Network Design for Wafer-Scale Systems with Wafer-on-Wafer Hybrid Bonding

Patrick Iff
ETH Zurich
Zurich, Switzerland
iffp@inf.ethz.ch

Tommaso Bonato
ETH Zurich
Zurich, Switzerland

Maciej Besta
ETH Zurich
Zurich, Switzerland

Luca Benini
ETH Zurich
Zurich, Switzerland

Torsten Hoefler
ETH Zurich
Zurich, Switzerland
htor@inf.ethz.ch

Abstract

Transformer-based large language models are increasingly constrained by data movement as communication bandwidth drops sharply beyond the chip boundary. Wafer-scale integration using wafer-on-wafer hybrid bonding alleviates this limitation by providing ultra-high bandwidth between reticles on bonded wafers. In this paper, we investigate how the physical placement of reticles on wafers influences the achievable network topology and the resulting communication performance. Starting from a 2D mesh-like baseline, we propose four reticle placements (*Aligned*, *Interleaved*, *Rotated*, and *Contoured*) that improve throughput by up to 250%, reduce latency by up to 36%, and decrease energy per transmitted byte by up to 38%.

Keywords

Wafer-Scale Integration, 3D Integration, Interconnect

1 Introduction

Transformer-based large language models (LLMs) power today's most advanced AI systems, enabling breakthroughs in reasoning, generation, and multimodal understanding. Training these models is increasingly constrained by data movement [16]. Communication bandwidth declines sharply across hierarchy levels, from on-chip interconnects (multiple TB/s) to intra-node connections such as NVLink (≈ 900 GB/s) and inter-node fabrics like NVIDIA's NDR InfiniBand (≈ 100 GB/s) [13]. For decades, transistor scaling continually increased on-chip compute density and mitigated communication bottlenecks, but the slowdown of Moore's law and the end of Dennard scaling has largely curtailed these gains. Wafer-scale integration (WSI) provides an alternative path by scaling up the physical chip size itself, enabling entire wafers to function as unified substrates with high-bandwidth internal communication.

Wafer-on-wafer integration with hybrid bonding [24] is a promising and commercially available approach for building wafer-scale systems, exemplified by TSMC's SoIC-WoW [9]. In these systems, two silicon wafers are bonded face-to-face (F2F) using hybrid bonding, enabling high-density inter-wafer connections. Unlike reticle stitching, adjacent reticles on the same wafer cannot communicate directly; instead, reticles must be placed so that connecting any two overlapping reticles on opposite wafers yields a fully connected system. This integration scheme introduces a new and unexplored design space for on-chip interconnects, where the physical placement of reticles on wafers dictates the achievable network topologies, a key factor in communication performance.

In this paper, we explore how to place reticles on the top and bottom wafers to achieve efficient network topologies. Starting from a 2D mesh-like topology with up to four neighbors per reticle, we investigate how alternative reticle placements reduce the average path length by enabling up to seven neighbors per reticle. We introduce four novel reticle placements (*Aligned*, *Interleaved*, *Rotated*, and *Contoured*) that are tailored to different architectural configurations and present different trade-offs in terms of design and manufacturing complexity, and performance. Our evaluation shows that these placements improve overall throughput by up to 250%, while reducing average packet latency and energy per transmitted byte by up to 36% and 38%, respectively, compared to the baseline 2D mesh-like topology.

2 Background on Wafer-Scale Integration

2.1 Approaches to Wafer-Scale Integration

Several approaches exist to achieve wafer-scale integration. Tesla's Dojo [35], integrates 25 silicon dies of 645 mm^2 into a single wafer-scale system by placing individual chiplets on a fan-out wafer, a method known as **chiplet-based wafer-scale integration**. Cerebras [28] employs **field stitching** [10] to overcome the reticle limit of 26×33 mm. Field stitching introduces a small, intentional overlap between neighboring reticle exposures to align circuit patterns across seams and create continuous wires that cross reticle boundaries. In this work, we focus on a third approach, **wafer-on-wafer hybrid bonding**, as offered by TSMC's SoIC-WoW process [9], where two wafers are patterned with reticles and bonded F2F to form a single wafer-scale chip. Reticles on the same wafer cannot be connected directly, but by interleaving reticles on both wafers and vertically connecting them through hybrid bonds (HBs), a fully connected network among all reticles is built.

2.2 Wafer-on-Wafer Hybrid Bonding

A key advantage of wafer-on-wafer hybrid bonding over chiplet-based WSI is the extremely small pitch of HBs, which is below 10 μm in production [27] and reaches 1 μm in research prototypes [19]. Unlike die-to-die (D2D) links in chiplet-based wafer-scale integration, which require dedicated, area- and power-intensive physical layers (PHYs) on both ends for protocol, frequency, and voltage translation, hybrid bonding requires no PHYs because its electrical characteristics resemble those of the upper metal layers. Moreover, while D2D link bandwidth is typically limited by the number of available microbumps [14, 15], the fine pitch of hybrid bonding shifts the bottleneck to wire routing from the HBs to the router or to the router area itself. We use the term *vertical connector* to describe a collection of HBs that enable one link between wafers. To build a link between two reticles, the reticles must be on opposite wafers, and their vertical connectors must be precisely aligned.

3 Architecture Overview

In this section, we describe the different architectural choices and routing strategies we consider in this work.

3.1 System Architecture

We target machine learning (ML) and high-performance computing (HPC) workloads suitable for acceleration by GPU-like architectures. Each 26 × 33 mm reticle contains a GPU with eight graphics processing clusters (GPCs) and local SRAM.

Integration Level: We explore two levels of vertical integration. In **logic-on-interconnect (LoI)**, only the top wafer contains compute reticles (i.e., GPUs), while the bottom wafer serves purely as an interconnect layer. Limiting the system to a single compute wafer directly attached to the heat sink simplifies thermal management and power delivery. The second integration level, called **logic-on-logic (LoL)**, places compute reticles on both wafers, with the interconnect integrated into the compute reticles. While technically feasible today, power and thermal constraints remain major challenges. We expect LoL to become viable within a few years through improved power efficiency or advanced cooling, such as thermal through-silicon vias [1] or microfluidic cooling [7].

Wafer Diameter: We analyze **300 mm** wafers, which represent the current mainstream in semiconductor manufacturing, and **200 mm** wafers, still used in some older fabs.

Wafer Utilization: We analyze two levels of wafer utilization. The common approach in literature is to assemble a **rectangular** 2D grid of chiplets or reticles on a wafer [35, 38, 41]. We also consider the case where wafer utilization is **maximized** by tightly packing the largest possible number of reticles onto the wafer. This enables more efficient use of silicon and, importantly, a system with increased compute capabilities and higher integration density. Since wafer-on-wafer hybrid bonding removes the need for dicing streets between reticles, we omit inter-reticle spacing from our model, as the remaining micrometer-scale spacing does not noticeably affect the results.

3.2 Network Architecture

We assume a packet-switched network with wormhole routing and credit-based flow control. Following Yin et al. [40], who abstract the global network into a single router to optimize a chiplet's local network in isolation, we abstract each compute reticle's local network into a single router to optimize the wafer-scale network independently. Thus, a compute reticle is modeled as one router connecting all GPCs and the reticle's vertical connectors. For interconnect reticles in LoI, we explicitly model routers and their connecting links.

The routing algorithm consists of two components: the routing function and the selection function, both invoked when a packet traverses a router. The routing function returns a list of output ports that ensure deadlock- and livelock-freedom, from which the selection function chooses one. Our routing algorithm applies Dijkstra's algorithm [8] to return only shortest paths, ensuring progress toward the destination and thus guaranteeing livelock-freedom. It also employs the simple cycle-breaking (SCB) algorithm [25], a turn model [11] variant for arbitrary topologies, to guarantee deadlock-freedom.

Selection Function: We evaluate two different selection functions: **random**, which chooses an output port at random, and local **adaptive**, which selects the port towards the router with the most available space in its input buffer. Since the input buffer occupancy of adjacent routers is available via credit-based flow control, the adaptive selection function can be implemented without additional overhead to communicate congestion information.

The network topology is arguably the most critical aspect of the network architecture. In wafer-on-wafer hybrid bonding, links can connect only overlapping reticles on opposite wafers; thus, the topology is dictated by the reticle placement, whose optimization is the central contribution of this work.

4 Optimization of Reticle Placement

In this section, we optimize the network topology. Previous work on network topologies, from supercomputers [3, 23] to network-on-chips (NoCs) [2, 18] and inter-chip interconnects (ICIs) [4, 14, 15, 21], focused on minimizing network diameter and average path length. Reducing these metrics decreases latency, mitigates congestion, and increases throughput. In wafer-on-wafer hybrid bonding systems, only links between overlapping reticles can be implemented, so optimizing the topology requires optimizing reticle placement.

Table 1: (§4) Comparison of different reticle placements.

Integration Level	Wafer Diameter	Wafer Utilization	Placement	Number of Compute Reticles (26×33mm)	Number of Interconnect Reticles (≈26×33mm)	Radix of Compute Reticles	Radix of Interconnect Reticles	Network Diameter	Average Path Length (Hops)	Total Bisection Bandwidth
Logic on Interconnect	200mm	Rec.	Baseline	20	26	4	4	8	4.08	16.00
			Ours Aligned	20	10	4	6	6	3.30	16.00
			Ours Interleaved	20	12	4	6	8	3.44	16.00
			Ours Rotated	20	20	7	7	6	2.84	32.00
		Max.	Baseline	26	26	4	4	12	4.80	16.00
			Ours Aligned	26	12	4	6	10	3.91	16.40
			Ours Interleaved	26	14	4	6	10	3.89	16.00
			Ours Rotated	27	25	7	7	6	3.20	38.00
	300mm	Rec.	Baseline	49	56	4	4	12	6.44	27.20
			Ours Aligned	49	28	4	6	12	5.53	28.00
			Ours Interleaved	49	26	4	6	12	5.57	24.00
			Ours Rotated	48	48	7	7	10	4.19	47.60
		Max.	Baseline	64	63	4	4	18	7.45	26.00
			Ours Aligned	64	31	4	6	14	5.83	31.20
			Ours Interleaved	64	31	4	6	14	6.04	28.20
			Ours Rotated	66	63	7	7	10	4.76	64.20
Logic on Logic	200mm	Rec.	Baseline	46	0	4	-	10	4.40	16.00
			Ours Contoured	40	0	5	-	8	3.52	16.00
		Max.	Baseline	52	0	4	-	12	4.71	16.00
			Ours Contoured	54	0	5	-	10	3.93	21.20
	300mm	Rec.	Baseline	105	0	4	-	14	6.66	27.20
			Ours Contoured	96	0	5	-	12	5.20	28.00
		Max.	Baseline	127	0	4	-	20	7.42	25.60
			Ours Contoured	132	0	5	-	16	6.01	36.00

Our approach is to minimize the average path length by maximizing the number of overlapping reticles between the two wafers (i.e., the network radix). While we present results for reticles at the lithographic limit of 26 × 33 mm, all optimization techniques apply to other reticle sizes as well. Table 1 lists the reticle count, radix, diameter, average path length, and estimated bisection bandwidth (averaged over ten METIS [22] runs with different random seeds) for all proposed reticle placements.

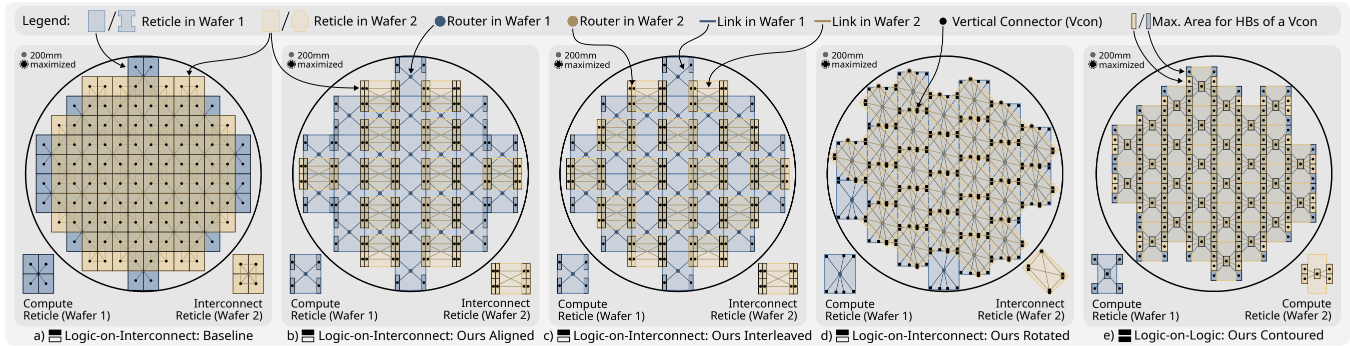

Figure 1: (§4) Different methods of building wafer-scale systems by optimizing the placement of reticles on the wafers.

4.1 Placements for Logic-on-Interconnect

Constraints. We assume that to minimize design cost and maximize manufacturing efficiency, all reticles on a given wafer are required to be identical.

Baseline. Most prior work on wafer-scale networks [29–31, 38, 39, 41] focuses on chiplet-based wafer-scale integration rather than wafer-to-wafer hybrid bonding, leaving no established baseline for this unexplored network design space. Since many chiplet-based systems use a 2D mesh topology as a baseline, we adopt a system that approximates a 2D mesh. In this baseline, the reticles on the interconnect wafer are shifted by half a reticle width and height so that each interconnect reticle connects to four neighboring compute reticles, and vice versa (see Fig. 1a). Each interconnect reticle contains a fully connected radix-4 network topology. Note that the resulting topology[1] does not exactly match a conventional 2D mesh network, so the XY-routing algorithm [5] cannot be applied.

Optimization "Ours Aligned". In our first optimization, we retain radix-4 compute reticles but rotate the interconnect reticles by 90 degrees and align them so that each interconnect reticle connects to up to six compute reticles (see Fig. 1b). This reduces both the average path length and the number of interconnect reticles, accelerating the manufacturing process. The overlapping area available per vertical connector decreases from 214.5 mm^2 to 45.5 mm^2, but even with a conservative 10 μm hybrid bond pitch, only 3.2 mm^2 is needed to implement a bidirectional 2 TB/s link at 1 GHz, so the overlap is more than sufficient. Each interconnect reticle provides eight vertical connectors and uses a fully connected intra-reticle topology of four routers with concentration 2.

Optimization "Ours Interleaved". This optimization slightly modifies the previous reticle placement by interleaving the interconnect reticles instead of aligning them (see Fig. 1c), resulting in a distinct network topology[1].

Optimization "Ours Rotated". We maximize the network radix of both compute and interconnect reticles. By reducing the interconnect reticle size to 22.98 × 32.53 mm and rotating them by 45 degrees, each interconnect reticle overlaps with up to seven compute reticles (see Fig. 1d). Although the overlapping area is smaller, it supports up to 6 TB/s links (assuming a HB pitch of 10 μm) with more than 10 mm^2 available per vertical connector. Unlike the previous placements with radix-4 compute reticles, this configuration increases the compute reticle radix to 7, slightly enlarging its area (see Section 5.2.2 for area evaluation details). Each interconnect reticle features seven vertical connectors and employs a fully connected topology of four routers with concentration 1 or 2. While a formal optimality proof is beyond the scope of this paper, an exhaustive search over all integer reticle positions and rotations found no configuration with a higher radix than seven.

4.2 Placements for Logic-on-Logic

Constraints. We again assume that all reticles on a given wafer are identical. While for LoI systems, only the compute reticles needed to tessellate the wafer plane and the interconnect reticles could be placed with spaces in between, in LoL systems, both wafers contain compute reticles and must tessellate the plane to maximize integration density.

Baseline. We use the same baseline placement as in LoI systems (see Fig. 1a). The only difference is that both wafers now contain identical radix-4 compute reticles.

Optimization "Ours Contoured". Because LoL systems prohibit spacing between reticles to maximize integration density, the three LoI optimizations relying on gaps between interconnect reticles are not applicable. We therefore propose a new radix-5 placement using contoured reticles on both wafers. The lower wafer features H-shaped reticles, while the upper wafer uses plus-shaped reticles (see Fig. 1e). By aligning the centers of these shapes, each reticle connects to up to five reticles on the opposite wafer. The placement illustrated in Fig. 1e is schematic, with exaggerated contouring that makes the total reticle area appear much smaller than the reticle limit. In practice, contouring is limited to the minimum required to achieve the target link bandwidth (e.g., for 2 TB/s links, the reticle area equals 98.5% of the reticle limit).

5 Evaluation

5.1 Experiment Setup

We use the cycle-accurate BookSim2 [17] NoC simulator to perform flit-level simulations of each wafer-scale architecture, providing zero-load latency, saturation throughput, and the average number of router-to-router hops per packet. BookSim2 models wormhole routing with virtual-channel flow control and a four-stage router pipeline (routing, virtual-channel allocation, switch allocation, and

[1]Visualizations of all network topologies as graphs are available in our repository: https://github.com/spcl/nw-design-for-wsi .

crossbar traversal). All simulations are repeated three times with different random seeds, and the results are averaged. Area and power estimates are obtained using the Orion3.0 [20] NoC power and area model. Because Orion3.0 supports only up to 45 nm technology, we scale the area and power results to 7 nm using DeepScaleTool [32].

5.1.1 Architectural Parameters. We model interconnects with bidirectional links providing 2 TB/s bandwidth per direction at 1 GHz, matching the link bandwidth in Tesla's Dojo [35]. Links are implemented as pipelined interconnects with one pipeline stage (register buffer) every 2 mm of physical wire length. Routers have a latency of four cycles. Through extensive performance exploration across different input buffer sizes, we found that large wafer-scale architectures with long pipelined links require 32 flit buffers to exploit the full throughput potential. Given this large buffer requirement, we consider virtual channels too costly and therefore assume a single virtual channel per physical channel in all experiments. However, our proposed network optimizations are fully compatible with configurations using multiple virtual channels.

5.1.2 Workloads. We use four synthetic traffic patterns: uniform (modeling all-to-all workloads such as mixture-of-experts (MoE) training [26]), random permutation (modeling shuffle-style workloads such as FFT or sorting [6]), neighbor (modeling stencil workloads such as fluid dynamics simulations [6]), and tornado (modeling long-stride communication).

In addition, we leverage the ATLAHS [34] toolchain to collect GOAL [12] formatted traces from Llama-7B [36] training, and extend BookSim2 to replay these traces on our wafer-scale architectures. These traces capture inter-GPU messages, message sizes, computation phase durations, and all dependencies between communication and computation events. We collect traces for training on 20, 24, 40, 48, 52, 64, 96, and 124 GPUs to obtain a suitable trace for each architecture under evaluation, noting that a few GPUs may be idle in some configurations. Messages, which can reach 1.8 MB, are split into 2 KB packets for network transmission. Because BookSim2's cycle-accurate flit-level model makes the simulation of a full LLM training epoch impractically slow, we instead run three independent simulations of eight hours each for every one of the 48 architecture–placement combinations, resulting in an average of 1.17 million packets transmitted per simulation.

5.1.3 Metrics and Measurement Methodology (Synthetic Traffic).

Latency. We use a BookSim2 simulation at a low injection rate to measure the zero-load latency, configuring each link's latency according to its physical length and assigning a latency of one cycle to each vertical connector between wafers.

Throughput. We progressively increase the injection rate in BookSim2 by 10%, 1%, 0.1%, and 0.01% increments to accurately determine the network's saturation throughput (the point where the latency exceeds twice the zero-load latency). Fig. 2 shows a latency vs. load curve, where each point represents a BookSim2 simulation at a specific injection rate.

Area. We use Orion3.0 to estimate the area of NoC routers, assuming that input buffers are implemented as SRAM rather than flip-flops. Since SRAM scaling has plateaued compared to logic scaling, we apply a scaling factor of 0.2 to scale SRAM area from 45 nm to 7 nm, which is more conservative than DeepScaleTool's area scaling factor of 0.0271.

Power and Energy. We use Orion3.0 to estimate the power consumption of each NoC router. For buffered links, we use BookSim2 results to estimate the average number of pipeline stages traversed per flit by subtracting the product of the average hop count and router latency from the zero-load latency. We assume a conservative energy of 2 pJ per bit per pipeline stage, consistent with prior work [37]. Because the energy consumption of HBs is negligible [37], we do not model it explicitly. Our analysis shows that in wafer-scale architectures, link power consumption exceeds router power by orders of magnitude, so we report only the total network power consumption without separating router and link power.

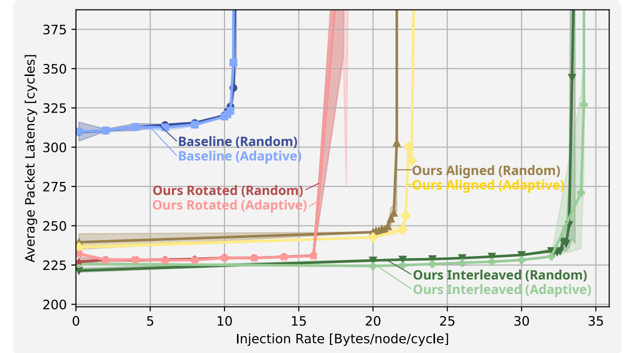

Figure 2: (§5.2.1) Latency vs. Load for LoI with 300 mm wafers and maximized utilization (permutation traffic).

5.2 Experiment Results on Synthetic Traffic

5.2.1 Latency and Throughput. Fig. 2 shows detailed latency vs. load curves for our four reticle placements and two selection functions on the LoI system with 300 mm wafers and maximized wafer utilization under random permutation traffic. Similar plots for the remaining 31 experiments are available in our open-source repository[2]. Due to space constraints, we summarize the latency and throughput results in heatmap plots showing improvements over the baseline placement in Figs. 3 to 6. Analyzing these heatmaps provides insights into the performance of our proposed placements across system configurations and traffic patterns.

Our *Aligned* and *Interleaved* placements increase throughput while reducing latency across all systems with maximized wafer utilization. With rectangular utilization, they still improve these two metrics in most cases, though they may underperform the *Baseline* for tornado and neighbor traffic. The *Rotated* placement consistently outperforms the *Baseline* across all architectures and traffic patterns. The *Contoured* placement for LoL systems improves throughput in most cases while maintaining similar latency as the *Baseline*. The adaptive selection function slightly increases throughput at comparable latency to the random selection function. Overall, our optimized placements achieve stronger improvements for maximized than for rectangular wafer utilization and for LoI than for LoL systems, with wafer diameter having only a minor effect. Performance gains are also more consistent for random uniform and random permutation traffic than for tornado and neighbor traffic.

[2] https://github.com/spcl/nw-design-for-wsi

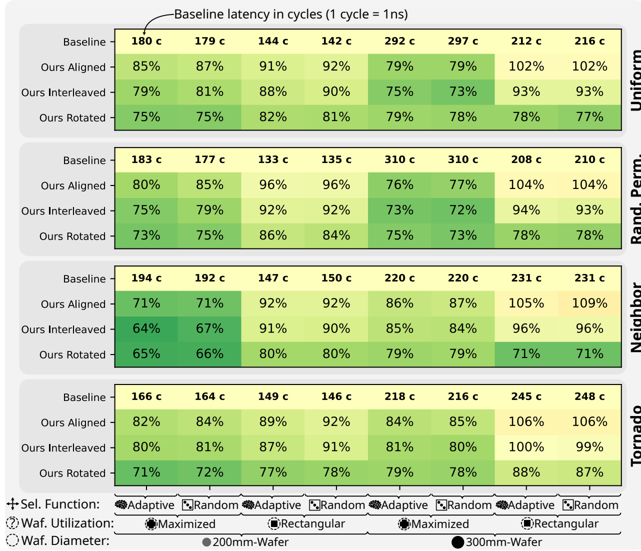

Figure 3: (§5.2.1) Latency of Logic-on-Interconnect.

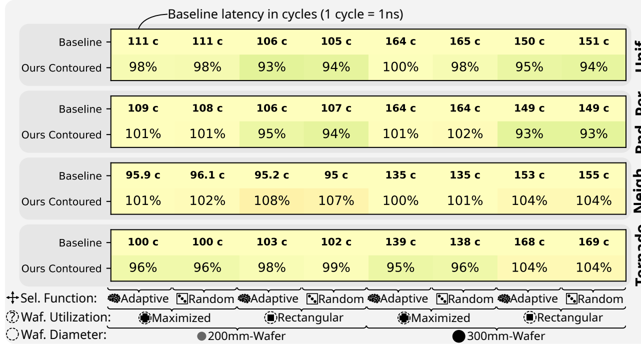

Figure 4: (§5.2.1) Latency of Logic-on-Logic.

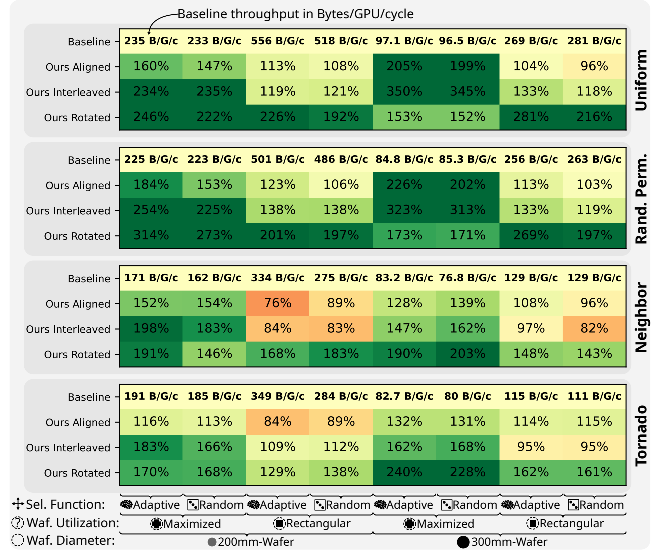

Figure 5: (§5.2.1) Throughput of Logic-on-Interconnect.

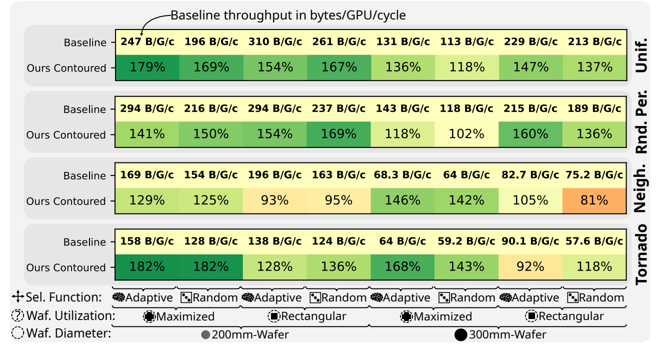

Figure 6: (§5.2.1) Throughput of Logic-on-Logic.

5.2.2 Area. Fig. 7 shows the area occupied by routers on compute and, where applicable, interconnect reticles. Because we report per-reticle area, these results are independent of wafer diameter and utilization and therefore apply to all LoI and LoL systems. We observe that routers occupy only a small fraction of the reticle area, and our proposed placements introduce little or no additional area overhead compared to the baseline. Router area is dominated by input buffers, with the remaining router logic contributing negligibly. A detailed study of reticle wiring resources is beyond the scope of this work, but global wiring usage can be expected to scale roughly with router area.

5.2.3 Power and Energy. Fig. 8 shows the total power consumption of the wafer-scale network (left) and the normalized energy per transferred byte (right) at saturation throughput for the two selection functions on the LoI system with 300 mm wafers and maximized wafer utilization under random permutation traffic. Equivalent plots for the remaining 31 experiments are available in our open-source repository[3]. We summarize the energy efficiency results in Figs. 9 and 10. Comparing the network power of about 4 kW to the reported 15 kW wafer-scale power budget [31] suggests that up to one quarter of the total power could be devoted to global data movement. Note that power is evaluated at saturation throughput. Since average network utilization is typically well below 100%, actual power consumption under normal operation will be much lower. The energy per byte shows that our optimized placements typically improve efficiency by shortening the average path length, while the higher total power mainly results from increased saturation throughput.

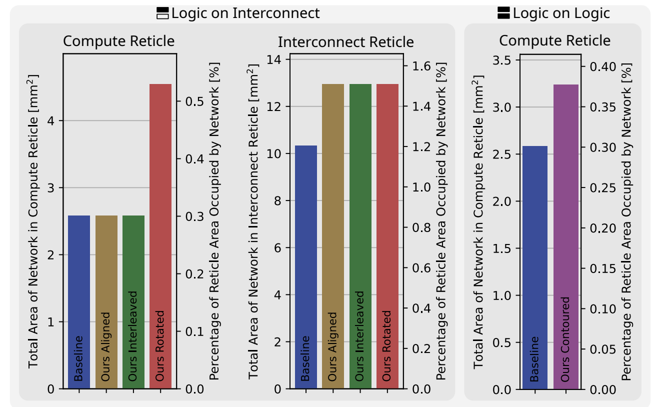

Figure 7: (§5.2.2) Area occupied by Network Routers.

[3] https://github.com/spcl/nw-design-for-wsi

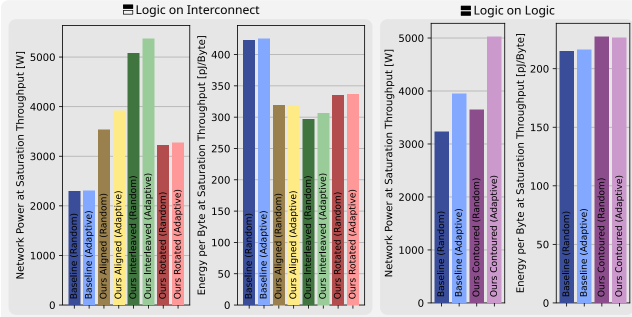

Figure 8: (§5.2.3) Power & energy for LoI with 300 mm wafers and maximized utilization (permutation traffic).

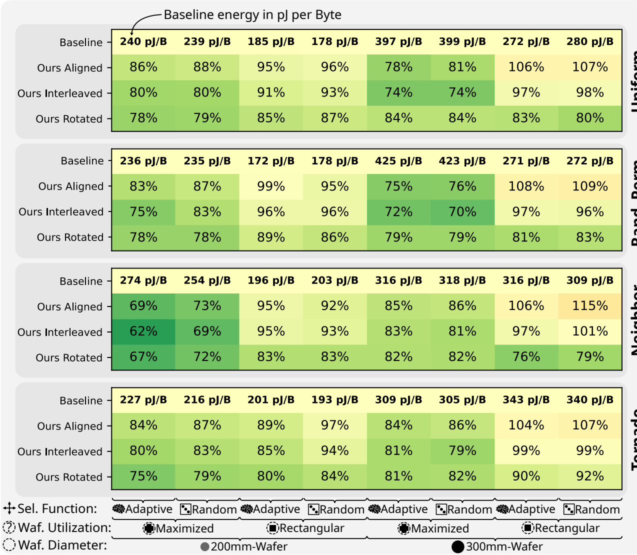

Figure 9: (§5.2.3) Energy of Logic-on-Interconnect.

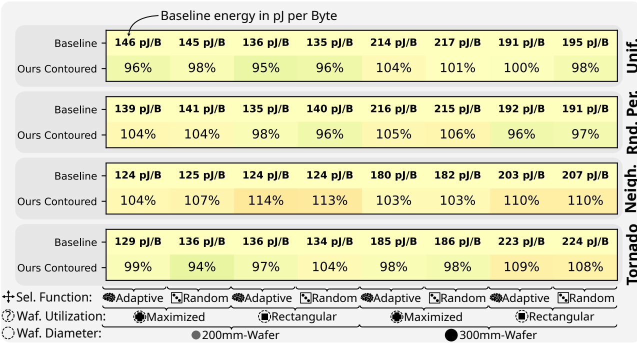

Figure 10: (§5.2.3) Energy of Logic-on-Logic.

5.3 Experiment Results on Application Traces

Fig. 11 shows the average network latency observed when running the Llama-7B training traces. During these simulations the network alternates between phases of high load with severe congestion and phases of lower load where computation dominates. This elevated congestion leads to substantially higher average latencies than in the synthetic traffic experiments where the zero-load latency captures the latency without contention. Our results indicate that for LLM training workloads the latency reductions achieved by our proposed placements exceed those measured with synthetic traffic. On average latency decreases to 60% of the baseline and in the best case to 37%. While beneficial for all architectures, our optimized placements yield larger improvements for LLM training on 300 mm wafers than on 200 mm wafers, and on LoI systems than on LoL systems.

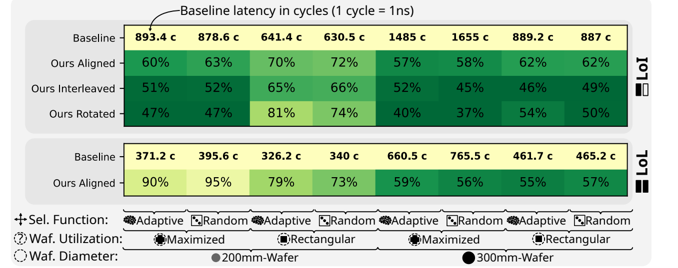

Figure 11: (§5.3) Network latency during trace simulation.

6 Related Work

With its SoIC technology, TSMC offers wafer-on-wafer hybrid bonding with fine-pitch connections, and its roadmap [9] projects a 2× increase in interconnect density every two years. Such rapid scaling has become feasible only through recent advances in hybrid bonding processes [19], comprehensively reviewed by Lau et al. [24]. For a broader overview of wafer-scale computing, we refer to Hu et al. [13].

While no prior work has addressed network design for WSI systems based on wafer-on-wafer hybrid bonding, several studies have explored related directions for chiplet-based WSI. FRED [31] employs a Clos-like topology for wafer-scale systems to accelerate collective operations in DNN training, but such topologies are infeasible under the geometric constraints of wafer-on-wafer hybrid bonding. Network-on-Wafer [39] co-designs topology, routing, and collective operations for wafer-scale systems and assumes per-link bandwidths of 2 TB/s, similar to Tesla Dojo [35] and our work. WSC-LLM [38] explores joint architectural and scheduling optimization using a 2D mesh topology for inter-die communication. Other studies adopting 2D mesh topologies and motivating our mesh-like baseline include the wafer-scale AI accelerator Simba [33], a wafer-scale GPU architecture [30], and a 2048-chiplet wafer-scale system developed by UCLA and UIUC [29].

7 Conclusion

Wafer-on-wafer hybrid bonding is a promising and readily available technology for realizing WSI with high-bandwidth interconnects. The constraint that network topology must emerge from connecting overlapping reticles on opposite wafers creates a new and unexplored design space. In this work, we explore said design space by optimizing the placement of reticles on the top and bottom wafers to maximize the number of neighbors per reticle and thereby minimize the network's average path length.

Our comprehensive evaluation shows that our proposed reticle placements significantly improve throughput, latency, and energy efficiency across almost all integration levels, wafer diameters, wafer utilizations, and workloads considered. We achieve throughput improvements of up to 250%, latency reductions of up to 36%, and energy reductions of up to 38%.

Acknowledgements

We thank Shuhao Li, Timo Schneider, and Siyuan Shen for their support with the ATLAHS toolchain and we thank Jiho Kim for insightful discussions on area and power modeling. Furthermore, we thank the Swiss National Supercomputing Centre (CSCS) for access to their Alps system. This work was supported by the ETH Future Computing Laboratory (EFCL), financed by a donation from Huawei Technologies. It also received funding from the European Research Council erc (Project PSAP, No. 101002047) and from the European Union's HE research and innovation programme (grant agreement No. 101070141; Project GLACIATION).

References

[1] Ayed Alqahtani, Zongqing Ren, Jaeho Lee, and Nader Bagherzadeh. 2018. System-level analysis of 3D ICs with thermal TSVs. *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 14, 3 (2018), 1–16.

[2] Maciej Besta, Syed Minhaj Hassan, Sudhakar Yalamanchili, Rachata Ausavarungnirun, Onur Mutlu, and Torsten Hoefler. 2018. Slim noc: A low-diameter on-chip network topology for high energy efficiency and scalability. *ACM SIGPLAN Notices* 53, 2 (2018), 43–55.

[3] Maciej Besta and Torsten Hoefler. 2014. Slim fly: A cost effective low-diameter network topology. In *SC'14: proceedings of the international conference for high performance computing, networking, storage and analysis*. IEEE, 348–359.

[4] Srikant Bharadwaj, Jieming Yin, Bradford Beckmann, and Tushar Krishna. 2020. Kite: A family of heterogeneous interposer topologies enabled via accurate interconnect modeling. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 1–6.

[5] Shubhangi D Chawade, Mahendra A Gaikwad, and Rajendra M Patrikar. 2012. Review of XY routing algorithm for network-on-chip architecture. *International Journal of Computer Applications* 43, 21 (2012), 975–8887.

[6] William James Dally and Brian Patrick Towles. 2004. *Principles and practices of interconnection networks*. Elsevier.

[7] Bing Dang, Muhannad S Bakir, Deepak Chandra Sekar, Calvin R King, and James D Meindl. 2010. Integrated microfluidic cooling and interconnects for 2D and 3D chips. *IEEE Transactions on Advanced Packaging* 33, 1 (2010), 79–87.

[8] Edsger W Dijkstra. 2022. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: his life, work, and legacy*.

[9] CH Douglas, Chuei-Tang Wang, and Harry Hsia. 2021. Foundry perspectives on 2.5 D/3D integration and roadmap. In *2021 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 3–7.

[10] WW Flack and GE Flores. 1992. Lithographic manufacturing techniques for wafer scale integration. In *[1992] Proceedings International Conference on Wafer Scale Integration*. IEEE, 4–13.

[11] Christopher J Glass and Lionel M Ni. 1992. The turn model for adaptive routing. *ACM SIGARCH Computer Architecture News* 20, 2 (1992).

[12] Torsten Hoefler, Christian Siebert, and Andrew Lumsdaine. 2009. Group operation assembly language-a flexible way to express collective communication. In *2009 International Conference on Parallel Processing*. IEEE, 574–581.

[13] Yang Hu, Xinhan Lin, Huizheng Wang, Zhen He, Xingmao Yu, Jiahao Zhang, Qize Yang, Zheng Xu, Sihan Guan, Jiahao Fang, et al. 2024. Wafer-scale computing: advancements, challenges, and future perspectives [Feature]. *IEEE Circuits and Systems Magazine* 24, 1 (2024), 52–81.

[14] Patrick Iff, Maciej Besta, Matheus Cavalcante, Tim Fischer, Luca Benini, and Torsten Hoefler. 2023. Hexamesh: Scaling to hundreds of chiplets with an optimized chiplet arrangement. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 1–6.

[15] Patrick Iff, Maciej Besta, and Torsten Hoefler. 2025. FoldedHexaTorus: An Inter-Chiplet Interconnect Topology for Chiplet-based Systems using Organic and Glass Substrates. *arXiv preprint arXiv:2504.19878* (2025).

[16] Andrei Ivanov, Nikoli Dryden, Tal Ben-Nun, Shigang Li, and Torsten Hoefler. 2021. Data movement is all you need: A case study on optimizing transformers. *Proceedings of Machine Learning and Systems* 3 (2021), 711–732.

[17] Nan Jiang, Daniel U Becker, George Michelogiannakis, James Balfour, Brian Towles, David E Shaw, John Kim, and William J Dally. 2013. A detailed and flexible cycle-accurate network-on-chip simulator. In *2013 IEEE international symposium on performance analysis of systems and software (ISPASS)*. IEEE, 86–96.

[18] Dai Cheol Jung, Scott Davidson, Chun Zhao, Dustin Richmond, and Michael Bedford Taylor. 2020. Ruche networks: Wire-maximal, no-fuss nocs: Special session paper. In *2020 14th IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*. IEEE, 1–8.

[19] Yoshihisa Kagawa, Takumi Kamibayashi, Yuriko Yamano, Kenya Nishio, Akihisa Sakamoto, Taichi Yamada, Kan Shimizu, Tomoyuki Hirano, and Hayato Iwamoto. 2022. Development of face-to-face and face-to-back ultra-fine pitch Cu-Cu hybrid bonding. In *2022 IEEE 72nd Electronic Components and Technology Conference (ECTC)*. IEEE, 306–311.

[20] Andrew B Kahng, Bill Lin, and Siddhartha Nath. 2015. ORION3. 0: A comprehensive NoC router estimation tool. *IEEE Embedded Systems Letters* 7, 2 (2015), 41–45.

[21] Ajaykumar Kannan, Natalie Enright Jerger, and Gabriel H Loh. 2015. Enabling interposer-based disintegration of multi-core processors. In *Proceedings of the 48th international symposium on Microarchitecture*. 546–558.

[22] George Karypis and Vipin Kumar. 1997. METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. (1997).

[23] John Kim, Wiliam J Dally, Steve Scott, and Dennis Abts. 2008. Technology-driven, highly-scalable dragonfly topology. *ACM SIGARCH Computer Architecture News* 36, 3 (2008), 77–88.

[24] John H Lau. 2023. Recent advances and trends in Cu–Cu hybrid bonding. *IEEE Transactions on Components, Packaging and Manufacturing Technology* 13, 3 (2023), 399–425.

[25] Lev Levitin, Mark Karpovsky, and Mehmet Mustafa. 2009. Deadlock prevention by turn prohibition in interconnection networks. In *2009 IEEE international symposium on parallel & distributed processing*. IEEE.

[26] Jiamin Li, Yimin Jiang, Yibo Zhu, Cong Wang, and Hong Xu. 2023. Accelerating distributed {MoE} training and inference with lina. In *2023 USENIX Annual Technical Conference (USENIX ATC 23)*. 945–959.

[27] Shenggao Li, Mu-Shan Lin, Wei-Chih Chen, and Chien-Chun Tsai. 2024. High-bandwidth chiplet interconnects for advanced packaging technologies in AI/ML applications: Challenges and solutions. *IEEE Open Journal of the Solid-State Circuits Society* (2024).

[28] Sean Lie. 2022. Cerebras architecture deep dive: First look inside the hw/sw co-design for deep learning: Cerebras systems. In *2022 IEEE Hot Chips 34 Symposium (HCS)*. IEEE Computer Society, 1–34.

[29] Saptadeep Pal, Jingyang Liu, Irina Alam, Nicholas Cebry, Haris Suhail, Shi Bu, Subramanian S Iyer, Sudhakar Pamarti, Rakesh Kumar, and Puneet Gupta. 2021. Designing a 2048-chiplet, 14336-core waferscale processor. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 1183–1188.

[30] Saptadeep Pal, Daniel Petrisko, Matthew Tomei, Puneet Gupta, Subramanian S Iyer, and Rakesh Kumar. 2019. Architecting waferscale processors-a gpu case study. In *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 250–263.

[31] Saeed Rashidi, William Won, Sudarshan Srinivasan, Puneet Gupta, and Tushar Krishna. 2025. FRED: A Wafer-scale Fabric for 3D Parallel DNN Training. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture*. 34–48.

[32] Satyabrata Sarangi and Bevan Baas. 2021. DeepScaleTool: A tool for the accurate estimation of technology scaling in the deep-submicron era. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1–5.

[33] Yakun Sophia Shao, Jason Clemons, Rangharajan Venkatesan, Brian Zimmer, Matthew Fojtik, Nan Jiang, Ben Keller, Alicia Klinefelter, Nathaniel Pinckney, Priyanka Raina, et al. 2019. Simba: Scaling deep-learning inference with multi-chip-module-based architecture. In *Proceedings of the 52nd annual IEEE/ACM international symposium on microarchitecture*. 14–27.

[34] Siyuan Shen, Tommaso Bonato, Zhiyi Hu, Pasquale Jordan, Tiancheng Chen, and Torsten Hoefler. 2025. ATLAHS: An Application-centric Network Simulator Toolchain for AI, HPC, and Distributed Storage. *arXiv preprint arXiv:2505.08936* (2025).

[35] Emil Talpes, Douglas Williams, and Debjit Das Sarma. 2022. Dojo: The microarchitecture of tesla's exa-scale computer. In *2022 IEEE Hot Chips 34 Symposium (HCS)*. IEEE Computer Society, 1–28.

[36] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[37] Tony F Wu, Huichu Liu, H Ekin Sumbul, Lita Yang, Dipti Baheti, Jeremy Coriell, William Koven, Anu Krishnan, Mohit Mittal, Matheus Trevisan Moreira, et al. 2024. 11.2 a 3D integrated prototype system-on-chip for augmented reality applications using face-to-face wafer bonded 7nm logic at< $2\mu m$ pitch with up to 40% energy reduction at iso-area footprint. In *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. 67. IEEE, 210–212.

[38] Zheng Xu, Dehao Kong, Jiaxin Liu, Jinxi Li, Jingxiang Hou, Xu Dai, Chao Li, Shaojun Wei, Yang Hu, and Shouyi Yin. 2025. WSC-LLM: Efficient LLM Service and Architecture Co-exploration for Wafer-scale Chips. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture*. 1–17.

[39] Qize Yang, Taiquan Wei, Sihan Guan, Chengran Li, Haoran Shang, Jinyi Deng, Huizheng Wang, Chao Li, Lei Wang, Yan Zhang, et al. 2025. PD Constraint-aware Physical/Logical Topology Co-Design for Network on Wafer. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture*. 49–64.

[40] Jieming Yin, Zhifeng Lin, Onur Kayiran, Matthew Poremba, Muhammad Shoaib Bin Altaf, Natalie Enright Jerger, and Gabriel H Loh. 2018. Modular

routing design for chiplet-based systems. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 726–738.

[41] Xingmao Yu, Dingcheng Jiang, Jinyi Deng, Jingyao Liu, Chao Li, Shouyi Yin, and Yang Hu. 2025. Cramming a Data Center into One Cabinet, a Co-Exploration of Computing and Hardware Architecture of Waferscale Chip. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture*. 631–645.

Received 18 November 2025; revised XX Month 2025; accepted XX Month 2025