, that require large w_{ubump}.

3) Interconnect Performance Oracle: We build an interconnect performance oracle that analyzes the maximum reachable length of interchiplet link for a given operating voltage and frequency, rise-time constraint, and propagation time constraint in the unit of cycles. We use HSPICE [34] to simulate the link models discussed in Section IV-B3. 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 [38], and scale it down to 22-nm technology. The area of TX and RX logic (A_{TXRX}) takes up less than 1% of the μbump area. The interposer wire resistance is 14.666×10^{-3} Ω/μm and the capacitance is 114.726×10^{-3} fF/μm, for the wire dimensions provided in Table III for 65-nm technology. Since the interchiplet 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 interchiplet link power is up to 22 W, which is insignificant compared to the total average system power of 508 W. Hence, interchiplet link power has negligible influence on chiplet placement.[2]

4) Thermal Simulation: We use HotSpot [35] to simulate thermal profiles for given chiplet placement choices and core power values. We use an extension of HotSpot [45] that provides detailed heterogeneous 3-D modeling features. To model our 2.5-D 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 2.5-D system model follows the properties (such as layer thickness, materials, dimensions of bumps, and TSVs) of real systems [10], [11]. We use the HotSpot default conventions for the thermal interface material properties, the ambient temperature of 45 °C, 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.

We implement a linear model of temperature-dependent leakage power based on published data of Intel 22-nm processors [46]. We assume 30% of power is due to leakage at 60 °C [42]. 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.

Fig. 8 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*

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

TABLE V
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.

TABLE VI
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.

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 [47]. The temperature benefit shown in Fig. 8 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.

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 μbump 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 μbumps along the chiplet periphery into pin clumps to limit the problem size and the MILP runtime. We use four pin clumps per chiplet in our experiments. We frame the delivery of required number of wires between chiplets as multicommodity flow, and formulate the MILP to find optimal routing solutions that comprehend the finite availability of μbumps in each pin clump.

Table V 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_{ihjk}^{n}, which describe the routing solution and μbump assignment to pin clumps.

Based on the inputs to the routing optimization step (see Table VI), 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 (14). Equation (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 (16) ensures that the flow variable f_{ihjk}^{n} is a non-negative number. Equation (17) is the flow constraint governing the flow variables f_{ihjk}^{n}. 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, nonsink) 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 (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, (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 (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 (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 (22). Equation (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

$$d_{ihjk} = \left|X_i + x_h - X_j - x_k\right| + \left|Y_i + y_h - Y_j - y_k\right| \tag{14}$$

$$\text{Minimize:} \quad \theta \cdot L + \varphi \cdot \sum_{i\in C, h\in P, j\in C, k\in P, n\in N} d_{ihjk} \cdot f_{ihjk}^{n} \tag{15}$$

$$\text{Subject to:} \quad f_{ihjk}^{n} \geq 0 \quad \forall i \in C, h \in P, j \in C, k \in P, n \in N \tag{16}$$

$$\sum_{h\in P, j\in C, k\in P} f_{ihjk}^{n} - \sum_{h\in P, j\in C, k\in P} \times f_{jkih}^{n} = \begin{cases} R_{s_n t_n}, & \text{if } i = s_n \ \forall n \in N \\ -R_{s_n t_n}, & \text{if } i = t_n \ \forall n \in N \\ 0 & \forall i \neq s_n || t_n \ \forall n \in N \end{cases} \tag{17}$$

$$f_{jks_nh}^{n} = 0 \qquad \forall n \in N \ \forall h \in P \ \forall j \in C \ \forall k \in P \tag{18}$$

$$f_{t_nhjk}^{n} = 0 \qquad \forall n \in N \ \forall h \in P \ \forall j \in C \ \forall k \in P \tag{19}$$

$$\sum_{j\in C, k\in P, n\in N} f_{ihjk}^{n} + \sum_{j\in C, k\in P, n\in N} f_{jkih}^{n} \leq P_{ih}^{max} \quad \forall i \in C, h \in P \tag{20}$$

$$\lambda_{ihjk}^{n} = \begin{cases} 1 & \text{if } f_{ihjk}^{n} > 0 \ \forall i \in C, h \in P, j \in C, k \in P, n \in N \\ 0 & \text{otherwise } \forall i \in C, h \in P, j \in C, k \in P, n \in N \end{cases} \tag{21}$$

$$L \geq d_{ihjk} \cdot \lambda_{ihjk}^{n} \ \forall i \in C, h \in P, j \in C, k \in P, n \in N \tag{22}$$

$$\sum_{i\in C,h\in P,j\in C,k\in P} f^n_{ihjk} \leq \begin{cases} R_{s_nt_n}, \text{ if } S_{\max}=1 \\ 2\cdot R_{s_nt_n}-\sum_{h\in P,k\in P} f^n_{s_nht_nk}, \text{ if } S_{\max}=2 \\ 3\cdot R_{s_nt_n}-2\cdot\sum_{h\in P,k\in P} f^n_{s_nht_nk}- \\ \sum_{i\in C|i\neq s_n||t_n} \min\Big(\sum_{h\in P,k\in P} f^n_{s_nhik} \\ \sum_{h\in P,k\in P} f^n_{ikt_nh}\Big) \text{if } S_{\max}=3. \end{cases} \quad (23)$$

D. Thermally Aware Placement Algorithm

Our thermally aware PNR tool supports arbitrary chiplet placements that consider nonmatrix 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}) placement options with 1 mm granularity. It is impractical to exhaustively search such a vast space. In addition, the solution space is nonconvex. Approaches like gradient descent or greedy search [7] 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.

Placement Description: Prior works [1], [7] 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 fourfold rotational symmetry does not allow us to ever find the optimal placement for some topologies. For Butterdonut and Butterfly networks, because of the fourfold 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 twofold 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 [11].

Neighbor Placement: A neighbor placement is the placement obtained by either moving a chiplet by the minimum step size in any of the eight 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. For instance, a chiplet cannot move in some directions because other chiplets block the way, especially, when the interposer size is small.

Acceptance Probability: The decision of whether a neighbor placement is accepted or not depends on the delta calculated using (24). Here, T_{curr}, L_{curr}, T_{nei}, and 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 (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

$$\text{delta} = \begin{cases} 0.9\times(T_{\text{curr}}-T_{\text{nei}})+0.1\times(L_{\text{curr}}-L_{\text{nei}}) \\ \quad \text{if } L_{\text{curr}}\leq L_{\text{th}} \text{ and } L_{\text{nei}}\leq L_{\text{th}} \\ 0.1\times(T_{\text{curr}}-T_{\text{nei}})+0.9\times(L_{\text{curr}}-L_{\text{nei}}) \\ \quad \text{if } L_{\text{curr}}> L_{\text{th}} \text{ or } L_{\text{nei}}> L_{\text{th}} \end{cases} \quad (24)$$

$$\text{AP} = e^{\frac{\text{delta}}{K}}, \text{ accept if AP} > \text{rand}(0,1). \quad (25)$$

Multistart and Multiphase 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 nonconvex 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 multistart and multiphase techniques. For multistart, 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 multistart technique is helpful in reducing the chance of getting a poor solution. We can run the multiple starts of the multistart technique in parallel, so as not to increase the time required to arrive at the solution. For multiphase, 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 multiphase 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 multiphase step size to 4 mm, which provides a good balance between the simulation time and the solution quality.

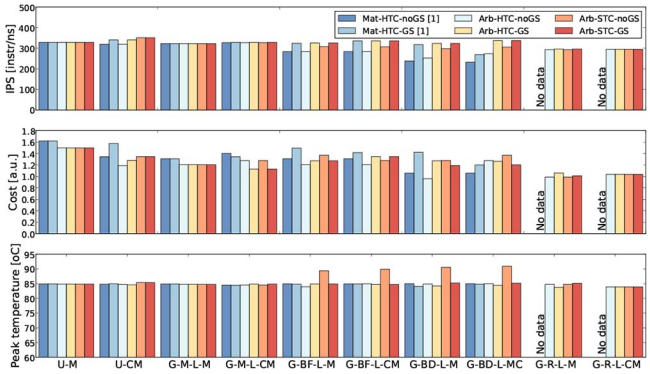

Fig. 9. Maximum performance, the corresponding cost, and the corresponding peak temperature for various networks with and without *gas-station* links when running *Cholesky* benchmark. Here, the optimization goal is to maximize performance; the cost values are normalized to the cost of a 2-D system.

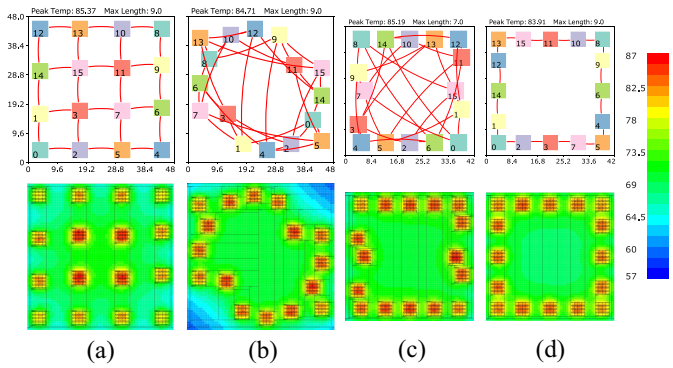

Fig. 10. Optimal chiplet placement for maximum performance and corresponding thermal maps when running the *Cholesky* benchmark in 2.5-D systems with different network topologies. (a) U-CM noGS. (b) G-BF-L-CM 2-stage GC. (c) G-BD-L-CM 3-stage GC. (d) G-R-L-CM noGS.

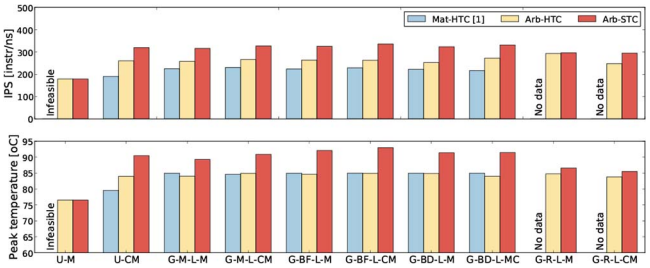

Fig. 11. Iso-cost performance and the corresponding peak temperature when running *Cholesky* benchmark for various networks, while not exceeding the cost budget of a 2-D system.

V. 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 [1], 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. Finally, 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.

A. Optimal Chiplet Placement Analyses

Fig. 9 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 [1] that only considers matrix-style chiplet placement (*Mat*) and an HTC of 85 °C, 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 °C 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 °C and arbitrary placement, as described in Section IV-D. We 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 [1]. 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 °C. 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 Fig. 10(a).

For Butterfly networks, the Arb-STC-GS approach achieves the same maximum performance as achieved using Mat-HTC-GS approach [1] and reduces the cost by 5% (see Fig. 9). The optimal placement for the Butterfly network is shown in Fig. 10(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 Fig. 9). Fig. 10(c) shows the optimal placement for Butterdonut network. The Ring networks (*G-R-L-M/CM*) are not included in the prior work [1], 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 Fig. 10(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 Fig. 9).

B. Iso-Cost and Iso-Performance Analyses

Fig. 11 shows the iso-cost performance for various networks running *Cholesky* benchmark, while not exceeding the cost of

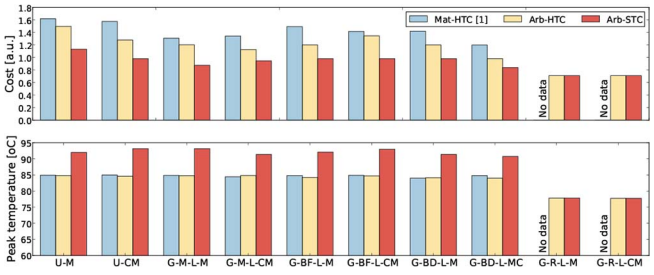

Fig. 12. Iso-performance cost and the corresponding peak temperature for each network. Here, the performance is equal to the maximum performance achieved using Mat-HC-GS [1] when running *Cholesky* benchmark. The cost values are normalized to the cost of a 2-D system.

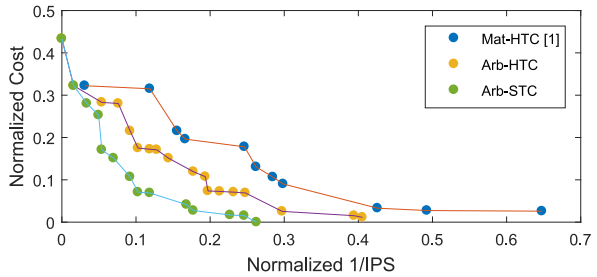

Fig. 13. Pareto frontier curve of normalized performance (1/IPS) and normalized cost using Mat-HTC approach [1], Arb-HTC approach, and Arb-STC approach.

a 2-D 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 [1]. The previous work [1] shows that the *U-M* network cannot be implemented feasibly due to the large μbump 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.

Fig. 12 shows the iso-performance cost and the corresponding peak temperature for each network. Here, for each network, we match the performance of the 2.5-D system designed using our proposed approach with the corresponding maximum performance of the 2.5-D system designed using prior Mat-HTC approach [1] when running *Cholesky* benchmark. The cost values are normalized to the cost of a 2-D system. Under the same HTC as the prior work [1], 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 °C overall system peak temperature.

Fig. 13 shows the Pareto frontier curve of normalized performance (1/IPS) and normalized cost using Mat-HTC approach [1], Arb-HTC approach, and Arb-STC approach. Our arbitrary placement pushes the Pareto frontier curve toward higher performance and lower cost, and the STC approach pushes the frontier further.

C. Analyses of Different Types of Applications

Fig. 14 shows the thermal maps of 2.5-D systems designed for high-power (*Cholesky*), medium-power (*Streamcluster*), and low-power (*Lu.cont*) applications using Mat-HTC [1], Arb-HTC and Arb-STC approaches. For comparison, we

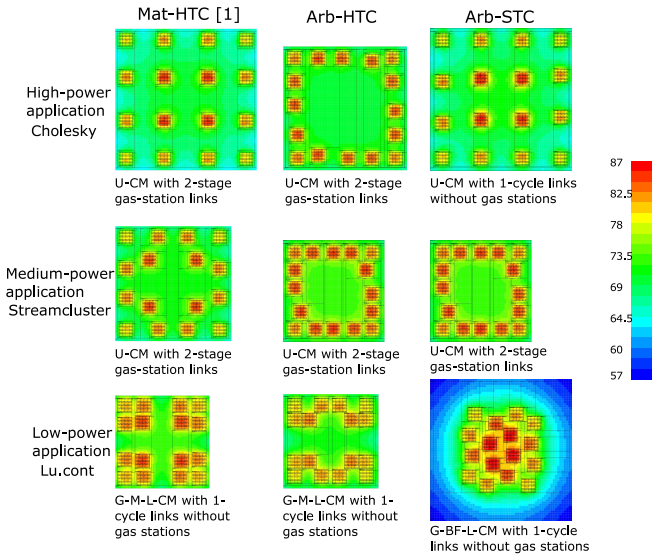

Fig. 14. Thermal maps of 2.5-D systems designed for high-power, medium-power, and low-power applications using Mat-HTC [1], Arb-HTC, Arb-STC approaches. The figures are scaled to the interposer sizes.

choose the same optimization objective as in the prior work [1], 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 [1] and reduce the manufacturing cost by 19%, 14%, and 3% for high-power, medium-power, and low-power applications, respectively. The 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 the high-power application, we can achieve the maximum possible performance (3% higher than both Mat-HTC approach [1] and Arb-HTC approach) and 15% lower cost. The improvement is achieved by violating the temperature threshold by 0.5 °C and using single-cycle interchiplet 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 °C. This improvement also comes with 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 Fig. 14 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 applications, 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

TABLE VII
CROSS-LAYER VERSUS SINGLE-LAYER OPTIMIZATION

Cases	Perf Improvement	Cost Increase	Temperature [^oC]	Perf/Unit Cost
Cross-layer	0%	0%	86	3.10
OOW	4%	-8%	99.9	3.50
OWO	0%	-22%	108.0	3.97
WOO	-20%	56%	84.2	1.59
FFB	-39%	-34%	100.9	2.88
FBF	4%	11%	102.5	2.92
BFF	-16%	-36%	103.4	4.09
FBB	-9%	-4%	85.8	2.94
BFB	-35%	-34%	100	3.09
BBF	2%	3%	86.2	3.06

size, chiplet placement, and interchiplet 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 the high-power application, we observe the same performance, 23% higher cost, and 6 °C lower temperature compared to that of a system custom designed for medium-power application. When running low-power application on the system designed for the high-power application, we observe 5% lower performance, 5% higher cost, and 12 °C lower temperature compared to that of a system custom designed for low-power application.

D. Analyses of Cross-Layer Co-Optimization Benefits

To understand the benefits of co-optimizing across multiple design layers simultaneously, we conduct a comparison between cross-layer and single-layer methodologies while running the *Blackscholes* benchmark. We compare multiple cases in Table VII. The baseline is the optimal solution of our cross-layer co-optimization methodology. We use three letters to represent the choices at each of the logical, physical, and circuit layers, for the remaining nine cases. Here, *O* means optimal, *W* means worst possible, *F* means prefixed, and *B* means best possible. So for example, the OOW case corresponds to the 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 shows the contribution of the circuit layer in our cross-layer co-optimization methodology. In the FFB case, we fix the design choices at the logical and physical layers, and only optimize the circuit layer. 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 the OOW and OWO cases, we observed a cost reduction and/or slight performance improvement, but at a high infeasible peak temperature. For the case of WOO, the temperature is acceptable but we get 20% lower performance and 50% higher cost. 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.

E. Sensitivity Analysis

We conduct a sensitivity analysis (see Fig. 15) to show the optimal combinations of performance, cost, and peak temperature, and the corresponding objective function values

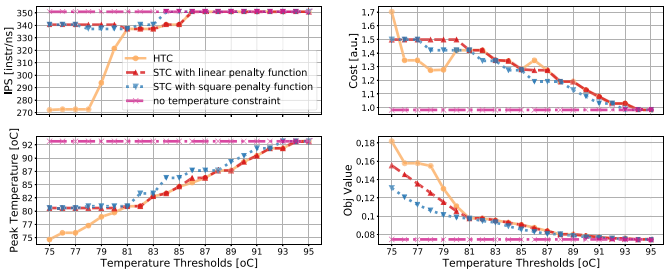

Fig. 15. Sensitivity analysis comparing HTC, STCs with linear function and square function, and no temperature constraint of various temperature thresholds from 75-95°C.

with respect to different temperature thresholds from 75 °C to 95 °C and different temperature constraint choices (including HTC, STC 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 °C. Thus, with a temperature threshold of 94 °C or higher, the optimal performance, cost, and temperature combinations with different constraint choices are the same. With an HTC, 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 the objective function value decreases. An STC allows violating the temperature threshold and translates the violation into a penalty in the objective function. The STC approach provides more choices and thus is guaranteed to have a solution that better or equal to that obtained using the HTC approach. For the STC 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 HTC 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 an STC 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 °C, the result with the HTC has lowest performance. With the STC with the linear penalty function, we violate the temperature threshold by 0.59 °C and achieve 6% higher performance but at 5% higher cost compared to the HTC approach. With the STC with the square penalty function, we violate the temperature threshold by 0.93 °C and achieve 5% higher performance at the same cost compared to the HTC approach.

VI. CONCLUSION

In this article, we introduced a cross-layer co-optimization methodology for network design and chiplet placement in 2.5-D systems. Our methodology optimizes network topology design, interchiplet link design, and chiplet placement across logical, physical, and circuit layers to jointly improve performance, lower manufacturing cost, and reduce operating temperature. Compared to our prior work, we improved the optimization methodology by enhancing the cost model, including operating temperature in the optimization goal, applying an STC, and improving the optimization algorithm to enable arbitrary chiplet placement. Our new methodology

shifts the performance-cost Pareto tradeoff curve for 2.5-D systems substantially. 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 2-D systems. Compared to our prior work [1], for the same HTC our enhanced placement algorithm with arbitrary placement improves iso-cost performance by 13%–37% (20% on average) and reduces iso-performance cost by 5%–20% (14% on average). Overall, our new optimization methodology with an STC and arbitrary placement achieves 40%–68% (49% on average) higher iso-cost performance and 30%–38% (32% on average) lower iso-performance cost over our prior work [1].

REFERENCES

[1] A. K. Coskun *et al.*, "A cross-layer methodology for design and optimization of networks in 2.5D systems," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2018, pp. 101–108.

[2] *Heterogeneous Integration Roadmap 2019 Edition*. Accessed: Feb. 7, 2020. [Online]. Available: https://eps.ieee.org/technology/heterogeneous-integration-roadmap/2019-edition.html

[3] (2015). *Heterogeneous Integration Chapter in ITRS 2.0*. [Online]. Available: http://www.itrs2.net/itrs-reports.html

[4] *DARPA CHIPS*. Accessed: Feb. 7, 2020. [Online]. Available: http://www.darpa.mil/news-events/2016-07-19

[5] A. Kannan *et al.*, "Enabling interposer-based disintegration of multi-core processors," in *Proc. IEEE/ACM 48th Annu. Int. Symp. Microarchit. (MICRO)*, 2015, pp. 546–558.

[6] G. H. Loh *et al.*, "Processor design in 3D die-stacking technologies," *IEEE Micro*, vol. 27, no. 3, pp. 31–48, May/Jun. 2007.

[7] F. Eris *et al.*, "Leveraging thermally-aware chiplet organization in 2.5D systems to reclaim dark silicon," in *Proc. Design Autom. Test Europe Conf. Exhibit. (DATE)*, 2018, pp. 1441–1446.

[8] D. C. Stow *et al.*, "Cost analysis and cost-driven IP reuse methodology for SoC design based on 2.5D/3D integration," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2016, pp. 56–61.

[9] P. Grani *et al.*, "Photonic interconnects for interposer-based 2.5D/3D integrated systems on a chip," in *Proc. 2nd Int. Symp. Memory Syst. (MEMSYS)*, 2016, pp. 377–386.

[10] J. Charbonnier *et al.*, "High density 3D silicon interposer technology development and electrical characterization for high end applications," in *Proc. 4th Electron. Syst. Integr. Technol. Conf. (ESTC)*, 2012, pp. 1–7.

[11] R. Chaware *et al.*, "Assembly and reliability challenges in 3D integration of 28nm FPGA die on a large high density 65nm passive interposer," in *Proc. IEEE 62nd Electron. Compon. Technol. Conf. (ECTC)*, San Diego, CA, USA, 2012, pp. 279–283.

[12] *FPGA VC707 Evaluation Kit Virtex-7*, Xilinx, San Jose, CA, USA, 2012.

[13] J. Macri, "AMD's next generation GPU and high bandwidth memory architecture: FURY," in *Proc. IEEE Hot Chips 27 Symp. (HCS)*, 2015, pp. 1–26.

[14] *Nvidia: NVIDIA Tesla P100*. Accessed: Feb. 7, 2020. [Online]. Available: https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf

[15] (2018). *Intel Introduces FOVEROS: 3D Die Stacking for More Than Just Memory*. [Online]. Available: https://arstechnica.com/gadgets/2018/12/intel-introduces-foveros-3d-die-stacking-for-more-than-just-memory/

[16] G. Parès, *3D Interposer for Silicon Photonics*, LETI, Grenoble, France, 2013.

[17] D. C. Stow *et al.*, "Cost-effective design of scalable high-performance systems using active and passive interposers," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2017, pp. 728–735.

[18] N. E. Jerger *et al.*, "NoC architectures for silicon interposer systems: Why pay for more wires when you can get them (from your interposer) for free?" in *Proc. IEEE/ACM 47th Annu. Int. Symp. Microarchit. (MICRO)*, 2014, pp. 458–470.

[19] *Hot Chips 2017: Intel Deep Dives Into EMIB*. Accessed: Feb. 7, 2020. [Online]. Available: https://www.tomshardware.com/news/intel-emib-interconnect-fpga-chiplet,35316.html

[20] S. Ramalingam, "HBM package integration: Technology trends, challenges and applications," in *Proc. IEEE Hot Chips 28 Symp. (HCS)*, 2016, pp. 1–17.

[21] R. Mahajan *et al.*, "Embedded multi-die interconnect bridge (EMIB)—A high density, high bandwidth packaging interconnect," in *Proc. IEEE 66th Electron. Compon. Technol. Conf. (ECTC)*, 2016, pp. 557–565.

[22] M. M. Ahmed *et al.*, "Increasing interposer utilization: A scalable, energy efficient and high bandwidth multicore-multichip integration solution," in *Proc. 8th Int. Green Sustain. Comput. Conf. (IGSC)*, 2017, pp. 1–6.

[23] I. Akgun *et al.*, "Scalable memory fabric for silicon interposer-based multi-core systems," in *Proc. IEEE 34th Int. Conf. Comput. Design (ICCD)*, 2016, pp. 33–40.

[24] S. Osmolovskyi *et al.*, "Optimal die placement for interposer-based 3D ICs," in *Proc. 23rd Asia South Pac. Design Autom. Conf. (ASP-DAC)*, 2018, pp. 513–520.

[25] C. Ravishankar *et al.*, "Placement strategies for 2.5D FPGA fabric architectures," in *Proc. 28th Int. Conf. Field Program. Logic Appl. (FPL)*, 2018, pp. 16–164.

[26] D. P. Seemuth *et al.*, "Automatic die placement and flexible I/O assignment in 2.5D IC design," in *Proc. 16th Int. Symp. Qual. Electron. Design (ISQED)*, 2015, pp. 524–527.

[27] E. J.-W. Fang *et al.*, "IR to routing challenge and solution for interposer-based design," in *Proc. 20th Asia South Pac. Design Autom. Conf. (ASP-DAC)*, 2015, pp. 226–230.

[28] W.-H. Liu *et al.*, "Metal layer planning for silicon interposers with consideration of routability and manufacturing cost," in *Proc. Design Autom. Test Europe Conf. Exhibit. (DATE)*, 2014, pp. 359–364.

[29] M. A. Karim *et al.*, "Power comparison of 2D, 3D, and 2.5D interconnect solutions and power optimization of interposer interconnects," in *Proc. IEEE 63rd Electron. Compon. Technol. Conf. (ECTC)*, 2013, pp. 860–866.

[30] M. S. Shamim *et al.*, "A wireless interconnection framework for seamless inter and intra-chip communication in multichip systems," *IEEE Trans. Comput.*, vol. 66, no. 3, pp. 389–402, Mar. 2017.

[31] D.-W. Kim *et al.*, "2.5D silicon optical interposer for 400 Gbps electronic-photonic integrated circuit platform packaging," in *Proc. IEEE 19th Electron. Packag. Technol. Conf. (EPTC)*, 2017, pp. 1–4.

[32] T. E. Carlson *et al.*, "SNIPER: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation," in *Proc. Int. Conf. High Perform. Comput. Netw. Stor. Anal. (SC)*, 2011, pp. 1–12.

[33] S. Li *et al.*, "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Proc. IEEE/ACM 42nd Annu. Int. Symp. Microarchit. (MICRO)*, 2009, pp. 469–480.

[34] *HSPICE User's Manual*, Meta Softw. Inc., Campbell, CA, USA, 1996.

[35] R. Zhang *et al.*, "HotSpot 6.0: Validation, acceleration and extension," Dept. Comput. Sci., Univ. Virginia, Charlottesville, VA, USA, Rep. CS-2015-04, 2015.

[36] R. Radojcic, *More-Than-Moore 2.5D and 3D SiP Integration*. Cham, Switzerland: Springer Int., 2017.

[37] G. Chen *et al.*, "A 340mV-to-0.9V 20.2Tb/s source-synchronous hybrid packet/circuit-switched 16×16 network-on-chip in 22nm tri-gate CMOS," *IEEE J. Solid-State Circuits*, vol. 50, no. 1, pp. 59–67, Jan. 2015.

[38] J. Knudsen, "NanGate 45nm open cell library," in *Proc. CDNLive EMEA*, 2008.

[39] E. Consoli *et al.*, "Conditional push-pull pulsed latches with 726fJ ps energy-delay product in 65nm CMOS," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2012, pp. 482–484.

[40] S.-C. Wong *et al.*, "Modeling of interconnect capacitance, delay, and crosstalk in VLSI," *IEEE Trans. Semicond. Manuf.*, vol. 13, no. 1, pp. 108–111, Feb. 2000.

[41] J. Howard *et al.*, "A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 173–183, Jan. 2011.

[42] T. Zhang *et al.*, "Thermal management of manycore systems with silicon-photonic networks," in *Proc. Design Autom. Test Europe Conf. Exhibit. (DATE)*, 2014, pp. 307–312.

[43] S. C. Woo *et al.*, "The SPLASH-2 programs: Characterization and methodological considerations," in *Proc. 22nd Annu. Int. Symp. Comput. Archit. (ISCA)*, 1995, pp. 24–36.

[44] K. Tran, "High-bandwidth memory white paper: Start your HBM/2.5D design today," Tempe, AZ, USA, Amkor Technol. Inc., White Paper, 2016. [Online]. Available: https://www.esilicon.com/wp-content/uploads/high-bandwidth-memory-white-paper-start-your-hbm-25d-design-today-20160329.pdf

[45] J. Meng *et al.*, "Optimizing energy efficiency of 3-D multicore systems with stacked DRAM under power and thermal constraints," in *Proc. Design Autom. Conf. (DAC)*, 2012, pp. 648–655.

[46] H. Wong. (2012). *A Comparison of Intel's 32nm and 22nm Core I5 CPUs: Power, Voltage, Temperature, and Frequency*. [Online]. Available: http://blog.stuffedcow.net/2012/10/intel32nm-22nm-core-i5-comparison/

[47] Y. Zhang *et al.*, "Thermal evaluation of 2.5D integration using bridge-chip technology: Challenges and opportunities," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 7, no. 7, pp. 1101–1110, Jul. 2017.