

Switch-Less Dragonfly on Wafers: A Scalable Interconnection Architecture based on Wafer-Scale Integration

Yinxiao Feng and Kaisheng Ma*

Institute for Interdisciplinary Information Sciences (IIIS)
Tsinghua University, Beijing, China

Abstract—Existing high-performance computing (HPC) interconnection architectures are based on high-radix switches, which limits the injection/local performance and introduces latency/energy/cost overhead. The new wafer-scale packaging and high-speed wireline technologies provide high-density, low-latency, and high-bandwidth connectivity, thus promising to support direct-connected high-radix interconnection architecture.

In this paper, we propose a wafer-based interconnection architecture called *Switch-Less-Dragonfly-on-Wafers*. By utilizing distributed high-bandwidth networks-on-chip-on-wafer, costly high-radix switches of the *Dragonfly* topology are eliminated while increasing the injection/local throughput and maintaining the global throughput. Based on the proposed architecture, we also introduce baseline and improved deadlock-free minimal/non-minimal routing algorithms with only one additional virtual channel. Extensive evaluations show that the *Switch-Less-Dragonfly-on-Wafers* outperforms the traditional switch-based *Dragonfly* in both cost and performance. Similar approaches can be applied to other switch-based direct topologies, thus promising to power future large-scale supercomputers.

Index Terms—wafer-scale integration, HPC interconnection network, *Dragonfly*, network-on-chip, routing algorithm.

I. INTRODUCTION

Mainstream high-performance computing (HPC) interconnection architectures are based on switches/routers. High-radix IO modules and switches enable very low-diameter network topologies, e.g., 2 switch-to-switch hops for Slim Fly [1] and PolarFly [2], 3 hops for Dragonfly [3], and 4 hops for three-stage Fat-Tree [4]. However, high-radix switches are limited in the port number and bandwidth per link. 400G/800G is the maximum bandwidth provided by current Ethernet or InfiniBand adapters/switches [5–7]. The limited physical channels connecting endpoints to the switch significantly constrain the local performance (injection bandwidth), which is critical for some workloads such as AI [8]. Besides, high-radix switches are expensive and introduce additional latency and energy overhead [9–12]. On the other hand, modern computing chips by themselves can provide abundant IO and switching bandwidth no weaker than a regular switching chip [13–15], thus introducing the motivation to fully utilize the local bandwidth of computing chips [8].

In recent years, a new advanced packaging technology called *wafer-scale-integration* promises to densely integrate

tens of chips and provide ultra-high on/off-wafer bandwidth [16, 17]. For example, a tile of DOJO achieves 10TB/s on-wafer bisection bandwidth and 36 TB/s off-wafer aggregate bandwidth [18], which is far beyond any existing switch. Therefore, if the chips can be directly interconnected with high-bandwidth and low-latency, it not only improves the network performance but also promises to avoid using costly high-radix switches. However, scaling wafer-scale systems out for large-scale supercomputers still faces many challenges. 1) Existing wafer-based systems, including *Wafer-scale Processor* [19], *Wafer-Scale GPU* [20], *Wafer-Scale Engine* (WSE) [21–23], and *DOJO* [18, 24–26], are based on the 2D-mesh topology, which is not scalable due to the large diameter. 2) The off-wafer bandwidth has a significant gap with the on-wafer bandwidth, which places higher demands on the hierarchy and configurability. 3) Besides, interconnecting 2D-mesh-on-wafer by high-radix topologies introduces serious routing problems. The on-chip and off-chip routing must be designed and evaluated jointly rather than separately.

Motivated by these, we propose a new interconnection architecture called *Switch-less Dragonfly on Wafers*. By utilizing distributed high-bandwidth networks-on-chip-on-wafer, we build a scalable wafer-based Dragonfly network without high-radix switches. The critical issues, including scalability, throughput, diameter, latency, energy, and cost, are quantitatively analyzed and discussed. We also give a simple minimal/non-minimal routing algorithm and a method to reduce the virtual-channel number. Extensive evaluations, including physical layout and cycle-accurate simulations on various workloads, are conducted based on the architecture. The contributions of this paper can be summarized as follows:

- We propose a switch-less method to build the Dragonfly topology. Costly high-radix switches are eliminated while improving injection/local throughput and maintaining global throughput.
- The wafer-based interconnection architecture is a whole new frontier. We scale out existing 2D-mesh-on-wafer to large-scale high-radix network-of-wafers, achieving much better scalability than any existing wafer-based network.
- We introduce a simple baseline minimal/non-minimal routing algorithm, and novel labeling and interconnection methods are used to reduce the VC number. Only one

*Corresponding author. Email: kaisheng@mail.tsinghua.edu.cn

additional virtual channel against traditional Dragonfly is required to achieve deadlock-free routing in the switchless Dragonfly.

- Similar approaches can be applied to other switch-based direct topologies, including but not limited to Slim Fly [1], PolarFly [2], and HyperX [27].

II. BACKGROUND & MOTIVATION

A. Wafer-Scale Integration

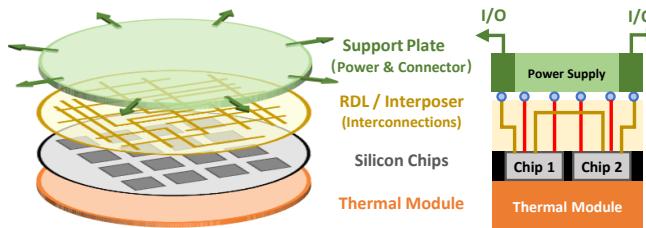


Fig. 1. Profile of the InFO-SoW integration technology. Connectors and power modules are solder-joined to the InFO wafer [17].

1) *Technology Introduction*: The traditional chip is implemented on a monolithic die, whose area is limited by the lithographic reticle (e.g. $26mm \times 33mm$ for ASML lithography [28, 29]). Advanced packaging technologies integrate multiple chiplets within a package thus breaking through the “Area Wall”. As shown in Fig. 1, by using *Integrated-Fan-Out-System-on-Wafer (InFO-SoW)* technology [16, 17], tens of known-good chiplets, as well as power and thermal modules, are integrated into a whole wafer (diameter $300mm$). Compared with the traditional system, the wafer-scale integration eliminates using substrates and PCBs while achieving higher integration/interconnection density and energy efficiency.

2) *Wafer-based interconnection*: In the past few years, many fantastic wafer-scale systems have emerged. The Tesla *DOJO* integrates 25 D1 dies with an area of $645 mm^2$ [15], resulting in a total silicon area exceeding $16,000 mm^2$ [26]. The *WSE-2* designed by *Cerebras* uses field stitching and achieves 850,000 cores (2.6 trillion transistors) on a wafer [23]. All existing systems adopt 2D-mesh as the on-wafer topology because it is implementation-friendly and scheduling-friendly. However, planar topologies are insufficient to scale out. For example, the *DOJO* supercomputer scales out the system by a larger 2D-mesh of wafers, resulting in a large diameter of up to 30 wafer-to-wafer hops [18]. To reduce the diameter, a centralized switch is used to connect all the edges of the enormous 2D-mesh, which leads to limited scalability and a fault-tolerance problem [24].

B. HPC Network Fabric

Almost all current HPC network architectures are based on switches. However, high-radix switches are very costly. A switch with $10\times$ the bisection bandwidth often costs about $100\times$ more [9]. An InfiniBand switch with 64 400G ports is priced over \$40,000 [7]. The latency and power consumption of high-performance switches cannot be ignored either. The port-to-port latency of an InfiniBand switch is up to

200ns [30], and the power consumption of the switch can be up to $1.7 KW$ [7]. Meanwhile, the single physical link limits the injection bandwidth and local bandwidth between two terminals. For example, two servers are connected to a 64-port 400G switch, whose total switching bandwidth is 25.6Tb/s, but the communication bandwidth between the two servers is only 400Gb/s.

TABLE I
EXTERNAL COMMUNICATION AND SWITCHING CAPABILITY
OF SEVERAL DATACENTER CHIPS

Category	Switching Chip			Computing Chip		
	NVSwitch [31]	Tofino2 [32]	Rosetta [33]	H100 [14, 34]	EPYC [35, 36]	DOJO D1 [15]
Physical Lanes	128	256	256	36	128	576
Data-rate (Gbps)	100	50	50	100	32	112
Throughput (Tb/s)	12.8	12.8	12.8	3.6	4	63

In recent years, with advances in high-speed wireline and packaging technologies, computing chips have become more powerful in NoC and IO throughput. As shown in TABLE I, the NVIDIA H100 chip has 36 lanes of 100G link (3.6Tb/s IO bandwidth in total) [31, 34], and the Tesla *DOJO* D1 chip has 576 lanes of 112G-SerDes (63Tb/s IO bandwidth in total)[15]. The total external bandwidth and NoC throughput of current high-end computing chips are already at the same level as mainstream switching chips and even exceed some high-end switches. Therefore, many interconnection networks, including TofuD [37], TPU [38], Wormhole [13], and *DOJO* [26], are using local interfaces and on-chip networks to scale out through direct (but low-radix) topologies. The injection/local bandwidth of these networks can be much higher than the limited bandwidth through a switch.

C. State-of-the-Art Interconnection Networks

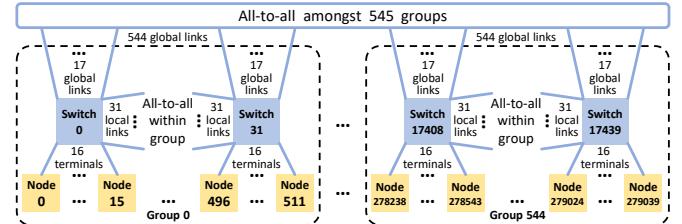


Fig. 2. The Dragonfly-based Slingshot topology. Switches are fully connected within groups, and groups are also all-to-all connected.

1) *Dragonfly*: Three supercomputers of Top5, Frontier (#1), Aurora (#2), and LUMI (#5), all adopt the Slingshot interconnect [39]. As shown in Fig. 2, the *Dragonfly* is the default topology for Slingshot [3, 33]. Several switches are fully connected between each other, forming a group, and multiple groups are also all-to-all connected.

2) *Diameter 2 Topologies*: Slim Fly [1] and PolarFly [2] are two topologies towards Moore bound. PolarFly leverages silicon-photonics co-package [6, 40] to achieve more than 96% of the theoretical peak with cost-effectiveness, which is a good example of innovating interconnection architecture through new technologies.

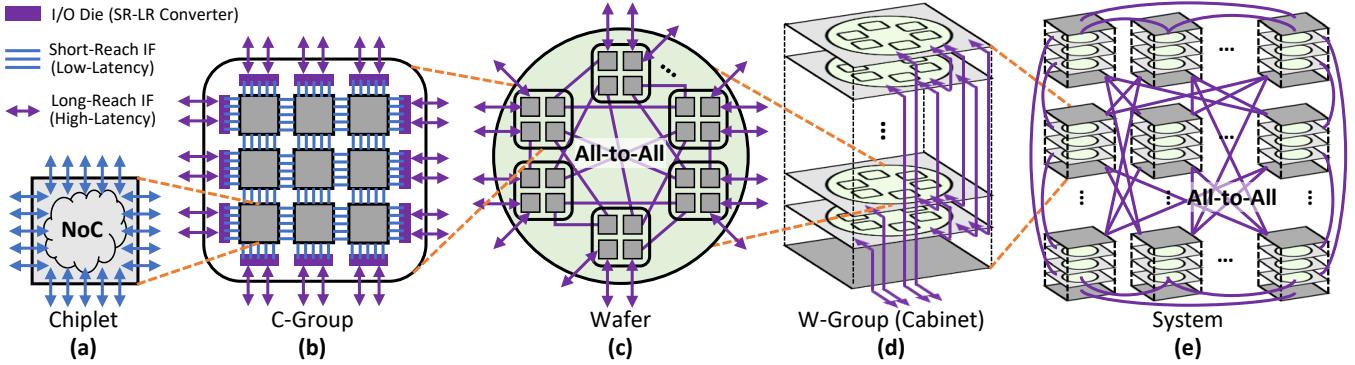


Fig. 3. Hierarchical architecture of the wafer-based switch-less Dragonfly. (a) A chiplet has an on-chip network and several short-reach low-latency interfaces used for interconnection. (b) Several chiplets are connected by a planar topology (2D-mesh as the default), forming a C-group. The remaining short-reach interfaces at the edges of the C-group are converted to long-reach interfaces for upper-level high-radix interconnection. (c)(d) Each wafer consists of several C-groups, and several wafers form a W-group. All C-groups in a W-group are fully-connected. (e) All the w-groups in the system are also fully-connected, just as the Dragonfly topology.

3) *HammingMesh*: People have noted that the local bandwidth of existing switch-based networks is under-provisioned while current high-end chips have abundant IO and switching capability [8], which is also the major motivation of this paper. Using local 2D-mesh networks and a global Fat-Tree, the *HammingMesh* provides high local bandwidth at a low cost with high scheduling flexibility.

III. ARCHITECTURE

The following symbols are used in the description:

- n the number of interfaces (IO ports) of a chiplet
- m the scale of the 2D-mesh of chiplets in a C-group
- k the number of external interfaces of a C-group
- a the number of C-groups in a wafer
- b the number of wafers in a W-group
- h the number of global ports of a C-group used to connect to other W-groups
- g the number of W-groups in the system
- N the total number of terminals/endpoints/chiplets

A. Topology Description

As shown in Fig. 3, the wafer-based switch-less Dragonfly architecture consists of 5 physical levels: chiplet, C-group, wafer, w-group, and system. Compared with the traditional switch-based Dragonfly [3], the chiplet is equivalent to the terminal (processor), the C-group is equivalent to the Dragonfly switch (router), and the W-group is equivalent to the Dragonfly router group.

1) *Chiplet*: As shown in Fig. 3(a), the chiplet is the smallest component of the system. Each chiplet has an on-chip network and n interconnection interfaces. The total IO ports, including memory and other peripherals, can be much more, but we focus only on the interconnection interfaces. These physical links are originally short-reach (e.g., PCIe [41] or XSR SerDes [42]) but have low latency and power consumption.

2) *C-Group*: Chiplets are clustered into a chiplet-group by an on-wafer planar network as shown in Fig. 3(b). We adopt 2D-mesh as the default topology in the C-group because it

is shortly-connected and implementation-friendly. A C-group consists of $m \times m$ chiplets. If each chiplet has $n/4$ ports at each edge, then a C-group has a total of $k = nm$ peripheral external ports. A C-group is equivalent to a switch in the traditional Dragonfly topology, with switching functionality realized through on-chip and intra-C-group interconnections. All the k short-reach (SR) external interfaces of the C-group are converted to long-reach (LR) interfaces (e.g., LR SerDes [42] and optic [40]) through conversion modules to support the high-radix connectivity of the upper level.

3) *Wafer & W-Group*: As shown in Fig. 3(c)(d), each wafer consists of a C-groups, and each W-group consists of b wafers. All ab C-groups in a W-group are fully connected: each C-group connects to every other $a - 1$ C-groups on the same wafer and every other $a(b - 1)$ C-groups on the other wafers. It is feasible for $a = 1$, then a whole wafer is a C-group, and there is no on-wafer all-to-all interconnection. When $a > 1$, due to the wiring distance limitation, the logical on-wafer all-to-all connections are implemented off-wafer physically, which is further illustrated in Sec. III-E. The W-group is equivalent to the group with ab switches in the traditional Dragonfly topology. Due to the ultra-high density of wafer-scale integration, one cabinet can hold an entire group that occupies a dozen cabinets in the traditional datacenter.

4) *System*: The entire system has g W-groups. As shown in Fig. 3(e), all W-groups are also fully connected: each connects to the other $g - 1$ W-group by at least one link. Subtracting $ab - 1$ interfaces used for local intra-W-group connections, the maximum number of global ports of a C-group is $h = k - ab + 1$, and the total number of W-groups in the system is $g = abh + 1$.

B. Analysis

1) *Scalability*: The total number of terminals (chiplets) in the wafer-based switch-less Dragonfly network described in Sec. III-A is:

$$N = abm^2 \times g = abm^2[ab(mn - ab + 1) + 1]. \quad (1)$$

Using a very small configuration $(a, b, m, n) = (2, 4, 2, 6)$, the total chiplet number can reach 1K. The scale of the traditional Dragonfly network is bounded by the switch radix. However, in the switch-less Dragonfly, the functionality of the switch is realized by the network-of-chiplet in the C-group; therefore, the network scale can be very huge. Nevertheless, the scalability of the switch-less Dragonfly is constrained by two main factors:

- **The physical scale of the wafer.** The maximum number of terminals (chiplets) that can be integrated within a C-group is limited by the area of the wafer (diameter 300mm). With current technologies, a wafer can fit more than 64 server chips [43], which is a considerable scale.
- **The performance of the chiplet network within the C-group.** Forwarding through the network is not as straightforward as forwarding through a non-blocking switch. Therefore, as the scale increases, the intra-C-group network may become the bottleneck due to the competition of the intra/inter-C/W-group traffic. Related issues are further discussed in the following subsections.

2) *Throughput*: If the bandwidth of all physical links is 1 flit/cycle, the global saturation throughput (injection rate) T_{global} of the switch-less Dragonfly can be estimated by the bisection bandwidth B_C and the topology [44]:

$$\begin{aligned} T_{\text{global}} &< \frac{2B_C}{N} = \frac{(g/2)^2 \times 2 \times 2}{N} \\ &= \frac{(mn - ab + 1)}{m^2} \text{ [flits/cycle/chip].} \end{aligned} \quad (2)$$

For the traditional Dragonfly, the global-local ratio $h/t \approx 1/2$ maintains load-balance because each packet traverses one global and two local channels [3]. In the switch-less Dragonfly, the global-local ratio can also be adjusted to about 1/2 when $ab \approx (2/3)k = (2/3)mn$, $m^2 \approx (1/2)ab$. In this case, the theoretical global throughput limit in Equation (2) reaches 1 flit/cycle/chip, the same as the traditional Dragonfly. Therefore, a reasonable configuration to achieve both globally load-balance and high-throughput is:

$$\begin{cases} n = 3m, \\ ab = 2m^2, \end{cases} \quad (3)$$

As for local throughput, the injection rate in the switch-based Dragonfly is bounded by the single physical link between the chip and the switch (1 flit/cycle/chip). In the switch-less Dragonfly, chiplets in the C-group are connected through a network with multiple physical links, thus can achieve higher local throughput. The local intra-W-group saturation injection rate T_{local} can be estimated as Equation (4):

$$T_{\text{local}} < \frac{(ab/2)^2 \times 2 \times 2}{abm^2} = \frac{ab}{m^2} = 2 \text{ [flits/cycle/chip]}, \quad (4)$$

twice as much as the throughput of the switch-based Dragonfly with the configuration of Equation (3). Since 2D-mesh is adopted in the C-group, the theoretical intra-C-group saturation throughput T_{cg} can be estimated as Equation (5):

$$T_{\text{cg}} < \frac{(nm/4) \times 2 \times 2}{m^2} = \frac{n}{m} = 3 \text{ [flits/cycle/chip]}, \quad (5)$$

which is also much better than the traditional switch-based Dragonfly. Therefore, the wafer-based switch-less Dragonfly can achieve higher injection/local throughput than the traditional switch-based Dragonfly. **However, bottlenecks can still exist due to the competition for the intra-C-group bandwidth and the imbalance of traffic distribution.** The total full-duplex bisection bandwidth B_{cg} of the 2D-mesh-in-C-group is

$$B_{\text{cg}} = \frac{nm}{2} = \frac{k}{2} \text{ [flits/cycle]}, \quad (6)$$

which is half of the k -port non-blocking switch (k flits/cycle). As a result, the inter-C-group traffic will compete with the intra-C-group traffic for the bandwidth provided by 2D-mesh. Therefore, to prevent the intra-C-group network from becoming the bottleneck under extreme traffic, a larger intra-C-group link bandwidth or higher-bandwidth topology, such as HexaMesh [45], is required. Higher intra-C-group bandwidth is easy and affordable to achieve by wafer-level integration. For example, the PCIe die-to-die interface can provide 1317 GB/s/mm die edge density (947 GB/s/mm² area density) on the wafer [41], much larger than traditional off-chip links.

3) *Diameter*: The diameter of the Dragonfly network consists of one global hop and two local hops. Therefore, in the worst case, a packet in the switch-less Dragonfly goes through four C-groups: source C-group, destination C-group, and two intermediate C-groups. Each 2D-mesh-based C-groups has a diameter of $2(m - 1)$ chiplet-to-chiplet hops. At the same time, each inter-C-group hop requires two additional SR-LR conversion hops. Therefore, the diameter (only off-chip hops are counted) of the wafer-based switch-less Dragonfly can be described as Equation (7):

$$D = \underbrace{H_g + 2H_l}_{\text{Dragonfly hops}} + \underbrace{(8m - 2)H_{sr}}_{\text{intra-C-group hops}}, \quad (7)$$

where H_g is a global hop, H_l is a local hop, H_{sr} is an on-wafer short-reach hop or a SR-LR hop. For comparison, the diameter of the traditional switch-based Dragonfly is $H_g + 2H_l + 2H_l^*$, where H_l^* is a hop from the terminal (processor) to the switch, whose typical cost is similar to a local hop. The rough cost of these hops is compared in TABLE II.

TABLE II
COMPARISON OF HOP COST [30, 33, 42, 46–51]

	H_g	H_l	H_{sr}	$H_{\text{on-chip}}$
Physical Medium	Optical Cable	Copper Cable	RDL	Metal Layer
Latency (ns)	150 + ToF	150 + ToF	~5	~1
Energy (pj/bit)	20+	20+	~2	~0.1

Ignoring protocol layers and considering only the physical layer, the latency of a short-reach hop generally comes from the PHY (e.g. PCIe and XSR SerDes [42]). When the transmission distance exceeds 100mm, forward error correction (FEC) must be introduced, significantly increasing the latency by tens of nanoseconds [46]. Above 10m, electro-optical (E-O) conversion is necessary, and time-of-flight (ToF) in fiber can no longer be ignored. For instance, the latency of a 10m

TABLE III
COMPARISON OF KEY SPECIFICATIONS BETWEEN THE SWITCH-LESS DRAGONFLY AND OTHER TOPOLOGIES

Interconnection Network	Chip-radix	SW-radix	#Switch	#Cabinet	#Processor	Cable Number/Length	T _{local}	T _{global}	Diameter
2D-Mesh & Switch (DOJO)	8	60	1	2	450	/	1.6	0.53	$2H_l^* + 18H_{sr}$
Three-Stage Fat-Tree	1		5120	608	65536	$N = 197K$	1	1	
Three-Stage Fat-Tree	4	64	20480	896	65536	$N = 786K$	4	4	$2H_g + 2H_l + 2H_l^*$
Three-Stage F-T (3:1 Taper)	4		14336	960	98304	$N = 655K$	4	4/3	
1-Plane Hx4Mesh	4		5120	352	65536	$N = 197K$	2	1/2	$2H_g + 2H_l + 2H_l^* + 4H_{sr}$
4-Plane Hx4Mesh	16	64	20480	640		$N = 786K$	8	2	
Co-Packaged PolarFly (p=32)	1	64	4033	504	129056	$N = 129K$	1	1	$2H_g + 2H_{sr}$
Dragonfly (Slingshot)	1	64	17440	2180	279040	$N=698K / L=154K \cdot E$	1(1)	1	$H_g + 2H_l + 2H_l^*$
Switch-less Dragonfly	12	/	0	545	279040	$N=419K / L=73K \cdot E$	3(2)	1	$H_g + 2H_l + 30H_{sr}$

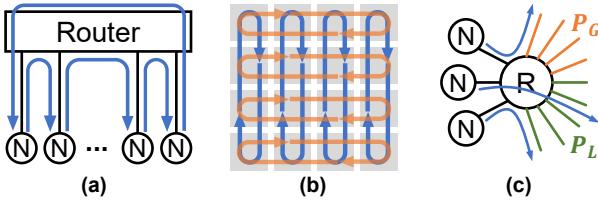


Fig. 4. Bottleneck of the switch-less Dragonfly in collective communication. (a) Ring AllReduce algorithm; (b) 2D algorithm for AllReduce within the 2D-mesh-based C-group; (c) Local/global link underutilization due to injection bandwidth limit.

optical link can easily be up to 200ns, which is approximately 40× higher than the on-wafer short-reach link. Besides the latency, the energy cost of long-distance hops is also much larger than the on-wafer hops. In the traditional Dragonfly, each packet must traverse these two local hops; however, in the switch-less Dragonfly, the number of short-reach hops is not always high.

4) *Collective Communication*: The throughput analysis in Sec. III-B2 is based on the assumption that the traffic is uniformly distributed across the bisection links. Under real workloads, the bottleneck of the switch can be more visible. As shown in Fig. 4, if the ring-based AllReduce algorithm is performed on a switch-based topology, the maximum bandwidth of the ring is 1 flit/chip/cycle, and the latency of N nodes is $O(N)$. On the 2D-mesh, as shown in Fig. 4(b), 2D algorithms can be performed to reduce the latency to $O(\sqrt{N})$ [52–54]. Besides, *bidirectional pipelined rings* can also be used to further reduce the latency [8]. For inter-router communication, the injection bandwidth can also become the bottleneck. As shown in Fig. 4(c), in a typical Dragonfly, terminals take up only a quarter of the switch ports (bandwidth). As a result, it is hard for a collective algorithm to fully utilize all the bandwidth, especially for small-scale jobs or hierarchical algorithms [55]. For the 2D-mesh-based C-group, the injection bandwidth is adequate thus the total off-C-group bandwidth can be fully utilized.

C. Comparison by Case Study

We compare the specifications of several typical HPC interconnection networks under specific configurations in TABLE III. All links are assumed to have the same bandwidth (normalized as 1), and T_{local} is the theoretical throughput of a subset of processors (e.g., a group of the Dragonfly

and a Hx4Mesh board of HammingMesh). All the topologies attempt to fully utilize the 64-port switch. We use a switch-less Dragonfly of the same scale as the Slingshot shown in Fig. 2 for comparison [33]. The configuration of the switch-less Dragonfly is as follows:

- $n = 12, m = 4$, Every chiplet has 3 external ports at each edge, and chiplets form the C-group by a 4×4 2D-mesh.
- $a = 4, b = 8$, Each wafer has 4 C-groups (64 chiplets), and eight wafers form a W-group (512 chiplets).
- Each W-group has a total of 544 off-W-group ports, so there are up to $g = 545$ W-groups and a total of $N = 279040$ chiplets.

1) *Bandwidth Trade-off*: The injection bandwidth can become the bottleneck for most existing switch-based topologies, including Fat-Tree, Dragonfly, and PolarFly. However, it is not easy to simply increase injection/ejection channels because available terminal ports are limited by the switch radix and network scale. Doubling the ports of a traditional endpoint results in doubling the requirement for the network building blocks. If we are willing to sacrifice the diameter and scalability, mesh/torus or DOJO-like topologies can provide adequate bandwidth for a small-scale system (hundreds of chips). Or, if we are willing to sacrifice the global throughput, the tapered Fat-Tree is a potential choice. Alternatively, the HammingMesh enables flexible configurations for different scales, diameters, bandwidth, and costs; however, it is still constrained by the Fat-Tree backbone. The *switch-less Dragonfly on wafer* provides another approach to directly build high-radix networks without switches. The intra-C-group and intra-W-group local throughput reaches 3 and 2 flits/cycle/chip, respectively, which is much higher than the traditional switch-based networks. With high-bandwidth on-wafer interconnects, the throughput can be even higher; at the same time, the global throughput is maintained. In summary, we achieve high injection/local/global bandwidth, low diameter, low cost, and high scalability, simultaneously.

2) *PolarFly*: The co-packaged PolarFly achieves the lowest diameter with integrated high-radix optical IO modules (OMs). PolarFly [2] does not discuss the in-package network in detail though it is critical for the overall performance. If there are multiple processors and OM in each package, besides all external IO ports, additional processor-to-OM and OM-OM ports inside the package are required. These intra-package

hops are regarded as short-reach hops, equivalent to on-wafer hops. With current technologies, it is hard to integrate 32 high-performance processors and multiple centralized high-radix IO modules in a single package. However, with a wafer-scale integration and a similar switch-less approach, the *switch-less PolarFly on wafer* promises to provide a more scalable and cost-effective solution.

3) *Cost*: The switch-less Dragonfly avoids using costly high-radix switches, thus significantly reducing the overall cost, including switches themselves and related power/cooling infrastructure. With wafer-scale integration, substrates and PCBs are also eliminated while providing affordable high-bandwidth interconnects. 1mm^2 silicon-on-wafer (< \$1) provides more than 800 GB/s [41] on-wafer bandwidth, much cheaper than the traditional inter-rack IOs and cables. Besides, wafer-scale integration also increases the density, thus reducing the physical size of the entire system. According to [56], one cabinet can host 64 blades, each consisting of 2 nodes; therefore, assuming 8 switches are at the top-of-rack (ToR), the Slingshot system requires 2180 cabinets in total. Besides, we also assume 32 core switches (except the ToR switch) can be placed in a cabinet for Fat-Tree-based networks. Short-reach 2D-mesh-on-PCB and co-package can increase the density, thus each cabinet is supposed to host 16 Hx4Mesh boards or 8 PolarFly co-packages (twice chips per cabinet). Conservative estimation suggests that the density of a single cabinet can increase by at least $4\times$ through wafer-scale integration [18, 22, 57]. As a result, the wafer-based switch-less Dragonfly only requires 545 cabinets (8 wafers per cabinet) to hold a system as large as the maximum Slingshot. If the Slingshot is flatly laid out in the datacenter at scale $E \times E$, the total cable length of inter-cabinet links can be estimated by cabinet-to-cabinet distance at $154K \cdot E$. For comparison, the local cable of switch-less Dragonfly is very short (intra-cabinet), and the total cable length is only $73K \cdot E$, less than half of the switch-based Dragonfly. Besides, all the terminal adapters and cables are also eliminated. In summary, the benefits of wafer-level integration and switch-less are all-encompassing, saving numerous datacenter building blocks.

D. Architecture Variations

1) *Small-Scale Networks*: HPC systems are not always very large. A single-chiplet C-group with only 12 external ports can be used to build a system of up to 333 chips (nodes). In this case, short-reach interfaces and conversion modules are not necessary. Besides, the inter-W-group interconnection can be eliminated; that is to say, the system is a single fully-connected W-group, whose diameter is only $H_l + (4m - 2)H_{sr}$.

2) *Topology Variations*: For many domain-specific workloads such as AI-training, the requirement for networks can be various [8]. Therefore, the topology is supposed to be adjustable. First, the parameters (a, b, m, n) of the switch-less Dragonfly can be changed to achieve unbalanced local/global bandwidth. Second, the topology of the intra-C-group network can be changed to HexaMesh [45] or other topologies. Third, C-groups within the W-group can be connected by a flatter

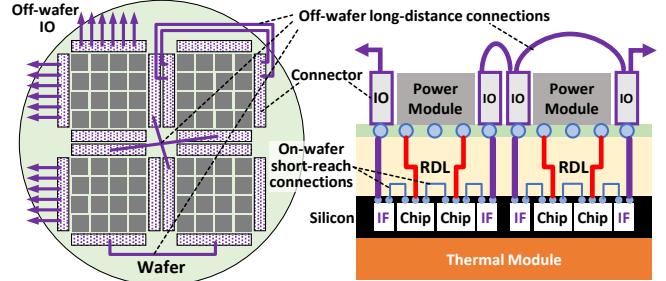


Fig. 5. Wafer-level long-distance connectivity. All the edge IOs of each C-group are fanned out, and the long-distance wafer-level logical links are connected off-wafer physically.

topology (e.g. 2D-flattened-butterfly), which consumes fewer local ports and is easier to lay out. Besides, other state-of-the-art topologies including but not limited to Slim Fly [1], PolarFly [2], and HyperX [27], can also be built by integrating endpoints under a switch through a planar topology on the wafer.

E. Wafer-Level Long-Distance Interconnection

As discussed above, when higher-radix topologies are used intra-C-group, or when there is more than one C-group on each wafer, wafer-level long-distance interconnections are required. However, due to the limitations of manufacturing, traditional technologies, such as field stitching [22, 58], only allow short-distance wiring within a lithographic reticle. The advanced mask stitching technologies [29, 59, 60] allows cross-reticle redistribution layer (RDL), and the reliability/quality of the wires across the stitching boundary is fine (negligible resistance contribution). However, though the stitched RDL promises to allow long-distance (> 100 mm) wiring, the high-speed electrical signals may not be able to travel that far. Therefore, other technologies such as on-wafer repeaters [61–63] are necessary.

Nevertheless, the switch-less Dragonfly is still practical without any physical on-wafer long-distance wires, because the inter-C-group interconnections do not require high-density on-wafer wiring. For a wafer with 9 C-groups (smaller C-groups do not require wafer-scale integration), there are only 36 inter-C-group wafer-level channels, which can be implemented off-wafer by standard packaging and interconnections. As shown in Fig. 5, each C-group is manufactured as a single unit with high-density short-reach on-wafer wiring, but all edge IOs, no matter whether for on-wafer or off-wafer interconnection, are fanned out to off-wafer electrical/optical connectors [64–66]. Then, the long-distance wafer-level logical links are connected off-wafer physically by backplane or cables. For the system discussed in Sec. III-C, the total number of IO channels for a wafer is 192, and a practical layout of the C-group is presented in Fig. 9.

IV. INTERCONNECTION AND ROUTING DESIGN

Routing is one of the core problems of interconnection networks. In traditional switch-based Dragonfly, the minimal path is unique, and all ports of a switch are equivalent and directly connected; thus, the routing is simple. Kim *et al.* achieved

deadlock-free minimal routing by two virtual channels (VCs) and non-minimal routing by three VCs [3]. However, in the switch-less Dragonfly, the switching functionality is realized by the distributed networks-on-chiplet; therefore, the ports of a C-group are non-equivalent, and channel dependencies among on-chip and off-chip networks can lead to potential deadlocks. Therefore, it is essential to illustrate the routing design of the entire network. In this section, we first introduce a simple baseline routing algorithm, and then present methods to reduce the number of virtual channels. Besides, the impact of intra-C-group networks is also discussed.

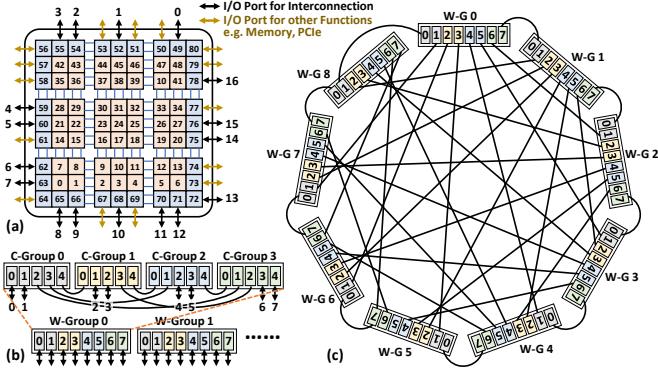


Fig. 6. Intra/inter-C/W-group interconnection. (a) Each in-C-group node has a unique label; k ports used for interconnection are also labeled. (b) C-groups are connected into multiple W-groups by local ports; The remaining ports are led out and re-labeled for global interconnection. (c) W-groups are fully connected.

A. Baseline Virtual-Channel-based Routing

The interconnection is shown in Fig. 6. In brief, the network is built in two steps: **1)** Label the port and fully connect C-groups into multiple W-groups. **2)** Relabel the remaining ports and fully connect all W-groups into a Dragonfly.

As shown in Algorithm 1, the minimal routing algorithm in the switch-less Dragonfly from the source node n_s of the source C-group C_s of the source W-group W_s to the destination node n_d of the destination C-group C_d of the destination W-group W_d is accomplished in seven steps: three

Algorithm 1 MINIMAL ROUTING IN SW-LESS DRAGONFLY

Input: Source: (W_s, C_s, n_s) ,

Destination: (W_d, C_d, n_d) ;

RWC(n_i, n_j): Routing within C-group from node n_i to n_j ;

Step 1: RWC(n_s, n_a). $n_a \in C_s$ is the node that has the local channel to C_b , which has the global channel to W_d .

Step 2: Traverse the local channel from n_a to $n_{b0} \in C_b$.

Step 3: RWC(n_{b0}, n_{b1}). $n_{b1} \in C_b$ is the node that has the global channel to $C_c \in W_d$.

Step 4: Traverse the global channel from n_{b1} to $n_{c0} \in C_c$.

Step 5: RWC(n_{c0}, n_{c1}). $n_{c1} \in C_c$ is the node that has the local channel to $C_d \in W_d$.

Step 6: Traverse the local channel from n_{c1} to $n_{d0} \in C_d$.

Step 7: RWC(n_{d0}, n_d).

inter-C-group routing steps and four intra-C-group routing steps. The non-minimal routing is similar to the minimal routing but with two additional inter-C-group steps and two additional intra-C-group steps at an intermediate W-group. Deadlock-free routing within 2D-mesh-based C-group can simply follow existing algorithms (*e.g.*, dimension-order and negative-first routing). Virtual channels (VCs) are used to avoid cross-C-group deadlocks in the switch-less Dragonfly. There are four kinds of situations for a minimal-routed packet in the C-group: source C-group C_s , intermediate C-group C_b , C_c , and destination C-group C_d . Therefore, we can simply use four VCs to avoid any cross-C-group deadlock by increasing the VC at each C-group. Similarly, six VCs can be used for deadlock-free non-minimal routing.

B. VC Number Reduction

When the VC number is limited, we also present methods to reduce the VC number. The basic idea is to achieve up*/down* deadlock-free routing [67] in a larger subnetwork beyond a C-group. If there is a valid up-first path for any source-destination pair within a W-group, the two VCs of the two C-groups can be merged into one VC. The up*/down* routing relies on proper labeling and interconnection. Definition 1 gives the type of all channels and ports. A feasible labeling method is stated in Property 1, which makes all ports consistently ordered and higher than the cores. The corresponding interconnection method is stated in Property 2, which organizes the different types of ports consistently from low to high: local ports to lower C-groups, global ports, and local ports to higher C-groups.

Definition 1. A physical or virtual channel from node (w_i, c_i, n_i) to node (w_j, c_j, n_j) is *up* if:

- $w_i < w_j$, or
- $w_i = w_j, c_i < c_j$, or
- $w_i = w_j, c_i = c_j, n_i < n_j$;

otherwise, the channel is *down*. A port P_s of a C-group or W-group is *up* if the channel from P_s to P_d is *up*; otherwise, the port is *down*.

Property 1. For the intra-C-group network,

- c1. \forall port-core pair (p, n) , \exists a down-only path from p to n (*i.e.* an up-only path from n to p).
- c2. \forall port-port pair with label $(i, j), i < j, \exists$ an up-only path from i to j (*i.e.* a down-only path from j to i), and

Property 2. As shown in Fig. 6(b), \forall global port of the C-group, all *down* local ports are at lower position, and all *up* local ports are at higher position.

As a result, any packet at the destination W-group has a valid up-first path to the destination: **1)** If the packet is at the core, it can reach the local port through an up-only path by Property 1(c1); and no matter the next local inter-C-group hop is *up* or *down*, it can then reach the destination core through a *down-only* path. **2)** As shown in Fig. 7, if the packet reaches the port node through a global channel, according to

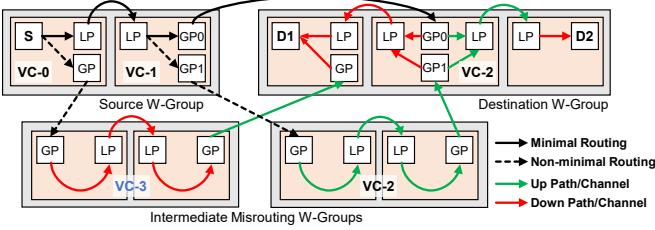


Fig. 7. Minimal/non-minimal routing and virtual channel assignment in the switch-less Dragonfly. S is the source node, and D1/D2 are the destination nodes.

Property 2 and Property 1(c2), there is a *down-only* or *up-only* path to the local port of the destination C-group; and then, according to Property 1(c1), there is a *down-only* path to the destination core. Therefore, one VC can be reduced for minimal/non-minimal routing at the destination W-group.

Similarly, any packet that reaches the intermediate W-group by non-minimal routing has a consistent path from the entering global port to the leaving global port: According to Property 2, if the leaving C-group is higher than the entering C-group, the path is *up-only*; otherwise, the path is *down-only*. As shown in Fig. 7, if we only allow non-minimal routing to a lower W-group from which there exists an *up-only* path to the destination W-group, then the routing among the intermediate and destination W-group can be merged with unified *up*/down** routing. If allowing non-minimal routing to other W-groups, one more VC is still required for the intermediate W-group.

In summary, the minimal routing requires three VCs: VC-0 and VC-1 for the source and intermediate C-groups of the source W-group, and VC-2 for the destination W-group. No more VC is required if only misrouting to a valid lower W-group; otherwise, one more VC-3 is required at the intermediate W-group.

C. Intra-C-group Networks

As stated in Property 1, two conditions for the intra-C-group network are required for *up*/down** routing. Various intra-C-group network architectures can meet the conditions by trading off performance and complexity.

The IO-router-based NoCs shown in Fig. 8(a) are adopted by many chips, including the EPYC [35, 36], TofuD [37], H100 [34], and TPU [68]. The advantages of the IO-router-based NoCs are the isolation of on/off-chip traffic and the simplification of intra-C-group interconnection. However, the IO router can become the bottleneck, and the chip-to-chip bandwidth does not scale with the chip scale. Fig. 8(a) shows a valid intra-C-group interconnection and labeling method for IO-router-based chiplets by four physical channels.

The mesh-based NoCs can provide a more scalable injection bandwidth. Many recent multi-chip systems, including the *Sapphire Rapids* [69], *Wormhole* [13], and *DOJO* [26], adopt such an architecture. Fig. 8(b) shows a labeling method consistent with the on-chip routing, and Fig. 8(c) shows another novel polar-system-based labeling method. Both two labeling methods meet the condition in Property 1 but are

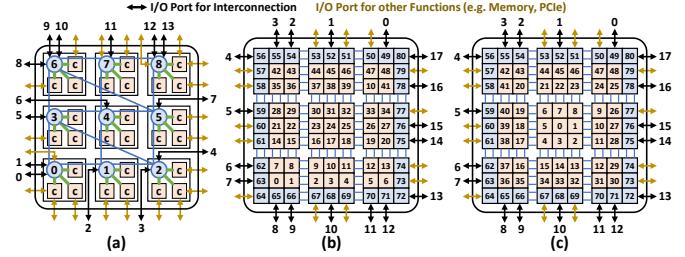


Fig. 8. Network-in-C-group architectures and the labeling. (a) IO-router-based: all interconnection ports are connected to one on-chip router; (b)(c) Mesh-based: interconnection ports are distributed at the edge of the NoC.

different in design detail. For example, router-less rings can be implemented on the polar-system-labeled NoCs to reduce the complexity and detour [70, 71]. A potential issue is the asymmetry of any such labeling method; however, since our labeling is software-based (the physical 2D-mesh is symmetric), it is possible to change the labeling method or mapping policy for different applications. More details are beyond the scope of this paper.

V. EVALUATION & DISCUSSION

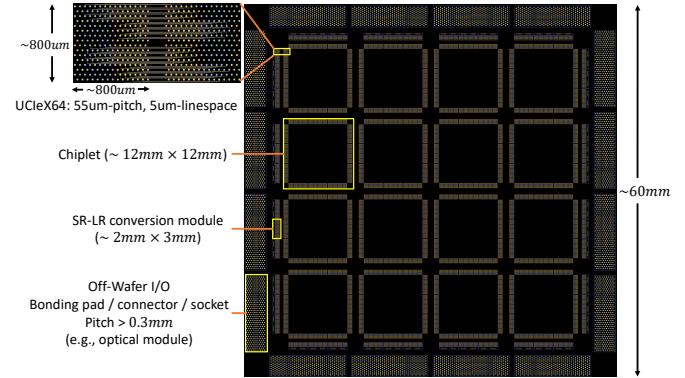


Fig. 9. Layout of PHYs, chiplets, and IO connectors of a C-group.

A. Methodology

1) *Layout*: To evaluate the feasibility of the implementation, we try to place and route a C-group on the wafer. The bump pitch and line space are assumed to be 55um and 5um on the wafer [16]. As shown in Fig. 9, the layout includes placement of PHYs, chiplets, and IO connectors. Assuming the C-group consists of 16 chiplets, each chiplet has 6 physical channels at each edge. In our layout, 128 lanes of UCIe (two 64× PHY [41]) are adopted at each on-wafer channel, achieving 4096 Gb/s/port intra-C-group short-reach bandwidth. 8 lanes of 112G SerDes (differential signal) are adopted at each off-C-group channel, achieving 896 Gb/s/port long-reach bandwidth [42, 47]. As a result, a C-group of 60mm × 60mm size leads out 1536 pairs of differential ports (~ 5500 IOs including the power and ground) in total. The total bisection and aggregation bandwidth of the on-wafer C-group is 12TB/s and 20.9TB/s, much larger than the highest-end switches. The layout also suggests that it is feasible

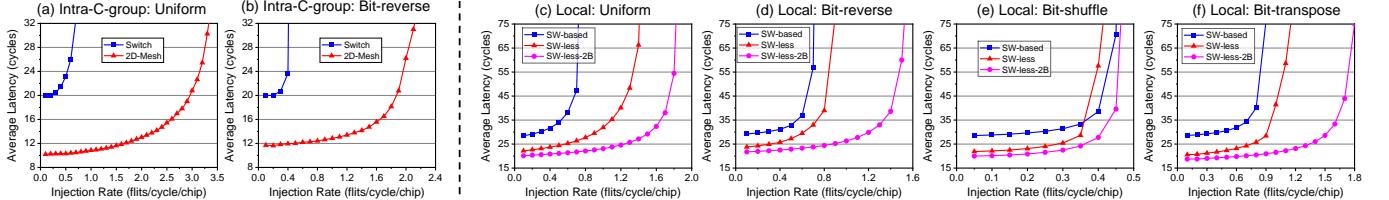


Fig. 10. (a-b) Intra-C-group (intra-switch) and (c-f) local (intra-Dragonfly-group) performance under different traffic patterns.

to achieve multiples of bandwidth on-wafer with advanced packaging and interface technologies.

TABLE IV
DEFAULT PARAMETERS

Parameter	Value
Packet Length	4 flits
Input Buffer Size	32 flits
Base Link Bandwidth	1 flit/cycle
Short-Reach Link Delay	1 cycle
Long-Reach Link Delay	8 cycles
Simulation Time	10000 cycles after 5000 cycles warming up

2) *Simulator*: CNSim [72] is used to evaluate the performance. The default parameters used in simulations are shown in TABLE IV. We do not set the long-reach link delay at the real value (hundreds of cycles), otherwise, the switch-less Dragonfly will always have a much lower latency due to the shorter diameter (3 v.s. 5 hops).

3) *Workloads*: The evaluations use three kinds of network workloads: (a) **Unicast traffic patterns**. The *uniform* and other permutation patterns [44], including bit-reverse, bit-shuffle, and bit-transpose, are evaluated. (b) **Adversarial traffic patterns**. We evaluate the *hotspot* traffic pattern, which conducts communications within four of all W-groups, and the *worst-case (WC)* traffic pattern, where each node in W-group W_i sends traffic to a random node in W-group W_{i+1} [3]. (c) **Collective traffic patterns**. We also evaluate the *ring-based AllReduce* traffic pattern, where each chip (process) i sends the $1/N$ segment to chip $(i + 1) \bmod N$ or sends two $1/2N$ segments to $(i - 1) \bmod N$ and $(i + 1) \bmod N$ [8, 55].

4) *Experiment Setup*: The baseline is the standard switch-based Dragonfly. A switch's terminal, local, and global ports are configured at 4:7:5 for radix-16 and 8:15:9 for radix-32. As a result, the total (group, chip) numbers are (41, 1312) for radix-16 and (145, 18560) for radix-32. For the switch-less Dragonfly, local and global ports are configured as the same number but no terminal ports. All nodes in a C-group of the switch-less Dragonfly are connected by a 2D-mesh with low-latency on-wafer links. The links between C-groups and W-groups are configured the same as the switch-based Dragonfly. As discussed in Sec. III-B2, the 2D-mesh with uniform link bandwidth has limited bisection bandwidth compared with a non-blocking switch. Therefore, we also evaluate the configuration of higher intra-C-group bandwidth (labeled as “2B/4B” for $2 \times 4 \times$ intra-C-group bandwidth). It is also important to note that all the switches are modeled

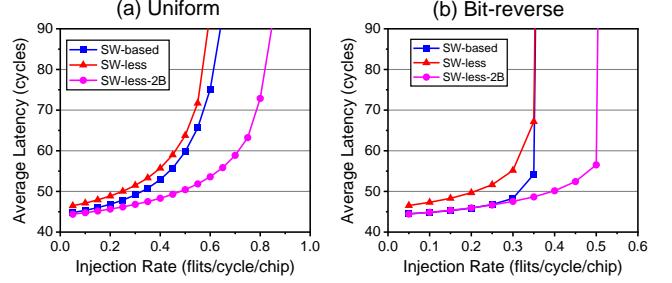


Fig. 11. Global performance under the uniform and bit-reverse traffic patterns.

as single ideal high-radix routers; however, they are actually also implemented by distributed networks-on-chip [33, 73]. The performance/energy overhead of the high-radix switches is underestimated in this paper.

B. Performance

1) *Local Throughput*: Rather than connecting to the switch by a single physical channel, the switch-less Dragonfly adopts a 2D-mesh within the C-group. As a result, the theoretical local throughput of the switch-less Dragonfly is more than 1 flit/cycle/chip. We evaluate the architecture by adopting a 2D-mesh of 2×2 chiplets with 2×2 on-chiplet network in the C-group (4×4 on-chip routers in total). The C-group has 12 external ports (7 for local and 5 for global, equivalent to the radix-16 switch); therefore, each W-group has 8 fully-connected C-groups (32 chips in total). As shown in Fig. 10(a), the saturation injection rate intra-C-group under uniform and bit-reverse traffic reaches 3.0 and 2.0 flits/cycle/chip, which is over 3× more than connecting to a switch. As for the intra-W-group throughput, although a traditional Dragonfly switch has 2× local ports than the terminal ports, the injection rate is still bounded by the single injection channel connecting to the switch. As shown in Fig. 10(c-f), except for the bit-shuffle pattern, the saturation injection rate intra-W-group can be 1.2 – 2× larger. With double on-wafer bandwidth, the performance can be even better. However, the performance is not improved if the bottleneck is the inter-C-group links rather than the intra-C-group links (e.g. bit-shuffle pattern shown in Fig. 10(e)). In summary, the switch-less Dragonfly achieves better local throughput without doubling the intra-C-group bandwidth.

2) *Global Throughput*: We evaluate the global performance of the same radix-16 network. The whole network has 1312 chips (5248 on-chip nodes) in total. As shown in Fig. 11(a), if the intra-C-group link bandwidth is the same as the

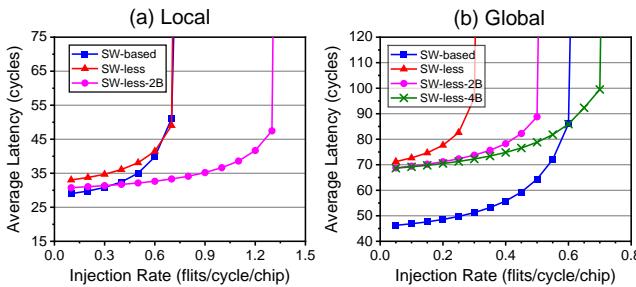


Fig. 12. Performance scalability under the uniform traffic.

local/global link bandwidth, the overall performance under uniform traffic for the switch-less Dragonfly is slightly worse than the switch-based Dragonfly due to the limited bisection bandwidth of the 2D-mesh-in-C-group. If the intra-C-group link bandwidth is doubled, the bottleneck on the bisection bandwidth is eliminated; thus, the switch-less Dragonfly performs much better than the traditional Dragonfly. For the bit-reverse traffic pattern shown in Fig. 11(b), the result is similar. For small-scale networks, the switch-less Dragonfly maintains the global performance with uniform bandwidth and achieves better performance with higher intra-C-group bandwidth.

3) *Scalability*: We also evaluate the scalability of the switch-less Dragonfly by simulating large-scale networks. It is important to note that the absolute value of the latency of the switch-based Dragonfly is greatly underestimated for easier comparison. We build a large-scale system of 18560 chips (radix-32). As shown in Fig. 12(a), the local performance of the large-scale switch-less Dragonfly is not as good as small-scale networks without doubling intra-C-group bandwidth. As shown in Fig. 12(b), the global performance of the uniform-bandwidth switch-less Dragonfly is severely constrained by the limited bisection bandwidth of the 2D-mesh-in-C-group. That is intuitive and inevitable since we have eliminated thousands of powerful switches with non-blocking switching capability. Higher intra-C-group bandwidth is critical for removing the bottlenecks for extreme global traffic. As shown in Fig. 12(b), the global throughput can be maintained or even improved after increasing the intra-C-group bandwidth. As we have analyzed and validated in Section III-B2, III-C3, and V-A1, it is feasible and affordable to achieve higher bandwidth on the wafer; or from another perspective, off-wafer bandwidth is reduced compared to the on-wafer bandwidth, just as the DOJO [24].

4) *Misrouting*: The minimal routing on the Dragonfly topology is insufficient for some unbalanced traffic; thus, non-minimal routing is required. We evaluate the non-minimal routing algorithm under the hotspot and worst-case traffic patterns. As shown in Fig. 13, the performance by minimal routing is poor because only 3/40 global links are used for the hotspot traffic, and only 1/40 global links are used for the worst-case traffic. Therefore, distributing traffic to more global channels by non-minimal routing can reduce congestion. The simulation results show that the saturation injection rate by non-minimal routing is tens of times larger than the minimal routing. As shown in Fig. 13(a), increasing the intra-C-group

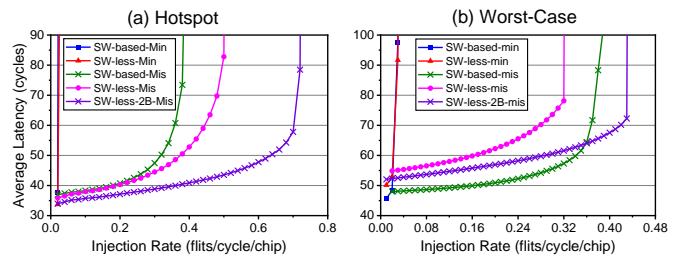


Fig. 13. Performance of the minimal and non-minimal routing under the hotspot and worst-case traffic patterns.

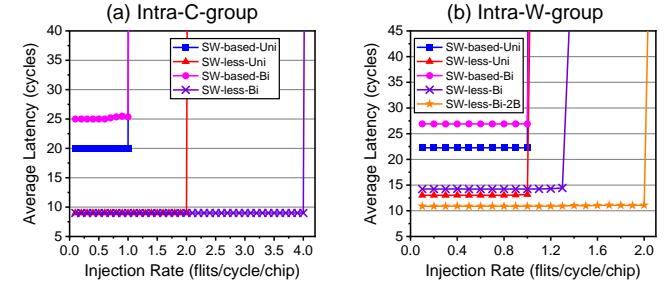


Fig. 14. Performance of ring-based AllReduce algorithm within C-group and W-group.

bandwidth can significantly improve the performance of the hotspot pattern because traffic congestion is also within the C-group.

5) *AllReduce Traffic*: We also evaluated the AllReduce traffic based on the unidirectional and bidirectional rings. As shown in Fig. 14(a), the saturation injection rate of the switch-based Dragonfly reaches 1 flit/cycle/chip for intra-C-group AllReduce. The bidirectional ring does not improve the performance but introduces congestion at the ejection port and leads to higher latency. Meanwhile, since there are four injection/ejection ports per chip in the switch-less Dragonfly, the saturation throughput can reach 2 and 4 flits/cycle/chip through the unidirectional and bidirectional rings. If considering the on-wafer bandwidth can be multiple times more, the expected performance can be even higher. As shown in Fig. 14(b), the performance of the intra-W-group AllReduce is bounded by the inter-C-group links. Without bidirectional rings, both the switch-based and switch-less Dragonfly reach the same throughput (1 flit/cycle/chip). With bidirectional rings, the switch-less Dragonfly can achieve a higher throughput of 1.3 flits/cycle/chip, but still lower than the theoretical value due to the competition on the intra-C-group networks. By doubling the intra-C-group bandwidth, the intra-C-group bandwidth bottleneck is eliminated, thus the performance of inter-C-group AllReduce can reach 2 flits/cycle/chip, twice that of the switch-based Dragonfly.

C. Power Consumption

Since the switching functionality is achieved by the intra-C-group network with numerous short-reach hops, it is not clear how the power consumption is affected. Considering modern switches have powerful software features, we evaluate the power consumption based on the energy per physical channel rather than directly comparing the chip power. As

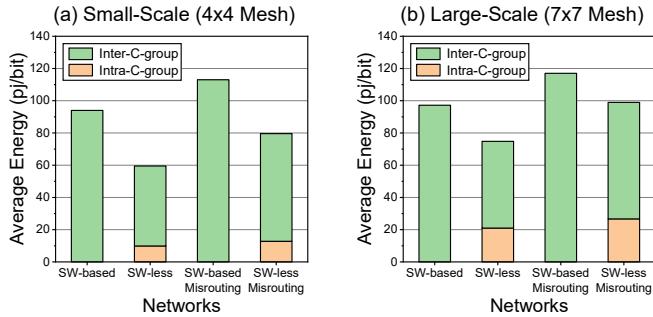


Fig. 15. Average energy consumption per data transmission of minimal/non-minimal routing for small-scale and large-scale Dragonfly.

shown in TABLE II, the energy consumption of H_l/H_g , H_{sr} , and $H_{on-chip}$ is estimated at 20, 2, and 0.1 pj/bit, respectively. For simplicity, we assume an intra-C-group hop takes 1pj/bit on average. Uniform traffic is performed on topologies of different scales, and the trace of each packet is collected. As shown in Fig. 15, the average energy consumption per data transmission is calculated based on the average hop count. For the small-scale Dragonfly, the energy overhead on the 4×4 2D-mesh-on-wafer is small compared with the energy reduction from eliminating switches. For large-scale Dragonfly, since the diameter of 2D-mesh-on-wafer is larger, the energy overhead can be significant, especially for non-minimal routing. However, high-radix switches are also based on NoCs [33, 73], which also introduce extra energy consumption. In conclusion, eliminating switches can reduce the total energy consumption for both small/large-scale networks and minimal/non-minimal routing.

VI. SUMMARY

Wafer-scale integration provides high-density, low-latency, and high-bandwidth connectivity among tens of chips, thus promising to support direct high-radix networks without high-radix switches. In this paper, we propose a scalable wafer-based interconnection architecture for large-scale supercomputers. By utilizing distributed high-bandwidth networks-on-chip-on-wafer, costly high-radix switches of the Dragonfly topology are eliminated while increasing local throughput and maintaining global throughput. We also introduce baseline and improved deadlock-free minimal/non-minimal routing algorithms with only one additional virtual channel against traditional Dragonfly. Discussion and evaluations show that the switch-less Dragonfly is implementable, cost-effective, high-performance and scalable. The proposed wafer-based switch-less approach can be applied to other switch-based direct topologies and is promising to power future large-scale supercomputers.

VII. ACKNOWLEDGMENTS

This work is partially supported by the Wafer-Scale Silicon-Optic Interconnected System (2022YFB2804100) and the National Natural Science Foundation of China (20211710187).

REFERENCES

- [1] M. Besta and T. Hoefler, “Slim fly: A cost effective low-diameter network topology,” in *SC14: International Conference for High Performance Computing, Networking, Storage and Analysis*. New Orleans, LA, USA: IEEE, Nov. 2014, pp. 348–359.
- [2] K. Lakhota, M. Besta, L. Monroe, K. Isham, P. Iff, T. Hoefler, and F. Petrini, “Polarfly: A cost-effective and flexible low-diameter topology,” in *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*. Dallas, TX, USA: IEEE, Nov. 2022, pp. 1–15.
- [3] J. Kim, W. J. Dally, S. Scott, and D. Abts, “Technology-driven, highly-scalable dragonfly topology,” in *2008 International Symposium on Computer Architecture*. Beijing, China: IEEE, Jun. 2008, pp. 77–88.
- [4] C. B. Stunkel, R. L. Graham, G. Shainer, M. Kagan, S. S. Sharkawi, B. Rosenburg, and G. A. Chochia, “The high-speed networks of the summit and sierra supercomputers,” *IBM Journal of Research and Development*, vol. 64, no. 3/4, pp. 3:1–3:10, May 2020.
- [5] S. K. Routray, A. Javali, L. Sharma, J. Gupta, and A. Sahoo, “The new frontiers of 800g high speed optical communications,” in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. Coimbatore, India: IEEE, Nov. 2020, pp. 821–825.
- [6] C. Minkenberg, R. Krishnaswamy, A. Zilkie, and D. Nelson, “Co-packaged datacenter optics: Opportunities and challenges,” *IET Optoelectronics*, vol. 15, no. 2, pp. 77–91, Apr. 2021.
- [7] “Nvidia mqm9700-ns2f quantum 2 ndr infiniband switch,” <https://store.nvidia.com/en-us/networking/store/product/mqm9700-ns2f/nvidia-quantum-2-ndr-infiniband-switch/>.
- [8] T. Hoefler, T. Bonato, D. De Sensi, S. Di Girolamo, S. Li, M. Hedges, J. Belk, D. Goel, M. Castro, and S. Scott, “Hammingmesh: A network topology for large-scale deep learning,” in *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*. Dallas, TX, USA: IEEE, Nov. 2022, pp. 1–18.
- [9] L. A. Barroso, U. Hözle, and R. Parthasarathy, *The Datacenter as a Computer: Designing Warehouse-Scale Machines*, 3rd ed. Cham, Switzerland: Springer, 2019.
- [10] L. Popa, S. Ratnasamy, G. Iannaccone, A. Krishnamurthy, and I. Stoica, “A cost comparison of datacenter network architectures,” in *Proceedings of the 6th International Conference*. Philadelphia Pennsylvania: ACM, Nov. 2010, pp. 1–12.
- [11] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, “The cost of a cloud: Research problems in data center networks,” *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 68–73, Dec. 2008.
- [12] O. Popoola and B. Pranggono, “On energy consumption

- of switch-centric data center networks,” *The Journal of Supercomputing*, vol. 74, no. 1, pp. 334–369, Jan. 2018.
- [13] D. Ignjatovic, D. W. Bailey, and L. Bajic, “The wormhole ai training processor,” in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2022, pp. 356–358.
- [14] A. C. Elster and T. A. Haugdahl, “Nvidia hopper gpu and grace cpu highlights,” *Computing in Science & Engineering*, vol. 24, no. 2, pp. 95–100, Mar. 2022.
- [15] T. C. Fischer, A. K. Nivarti, R. Ramachandran, R. Bharti, D. Carson, A. Lawrendra, V. Mudgal, V. Santhosh, S. Shukla, and T.-C. Tsai, “9.1 d1: A 7nm ml training processor with wave clock distribution,” in *2023 IEEE International Solid-State Circuits Conference (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2023, pp. 8–10.
- [16] Douglas Yu, “Tsmc packaging technologies for chiplets and 3d,” in *Proceedings of the 2021 IEEE Hot Chips (HCS)*, 2021.
- [17] S.-R. Chun, T.-H. Kuo, H.-Y. Tsai, C.-S. Liu, C.-T. Wang, J.-S. Hsieh, T.-S. Lin, T. Ku, and D. Yu, “Info_sow (system-on-wafer) for high performance computing,” in *2020 IEEE 70th Electronic Components and Technology Conference (ECTC)*. Orlando, FL, USA: IEEE, Jun. 2020, pp. 1–6.
- [18] B. Chang, R. Kurian, D. Williams, and E. Quinnell, “Dojo: Super-compute system scaling for ml training,” in *2022 IEEE Hot Chips 34 Symposium (HCS)*. Cupertino, CA, USA: IEEE, Aug. 2022, pp. 1–45.
- [19] S. Pal, J. Liu, I. Alam, N. Cebry, H. Suhail, S. Bu, S. S. Iyer, S. Pamarti, R. Kumar, and P. Gupta, “Designing a 2048-chiplet, 14336-core waferscale processor,” in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. San Francisco, CA, USA: IEEE, Dec. 2021, pp. 1183–1188.
- [20] S. Pal, D. Petisko, M. Tomei, P. Gupta, S. S. Iyer, and R. Kumar, “Architecting waferscale processors - a gpu case study,” in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. Washington, DC, USA: IEEE, Feb. 2019, pp. 250–263.
- [21] Cerebras, “Wafer-scale deep learning,” in *2019 IEEE Hot Chips 31 Symposium (HCS)*. Cupertino, CA, USA: IEEE, Aug. 2019, pp. 1–31.
- [22] G. Lauterbach, “The path to successful wafer-scale integration: The cerebras story,” *IEEE Micro*, vol. 41, no. 6, pp. 52–57, Nov. 2021.
- [23] S. Lie, “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)*. Cupertino, CA, USA: IEEE, Aug. 2022, pp. 1–34.
- [24] E. Talpes, D. Williams, and D. D. Sarma, “Dojo: The microarchitecture of tesla’s exa-scale computer,” in *2022 IEEE Hot Chips 34 Symposium (HCS)*. Cupertino, CA, USA: IEEE, Aug. 2022, pp. 1–28.
- [25] Ganesh Venkataraman, “Beyond compute - enabling ai through system integration,” Cupertino, CA, USA, Aug. 2022.
- [26] E. Talpes, D. D. Sarma, D. Williams, S. Arora, T. Kunjan, B. Floering, A. Jalote, C. Hsiong, C. Poorna, V. Samant, J. Sicilia, A. K. Nivarti, R. Ramachandran, T. Fischer, B. Herzberg, B. McGee, G. Venkataraman, and P. Banon, “The microarchitecture of dojo, tesla’s exascale computer,” *IEEE Micro*, vol. 43, no. 3, pp. 31–39, May 2023.
- [27] J. H. Ahn, N. Binkert, A. Davis, M. McLaren, and R. S. Schreiber, “Hyperx: Topology, routing, and packaging of efficient large-scale networks,” in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*. Portland Oregon: ACM, Nov. 2009, pp. 1–11.
- [28] “Mask / reticle - wikichip,” <https://en.wikichip.org/wiki/mask>.
- [29] P. K. Huang, C. Y. Lu, W. H. Wei, C. Chiu, K. C. Ting, C. Hu, C. Tsai, S. Y. Hou, W. C. Chiou, C. T. Wang, and D. Yu, “Wafer level system integration of the fifth generation cowos®-s with high performance si interposer at 2500 mm²,” in *2021 IEEE 71st Electronic Components and Technology Conference (ECTC)*. San Diego, CA, USA: IEEE, Jun. 2021, pp. 101–104.
- [30] M. R. S. Katebzadeh, P. Costa, and B. Grot, “Evaluation of an infiniband switch: Choose latency or bandwidth, but not both,” in *2020 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. Boston, MA, USA: IEEE, Aug. 2020, pp. 180–191.
- [31] A. Ishii and R. Wells, “The nvlink-network switch: Nvidia’s switch chip for high communication-bandwidth superpods,” in *2022 IEEE Hot Chips 34 Symposium (HCS)*. Cupertino, CA, USA: IEEE, Aug. 2022, pp. 1–23.
- [32] A. Agrawal and C. Kim, “Intel tofino2 – a 12.9tbps p4-programmable ethernet switch,” in *2020 IEEE Hot Chips 32 Symposium (HCS)*. Palo Alto, CA, USA: IEEE, Aug. 2020, pp. 1–32.
- [33] D. De Sensi, S. Di Girolamo, K. H. McMahon, D. Roweth, and T. Hoefer, “An in-depth analysis of the slingshot interconnect,” in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. Atlanta, GA, USA: IEEE, Nov. 2020, pp. 1–14.
- [34] J. Choquette, “Nvidia hopper h100 gpu: Scaling performance,” *IEEE Micro*, vol. 43, no. 3, pp. 9–17, May 2023.
- [35] S. Naffziger, K. Lepak, M. Paraschou, and M. Subramony, “2.2 amd chiplet architecture for high-performance server and desktop products,” in *2020 IEEE International Solid State Circuits Conference (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2020, pp. 44–45.
- [36] K. Troester and R. Bhargava, “Amd next generation “zen 4” core and 4th gen amd epyc™ 9004 server cpu,” in *2023 IEEE Hot Chips 35 Symposium (HCS)*. Palo Alto, CA, USA: IEEE, Aug. 2023, pp. 1–25.
- [37] Y. Ajima, T. Kawashima, T. Okamoto, N. Shida, K. Hirai, T. Shimizu, S. Hiramoto, Y. Ikeda, T. Yoshikawa, K. Uchida, and T. Inoue, “The tofu interconnect d,” in

- 2018 IEEE International Conference on Cluster Computing (CLUSTER).* Belfast: IEEE, Sep. 2018, pp. 646–654.
- [38] N. P. Jouppi, G. Kurian, S. Li, P. Ma, R. Nagarajan, L. Nai, N. Patil, A. Swing, B. Towles, C. Young, X. Zhou, Z. Zhou, and D. Patterson, “Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings,” in *2023 ACM/IEEE 50th Annual International Symposium on Computer Architecture (ISCA)*, 2023.
- [39] “November 2023 | top500,” <https://www.top500.org/lists/top500/2023/11/>.
- [40] P. Maniotis, L. Schares, B. G. Lee, M. A. Taubenblatt, and D. M. Kuchta, “Scaling hpc networks with co-packaged optics,” in *Optical Fiber Communication Conference (OFC) 2020*. San Diego, California: Optica Publishing Group, 2020, p. T3K.7.
- [41] “Universal chiplet interconnect express (ucie) specification revision 1.1,” Jul. 2023.
- [42] “Common electrical i/o (cei) - electrical and jitter interoperability agreements for 6g+ bps, 11g+ bps, 25g+ bps, 56g+ bps and 112g+ bps i/o,” Dec. 2022.
- [43] SkyJuice, “Monolithic sapphire rapids,” <https://www.angstromomics.com/p/monolithic-sapphire-rapids>, Sep. 2022.
- [44] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Amsterdam ; San Francisco: Morgan Kaufmann Publishers, 2004.
- [45] P. Iff, M. Besta, M. Cavalcante, T. Fischer, L. Benini, and T. Hoefer, “Hexamesh: Scaling to hundreds of chiplets with an optimized chiplet arrangement,” in *2023 60th ACM/IEEE Design Automation Conference (DAC)*. San Francisco, CA, USA: IEEE, Jul. 2023, pp. 1–6.
- [46] O. S. Sella, A. W. Moore, and N. Zilberman, “Fec killed the cut-through switch,” in *Proceedings of the 2018 Workshop on Networking for Emerging Applications and Technologies*. Budapest Hungary: ACM, Aug. 2018, pp. 15–20.
- [47] “Designware die-to-die 112g usr/xsr phy & die-to-die controller,” Apr. 2021.
- [48] M. Y. Frankel, “Prospects for optical transceivers expanding to access, metro and long-haul,” in *Optical Fiber Communication Conference (OFC) 2021*. Washington, DC: Optica Publishing Group, 2021, p. Tu5A.2.
- [49] Davide Tonietto, “Energy efficiency in serial links,” in *The 30th IEEE Hot Interconnects Symposium (HotI30)*, 2023.
- [50] J. Navaridas, M. Kynigos, J. A. Pascual, M. Luján, J. Miguel-Alonso, and J. Goodacre, “Understanding the impact of arbitration in mzi-based beneš switching fabrics,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 35, no. 2, pp. 338–348, Feb. 2024.
- [51] Y. Feng, D. Xiang, and K. Ma, “Heterogeneous die-to-die interfaces: Enabling more flexible chiplet interconnection systems,” in *56th Annual IEEE/ACM International Symposium on Microarchitecture*. Toronto ON Canada: ACM, Oct. 2023, pp. 930–943.
- [52] S. Kumar and N. Jouppi, “Highly available data parallel ml training on mesh networks,” Nov. 2020.
- [53] P. Luczynski, L. Gianinazzi, P. Iff, L. Wilson, D. De Sensi, and T. Hoefer, “Near-optimal wafer-scale reduce,” May 2024.
- [54] D. D. Sensi, T. Bonato, D. Saam, and T. Hoefer, “Swing: Short-cutting rings for higher bandwidth allreduce,” in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, 2024, pp. 1445–1462.
- [55] G. Feng, D. Dong, and Y. Lu, “Optimized mpi collective algorithms for dragonfly topology,” in *Proceedings of the 36th ACM International Conference on Supercomputing*. Virtual Event: ACM, Jun. 2022, pp. 1–11.
- [56] “Hewlett packard enterprise ushers in new era with world’s first and fastest exascale supercomputer “frontier” for the u.s. department of energy’s oak ridge national laboratory,” <https://www.hpe.com/us/en/newsroom/press-release/2022/05/hewlett-packard-enterprise-ushers-in-new-era-with-worlds-first-and-fastest-exascale-supercomputer-frontier-for-the-us-department-of-energys-oak-ridge-national-laboratory.html>.
- [57] Tesla, “Tesla ai day 2022,” Oct. 2022.
- [58] W. Flack and G. Flores, “Lithographic manufacturing techniques for wafer scale integration,” in *[1992] Proceedings International Conference on Wafer Scale Integration*. San Francisco, CA, USA: IEEE Comput. Soc. Press, 1992, pp. 4–13.
- [59] S. Y. Hou, W. C. Chen, C. Hu, C. Chiu, K. C. Ting, T. S. Lin, W. H. Wei, W. C. Chiou, V. J. C. Lin, V. C. Y. Chang, C. T. Wang, C. H. Wu, and D. Yu, “Wafer-level integration of an advanced logic-memory system through the second-generation cowos technology,” *IEEE Transactions on Electron Devices*, vol. 64, no. 10, pp. 4071–4077, Oct. 2017.
- [60] S. Y. Hou, C. H. Lee, T.-D. Wang, H. C. Hou, and H.-P. Hu, “Supercarrier redistribution layers to realize ultra large 2.5d wafer scale packaging by cowos,” in *2023 IEEE 73rd Electronic Components and Technology Conference (ECTC)*. Orlando, FL, USA: IEEE, May 2023, pp. 510–514.
- [61] Y. Han, H. Xu, M. Lu, H. Wang, J. Huang, Y. Wang, Y. Wang, F. Min, Q. Liu, M. Liu, and N. Sun, “The big chip: Challenge, model and architecture,” *Fundamental Research*, p. S2667325823003709, Dec. 2023.
- [62] B. Vaisband and S. S. Iyer, “Communication considerations for silicon interconnect fabric,” in *2019 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP)*. Las Vegas, NV, USA: IEEE, Jun. 2019, pp. 1–6.
- [63] S. Chen, S. Pal, and R. Kumar, “Waferscale network switches,” in *2024 ACM/IEEE 51th Annual International Symposium on Computer Architecture (ISCA)*, 2024.
- [64] M. Wade, E. Anderson, S. Ardalani, P. Bhargava, S. Buchbinder, M. L. Davenport, J. Fini, H. Lu, C. Li, R. Meade, C. Ramamurthy, M. Rust, F. Sedgwick, V. Stojanovic,

- D. van Orden, C. Zhang, C. Sun, S. Y. Shumarayev, C. O'Keeffe, T. T. Hoang, D. Kehlet, R. V. Mahajan, M. T. Guzy, A. Chan, and T. Tran, “Teraphy: A chiplet technology for low-power, high-bandwidth in-package optical i/o,” *IEEE Micro*, vol. 40, no. 2, pp. 63–71, Jan. 2020.
- [65] H. Hsia, S. P. Tai, C. S. Liu, C. W. Tseng, S. Lu, Y. Wu, C. C. Chang, J. Wu, K. C. Yee, C. Y. Wu, C. H. Tung, and D. C. Yu, “Integrated optical interconnect systems (iois) for silicon photonics applications in hpc,” in *2023 IEEE 73rd Electronic Components and Technology Conference (ECTC)*. Orlando, FL, USA: IEEE, May 2023, pp. 612–616.
- [66] C. Kopp, S. Bernabé, B. B. Bakir, J. Fedeli, R. Orobouchouk, F. Schrank, H. Porte, L. Zimmermann, and T. Tekin, “Silicon photonic circuits: On-cmos integration, fiber optical coupling, and packaging,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 17, no. 3, pp. 498–509, May 2011.
- [67] M. Schroeder, A. Birrell, M. Burrows, H. Murray, R. Needham, T. Rodeheffer, E. Satterthwaite, and C. Thacker, “Autonet: A high-speed, self-configuring local area network using point-to-point links,” *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 8, pp. 1318–1335, Oct./1991.
- [68] T. Norrie, N. Patil, D. H. Yoon, G. Kurian, S. Li, J. Laudon, C. Young, N. Jouppi, and D. Patterson, “The design process for google's training chips: Tpuv2 and tpuv3,” *IEEE Micro*, vol. 41, no. 2, pp. 56–63, Mar. 2021.
- [69] N. Nassif, A. O. Munch, C. L. Molnar, G. Pasdast, S. V. Lyer, Z. Yang, O. Mendoza, M. Huddart, S. Venkataraman, S. Kandula, R. Marom, A. M. Kern, B. Bowhill, D. R. Mulvihill, S. Nimmagadda, V. Kalidindi, J. Krause, M. M. Haq, R. Sharma, and K. Duda, “Sapphire rapids: The next-generation intel xeon scalable processor,” in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2022, pp. 44–46.
- [70] S. Liu, T. Chen, L. Li, X. Feng, Z. Xu, H. Chen, F. Chong, and Y. Chen, “Imr: High-performance low-cost multi-ring nocs,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 6, pp. 1700–1712, 2016.
- [71] F. Alazemi, A. AziziMazreah, B. Bose, and L. Chen, “Routerless network-on-chip,” in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. Vienna: IEEE, Feb. 2018, pp. 492–503.
- [72] Y. Feng, Y. Wei, D. Xiang, and K. Ma, “Evaluating chiplet-based large-scale interconnection networks via cycle-accurate packet-parallel simulation,” in *2024 USENIX Annual Technical Conference (USENIX ATC 24)*. Santa Clara, CA: USENIX Association, Jul. 2024, pp. 731–747.
- [73] J. H. Ahn, Y. H. Son, and J. Kim, “Scalable high-radix router microarchitecture using a network switch organization,” *ACM Transactions on Architecture and*