A Case for Bufferless Routing in On-Chip Networks

Thomas Moscibroda Onur Mutlu
Microsoft Research
{moscitho, onur}@microsoft.com

***Abstract*—Buffers in on-chip networks consume significant energy/power, occupy chip area, and increase design complexity. In this paper, we make a case for a new approach to designing on-chip interconnection networks that eliminates the need for buffers for routing or flow control. Our preliminary evaluations show that routing without buffers significantly reduces the energy consumption of the on-chip network, while providing similar performance to that of existing buffered routing algorithms at low network utilization. We conclude that bufferless routing can be an attractive and energy-efficient interconnect design option for on-chip networks where network utilization is relatively low.**

I. Introduction

Interconnection networks are commonly used to connect different computing components [9]. With the arrival of chip multiprocessor systems, on-chip interconnection networks have started to form the backbone of communication between cores and cores and memory within a microprocessor chip [24], [19], [14], [4]. As power/energy consumption has already become a limiting constraint in the design of high-performance processors [11] and future on-chip networks in many-core processors are estimated to consume hundreds of watts of power [4], simple energy- and area-efficient on-chip interconnection network designs are especially desirable.

Previous on-chip interconnection network designs commonly assumed that each router in the network needs to contain buffers to buffer the packets (or flits) transmitted within the network. While buffering within each router improves the bandwidth efficiency in the network,[1] it has several disadvantages. First, buffers consume significant energy/power: they consume dynamic energy when read/written and static energy even when they are not occupied. Second, having buffers increases the complexity of the network design because logic needs to be implemented to place packets into and out of buffers. Third, buffers can consume significant chip area: even with a small number (64) of total buffer entries per node where each buffer can store 32 bytes of data, a network with 64 nodes requires 128KB of buffer storage. Energy consumption and hardware storage cost of buffers will increase as the number of network nodes increases in future chips.

In this paper, we propose to eliminate buffers in the design of on-chip networks to improve both energy- and area-efficiency. The basic idea of "bufferless routing" is to *always route a packet (or a flit) to an output port* regardless of whether or not that output port results in a lower distance to the destination of the packet. In other words, packets are deflected [2] or "misrouted" [9] by the router to a different output port if an output port that reduces the distance to the destination node is not available. Bufferless routing has also been called "hot-potato" routing in network theory [3], alluding to the scenario that the router immediately needs to pass the potato (i.e. the packet) on to some other router as the potato is too hot to keep (i.e. buffer).

We evaluate a set of simple and practical bufferless routing algorithms, and compare them against baseline buffered algorithms in terms of on-chip energy consumption and latency. We find that bufferless routing can yield substantial reductions in energy consumption, while incurring little extra latency (versus buffered algorithms) if the average injected traffic into the network is low, i.e. below the network saturation point.

II. Why Could It Work?

At first thought, the idea of completely eliminating buffers in on-chip interconnection networks might appear audacious, since it is clear that this will result in an increase in average packet latencies and a decrease in achievable network throughput compared to buffered routing schemes. Nonetheless, the approach could be suitable for on-chip networks, as we describe below.

Intuitively, bufferless deflection routing works well when network utilization is low. A packet is deflected only if a collision occurs in a router, i.e., if multiple packets arrive at a router at the same time, and if not all of these packets can be sent in a productive direction.[2] If there are only few packets simultaneously in the network, the number of collisions is low. Hence, most packets make fast forward progress and can be routed to their destination without being frequently deflected.

For larger traffic volumes, the fundamental effect of removing buffers is a *reduction of the total available bandwidth* in the network. In a buffered network, a packet waits idle in some buffer until it can be routed in a productive direction and therefore does not unnecessarily consume link bandwidth while it is buffered. In contrast, in a bufferless network all packets *always* consume link bandwidth because, in effect, links act like "buffers" for the packets. Therefore, beyond a certain packet injection rate into the network, bufferless routing algorithms will fail, while good buffered routing algorithms can still perform well. More precisely, the network saturation throughput $\Theta(B'less)$ of bufferless routing is less than the saturation throughput of buffered routing $\Theta(B)$.

The critical questions that determine the potential usefulness of bufferless routing in on-chip interconnection networks are therefore 1) how much energy reduction can be achieved by thus eliminating buffers, 2) how large is the gap between $\Theta(B'less)$ and $\Theta(B)$, and how well does bufferless routing perform compared to buffered routing at injection rates below $\Theta(B'less)$, and 3) how common are the realistic situations in which an interconnection network is operated at a traffic injection rate below $\Theta(B'less)$?

Our results in this paper show that the answers to the first two questions are very promising for bufferless routing in on-chip networks. Regarding the third, many on-chip interconnection networks are observed to be operating at relatively low packet injection rates [16], [15], which are significantly below their peak throughput. For instance, the L1 miss rate is typically below 10%, which, in a chip multiprocessor with a distributed shared L2 cache, results in very low packet injection rates for the network connecting L1 caches and L2 cache banks [7]. Hence, bufferless routing, which performs well at low packet injection rates can be a promising approach for on-chip networks that primarily operate at low utilization.

[1] This is because buffering reduces the number of dropped packets or "misrouted" packets [9], i.e. those that are sent to a less desirable destination port. Hence, buffering reduces the wasted network bandwidth.

[2] A productive direction is a direction (or output port) that brings the packet closer to its destination [21]. A non-productive direction brings the packet further away from its destination.

III. On-Chip Bufferless Routing

A. Overview

Since bufferless routers cannot store packets in transit, all packets that arrive at a router must immediately be forwarded to an adjacent router. A processor can safely inject a packet into its router when at least one incoming link (from other routers) is free. When there is contention between multiple packets that are destined for a particular direction, only one of these packets is actually sent to the corresponding output port, whereas the others are sent to other, undesirable output ports, i.e., they are "deflected." In contrast, traditional routing algorithms would temporarily store such packets in a buffer within the router. The idea in bufferless routing is that deflected packets will eventually reach their destination, and that the total extra latency due to the detours resulting from deflections is not too high.

Note that this kind of bufferless routing can be used on every network topology that satisfies the following two constraints: Every router 1) has an equal number of input and output ports towards other routers, and 2) is reachable from every other router. Many important topologies such as Meshes, Tori, Hypercubes, or Trees satisfy these criteria. However, bufferless routing cannot easily be applied to networks with directed links, such as the Butterfly network, as a deflected packet may no longer be able to reach its destination [9].

B. Algorithms

We start by presenting a very simple, easily implementable algorithm. All our algorithms are described for the 2-D Mesh topology, but they can be generalized to other topologies.

Bufferless Dimension-Order Oldest-First Routing (BL-DO-OF): This algorithm is the bufferless analogue of the classic (buffered) dimension-order routing algorithm. If there is at least one incoming packet in a cycle, the router routes these packets as shown in Algorithm 1.

Algorithm 1 BL-DO-OF Routing:

Arbitration: The packets are prioritized in decreasing order of age (oldest-first).

Routing: A packet is directed to a *free* output port in the following order of priority:

1: a free, productive output port in x-dimension
2: a free, productive output port in y-dimension
3: a free, non-productive output port in x-dim (deflection)
4: a free, non-productive output port in y-dim (deflection)

This algorithm tries to send a packet to an output port that brings the packet closer to its destination, prioritizing older packets over younger ones in case of contention. Once a packet is assigned to an output port, this port is no longer free for other packets.

The above description suggests that a bufferless routing algorithm is specified by two policies: contention resolution and routing. In BL-DO-OF, we use OF for the former, and DO for the latter. We have considered various different approaches for each of these policies.

Contention Resolution Policy: Instead of using age-based arbitration (Oldest-First (OF)), we have also evaluated the Closest-First (CF) policy, which prioritizes packets according to the distance to their destinations. The packet with the closest destination has the highest priority. Compared to OF, CF tends to yield better average latency as it prioritizes a packet that is already close to its destination, thus helping to quickly remove that packet from the network. The net effect is a decrease in the total number of packets that are in the network, and consequently, an increase in the available link bandwidth. On the other hand, a key advantage of the OF policy is that it prevents livelock, since at least the oldest packet in the network can always make progress. In addition to OF and CF, we have also evaluated Furthest-First, and Most-Deflections-First policies, but as they did not perform well in our evaluations, we do not present these results.

Routing Policy: One disadvantage of using the DO policy in bufferless routing is that some packets may be deflected unnecessarily, because DO strictly assigns the highest-priority packet to the x-dimension, even if this packet could also be routed productively in the y-dimension. Our Optimal-Local-Search (OLS) policy routes packets in such a way that *the maximum number of packets are sent to productive directions*. For instance, in a 2-D Mesh, consider two packets P_1 and P_2. The older packet, P_1, has productive directions in both x and y dimensions, whereas P_2 can only be productively routed in the x-dimension. The DO policy will route P_1 along the x-dimension, and may therefore have to deflect P_2, whereas the OLS policy can route both packets productively. In addition to DO and OLS, we have also evaluated policies that are based on two-phase ROMM algorithms [18]. However, in contrast with the corresponding baseline buffered algorithms, ROMM does not improve performance in the bufferless case.

C. Advantages and Disadvantages

While each of the above algorithms behaves differently, bufferless on-chip routing algorithms have common advantages and disadvantages.

No Buffers: This is the key advantage of our approach because it helps reduce both complexity and, as we show in our evaluation, energy consumption.

Purely Local and Simple Flow Control: Any buffered routing scheme inherently requires some kind of communication-based flow control mechanism or rate limitation in order to prevent the buffers in the routers from overflowing. Flow control is simpler in bufferless routing. A node safely injects a new packet into the network when at least one incoming link from another node is free, which can easily be detected locally without any need for communication between routers.

Absence of Deadlocks: Deflection-based bufferless routing is free of deadlock. Since the number of input and output ports are the same, every packet that enters a router is guaranteed to leave it.

Absence of Livelock: One of the potential challenges in bufferless routing is livelocks that could arise if a packet continuously gets deflected. In packet routing algorithms, however, preventing livelocks is easy if the oldest-first (OF) contention resolution mechanism is used. This is true because once a packet is the oldest in the network, it will eventually reach its destination. With the other congestion resolution policies, livelocks are theoretically possible, yet exceedingly unlikely, especially at low and moderate injection rates.

Adaptivity: Bufferless routing has the ability to be adaptive "on demand" to a certain degree. When there is no congestion, bufferless routing almost always routes packets along shortest paths. In congested areas, however, packets will be deflected away from local hotspots, which allows different links to be utilized and packets to be routed around congested areas. As such, bufferless routing automatically provides a form of adaptivity that buffered routing schemes must achieve using more sophisticated and potentially complex means.

For the same reason, bufferless routing can cope well with temporary bursty traffic. To a certain degree, the network itself (i.e., its links and routers) acts like a temporary buffer. In buffered routing, if a traffic burst occurs and many packets are sent to a router R, the buffers in routers in the neighborhood of R will gradually fill up. In bufferless routing, the packets

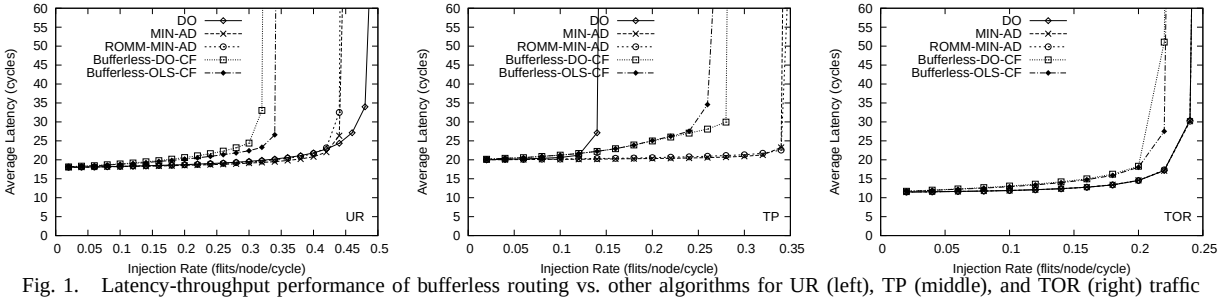

Fig. 1. Latency-throughput performance of bufferless routing vs. other algorithms for UR (left), TP (middle), and TOR (right) traffic

are continuously deflected in the extended neighborhood of R, until the burst completes and they can gradually reach R.

Disadvantages: Increased Latency & Reduced Bandwidth: The key downside of bufferless routing is that it can increase average packet latency because deflected packets will take a longer path to the destination than necessary. Also, bufferless routing effectively reduces the available network bandwidth as all in-network packets always consume link resources. Hence, the saturation throughput is reached at lower injection rates compared to conventional buffered routing. However, our evaluations show that for the kinds of low and moderate injection rates commonly seen in on-chip networks, the performance of bufferless routing is close to that of buffered routing. For such application domains, the advantages of bufferless routing can thus outweigh its disadvantages.

D. Other Issues

The above algorithms are simple and can be implemented using standard router implementation techniques, except that no buffers are needed, which reduces the complexity of the router. Also, without buffers, there is no need for virtual channels since every incoming flit is immediately sent out on a link. Bufferless routing can be modified to be used in conjunction with wormhole routing [8], to ensure that flits comprising a packet are required to be sent to the destination in strict succession on the same path [2], [20]. Due to space limitations, we do not discuss in detail how this is accomplished.

IV. Experimental Methodology

We evaluate the performance and energy-efficiency of bufferless routing using a cycle-accurate interconnection network simulator. We use the Orion energy model [23], assuming 100nm technology and 2GHz @ 1.2 V_{dd}. Link length of adjacent nodes is 2.5mm. Bufferless routing is compared to three different buffered routing algorithms in terms of average/maximum packet delivery latency, saturation throughput, and energy consumption: dimension-order routing (DO), minimal adaptive routing (MIN-AD) [12], and a minimal adaptive version of the ROMM algorithm [18] (ROMM-MIN-AD).

The modeled network configuration has a two-dimensional 8x8 mesh topology.[3] Each router has 5 input ports and 5 output ports, including the node injection/delivery ports. Each link is 32-byte wide and for our preliminary evaluations we assume that each packet has one flit. Packets are fixed length. Router latency is 2 cycles. The evaluated baseline buffered routing algorithms are simulated using infinite size buffers. However, their energy consumption is computed by assuming that each router has a small, 16-flit entry buffer in each of its input ports.

We use five different traffic patterns: uniform random (UR), transpose (TR), mesh-tornado (TOR), bit complement (BC), and hot-spot (HS) (see [21]). We present results for only the first three; the other two are qualitatively similar. Each simulation experiment is run for 100,000 packet injections.

[3]We choose the 2-D Mesh for our initial investigations because this topology has been implemented in the on-chip networks of several large-scale chip multi-processors or their prototypes [24], [19], [14].

V. Experimental Evaluation

A. Performance

Figure 1 shows the latency and throughput characteristics of each routing algorithm. Several observations are in order. First, at low packet injection rates (below 0.2 flits/cycle), bufferless routing provides similar average packet latencies as the baseline algorithms. For example, for the UR traffic pattern, bufferless routing increases the average packet latency by only 12% even with a large injection rate of 0.3. Second, with bufferless routing, the network saturates at a smaller injection rate than it does with buffered routing. This is because, at high network utilization, bufferless routing wastes significant network bandwidth by causing too many deflections. Even so, with the TOR traffic pattern, the saturation point of the bufferless network (inj. rate 0.22) is very close to that of a baseline buffered network (inj. rate 0.24). Third, bufferless routing provides adaptivity without requiring any explicit or global information about congestion. The non-adaptive DO baseline performs very poorly and saturates very early for the TP traffic pattern. MIN-AD and ROMM-MIN-AD algorithms significantly improve saturation throughput by providing explicitly adaptive routing. Bufferless routing performs in-between the non-adaptive DO and adaptive algorithms because it allows packets to avoid congested paths by deflecting them toward other parts of the network. We conclude that bufferless routing achieves good performance and can reduce congestion in on-chip networks if network utilization is not too high.

Figure 2 shows that the maximum packet latency increases with bufferless routing, which is expected as deflected packets take longer routes to their destinations. The degradation in maximum packet latency is severe when the closest-first (CF) contention resolution policy is used. This is because CF delays packets whose destinations are far away by prioritizing packets that have closer destinations. As a result, CF provides better average latency (shown in Figure 1), but higher maximum latency than the oldest-first (OF) policy. OF prioritization ensures that older packets are deflected less frequently, and thus reach their destination faster. We conclude that, for networks where maximum latency is a concern, the OF policy should be used in bufferless routing.

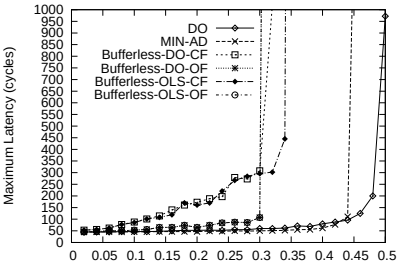

Fig. 2. Max. packet latency for UR

B. Energy Consumption

Figure 3 shows that bufferless routing significantly and consistently reduces energy consumption in the network compared to all buffered baselines for all injection rates before saturation. For UR traffic, the reduction in energy consumption ranges from 37% at low injection rates (0.02) to 25% at high injection rates (0.34). Thus, bufferless routing greatly improves

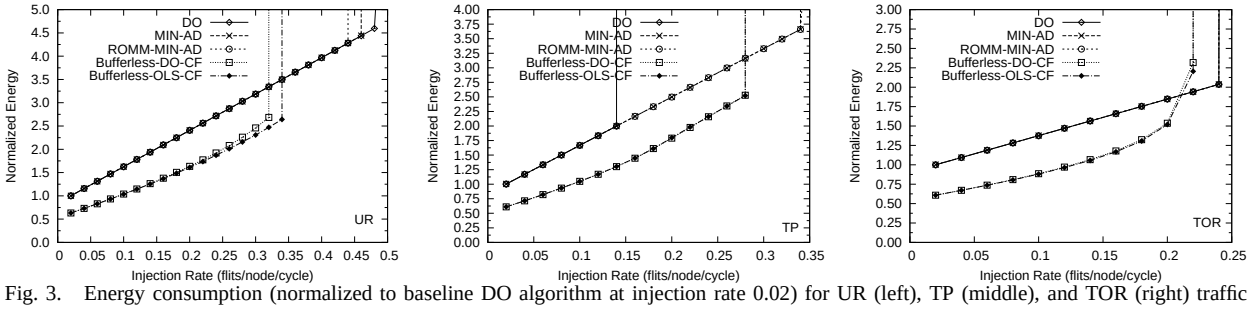

Fig. 3. Energy consumption (normalized to baseline DO algorithm at injection rate 0.02) for UR (left), TP (middle), and TOR (right) traffic

energy efficiency by eliminating the dynamic and static energy consumption due to buffers.

Figure 4(left) provides insight into the energy consumption behavior of bufferless routing by showing the breakdown of normalized network energy for different injection rates for DO and bufferless routing. At low injection rates, bufferless routing eliminates the energy consumed by buffers without significantly increasing the energy consumption in the links and the crossbar (including arbiter and routing energy). As the injection rate increases, bufferless routing causes an increase in link and crossbar energy consumption compared to the baseline because congestion in the network causes more deflections to happen and more routers and links to be utilized by the deflected packets. This is supported by the data shown in Figure 4(right), which shows that bufferless routing significantly increases link traversals as injection rate increases. However, the elimination of the buffer energy overcomes the increase in link and router energy, and therefore bufferless routing results in a net energy savings. We conclude that bufferless routing can effectively improve energy-efficiency in the on-chip network while preserving high network performance.

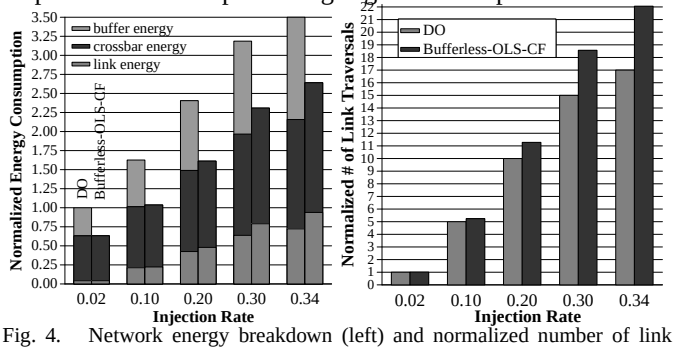

Fig. 4. Network energy breakdown (left) and normalized number of link traversals (right) for UR traffic (normalized to DO at injection rate 0.02)

VI. Related Work

Hot-potato routing was first described by Baran [3]. Several massively parallel machines, such as the HEP [22], the Tera [1], and the Connection Machine [13] have used deflection routing to connect off-chip components. These techniques are not disclosed in detail and, to our knowledge, have not been publicly evaluated in terms of energy consumption or performance. Moreover, their application was to large scale link-energy-dominated off-chip networks with large path diversity and long link latencies rather than on-chip networks with short link latencies and a lower fraction of link energy. The Chaos router [17] uses a form of deflection routing when a node is congested, however it still buffers packets in the router. Our main contributions beyond these techniques are: 1) we propose using bufferless routing in on-chip networks, which assumed buffered routing, 2) we provide preliminary energy and performance evaluations of bufferless on-chip routing, which were not available previously.

In the theory community, there has been a significant amount of work studying algorithms for hot-potato routing, e.g. [10], [2], [5]. However, most of these algorithms are *static*, i.e., all packets are injected at time zero, and the analysis examines the time needed to deliver the packets, which is not realistic in on-chip interconnection networks. One notable exception is the work in [6] which provides an analysis of a dynamic hot-potato routing algorithm and show that it achieves provable (probabilistic) worst-case delivery guarantees. However, the algorithm is not designed for efficient average case performance and its energy-efficiency remains unclear.

VII. Conclusions & Future Work

We make the case that bufferless routing could be used beneficially in on-chip interconnection networks. We show that, by getting rid of buffers completely, significant energy reductions can be achieved at modest performance loss compared to buffered routing algorithms, as long as the volume of injected traffic is not very high. We believe that bufferless routing algorithms are therefore a promising choice for on-chip interconnection networks that are known to run at below-peak throughput most of the time. In future work, we plan to optimize the performance of bufferless routing algorithms, while maintaining their simplicity and energy-efficiency.

References

[1] R. Alverson *et al.*, "The Tera computer system," in *ICS*, 1990.
[2] A. Bar-Noy *et al.*, "Fast deflection routing for packets and worms," in *PODC*, 1993.
[3] P. Baran, "On distributed communications networks," *IEEE Trans. on Communications*, Mar. 1964.
[4] S. Borkar, "Thousand core chips:A technology perspective," *DAC*, 2007.
[5] C. Busch *et al.*, "Hard-potato routing," in *STOC*, 2000.
[6] ——, "Routing without flow control," in *SPAA*, 2001.
[7] S. Cho and L. Jin, "Managing distributed, shared L2 caches through OS-level page allocation," in *MICRO*, 2006.
[8] W. J. Dally and C. L. Seitz, "The torus routing chip," *Distributed Computing*, vol. 1, pp. 187–196, 1986.
[9] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2004.
[10] U. Feige and P. Raghavan, "Exact analysis of hot-potato routing," in *STOC*, 1992.
[11] M. K. Gowan *et al.*, "Power considerations in the design of the Alpha 21264 microprocessor," in *DAC*, 1998.
[12] L. Gravano *et al.*, "Adaptive deadlock- and livelock-free routing with all minimal paths in torus networks," *IEEE TPDS*, vol. 12, no. 5, 1994.
[13] W. D. Hillis, *The Connection Machine*. MIT Press, 1989.
[14] Y. Hoskote *et al.*, "A 5 GHz mesh interconnect for a teraflops processor."*IEEE Micro*, vol. 27, no. 5, 2007.
[15] N. D. E. Jerger, L.-S. Peh, and M. H. Lipasti, "Circuit-switched coherence," in *NOCS*, 2008.
[16] J. Kim, J. D. Balfour, and W. J. Dally, "Flattened butterfly topology for on-chip networks," in *MICRO*, 2007.
[17] S. Konstantinidou and L. Snyder, "Chaos router: architecture and performance," in *ISCA*, 1991.
[18] T. Nesson and S. L. Johnsson, "ROMM: Routing on mesh and torus networks," in *SPAA*, 1995.
[19] J. D. Owens *et al.*, "Research challenges for on-chip interconnection networks," *IEEE Micro*, vol. 27, no. 5, 2007.
[20] A. Roberts and A. Symvonis, "A general method for deflection worm routing on meshes based on packet routing algorithms," *TPDS*, 1997.
[21] A. Singh, W. J. Dally, A. K. Gupta, and B. Towles, "GOAL: A load-balanced adaptive routing algorithm for torus networks," in *ISCA*, 2003.
[22] B. J. Smith, "A pipelined shared resource MIMD computer," in *ICPP*, 1978.
[23] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik, "Orion: a power-performance simulator for interconnection networks," in *MICRO*, 2002.
[24] D. Wentzlaff *et al.*, "On-chip interconnection architecture of the Tile processor," *IEEE Micro*, vol. 27, no. 5, 2007.