Designing Chip-Level Nanophotonic Interconnection Networks

Christopher Batten, *Member, IEEE*, Ajay Joshi, *Member, IEEE*, Vladimir Stojanović, *Member, IEEE*, and Krste Asanović, *Senior Member, IEEE*

***Abstract*—Technology scaling will soon enable high-performance processors with hundreds of cores integrated onto a single die, but the success of such systems could be limited by the corresponding chip-level interconnection networks. There have been many recent proposals for nanophotonic interconnection networks that attempt to provide improved performance and energy-efficiency compared to electrical networks. This paper discusses the approach we have used when designing such networks, and provides a foundation for designing new networks. We begin by briefly reviewing the basic silicon-photonic device technology before outlining design issues and surveying previous nanophotonic network proposals at the architectural level, the microarchitectural level, and the physical level. In designing our own networks, we use an iterative process that moves between these three levels of design to meet application requirements given our technology constraints. We use our ongoing work on leveraging nanophotonics in an on-chip title-to-tile network, processor-to-main-memory network, and dynamic random-access memory (DRAM) channel to illustrate this design process.**

***Index Terms*—Interconnection networks, multicore/manycore processors, nanophotonics, optical interconnect.**

I. Introduction

TODAY'S graphics, network, embedded, and server processors already contain multiple cores integrated onto a single chip, and this amount of integration will surely continue to increase over the next decade. Intra-chip and inter-chip communication networks are becoming critical components in such systems, affecting not only performance and power consumption, but also programmer productivity. Any future interconnect technology used to address these challenges must be judged on three primary metrics: bandwidth density, energy efficiency, and latency. Enhancements of current electrical technology might enable improvements in two metrics while sacrificing a third. Nanophotonics is a promising disruptive technology that can potentially achieve simultaneous improvements in all three metrics, and could therefore radically transform chip-level interconnection networks. Of course, there are many practical challenges involved in using any emerging technology including economic feasibility, effective system design, manufacturing issues, reliability concerns, and mitigating various overheads.

There has recently been a diverse array of proposals for chip-level network architectures that use nanophotonic devices to potentially improve performance and energy efficiency. These proposals explore different single-stage topologies from buses [1]–[5] to crossbars [1], [6]–[9] and different multistage topologies from quasi-butterflies [10]–[17] to tori [18]–[20]. Most proposals use different routing algorithms, flow control mechanisms, optical wavelength organizations, and physical layouts. This diversity can make it difficult to see relationships between different proposals and to identify promising directions for future network design.

In this paper, we describe our approach for designing nanophotonic interconnection networks, which is based on thinking of the design at the architectural level, the microarchitectural level, and the physical level. At each level, we illustrate design trade-offs and categorize previous work. Architectural-level design focuses on choosing the best logical network topology and routing algorithm. This early phase of design should also include a detailed design of an electrical baseline network to motivate the use of nanophotonic devices. Microarchitectural-level design considers which buses, channels, and routers should be implemented with electrical versus nanophotonic technology, and also explores how to best implement optical switching, techniques for wavelength arbitration, and effective flow control. Physical-level design determines where to locate transmitters and receivers, how to map wavelengths to waveguides, where to layout waveguides for intra-chip interconnect, and where to place optical couplers and fibers for inter-chip interconnect. We use an inherently iterative process to navigate these levels in order to meet application requirements given our technology constraints.

Before discussing nanophotonic interconnection network design, Section II briefly reviews the underlying nanophotonic technology. Section III provides more details about our three-level design process. Sections IV–VI discuss three case studies to illustrate this design process and to demonstrate the potential for nanophotonic interconnection networks. Section VII concludes with several design themes that can be applied when designing future nanophotonic interconnection networks.

Manuscript received December 31, 2011; accepted February 14, 2012. Date of publication June 04, 2012; date of current version June 07, 2012. This work was supported in part by DARPA awards W911NF-06-1-0449, W911NF-08-1-0134, W911NF-08-1-0139, and W911NF-09-1-0342. This work was also supported in part by Microsoft (Award #024263) and Intel (Award #024894) funding and by matching funding from U.C. Discovery (Award DIG07-10227). The authors acknowledge chip fabrication support from Texas Instruments. This paper was recommended by Guest Editor J. Kim.

C. Batten is with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: cbatten@cornell.edu).

A. Joshi is with the Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215 USA.

V. Stojanović is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

K. Asanović is with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA 94720 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JETCAS.2012.2193932

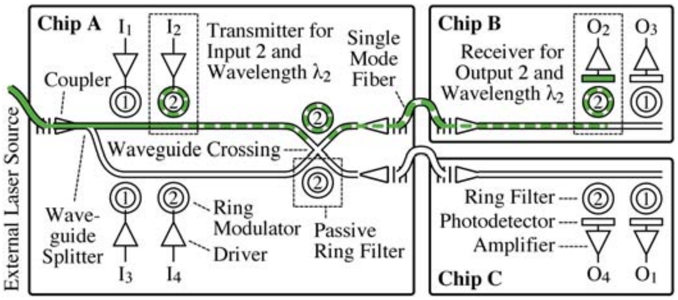


Fig. 1. Nanophotonic devices: Four point-to-point nanophotonic channels implemented with wavelength-division multiplexing. Number inside ring indicates resonant wavelength; link corresponding to $\mathrm{I}_2 \rightarrow \mathrm{O}_2$ on wavelength λ_2 is highlighted.

II. Nanophotonic Technology

Fig. 1 illustrates the devices in a typical wavelength-division multiplexed (WDM) nanophotonic link used to communicate between chips. Light from an off-chip two-wavelength (λ_1, λ_2) laser source is carried by an optical fiber and then coupled into an optical power waveguide on chip A. A splitter sends both wavelengths down parallel branches on opposite sides of the chip. Transmitters along each branch use silicon ring modulators to modulate a specific wavelength of light. The diameter of each ring sets its default resonant frequency, and the small electrical driver uses charge injection to change the resonant frequency and thus modulate the corresponding wavelength. Modulated light continues through the waveguides to the other side of the chip where passive ring filters can be used to shuffle wavelengths between the two waveguides. It is possible to shuffle multiple wavelengths at the same time with either multiple single-wavelength ring filters or a single multiple-wavelength comb filter. Additional couplers and single-mode fiber are used to connect the chips. On chips B and C, modulated light is guided to receivers that each use a passive ring filter to “drop” the corresponding wavelength from the waveguide into a photodetector. The photodetector turns absorbed light into current, which is sensed by the electrical amplifier. Ultimately, the example in Fig. 1 creates four point-to-point channels that connect the four inputs $(\mathrm{I}_1$–$\mathrm{I}_4)$ to the four outputs $(\mathrm{O}_1$–$\mathrm{O}_4)$, such that input I_1 sends data to output O_1, input I_2 sends data to output O_2, and so on. For higher bandwidth channels we can either increase the modulation rate of each wavelength, or we can use multiple wavelengths to implement a single logical channel (with the number of wavelengths denoted as b_λ). The same devices can be used for a purely intra-chip interconnect by simply integrating transmitters and receivers on the same chip.

As shown in Fig. 1, the silicon ring resonator is used in transmitters, passive filters, and receivers. Although other photonic structures (e.g., Mach–Zehnder interferometers) are possible, ring modulators are extremely compact (3–10 $\mu\mathrm{m}$ radius) resulting in reduced area and power consumption. Although not shown in Fig. 1, many nanophotonic interconnection networks also use active filtering to implement optical switching. For example, we might include multiple receivers with active filters for wavelength λ_1 on chip B. Each receiver’s ring filter would be detuned by default, and we can then actively tune a single receiver’s ring filter into resonance using charge injection. Some networks use active ring filters in the middle of the network itself. For example, we might replace the passive ring filters on chip A in Fig. 1 with active ring filters to create an optical switch. When detuned, inputs I_1, I_2, I_3, and I_4 are connected to outputs O_1, O_4, O_3, and O_2, respectively. When the ring filters are actively tuned into resonance, then the inputs are connected to the outputs with the corresponding subscripts. Of course, one of the challenges with these actively switched filters is in designing the appropriate electrical circuitry for routing and flow control that determines when to tune or detune each filter.

In the case studies presented later in this paper, we will be assuming a monolithic front-end-of-line integration strategy currently under development at the Massachusetts Institute of Technology (MIT) that attempts to integrate nanophotonics into state-of-the-art bulk-complementary-metal–oxide–semiconductor (CMOS) micro-electronic chips with no changes to the standard CMOS fabrication process. We use our experiences with a 65-nm test chip [21], our feasibility studies for a prototype 32-nm process, predictive electrical device models [22], and interconnect projections [23] to estimate both electrical and photonic device parameters for a target 22-nm technology node. Device-level details about the MIT nanophotonic technology assumed in the rest of this paper can be found in [21], and [24]–[27], although the technology is rapidly evolving such that more recent device-level work uses more advanced device and circuit techniques [28]–[30]. Details about the specific technology assumptions for each case study can be found in our previous system-level publications [3], [12], [13], [15].

We will focus on the design of networks using the devices shown in Fig. 1, but it is worth noting that some proposals use alternative devices such as vertical cavity surface emitting lasers combined with free-space optical channels [31], [32] or planar waveguides [20] that are not addressed in this paper.

III. Designing Chip-Level Nanophotonic Interconnection Networks

In this section, we describe three levels of nanophotonic interconnection network design: the architectural level, the microarchitectural level, and the physical level. At each level, we use insight gained from designing several nanophotonic networks to discuss the specific implications of using this emerging technology, and we classify recent nanophotonic network proposals to illustrate various different approaches.

A. Architectural-Level Design

The design of nanophotonic interconnection networks usually begins at the architectural level and involves selecting a logical network topology that can best leverage nanophotonic devices. A logical network topology connects a set of N input and output terminals through a collection of buses and routers interconnected by point-to-point channels. Fig. 2 illustrates several topologies for a 64-terminal network. At this preliminary phase of design, we can begin to determine the bus and channel bandwidths that will be required to meet application requirements assuming ideal routing and flow-control algorithms. Usually this analysis is in terms of theoretical upper-bounds on the network’s performance, but we can also begin to explore the impact of more realistic routing algorithms. When designing

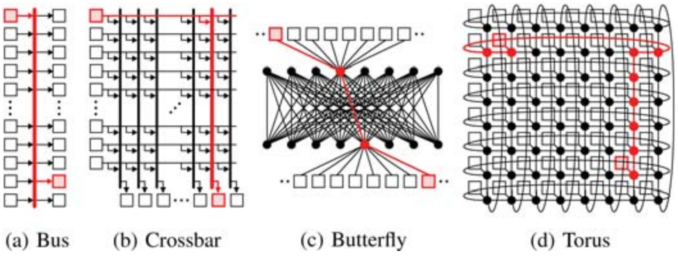


Fig. 2. Logical topologies for various 64 terminal networks: (a) 64-writer/64-reader single global bus; (b) 64 × 64 global nonblocking crossbar; (c) 8-ary 2-stage butterfly; (d) 8-ary 2-D torus. Squares = input and/or output terminals; dots = routers; in (c) inter-dot lines = uni-directional channels in (d) inter-dot lines = two channels in opposite directions.

nanophotonic networks, it is important to begin by characterizing state-of-the-art electrical networks to help motivate the best opportunities for leveraging nanophotonic devices.

A global bus is perhaps the simplest of logical topologies [see Fig. 2(a)]. Unfortunately, using a single shared medium often limits the performance of buses due to practical constraints on bus bandwidth and arbitration latency as the number of network terminals increases. There have been several nanophotonic bus designs that explore these trade-offs, mostly in the context of implementing efficient dynamic random-access memory (DRAM) channels [1]–[5], although there have also been proposals for specialized nanophotonic broadcast buses to improve the performance of cache-coherence protocols [1]. Multiple global buses can be used to improve system throughput, and such topologies have also been designed using nanophotonic devices [33].

A global crossbar topology includes one bus per terminal [see Fig. 2(b)]. Such topologies present a simple performance model to software and can sustain high-performance owing to their strictly nonblocking connectivity. This comes at the cost of many global buses crossing the network bisection and potentially long global arbitration delays. Nanophotonic crossbar topologies have been particularly popular in the literature [1], [6]–[9], [34], and we will see in the following sections that careful design at the microarchitectural and physical levels is required to help mitigate the challenges inherent in any global crossbar topology.

To avoid global buses and arbitration, we can move to a multi-stage topology such as a k-ary n-stage butterfly [see Fig. 2(c)]. Although multi-stage topologies increase the hop-count as compared to a global crossbar, each hop involves a localized lower-radix router that can be implemented more efficiently than a global crossbar. The reason for the butterfly topology's efficiency (distributed routing, arbitration, and flow-control), also leads to challenges in reducing zero-load latency and balancing channel load. Nanophotonic topologies have been proposed that are similar in spirit to the butterfly topology for multichip-module networks [10], on-chip networks [11], and processor-to-DRAM networks [12], [13]. Clos or fat-tree topologies add an extra $n-1$ stages to a basic butterfly topology and can offer the same nonblocking guarantees as global crossbars with potentially lower resource requirements. Clos and fat-tree topologies have been proposed that use nanophotonic devices in low-radix [14] and high-radix [15], [17] configurations. Nanophotonic Clos-like topologies that implement high-radix routers using a subnetwork of low-radix routers have also been explored [16].

A k-ary n-dimensional torus topology is an alternative multi-stage topology [see Fig. 2(d)]. A mesh topology eliminates the "wrap-around" channels in each dimension. Two-dimensional torus and mesh topologies map naturally to a planar chip substrate. Unfortunately, low-dimensional torus and mesh topologies have high hop counts resulting in longer latencies and possibly higher energy consumption. Moving to high-dimensional torus or mesh topologies reduces the network diameter, but requires long channels when mapped to a planar substrate and higher radix routers. There has been work investigating how to best use nanophotonics in both 2-D torus [18] and mesh [19], [20] topologies.

While many nanophotonic interconnection networks can be loosely categorized as belonging to one of the four categories shown in Fig. 2, there are also more radical alternatives. For example, Koohi *et al.* propose a hierarchical topology for an on-chip nanophotonic network where a set of global rings connect clusters each with their own local ring [35].

Table I illustrates the first-order analysis that can be performed at the architectural level. In this example, we compare six logical topologies for a 64-terminal on-chip network. We assume a 22-nm technology, 5-GHz clock frequency, and 400-mm^2 chip. The bus and channel bandwidths are sized so that each terminal can sustain 128 b/cycle under uniform random traffic assuming ideal routing and flow control. Even from this first-order analysis we can see that some topologies (e.g., crossbar, butterfly, and Clos) require fewer channels but they are often long, while other topologies (e.g., torus and mesh) require more channels but they are often short. Some topologies (e.g., crossbar and Clos) require more global bisection wiring resources, and others require higher-radix routers (e.g., crossbar, butterfly, Clos, and cmesh). First-order zero-load latency calculations can help illustrate trade-offs between hop count, router complexity, and serialization latency. Ultimately, this kind of rough analysis helps motivate the microarchitectural-level design discussed in the next section.

B. Microarchitectural-Level Design

For nanophotonic interconnection networks, micro-architectural-level design involves choosing how to implement each bus, channel, and router in the network. We must decide how to use active filters to implement nanophotonic routers, the best way to arbitrate for wavelengths, and how to manage electrical buffering at the edges of nanophotonic network components. We can use nanophotonic schematics to abstractly illustrate how various components are integrated (see Fig. 3). To focus on higher-level functionality, we assume as many wavelengths as necessary to meet our application requirements and defer practical issues related to mapping wavelengths to waveguides or waveguide layout until the physical level of design. Although this means detailed analysis of area overheads or optical power requirements is not yet possible, we can still make relative comparisons between various network microarchitectures. For example, we can compare the number of opto-electrical conversions along a given routing path, the total number of transmit-

TABLE I
ARCHITECTURAL-LEVEL ANALYSIS FOR VARIOUS 64 TERMINAL NETWORKS

		Buses & Channels				Routers		Latency				
Topology		N_C	N_{BC}	b_C	$N_{BC} \cdot b_C$	N_R	radix	H_R	T_R	T_C	T_S	T_0
Crossbar	64×64	64	64	128	8,192	1	64×64	1	10	n/a	4	14
Butterfly	8-ary 2-fly	64	32	128	4,096	16	8×8	2	2	2-10	4	10-18
Clos	(8,8,8)	128	64	128	8,192	24	8×8	3	2	2-10	4	14-32
Torus	8-ary 2-dim	256	32	128	4,096	64	5×5	2-9	2	2	4	10-38
Mesh	8-ary 2-dim	224	16	256	4,096	64	5×5	2-15	2	1	2	7-46
CMesh	4-ary 2-dim	48	8	512	4,096	16	8×8	1-7	2	2	1	3-25

Networks sized to sustain 128 b/cycle per input terminal under uniform random traffic. Latency calculations assume electrical implementation with an 8 × 8 grid of input/output terminals and the following parameters: 22-nm technology, 5-GHz clock frequency, and 400-mm^2 chip. N_c = number of channels or buses; b_C = bits/channel or bits/bus; N_{BC} = number of bisection channels or buses; N_R = number of routers; H_R = number of routers along minimal routes; T_R = router latency; T_C = channel latency; T_S = serialization latency; T_0 = zero load latency

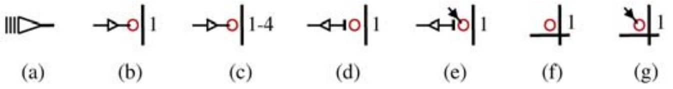
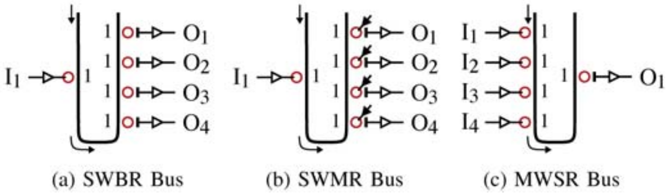


Fig. 3. Symbols used in nanophotonic schematics and layouts: (a) coupler for attaching fiber to on-chip waveguide; (b) transmitter including driver and ring modulator for λ_1; (c) multiple transmitters including drivers and ring modulators for each of λ_1–λ_4; (d) receiver including passive ring filter for λ_1 and photodetector; (e) receiver including active ring filter for λ_1 and photodetector; (f) passive ring filter for λ_1; (g) active ring filter for λ_1.



(a) SWBR Bus (b) SWMR Bus (c) MWSR Bus

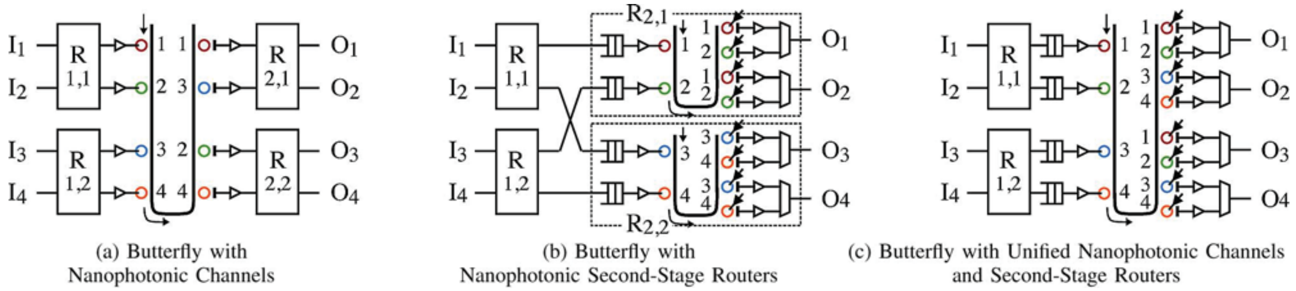
Fig. 4. Microarchitectural schematics for nanophotonic four-terminal buses: The buses connect one or more input terminals (I_1–I_4) to one or more output terminals (O_1–O_4) via a single shared wavelength: (a) single-writer broadcast-reader bus; (b) single-writer multiple-reader bus; (c) multiple-writer single-reader bus.

ters and receivers, the number of transmitters or receivers that share a single wavelength, the amount of active filtering, and design complexity. It should be possible to narrow our search in promising directions that we can pursue with a physical-level design, or to iterate back to the architectural level to explore other topologies and routing algorithms.

Nanophotonics can help mitigate some of the challenges with global electrical buses, since the electrical modulation energy in the transmitter is independent of both bus length and the number of terminals. However, the optical power strongly depends on these factors making it necessary to carefully consider the network's physical design. An efficient global bus arbitration may be required which is always challenging regardless of the implementation technology. A nanophotonic bus topology can be implemented with a single wavelength as the shared communication medium (see Fig. 4). Assuming a fixed modulation rate per wavelength, we can increase the bus bandwidth by using using multiple parallel wavelengths. In the single-writer broadcast-reader (SWBR) bus shown in Fig. 4(a), a single input terminal modulates the bus wavelength that is then broadcast to all four output terminals. An SWBR bus requires significant optical power to broadcast packets; if we wish to send a packet to only one of many outputs, then we can use active filters in each receiver. Fig. 4(b) shows a single-writer multiple-reader (SWMR) bus where by default the ring filters in each receiver are detuned such that none drop the bus wavelength. When the input terminal sends a packet to an output terminal, it first ensures that the ring filter at the destination receiver is actively tuned into the bus wavelength. The control logic for this active tuning usually requires additional optical or electrical communication. Fig. 4(c) illustrates a multiple-writer single-reader (MWSR) bus where four input terminals arbitrate to modulate the bus wavelength that is then dropped at a single output terminal. MWSR buses require global arbitration, which can be implemented either electrically or optically. Multiple-writer multiple-reader (MWMR) buses and multiple-writer broadcast-reader (MWBR) buses are also possible.

There are several examples of nanophotonic buses in the literature. A combination of nanophotonic SWBR and MWSR buses can be used to implement the command, write-data, and read-data buses in a DRAM memory channel [1]–[5]. Pan *et al.* proposed several techniques to help address scaling nanophotonic MWMR buses to larger numbers of terminals [33].

Nanophotonic crossbars use a dedicated nanophotonic bus per terminal to enable every input terminal to send a packet to a different output terminal at the same time. Fig. 6 illustrates three types of nanophotonic crossbars. In the SWMR crossbar shown in Fig. 6(a), there is one bus per input and every output can read from any bus. As an example, if I_2 wants to send a packet to O_3 it first arbitrates for access to the output terminal, then (assuming it wins arbitration) the receiver for wavelength λ_2 at O_3 is actively tuned, and finally the transmitter at I_2 modulates wavelength λ_2 to send the packet. SWBR crossbars are also possible where the packet is broadcast to all output terminals, and each output terminal is responsible for converting the packet into the electrical domain and determining if the packet is actually destined for that terminal. Note that SWMR crossbars usually include a low-bandwidth SWBR crossbar to implement distributed redundant arbitration at the output terminals and/or to determine which receivers at the destination should be actively tuned. A SWMR crossbar needs one transmitter per input, but requires $\mathrm{O}(N^2 b_\lambda)$ receivers. Fig. 6(b) illustrates modifications required for a buffered SWMR crossbar that avoids the need for any global or distributed arbitration. The MWSR



(a) Butterfly with Nanophotonic Channels
(b) Butterfly with Nanophotonic Second-Stage Routers
(c) Butterfly with Unified Nanophotonic Channels and Second-Stage Routers

Fig. 5. Microarchitectural schematics for nanophotonic 2-ary 2-stage butterflies: Networks connect all inputs $(\mathrm{I}_1–\mathrm{I}_4)$ to all outputs $(\mathrm{O}_1–\mathrm{O}_4)$: (a) electrical routers and nanophotonic channels; (b) electrical first-stage routers, electrical channels, and nanophotonic second-stage routers; (c) channels and intra-router crossbars are unified into a single stage of nanophotonic interconnect.

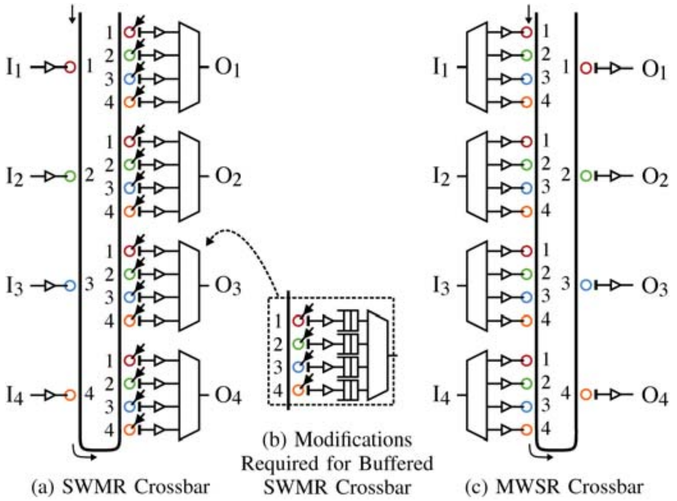


Fig. 6. Microarchitectural schematics for nanophotonic 4 × 4 crossbars: The crossbars connect all inputs $(\mathrm{I}_1-\mathrm{I}_4)$ to all outputs $(\mathrm{O}_1-\mathrm{O}_4)$ and are implemented with either: (a) four SWMR buses; (b) four SWMR buses with additional output buffering; or (c) four MWSR buses.

crossbar shown in Fig. 6(c) uses one bus per output and allows every input to write any of these buses. As an example, if I_2 wants to send a packet to O_3 it first arbitrates, and then (assuming it wins arbitration) it modulates wavelength λ_3. A MWSR crossbar needs one receiver per output, but requires $\mathrm{O}(N^2 b_\lambda)$ transmitters.

There have been several diverse proposals for implementing global crossbars with nanophotonics. Kırman *et al.* describe three on-chip SWBR nanophotonic crossbars for addresses, snoop responses, and data to implement a snoopy-based cache-coherence protocol [6]. Miller *et al.* describe a buffered SWBR nanophotonic crossbar for implementing a directory-based cache-coherence protocol, and the broadcast capabilities of the SWBR crossbar are used for invalidations [7]. Vantrease *et al.* describe a MWSR nanophotonic crossbar for implementing a directory-based cache-coherence protocol, and a separate MWBR nanophotonic bus for invalidations [1], [36].

There are additional design decisions when implementing a multi-stage topology, since each network component can use either electrical or nanophotonic devices. Fig. 5 illustrates various microarchitectures for a 2-ary 2-stage butterfly. In Fig. 5(a), the routers are all implemented electrically and the channels connecting the first and second stage of routers are implemented with point-to-point nanophotonic channels. This approach leverages the advantages of nanophotonics for implementing long global channels and uses electrical technology for buffering, arbitration, and switching. Note that even though these are point-to-point channels, we can still draw the corresponding nanophotonic implementations of these channels as being wavelength-division multiplexed. Similarly, the input and output terminals may be co-located in the physical design, but the schematic can use a more abstract representation. In Fig. 5(b), just the second stage of routers are implemented with nanophotonic devices and the channels are still implemented electrically. Such a microarchitecture seems less practical since the router crossbars are localized, and it will be difficult to outweigh the opto-electrical conversion overhead when working with short buses. Fig. 5(c) illustrates a microarchitecture where the nanophotonic channels and second-stage routers are unified and requires a single opto-electrical conversion. This forces the electrical buffering to the edge of the nanophotonic region of the network. It is also possible to implement all routers and all channels with nanophotonics to create a fully optical multi-stage network, although the microarchitecture for each router will need to be more complicated and a second control network is required to setup the active ring filters in each router.

Most proposals for nanophotonic butterfly-like topologies in the literature focus on high-radix, low-diameter butterflies and use electrical routers with nanophotonic point-to-point channels. Koka *et al.* explore both single-stage and 2-stage butterfly-like topologies as the interconnect for large multichip modules [10]. Kodi *et al.* proposed a 2-stage butterfly-like topology for a purely on-chip network [11]. Both of these proposals are not true butterfly topologies since they incorporate some amount of flattening as in the flattened butterfly topology. In addition, some of the configurations include shared nanophotonic buses instead of solely using point-to-point channels. In spite of these details, both microarchitectures are similar in spirit to that shown Fig. 5(a). Pan *et al.* proposed a three-stage high-radix Clos-like topology for an on-chip network to enable much better load balancing [16]. Gu *et al.* proposed a completely different Clos microarchitecture that uses low-radix 2 × 2 routers and implements all routers and channels with nanophotonic devices [14].

Designing nanophotonic torus topologies requires similar design decisions at the microarchitectural level as when designing

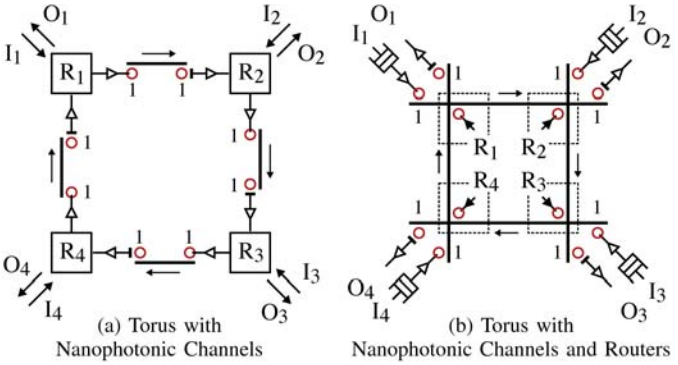

O1
I1
R1
R2
I2
O2
R3
R4
O4
I4
I3
O3
1
(a) Torus with Nanophotonic Channels
(b) Torus with Nanophotonic Channels and Routers

Fig. 7. Microarchitectural schematics for nanophotonic 4-ary 1-D torus: Networks connect all inputs $(\mathrm{I}_1-\mathrm{I}_4)$ to all outputs $(\mathrm{O}_1-\mathrm{O}_4)$ with each network component implemented with either electrical or nanophotonic technology: (a) electrical routers and nanophotonic channels or (b) nanophotonic routers and channels. Note that this topology uses a single unidirectional channel to connect each of the routers.

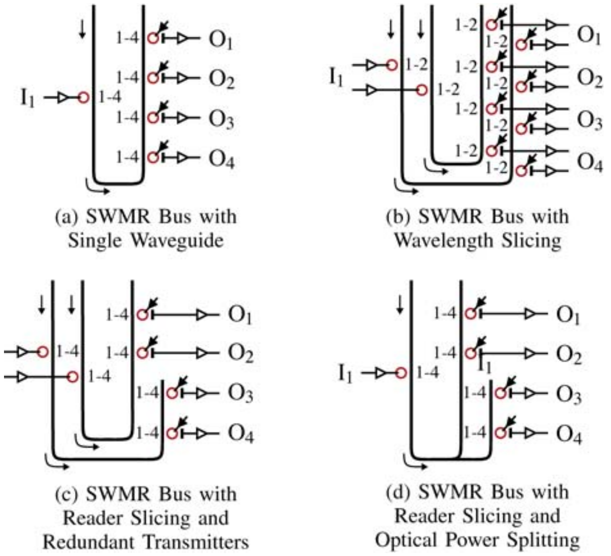

I1
O1
O2
O3
O4
1-4
1-2
(a) SWMR Bus with Single Waveguide
(b) SWMR Bus with Wavelength Slicing
(c) SWMR Bus with Reader Slicing and Redundant Transmitters
(d) SWMR Bus with Reader Slicing and Optical Power Splitting

Fig. 8. Physical design of nanophotonic buses. The four wavelengths for an example four-output SWMR bus are mapped to waveguides in various ways: (a) all wavelengths mapped to one waveguide; (b) wavelength slicing with two wavelengths mapped to one waveguide; (c) reader slicing with two readers mapped to one waveguide and two redundant sets of transmitters; (d) reader slicing with a single transmitter and optical power passively split between two branches.

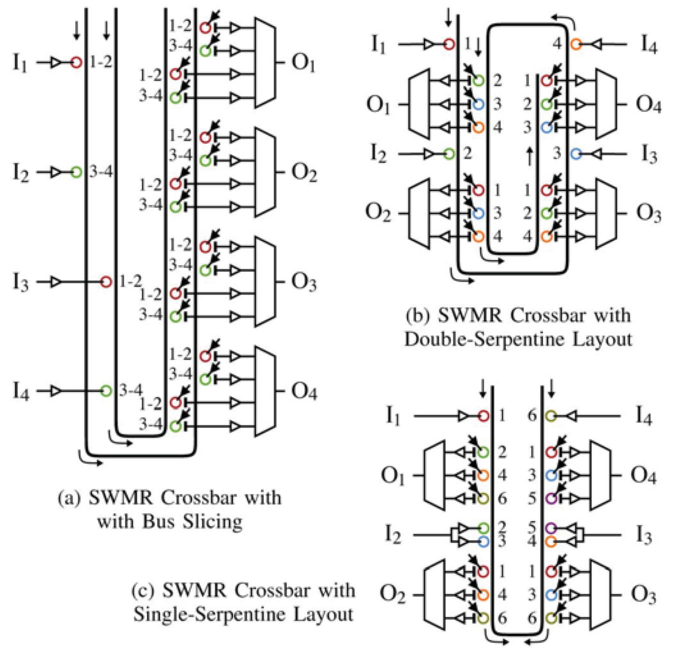
butterfly topologies. Fig. 7 illustrates two different microarchitectures for a 4-ary 1-D torus. In Fig. 7(a), the four radix-2 routers are implemented electrically and the channels between each pair of routers are implemented with nanophotonic devices. In Fig. 7(b), both the routers and the channels are implemented with nanophotonic devices. The active ring filters in each router determine whether the packet exits the network at that router or turns clockwise and continues on to the next router. Since this creates a fully optical multi-stage network, a separate control network, implemented either optically or electrically, will be required to setup the control signals at each router. As with the butterfly microarchitecture in Fig. 5(c), buffering must be pushed to the edge of the nanophotonic region of the network.

Proposals in the literature for chip-level nanophotonic torus and mesh networks have been mostly limited to 2-D topologies. In addition, these proposals use fully optical microarchitectures in the spirit of Fig. 7(b), since using electrical routers with short nanophotonic channels as in Fig. 7(a) yields little benefit. Shacham *et al.* proposed a fully optical 2-D torus with a combination of radix-4 blocking routers and specialized radix-2 injection and ejection routers [18], and others later explored radix-4 nonblocking routers [37]. Poon *et al.* survey a variety of designs for optical routers that can be used in on-chip multistage nanophotonic networks [38]. Cianchetti *et al.* proposed a fully optical 2-D mesh topology with packet-based flow control [19], [39].

C. Physical-Level Design

The final phase of design is at the physical level and involves mapping wavelengths to waveguides, waveguide layout, and placing nanophotonic devices along each waveguide. We can use abstract layout diagrams that are similar to microarchitectural schematics but include additional details to illustrate the physical design. Ultimately, we must develop a detailed layout specifying the exact placement of each device to calculate the area overhead and the total optical power requirements.

Fig. 8 illustrates general approaches for the physical design of a four-wavelength SWMR nanophotonic bus. Fig. 8(a) illustrates all four wavelengths multiplexed onto the same waveguide. This produces the most compact layout, but requires all nanophotonic devices to operate on the same waveguide which can increase the total optical loss per wavelength. In this example, each wavelength would experience one modulator insertion loss, $\mathrm{O}(Nb_\lambda)$ through losses in the worst case, and a drop loss at the desired output terminal. As the number of wavelengths increases, we will need to consider techniques for distributing those wavelengths across multiple waveguides both to stay within the waveguide's total bandwidth capacity and within the waveguide's total optical power limit. Fig. 8(b) illustrates wavelength slicing, where a subset of the bus wavelengths are mapped to distinct waveguides. In addition to reducing the number of wavelengths per waveguide, wavelength slicing can potentially reduce the number of through losses. Fig. 8(c) and (d) illustrates reader slicing, where a subset of the bus readers are mapped to distinct waveguides. The example shown in Fig. 8(c) doubles the number of transmitters, but the input terminal only needs to drive transmitters on the waveguide associated with the desired output terminal. Reader slicing does not reduce the number of wavelengths per waveguide, but it does reduce the number of through losses. Fig. 8(d) illustrates a split nanophotonic bus that uses a single set of transmitters, but requires more optical power since this power must be split between the multiple bus branches. A guided nanophotonic bus uses active ring filters instead of a passive splitter to guide the optical power down the desired bus branch. Guided buses require more control overhead but can significantly reduce the total optical power when the optical loss per branch is large. Reader slicing can be particularly effective in SWBR buses, since it can reduce the number of drop losses per wavelength. It is possible to implement MWSR buses using a similar technique called writer slicing, which can help reduce the number of modulator insertion losses per wavelength. More complicated

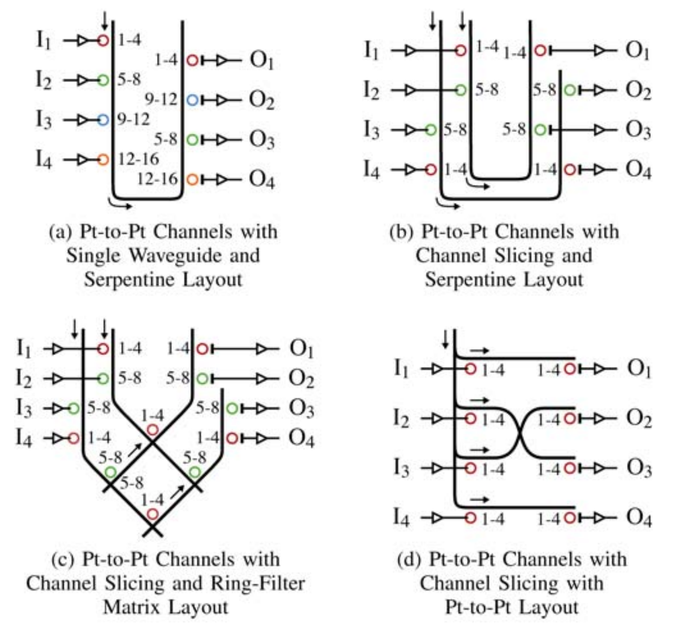


(a) SWMR Crossbar with with Bus Slicing

(b) SWMR Crossbar with Double-Serpentine Layout

(c) SWMR Crossbar with Single-Serpentine Layout

Fig. 9. Physical design of nanophotonic crossbars: (a) illustrates 4 × 4 SWMR crossbar with two wavelengths per bus and two buses per waveguide. Co-locating input and output terminals can impact the physical layout. For example, a 4 × 4 SWMR crossbar with one wavelength per bus and a single waveguide can be implemented with either: (b) a double-serpentine layout where the light travels in one direction or (c) a single-serpentine layout where the light travels in two directions.



(a) Pt-to-Pt Channels with Single Waveguide and Serpentine Layout

(b) Pt-to-Pt Channels with Channel Slicing and Serpentine Layout

(c) Pt-to-Pt Channels with Channel Slicing and Ring-Filter Matrix Layout

(d) Pt-to-Pt Channels with Channel Slicing with Pt-to-Pt Layout

Fig. 10. Physical design of nanophotonic point-to-point channels. Four point-to-point channels each with four wavelengths implemented with: (a) all wavelengths mapped to one waveguide; (b) partial channel slicing with all wavelengths from two channels mapped to one waveguide and a serpentine layout; (c) partial channel slicing with a ring-filter matrix layout to passively shuffle wavelengths between waveguides; (d) full channel slicing with each channel mapped to its own waveguide and a point-to-point layout.

physical design (e.g., redundant transmitters and optical power guiding) may have some implications on the electrical control logic and thus the network's microarchitecture, but it is important to note that these techniques are solely focused on mitigating physical design issues and do not fundamentally change the logical network topology. Most nanophotonic buses in the literature use wavelength slicing [1], [2], [4] and there has been some work on the impact of split nanophotonic buses [4], and guided nanophotonic buses [3].

Most nanophotonic crossbars use a set of shared buses, and thus wavelength slicing, reader slicing, and writer slicing are all applicable. Fig. 9(a) illustrates another technique called bus slicing, where a subset of the crossbar buses are mapped to each waveguide. In this example, a 4 × 4 SWMR crossbar with two wavelengths per bus is sliced such that two buses are mapped to each of the two waveguides. Bus-sliced MWSR crossbars are also possible. Bus slicing reduces the number of wavelengths per waveguide and the number of through losses in both SWMR and MWSR crossbars. In addition to illustrating how wavelengths are mapped to waveguides, Fig. 9(a) also illustrates a serpentine layout. Such layouts minimize waveguide crossings by "snaking" all waveguides around the chip, and they result in looped, U-shaped, and S-shaped waveguides. The example in Fig. 9(a) assumes that the input and output terminals are located on opposite sides of the crossbar, but it is also common to co-locate terminals. Fig. 9(b) illustrates a double-serpentine layout for a 4 × 4 SWMR crossbar with one wavelength per bus and a single waveguide. In this layout, waveguides are "snaked" by each terminal twice with light traveling in one direction. Fig. 9(c) illustrates an alternative single-serpentine layout where waveguides are "snaked" by each terminal once, and light travels in both directions. A single-serpentine layout can reduce waveguide length but requires additional transmitters to send the light for a single bus in both directions. A variety of physical designs for nanophotonic crossbars are proposed in the literature that use a combination of the basic approaches described above. Examples include fully wavelength-sliced SWBR crossbars with no bus slicing and a serpentine layout [6], [7], [9], partially wavelength-sliced and bus-sliced MWSR/SWMR crossbars with a double-serpentine layout [1], [16], and fully reader-sliced SWMR crossbars with multiple redundant transmitters and a serpentine layout [11].

Fig. 10 illustrates general approaches for the physical design of point-to-point nanophotonic channels that can be used in butterfly and torus topologies. This example includes four point-to-point channels with four wavelengths per channel, and the input and output terminals are connected in such a way that they could be used to implement the 2-ary 2-stage butterfly microarchitecture shown in Fig. 5(a). In Fig. 10(a), all 16 wavelengths are mapped to a single waveguide with a serpentine layout. As with nanophotonic buses, wavelength slicing can reduce the number of wavelengths per waveguide and total through losses. Fig. 10(b)–(d) illustrates channel slicing where all wavelengths from a subset of the channels are mapped to a single waveguide. Channel slicing reduces the number of wavelengths per waveguide, the through losses, and can potentially enable shorter waveguides. The example shown in Fig. 10(b), maps two channels to each waveguide but still uses a serpentine layout. The example in Fig. 10(c) has the same organization on the transmitter side, but uses a passive

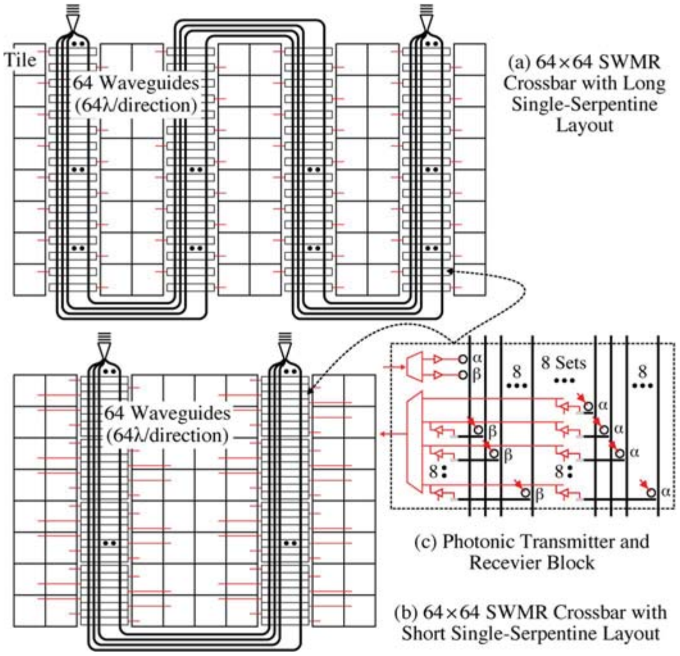


Fig. 11. Abstract physical layouts for 64 × 64 SWMR crossbar: (a) long single-serpentine layout; (b) shorter single-serpentine layout. Nanophotonic transmitter and receiver block shown in (c) illustrates bus slicing. One logical channel (128 b/cycle or 64λ per channel) is mapped to each waveguide; the channel is split into 64λ directed left to right and 64λ directed right to left. Each ring actually represents 64 rings each tuned to a different wavelength; $\alpha = \lambda_1$–λ_{64}; $\beta = \lambda_{64}$–λ_{128}; couplers indicate where laser light enters chip.

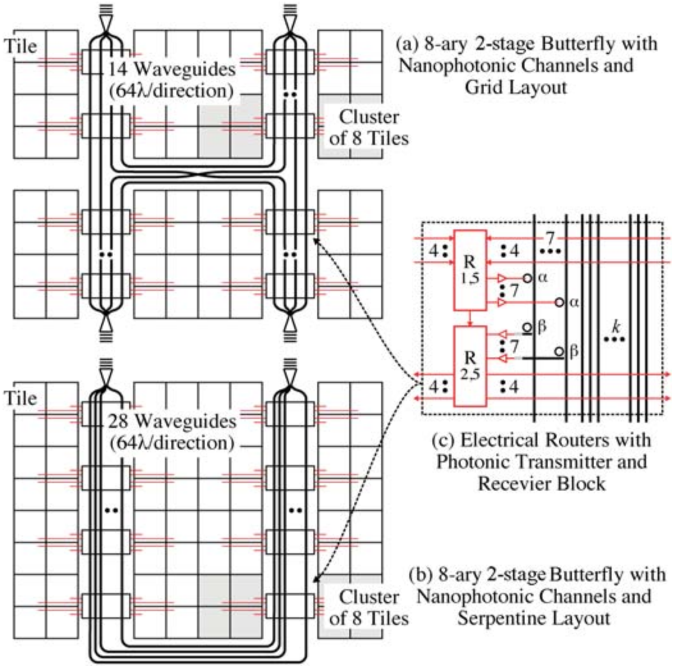
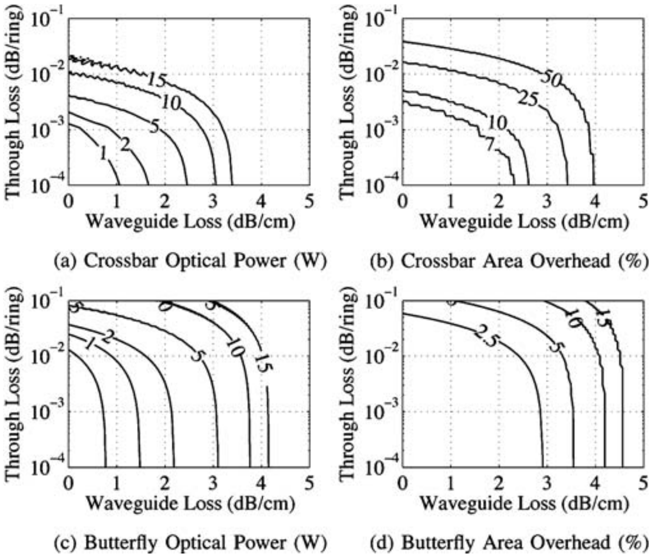


Fig. 12. Abstract physical layouts for 8-ary 2-stage butterfly with nanophotonic channels: (a) point-to-point layout; (b) serpentine layout. Nanophotonic transmitter and receiver block shown in (c) illustrates channel slicing. Two logical channels (128 b/cycle or 64λ per channel) are mapped to each waveguide. Each ring actually represents 64 rings each tuned to a different wavelength; $\alpha = \lambda_1$–λ_{64}; $\beta = \lambda_{64}$–λ_{128}; k is seven for point-to-point layout and 21 for serpentine layout; couplers indicate where laser light enters chip.

ring filter matrix layout to shuffle wavelengths between waveguides. Ring filter matrices can shorten waveguides at the cost of increased waveguide crossings and one or more additional drop losses. Fig. 10(d) illustrates a fully channel-sliced design with one channel per waveguide. This enables a point-to-point layout with waveguides directly connecting input and output terminals. Although point-to-point layouts enable the shortest waveguide lengths they usually also lead to the greatest number of waveguide crossings and layout complexity. One of the challenges with ring-filter matrix and point-to-point layouts is efficiently distributing the unmodulated laser light to all of the transmitters while minimizing the number of laser couplers and optical power waveguide complexity. Optimally allocating channels to waveguides can be difficult, so researchers have investigated using machine learning [34] or an iterative algorithm [40] for specific topologies. There has been some exploratory work on a fully channel-sliced physical design with a point-to-point layout for implementing a quasi-butterfly topology [10], and some experimental work on passive ring filter network components similar in spirit to the ring-filter matrix [41].

Much of the above discussion about physical-level design is applicable to multi-stage topologies. However, the physical layout in these designs is often driven more by the logical topology, leading to inherently channel-sliced designs with point-to-point layouts. For example, nanophotonic torus and mesh topologies are often implemented with regular grid-like layouts. It is certainly possible to map such topologies onto serpentine layouts or to use a ring filter matrix to pack multiple logical channels onto the same waveguide, but such designs would probably be expensive in terms of area and optical power. Wavelength slicing is often used to increase the bandwidth per channel. The examples in the literature for fully optical fat-tree networks [14], torus networks [18], and mesh networks [19], [20] all use channel slicing and regular layouts that match the logical topology.

Figs. 11 and 12 illustrate abstract layout diagrams for an on-chip nanophotonic 64 × 64 global crossbar network and an 8-ary 2-stage butterfly network. These layouts assume a 22-nm technology, 5-GHz clock frequency, and 400-mm^2 chip with 64 tiles. The network bus and channel bandwidths are sized according to Table I. The 64 × 64 crossbar topology in Fig. 11 uses a SWMR microarchitecture with bus slicing and a single-serpentine layout. Both layouts map a single bus to each waveguide with half the wavelengths directed from left to right and the other half directed from right to left. Fig. 11(a) uses a longer serpentine layout, while Fig. 11(b) uses a shorter serpentine layout which reduces waveguide lengths at the cost of increased electrical energy to communicate between the more distant tiles and the nanophotonic devices. The 8-ary 2-stage butterfly topology in Fig. 12 is implemented with 16 electrical routers (eight per stage) and 64 point-to-point nanophotonic channels connecting every router in the first stage to every router in the second stage. Fig. 12(a) uses channel slicing with no wavelength slicing and a point-to-point layout to minimize waveguide length. Note that although two channels are mapped to the same waveguide, those two channels connect routers in the same physical locations eliminating any need for a ring-filter matrix. Clever waveguide layout



(a) Crossbar Optical Power (W) (b) Crossbar Area Overhead (%)

(c) Butterfly Optical Power (W) (d) Butterfly Area Overhead (%)

Fig. 13. Waveguide comparison of 64 × 64 crossbar and 8-ary 3-stage butterfly networks: Contour plots show optical laser power in Watts and area overhead as a percentage of the total chip area for the layouts in Fig. 11(b) and Fig. 12(b). These metrics are plotted as a function of optical device quality (i.e., ring through loss and waveguide loss).

results in 16 waveguide crossings located in the middle of the chip. Fig. 12(b) uses a single-serpentine layout that increases waveguide lengths but eliminates waveguide crossings in the middle of the chip.

Fig. 13 illustrates the kind of quantitative analysis that can be performed at the physical level of design. Detailed layouts corresponding to the abstract layouts in Fig. 11(b) and Fig. 12(b) are used to calculate the total optical power and area overhead as a function of optical device quality given various assumptions for our specific nanophotonic technology. Higher optical losses increase the power per waveguide which eventually necessitates distributing wavelengths over more waveguides to stay within the waveguide's total optical power limit. Thus, higher optical losses can increase both optical power and area overhead. It is clear that for these layouts, the crossbar network requires more optical power and area for the same quality of devices compared to the butterfly network. This is simply a result of the cost of providing $\mathrm{O}(N^2 b_\lambda)$ receivers in the SWMR crossbar network versus the simpler point-to-point nanophotonic channels used in the butterfly network. We can also perform rough terminal tuning estimates based on the total number of rings in each layout. Given our technology assumptions, the crossbar network requires 500 000 rings and a fixed thermal tuning power of over 10 W. The butterfly network requires only 14 000 rings and a fixed thermal tuning power of 0.28 W. Although the crossbar is more expensive to implement, it should also have significantly higher performance since it is a single-stage nonblocking topology. Since nanophotonics is still an emerging technology, evaluating a layout as a function of optical device quality is critical for a fair comparison.

IV. Case Study #1: On-Chip Tile-to-Tile Network

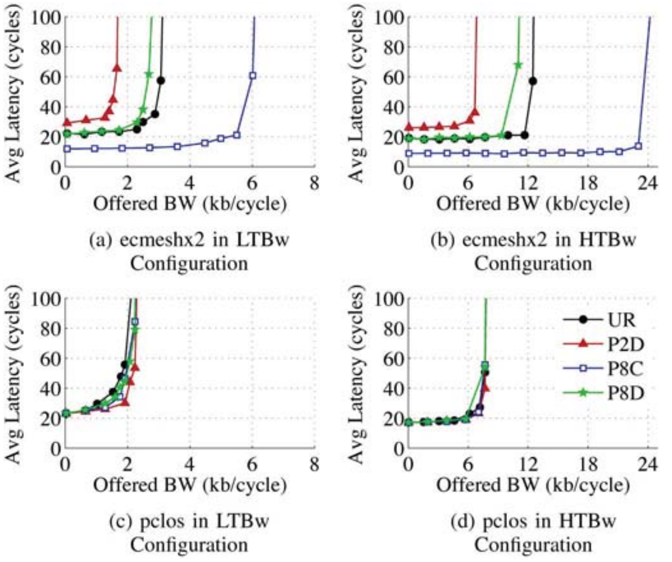
In this case study, we present a nanophotonic interconnection network suitable for global on-chip communication between 64 tiles in a chip-multiprocessor. This case study assumes a 22-nm technology, 5-GHz clock frequency, 512-bit packets, and 400-mm^2 chip. We examine networks sized for low (LTBw), medium (MTBw), and high (HTBw) target bandwidths which correspond to ideal throughputs of 64, 128, and 256 b/cycle per tile under uniform random traffic. More details on this case study can be found in [15].

A. Network Design

Table I shows configurations for various topologies that meet the MTBw target. Nanophotonic implementations of the 64 × 64 crossbar and 8-ary 2-stage butterfly networks were discussed in Section III. Our preliminary analysis suggested that the crossbar network could achieve good performance but with significant optical power and area overhead, while the butterfly network could achieve lower optical power and area overhead but might perform poorly on adversarial traffic patterns. This analysis motivated our interest in high-radix, low-diameter Clos networks. A classic three-stage (m, n, r) Clos topology is characterized by the number of routers in the middle stage (m), the radix of the routers in the first and last stages (n), and the number of input and output switches (r). For this case study, we explore a (8,8,8) Clos topology which is similar to the 8-ary 2-stage butterfly topology shown in Fig. 2(c) except with three stages of routers. This topology is nonblocking which can enable significantly higher performance than a blocking butterfly, but the Clos topology also requires twice as many bisection channels which requires careful design at the microarchitectural and physical level. We use an oblivious nondeterministic routing algorithm that efficiently balances load by always randomly picking a middle-stage router.

The 8-ary 2-stage butterfly in Fig. 12(b) has low optical power and area overhead due to its use of nanophotonics solely for point-to-point channels and not for optical switching. For the Clos network we considered two microarchitectures. The first uses two sets of nanophotonic point-to-point channels to connect three stages of electrical routers. All buffering, arbitration, and flow-control is done electrically. The second implements both the point-to-point channels and the middle stage of routers with nanophotonics. We chose to purse the first microarchitecture, since preliminary analysis suggested that the energy advantage of using nanophotonic middle-stage routers was outweighed by the increased optical laser power. We will revisit this assumption later in this case study. Note how the topology choice impacted our microarchitectural-level design; if we had chosen to explore a low-radix, high-diameter Clos topology then optical switching would probably be required to avoid many opto-electrical conversions. Here, we opt for a high-radix, low-diameter topology to minimize the complexity of the nanophotonic network.

We use a physical layout similar to that shown for the 8-ary 2-stage butterfly in Fig. 12(b) except that we require twice as many point-to-point channels and thus twice as many waveguides. For the Clos network, each of the eight groups of routers includes three instead of two radix-8 routers. The Clos network will have twice the optical power and area overhead as shown for the butterfly in Fig. 13(a) and (b). Note that even with twice the number of bisection channels, the Clos network still uses



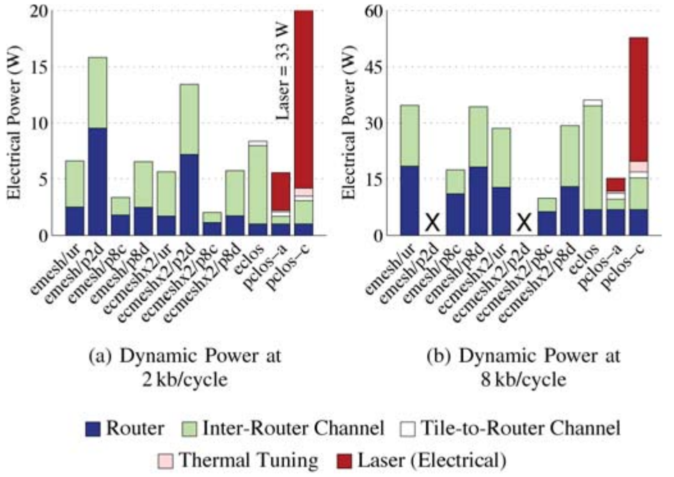
(a) ecmeshx2 in LTBw Configuration

(b) ecmeshx2 in HTBw Configuration

(c) pclos in LTBw Configuration

(d) pclos in HTBw Configuration

Fig. 14. Latency versus offered bandwidth for on-chip tile-to-tile networks: LTBw systems have a theoretical throughput of 64 b/cycle per tile, while HTBw systems have a theoretical throughput of 256 b/cycle both for the uniform random traffic pattern.



(a) Dynamic Power at 2 kb/cycle

(b) Dynamic Power at 8 kb/cycle

Fig. 15. Dynamic power breakdown for on-chip tile-to-tile networks: Power of *eclos* and pclos did not vary significantly across traffic patterns. (a) LTBw systems at 2 kb/cycle offered bandwidth (except for *emesh*$/p2d$ and *ecmeshx2*$/p2d$ which saturated before 2 kb/cycle, HTBw system shown instead); (b) HTBw systems at 8 kb/cycle offered bandwidth (except for *emesh/p2d* and *ecmeshx2/p2d* which are not able to achieve 8 kb/cycle). *pclos-c* (*pclos-a*) corresponds to conservative (aggressive) nanophotonic projections.

less than 10% of the chip area for a wide range of optical device parameters. This is due to the impressive bandwidth density provided by nanophotonic technology. The Clos network requires an order of magnitude fewer rings than the crossbar network resulting in a significant reduction in optical power and area overhead.

B. Evaluation

Our evaluation uses a detailed cycle-level microarchitectural simulator to study the performance and power of various electrical and nanophotonic networks. Our baseline includes three electrical networks: an 8-ary 2-D mesh *(emesh)*, a 4-ary 2-D concentrated mesh with two independent physical networks *(ecmeshx2)*, and an (8,8,8) Clos *(eclos)*. We use aggressive projections for the on-chip electrical interconnect. We also study a nanophotonic implementation of the Clos network as described in the previous section *(pclos)* with both aggressive and conservative nanophotonic technology projections. We use synthetic traffic patterns based on a partitioned application model. Each traffic pattern has some number of logical partitions, and tiles randomly communicate only with other tiles that are in the same partition. Although we studied various partition sizes and mappings, we focus on the following four representative patterns. A single global partition is identical to the standard uniform random traffic pattern (UR). The P8C pattern has eight partitions each with eight tiles optimally co-located together. The P8D pattern stripes these partitions across the chip. The P2D pattern has 32 partitions each with two tiles, and these two tiles are mapped to diagonally opposite quadrants of the chip.

Fig. 14 shows the latency as a function of offered bandwidth for a subset of the configurations. First note that the *pclos* network has similar zero-load latency and saturation throughput regardless of the traffic patterns, since packets are always randomly distributed across the middle-stage routers. Since to first order the nanophotonic channel latencies are constant, this routing algorithm does not increase the zero-load latency over a “minimal” routing algorithm. This is in contrast to *eclos*, which has higher zero-load latency owing to the nonuniform channel latencies. Our simulations show that on average, *ecmeshx2* has higher performance than *emesh* due to the path diversity provided by the two mesh networks and the reduced network diameter. Fig. 14 illustrates that *pclos* performs better than *ecmeshx2* on global patterns (e.g., P2D) and worse on local patterns (e.g., P8C). The hope is that a higher-capacity *pclos* configuration [e.g., Fig. 14(d)] will have similar power consumption as a lower-capacity *ecmeshx2* configuration [e.g., Fig. 14(a)]. This could enable a nanophotonic Clos network to have similar or better performance than an electrical network within a similar power constraint.

Fig. 15 shows the power breakdowns for various topologies and traffic patterns. Fig. 15(a) includes the least expensive configurations that can sustain an aggregate throughput of 2 kb/cycle, while Fig. 15(b) includes the least expensive configurations that can sustain an aggregate throughput of 8 kb/cycle. Compared to *emesh* and *ecmeshx2* at 8 kb/cycle, the *pclos* network with aggressive technology projections provides comparable performance and lower power dissipation for global traffic patterns, and comparable performance and power dissipation for local traffic patterns. The benefit is less clear at lower target bandwidths, since the nontrivial fixed power overhead of nanophotonics cannot be as effectively amortized. Notice the significant amount of electrical laser power; our analysis assumes a 33% efficiency laser. Although this electrical laser power is off-chip, it can impact system-level design and the corresponding optical laser power is converted into heat on-chip.

C. Design Themes

This case study illustrates several important design themes. First, it can be challenging to show a compelling advantage for purely on-chip nanophotonic interconnection networks if we include fixed power overheads, use a more aggressive electrical baseline, and consider local as well as global traffic patterns. Second, point-to-point nanophotonic channels (or at least a limited amount of optical switching) seems to be a more practical approach compared to global nanophotonic crossbars. Third, it is important to use an iterative design process that considers all levels of the design. For example, Fig. 15 shows that the router power begins to consume a significant portion of the total power at higher bandwidths in the nanophotonic Clos network, and in fact, follow up work by Kao *et al.* began exploring the possibility of using both nanophotonic channels and one stage of nanophotonic routers [17].

V. Case Study #2: Manycore Processor-to-DRAM Network

Off-chip main-memory bandwidth is likely to be a key bottleneck in future manycore systems. In this case study, we present a nanophotonic processor-to-DRAM network suitable for single-socket systems with 256 on-chip tiles and 16 DRAM modules. This case study assumes a 22-nm technology, 2.5-GHz clock frequency, 512-bit packets for transferring cache lines, and 400-mm^2 chip. More details on this case study can be found in [12], [13].

A. Network Design

We focus on high-radix, low-diameter topologies so that we can make use of simple point-to-point nanophotonic channels. The lack of path diversity in the butterfly topology is less of an issue in this application. We assume that the address space is interleaved across DRAM modules at a fine granularity to maximize performance, and any structure in the address stream from a single core is effectively lost when we consider hundreds of tiles arbitrating for tens of DRAM modules. A 2-stage butterfly topology for 256 tiles would require radix-16 routers which can be expensive to implement electrically. We could implement these routers with nanophotonics, but this increases complexity and risk. We could also increase the number of stages to reduce the radix, but this increases the amount of opto-electrical conversions or requires optical switching. We choose instead to use the local-meshes to global-switches (LMGS) topology shown in Fig. 16 where each high-radix router is implemented with an electrical mesh subnetwork also called a cluster. A generic (c, n, m, r) LMGS topology is characterized by the number of clusters (c), the number of tiles per cluster (n), the number of global switches (m), and the radix of the global switches $(c \times r)$. Some of the mesh routers in each cluster are access points, meaning they directly connect to the global routers. Each global router is associated with a set of memory controllers that manage an independent set of DRAM chips, and together this forms a DRAM module. In this case study, we explore LMGS topologies supporting 256 tiles and 16 DRAM modules with one, four, and 16 clusters. Since the DRAM memory controller design is not the focus of this case study, we ensure that the memory controller bandwidth is not a bottleneck.

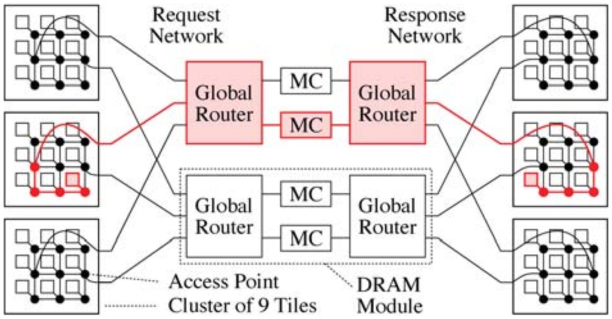


Fig. 16. Logical topology for processor-to-DRAM network: Two (3,9,2,2) LMGS networks are used for memory requests and responses. Each LMGS network includes three groups of nine tiles arranged in small 3-ary 2-D mesh clusters and two global 3 × 2 routers that interconnect the clusters and DRAM memory controllers (MC). Lines in cluster mesh network represent two unidirectional channels in opposite directions; other lines represent one unidirectional channel heading from left to right.

We use first-order analysis to size the nanophotonic point-to-point channels such that the memory system power consumption on uniform random traffic is less than a 20 W power constraint. Initially, we balance the bisection bandwidth of the cluster mesh networks and the global channel bandwidth, but we also consider overprovisioning the channel bandwidths in the cluster mesh networks to compensate for intra-mesh contention. Configurations with more clusters will require more nanophotonic channels, and thus each channel will have lower bandwidth to still remain within this power constraint.

Fig. 17 shows the abstract layout for a system with 16 clusters. Each cluster requires one dedicated global channel to each DRAM module, resulting in a total of 256 cluster-to-memory channels with one nanophotonic access point per channel. Our first-order analysis determined that 16λ (160 Gb/s) per channel should enable the configuration to meet the 20 W power constraint. A ring-filter matrix layout is used to passively shuffle the 16-λ channels on different horizontal waveguides destined for the same DRAM module onto the same set of four vertical waveguides. Each DRAM module includes a custom switch chip containing the global router for the request and response networks. Fig. 18 shows the laser power as a function of optical device quality for two different channel bandwidths. Systems with greater aggregate bandwidth have quadratically more waveguide crossings, making them more sensitive to crossing losses. Additionally, certain combinations of waveguide and crossing losses result in large cumulative losses and require multiple waveguides to stay within the waveguide power limit. These additional waveguides further increase the total number of crossings, which in turn continues to increase the power per wavelength, meaning that for some device parameters it is infeasible to achieve a desired aggregate bandwidth with a ring-filter matrix layout.

B. Evaluation

Our evaluation uses a detailed cycle-level microarchitectural simulator to study the performance and power of various electrical and nanophotonic networks. The modeled system includes two-cycle mesh routers, one-cycle mesh channels, four-cycle global point-to-point channels, and 100-cycle DRAM array access latency. For this study, we use a synthetic uniform random traffic pattern at a configurable injection rate.

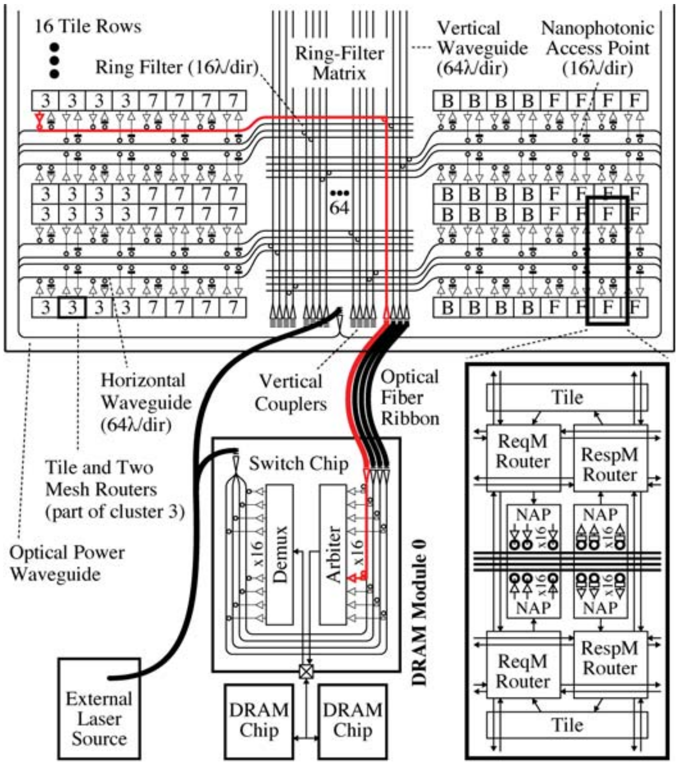
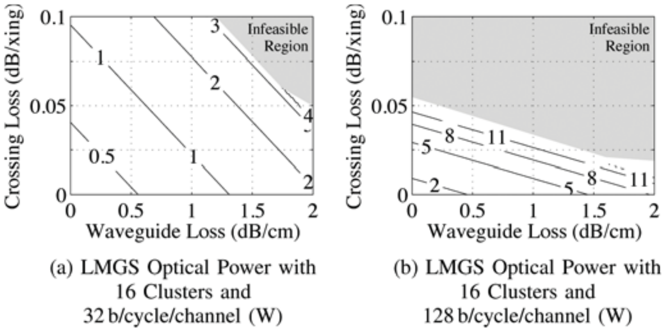


Fig. 17. Abstract physical layout for nanophotonic processor-to-DRAM network: Target (16,16,16,4) LMGS network with 256 tiles, 16 DRAM modules, and 16 clusters each with a 4-ary 2-D electrical mesh. Each tile is labeled with a hexadecimal number indicating its cluster. For simplicity the electrical mesh channels are only shown in the inset, the switch chip includes a single memory controller, each ring in the main figure actually represents 16 rings modulating or filtering 16 different wavelengths, and each optical power waveguide actually represents 16 waveguides (one per horizontal waveguide). ReqM = request mesh; RespM = response mesh; NAP = nanophotonic access point.

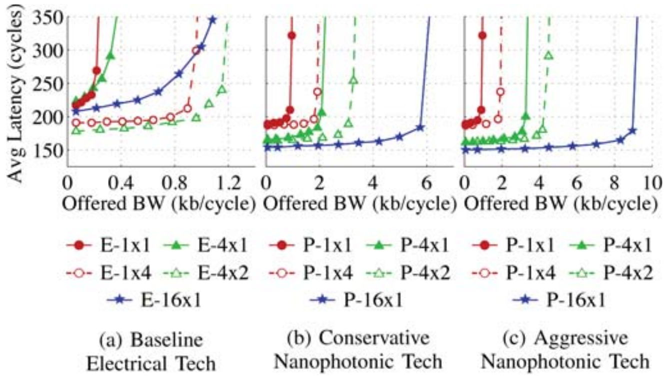


(a) LMGS Optical Power with 16 Clusters and 32 b/cycle/channel (W)

(b) LMGS Optical Power with 16 Clusters and 128 b/cycle/channel (W)

Fig. 18. Optical power for nanophotonic processor-to-DRAM networks: (16,16,16,4) LMGS topology with a ring-filter matrix layout and two different power constraints: (a) low power constraint and thus low aggregate bandwidth and (b) high-power constraint and thus high aggregate bandwidth.

Fig. 19 shows the latency as a function of offered bandwidth for 15 configurations. The name of each configuration indicates the technology used to implement the global channels (*E/P*), the number of clusters (*1/4/16*), and the over-provisioning factor (*x1/x2/x4*). Overall *E4x2* is the best electrical configuration and it consumes approximately 20 W near saturation. Just implementing the global channels with nanophotonics in a simple mesh topology results in a 2× improvement in throughput (e.g., *P1x4* versus *E1x4*). However, the full benefit of photonic interconnect only becomes apparent when we partition the on-chip mesh network into clusters and offload more traffic onto the energy-efficient nanophotonic channels. The *P16x1* configuration with aggressive projections can achieve a throughput of 9 kb/cycle (22 Tb/s), which is a ≈9× improvement over the best electrical configuration (*E4x2*) at comparable latency. The best optical configurations consume ≈16 W near saturation.



(a) Baseline Electrical Tech

(b) Conservative Nanophotonic Tech

(c) Aggressive Nanophotonic Tech

Fig. 19. Latency versus offered bandwidth for processor-to-DRAM networks: $E =$ electrical, $P =$ nanophotonics, *1/4/16*= number of clusters, *x1/x2/x4* = over-provisioning factor.

Table II shows the power breakdown for the *E4x2* and *P16x1* configurations near saturation. As expected, the majority of the power in the electrical configuration is spent on the global channels that connect the access points to the DRAM modules. By implementing these channels with energy-efficient photonic links we have a larger portion of our energy budget for higher-bandwidth on-chip mesh networks even after including the overhead for thermal tuning. The photonic configurations consume close to 15 W leaving 5 W for on-chip optical power dissipation as heat. Ultimately, photonics enables an 8–10× improvement in throughput at similar power consumption.

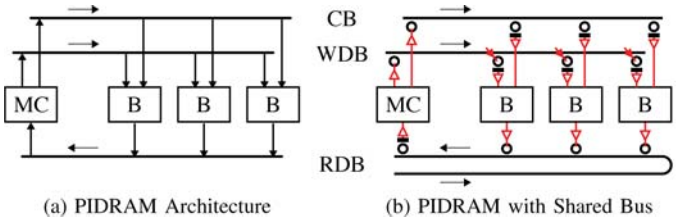
C. Design Themes

This case study suggests it is easier to show a compelling advantage for implementing an inter-chip network with nanophotonic devices, as compared to a purely intra-chip nanophotonic network. Additionally, our results show that once we have made the decision to use nanophotonics for chip-to-chip communication, it makes sense to push nanophotonics deeper into each chip (e.g., by using more clusters). This approach for using seamless intra-chip/inter-chip nanophotonic links is a general design theme that can help direct future directions for nanophotonic network research. Also notice that our nanophotonic LMGS network was able to achieve an order-of-magnitude improvement in throughput at a similar power constraint without resorting to more sophisticated nanophotonic devices, such as active optical switching. Again, we believe that using point-to-point nanophotonic channels offers the most promising approach for short term adoption of this technology. The choice of the ring-filter matrix layout was motivated by its regularity, short waveguides, and the need to aggregate all of the nanophotonic couplers in one place for simplified packaging. However, as shown in Fig. 18, this layout puts significant constraints on the maximum tolerable losses in waveguides and crossings. Alternate serpentine layouts can reduce the losses in crossings and waveguides, but

TABLE II
POWER BREAKDOWN FOR PROCESSOR-TO-DRAM NETWORKS

		Component Power (W)				
Configuration	**Throughput (kb/cycle)**	Mesh Routers	Mesh Channels	Global Channels	Thermal Tuning	**Total Power (W)**
E4x2	0.8	2.4	1.2	16.9	n/a	20.5
P16x1 (conservative)	6.0	5.9	3.2	3.1	3.9	16.2
P16x1 (aggressive)	9.0	8.0	4.5	1.5	2.6	16.7

These represent the best electrical and nanophotonic configurations. E4x2 is the electrical baseline with four clusters and an overprovisioning factor of two, while P16x1 uses nanophotonic global channels, 16 clusters, and no overprovisioning.



(a) PIDRAM Architecture (b) PIDRAM with Shared Bus

Fig. 20. PIDRAM architecture and microarchitecture: A single DRAM memory channel (MC) with three DRAM banks (B) interconnected with a command bus (CB), write-data bus (WDB), and read-data bus (RDB). (a) Logical topology for DRAM memory channel; (b) shared nanophotonic buses where optical power is broadcast to all banks along a shared physical medium.

would require couplers at multiple locations on the chip. An alternative would be to leverage the multiple nanophotonic devices layers available in monolithic back-end-of-line integration. Work by Biberman *et al.* has shown how multilayer deposited devices can significantly impact the feasibility of various network architectures [42], and this illustrates the need for a design process that iterates across the architecture, microarchitecture, and physical design levels.

VI. CASE STUDY #3: DRAM MEMORY CHANNEL

Both of the previous case studies assume a high-bandwidth and energy-efficient interface to off-chip DRAM. In this case study, we present photonically integrated DRAM (PIDRAM) which involves re-architecting the DRAM channel, chip, and bank to best use nanophotonics. This case study assumes a 32-nm DRAM technology, 512-bit access width, and timing constraints similar to those in contemporary Micron DDR3 SDRAM. More details on this case study can be found in [3].

A. Network Design

Fig. 20(a) illustrates the logical topology for a DRAM memory channel. A memory controller is used to manage a set of DRAM banks that are distributed across one or more DRAM chips. The memory system includes three logical buses: a command bus, a write-data bus, and a read-data bus. Fig. 20(b) illustrates a straightforward nanophotonic microarchitecture for a DRAM memory channel with a combination of SWBR, SWMR, and MWSR buses. The memory controller first broadcasts a command to all of the banks and each bank determines if it is the target bank for the command. For a PIDRAM write command, just the target bank will then tune-in its nanophotonic receiver on the write-data bus. For a PIDRAM read command, just the target bank will perform the read operation and then use its modulator on the read-data bus to send the data back to the memory controller. Unfortunately, the losses multiply together in this layout making the optical laser power an exponential function of the number of banks. Large coupler losses and exponential scaling combine to make the shared nanophotonic bus feasible only for connecting banks within a PIDRAM chip as opposed to connecting banks across PIDRAM chips. An alternative is to use a reader-/writer-sliced split nanophotonic bus layout, which divides the long shared bus into multiple branches. The split nature of the bus means that the total laser power is roughly a linear function of the number of banks at the cost of additional splitter and combiner losses in the memory controller. It also reduces the effective bandwidth density of the nanophotonic bus, by increasing the number of fibers for the same bandwidth. To further reduce the required optical power, we can use a reader-/writer-sliced guided nanophotonic bus layout. Since the optical power is always guided down a single branch, the total laser power is roughly constant and independent of the number of banks. The optical loss overhead due to the nanophotonic demultiplexers and the reduced bandwidth density due to the branching make a guided nanophotonic bus most attractive when working with relatively large per-bank optical losses.

Fig. 21 illustrates our proposed PIDRAM memory system. The figure shows a processor chip with multiple independent PIDRAM memory channels; each memory channel includes a memory controller and a PIDRAM DIMM, which in turn includes a set of PIDRAM chips. Each PIDRAM chip contains a set of banks, and each bank is completely contained within a single PIDRAM chip. We use a hybrid approach to implement each of the three logical buses. For example, both the write-data and read-data buses are implemented with a guided nanophotonic bus to actively guide optical power to a single PIDRAM chip within a DIMM, and then they are implemented with a shared nanophotonic bus to distribute the data within the PIDRAM chip.

Fig. 22 illustrates two abstract layouts for a PIDRAM chip. In the $P1$ layout shown in Fig. 22(a), the standard electrical I/O strip in the middle of the chip is replaced with a horizontal waveguide and multiple nanophotonic access points. The on-chip electrical H-tree command bus and vertical electrical data buses remain as in traditional electrical DRAM. In the $P2$ layout shown in Fig. 22(b), more of the on-chip portion of the data buses are implemented with nanophotonics to improve cross-chip energy-efficiency. The horizontal waveguides contain all of the wavelengths, and optically passive ring filter

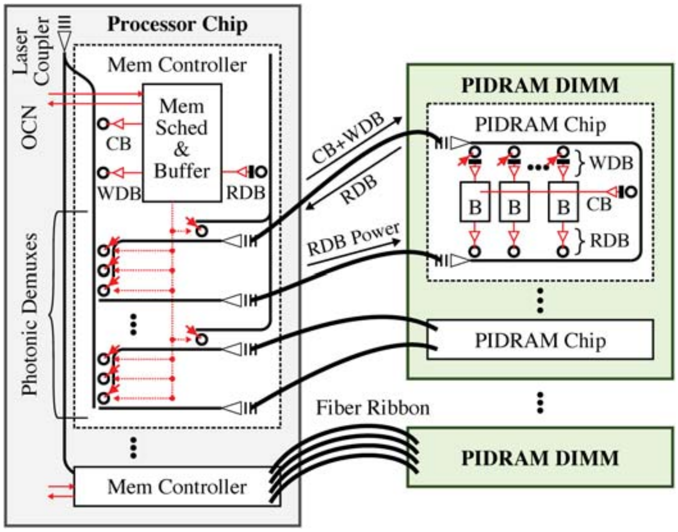


Fig. 21. PIDRAM memory system organization: Each PIDRAM memory channel connects to a PIDRAM DIMM via fiber ribbon. The memory controller manages the command bus (CB), write-data bus (WDB), and read-data bus (RDB). Nanophotonic demuxes guide power to the active PIDRAM chip. B = PIDRAM bank; each ring represents multiple rings for multiwavelength buses.

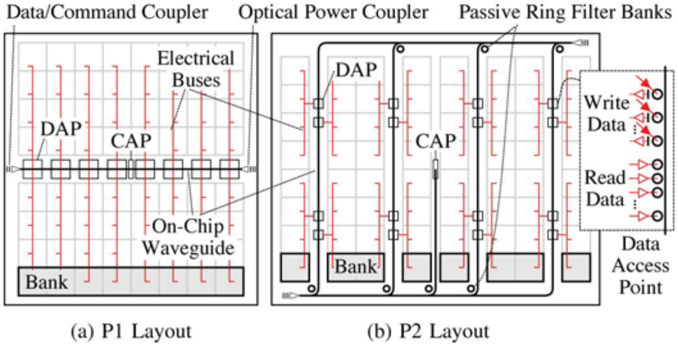


Fig. 22. Abstract physical layout for PIDRAM chip: two layouts with eight banks and eight array blocks per bank. The nanophotonic command bus ends at the command access point (CAP) and an electrical H-tree implementation (not shown) broadcasts control bits to all array blocks. (a) *P1* uses nanophotonic chip I/O for the data buses but fully electrical on-chip data bus implementations, and (b) *P2* uses nanophotonics to distribute the data bus to a group of four banks at the data access point (DAP).

banks at the bottom and top of the chip are used so that each vertical waveguide only contains a subset of the channel's wavelengths. More generally for a Pn layout, n indicates the number of partitions along each vertical electrical data bus.

B. Evaluation

To evaluate the energy efficiency and area trade-offs of the proposed DRAM channels, we use a heavily modified version of the Cacti-D DRAM modeling tool. We explore the space of possible results with both aggressive and conservative projections for nanophotonic devices. To quantify the performance of each DRAM design, we use a detailed cycle-level microarchitectural simulator. We use synthetic traffic patterns to issue loads and stores at a rate capped by the number of in-flight messages.

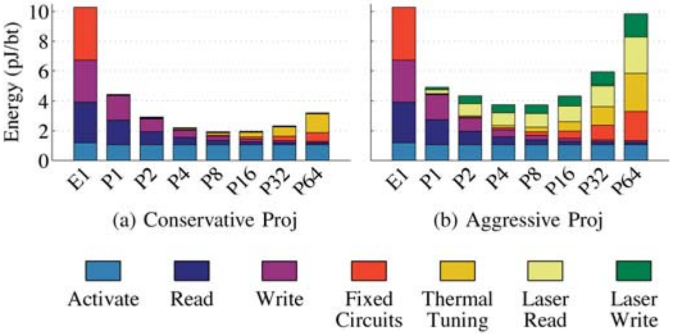


Fig. 23. Energy breakdown for DRAM memory channels: uniform random traffic with a peak bandwidth of ≈500 Gb/s. (a) Assumes conservative nanophotonic device projections, (b) assumes more aggressive projections. Fixed circuits energy includes clock and leakage. Read energy includes chip I/O read, cross-chip read, and bank read energy. Write energy includes chip I/O write, cross-chip write, and bank write energy. Activate energy includes chip I/O command, cross-chip row address energy, and bank activate energy.

Fig. 23 shows the energy-efficiency breakdown for a small subset of the studied configurations. Across all designs replacing the off-chip links with photonics is advantageous, but how far photonics is taken on chip is a much richer design space. To achieve the optimal energy efficiency requires balancing both the data-dependent and data-independent components of the overall energy. As shown in Fig. 23(a), *P1* spends the majority of the energy on intra-chip communication (write and read energy) because the data must traverse long global wires to get to each bank. Taking photonics all the way to each array block with *P64* minimizes the cross-chip energy, but results in a large number of photonic access points (since the photonic access points in *P1* are replicated 64 times in the case of *P64*), contributing to the large data-independent component of the total energy. The *P8* design better balances the data-dependent savings of using intra-chip photonics with the data-independent overheads due to electrical laser power, fixed circuit power, and thermal tuning power. Once the off-chip and cross-chip energies have been reduced (as in the *P8*), the activation energy becomes dominant. Additional experiments that increase the number of bits we read or write from each array core to 32 further reduces the activate energy cost, and overall this optimized design is 10× more energy efficient than the baseline electrical design.

In addition to these results, we also examined the energy as a function of utilization and the area overhead. The large fixed power in electrical DRAM interfaces helps mitigate the fixed power overhead in a nanophotonic DRAM interface at low utilization. The area overhead for a PIDRAM chip is actually quite minimal since any extra active area for the nanophotonic devices is compensated for the more area-efficient, higher-bandwidth array blocks. PIDRAM has comparable latency since this latency is mostly dominated by the intra-array-core access.

C. Design Themes

Point-to-point nanophotonic channels were a general theme in the first two case studies, but in this case study point-to-point channels were less applicable. DRAM memory channels usually use bus-based topologies to decouple bandwidth from capacity, so we use a limited form of active optical switching in reader-sliced SWMR and MWSR nanophotonic buses to reduce the required optical power. We see this as a gradual approach to nanophotonic network complexity: a designer can start with point-to-point nanophotonic channels, move to reader-sliced buses if there is a need to scale terminals but not the network bandwidth, and finally move to fully optical switching only if it is absolutely required to meet the desired application requirements. As in the previous case study, focusing on inter-chip nanophotonic networks and using a broad range of nanophotonic device parameters helps make a more compelling case for adopting this new technology compared to purely on-chip nanophotonic networks. Once we move to using nanophotonic inter-chip interfaces, there is a rich design space in how far into the chip we extend these nanophotonic links to help off-load global on-chip interconnect. In this specific application the fixed power overhead of nanophotonic interconnect is less of an issue owing to the significant amount of fixed power in the electrical baseline interfaces.

VII. Conclusion

Based on our experiences designing multiple nanophotonic networks and reviewing the literature, we have identified several common design guidelines that can aid in the design of new nanophotonic interconnection networks.

Clearly Specify the Logical Topology: A crisp specification of the logical network topology uses a simple high-level diagram to abstract away the details of the nanophotonic devices. Low-level microarchitectural schematics and physical layouts usually do a poor job of conveying the logical topology. For example, Fig. 11(b) and Fig. 12(b) have very similar physical layouts but drastically different logical topologies. In addition, it is easy to confuse passively WDM-routed wavelengths with true network routing; the former is analogous to routing wires at design time while the later involves dynamically routing packets at run time. A well-specified logical topology removes this ambiguity, helps others understand the design, enables more direct comparison to related proposals, and allows the application of well-know interconnection network techniques.

Iterate Through the Three-Levels of Design: There are many ways to map a logical bus or channel to nanophotonic devices and to integrate multiple stages of nanophotonic interconnect. Overly coupling the three design levels artificially limits the design space, and since this is still an emerging technology there is less intuition on which parts of the design space are the most promising. For example, restricting a design to only use optical switching eliminates some high-radix topologies. These high-radix topologies can, however, be implemented with electrical routers and point-to-point nanophotonic channels. As another example, only considering wavelength slicing or bus/channel slicing artificially constrains bus and channel bandwidths as opposed to using a combination of wavelength and bus/channel slicing. A final example was discussed in Section V; an honest evaluation of our results suggest that it may be necessary to revisit some of our earlier design decisions about the importance of waveguide crossings.

Use an Aggressive Electrical Baseline: There are many techniques to improve the performance and energy-efficiency of electrical chip-level networks, and most of these techniques are far more practical than adopting an emerging technology. Designers should assume fairly aggressive electrical projections in order to make a compelling case for chip-level nanophotonic interconnection networks. For example, with an aggressive electrical baseline technology in Section IV, it becomes more difficult to make a strong case for purely on-chip nanophotonic networks. However, even with aggressive electrical assumptions it was still possible to show significant potential in using seamless intra-chip/inter-chip nanophotonic links in Sections V and VI.

Assume a Broad Range of Device Parameters: Nanophotonics is an emerging technology, and any specific instance of device parameters are currently meaningless for realistic network design. This is especially true when parameters are mixed from different device references that assume drastically different fabrication technologies (e.g., hybrid integration versus monolithic integration). It is far more useful for network designers to evaluate a specific proposal over a range of device parameters. In fact, one of the primary goals of nanophotonic interconnection network research should be to provide feedback to device experts on the most important directions for improvement. In other words, are there certain device parameter ranges that are critical for achieving significant system-level benefits? For example, the optical power contours in Section V helped not only motivate alternative layouts but also an interest in very low-loss waveguide crossings.

Carefully Consider Fixed-Power Overheads: One of the primary disadvantages of nanophotonic devices are the many forms of fixed power (e.g., fixed transceiver circuit power, static thermal tuning power, optical laser power). These overheads impact the energy efficiency, on-chip power density, and system-level power; ignoring these overheads or only evaluating designs at high utilization can lead to overly optimistic results. For example, Section IV suggested that static power overhead could completely mitigate any advantage for purely on-chip nanophotonic networks, unless we assume relatively aggressive nanophotonic devices. This is in contrast to the study in Section VI, which suggests that even at low utilization, PIDRAM can achieve similar performance at lower power compared to projected electrical DRAM interfaces.

Motivate Nanophotonic Network Complexity: There will be significant practical risk in adopting nanophotonic technology. Our goal as designers should be to achieve the highest benefit with the absolute lowest amount of risk. Complex nanophotonic interconnection networks can require many types of devices and many instances of each type. These complicated designs significantly increase risk in terms of reliability, fabrication cost, and packaging issues. If we can achieve the same benefits with a much simpler network, then this increases the potential for realistic adoption of this emerging technology. Two of our case studies make use of just nanophotonic point-to-point channels, and our hope is that this simplicity can reduce risk. Once we decide to use nanophotonic point-to-point channels, then high-radix, low-diameter topologies seem like a promising direction for future research.

Acknowledgment

The authors would like to thank our co-authors on the various publications that served as the basis for the three case studies, including Y.-J. Kwon, S. Beamer, I. Shamim, and C. Sun. The authors would like to acknowledge the MIT nanophotonic device and circuits team, including J. S. Orcutt, A. Khilo, M. A. Popovič, C. W. Holzwarth, B. Moss, H. Li, M. Georgas, J. Leu, J. Sun, C. Sorace, F. X. Kärtner, J. L. Hoyt, R. J. Ram, and H. I. Smith.

References

[1] D. Vantrease *et al.*, "Corona: System implications of emerging nanophotonic technology," in *Int. Symp. Comput. Architecture*, Beijing, China, Jun. 2008, pp. 153–164.

[2] A. Hadke *et al.*, "OCDIMM: Scaling the DRAM memory wall using WDM-based optical interconnects," in *Symp. High-Performance Interconnects*, Stanford, CA, Aug. 2008, pp. 57–63.

[3] S. Beamer *et al.*, "Re-architecting DRAM memory systems with monolithically integrated silicon photonics," in *Int. Symp. Comput. Architecture*, Saint-Malo, France, Jun. 2010, pp. 129–140.

[4] A. N. Udipi *et al.*, "Combining memory and a controller with photonics through 3D-stacking to enable scalable and energy-efficient systems," in *Int. Symp. Comput. Architecture*, San Jose, CA, Jun. 2011, pp. 425–436.

[5] P. V. Mejia *et al.*, "Performance evaluation of a multicore system with optically connected memory modules," in *Int. Symp. Networks-on-Chip*, Grenoble, France, May 2011, pp. 215–222.

[6] N. Kırman *et al.*, "Leveraging optical technology in future bus-based chip multiprocessors," in *Int. Symp. Microarchitecture*, Orlando, FL, Dec. 2006, pp. 492–503.

[7] G. Kurian *et al.*, "ATAC: A 1000-core cache-coherent processor with on-chip optical network," in *Int. Conf. Parallel Architectures Compilat. Techn.*, Minneapolis, MN, Sep. 2010, pp. 477–488.

[8] M. Petracca *et al.*, "Photonic NoCs: System-level design exploration," *IEEE Micro*, vol. 29, no. 4, pp. 74–77, Jul./Aug. 2009.

[9] S. Pasricha and N. Dutt, "ORB: An on-chip optical ring bus communication architecture for multi-processor systems-on-chip," in *Asia South Pacific Design Automat. Conf.*, Seoul, Korea, Jan. 2008, pp. 789–794.

[10] P. Koka *et al.*, "Silicon-photonic network architectures for scalable, power-efficient multi-chip systems," in *Int. Symp. Comput. Architecture*, Saint-Malo, France, Jun. 2010, pp. 117–128.

[11] R. Morris and A. Kodi, "Exploring the design of 64 & 256 core power efficient nanophotonic interconnect," *J. Sel. Topics Quant. Electron.*, vol. 16, no. 5, pp. 1386–1393, Sep./Oct. 2010.

[12] C. Batten *et al.*, "Building manycore processor-to-DRAM networks with monolithic silicon photonics," in *Symp. High-Performance Interconnects*, Stanford, CA, Aug. 2008, pp. 21–30.

[13] C. Batten *et al.*, "Building manycore processor-to-DRAM networks with monolithic CMOS silicon photonics," *IEEE Micro*, vol. 29, no. 4, pp. 8–21, Jul./Aug. 2009.

[14] H. Gu, J. Xu, and W. Zhang, "A low-power fat-tree-based optical network-on-chip for multiprocessor system-on-chip," in *Design, Automat. Test Eur.*, Nice, France, May 2009, pp. 3–8.

[15] A. Joshi *et al.*, "Silicon-photonic Clos networks for global on-chip communication," in *Int. Symp. Networks-on-Chip*, San Diego, CA, May 2009, pp. 124–133.

[16] Y. Pan *et al.*, "Firefly: Illuminating on-chip networks with nanophotonics," in *Int. Symp. Comput. Architecture*, Austin, TX, Jun. 2009, pp. 429–440.

[17] Y.-H. Kao and J. J. Chao, "BLOCON: A bufferless photonic clos network-on-chip architecture," in *Int. Symp. Networks-on-Chip*, Pittsburgh, PA, May 2011, pp. 81–88.

[18] A. Shacham, K. Bergman, and L. P. Carloni, "Photonic networks-on-chip for future generations of chip multiprocessors," *IEEE Trans. Comput.*, vol. 57, no. 9, pp. 1246–1260, Sep. 2008.

[19] M. J. Cianchetti, J. C. Kerekes, and D. H. Albonesi, "Phastlane: A rapid transit optical routing network," in *Int. Symp. Comput. Architecture*, Austin, TX, Jun. 2009, pp. 441–450.

[20] Z. Li *et al.*, "Iris: A hybrid nanophotonic network design for high-performance and low-power on-chip communication," *J. Emerg. Technol. Comput. Syst.*, vol. 7, no. 2, p. 8, Jun. 2011.

[21] J. S. Orcutt *et al.*, "Demonstration of an electronic photonic integrated circuit in a commercial scaled bulk CMOS process," in *Conf. Lasers Electro-Optics*, San Jose, CA, May 2008, pp. 1–2.

[22] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45 nm early design exploration," *IEEE Trans. Electron Devices*, vol. 53, no. 11, pp. 2816–2823, Nov. 2006.

[23] B. Kim and V. Stojanović, "Characterization of equalized and repeated interconnects for NoC applications," *IEEE Design Test Comput.*, vol. 25, no. 5, pp. 430–439, Sep. 2008.

[24] C. W. Holzwarth *et al.*, "Localized substrate removal technique enabling strong-confinement microphotonics in bulk Si CMOS processes," in *Conf. Lasers Electro-Optics*, San Jose, CA, May 2008, pp. 1–2.

[25] J. S. Orcutt *et al.*, "Nanophotonic integration in state-of-the-art CMOS foundaries," *Opt. Exp.*, vol. 19, no. 3, pp. 2335–2346, Jan. 2011.

[26] J. S. Orcutt *et al.*, "Photonic integration in a commercial scaled bulk-CMOS process," in *Int. Conf. Photon. Switch.*, Pisa, Italy, Sep. 2009, pp. 1–2.

[27] J. S. Orcutt *et al.*, "Low-loss polysilicon waveguides suitable for integration witin a high-volume polysilicon process," in *Conf. Lasers Electro-Optics*, Baltimore, MD, May 2011, pp. 1–2.

[28] J. C. Leu and V. Stojanović, "Injection-locked clock receiver for monolithic optical link in 45 nm," in *Asian Solid-State Circuits Conf.*, Jeju, Korea, Nov. 2011, pp. 149–152.

[29] M. Georgas *et al.*, "A monolithically-integrated optical receiver in standard 45 nm SOI," in *Eur. Solid-State Circuits Conf.*, Sep. 2011.

[30] M. Georgas *et al.*, "Addressing link-level design tradeoffs for integrated photonic interconnects," in *Custom Integrated Circuits Conf.*, San Jose, Sep. 2011, pp. 1–8.

[31] J. Xue *et al.*, "An intra-chip free-space optical interconnect," in *Int. Symp. Comput. Architecture*, Saint-Malo, Jun. 2010, pp. 94–105.

[32] A. Abousamra, R. Melhem, and A. Jones, "Two-hop free-space based optical interconnects for chip multiprocessors," in *Int. Symp. Networks-on-Chip*, Pittsburgh, PA, May 2011, pp. 89–96.

[33] Y. Pan, J. Kim, and G. Memik, "FlexiShare: Energy-efficient nanophotonic crossbar architecture through channel sharing," in *Int. Symp. High-Performance Comput. Architecture*, Bangalore, India, Jan. 2010, pp. 1–12.

[34] N. Kırman and J. F. Martíinez, "A power-efficient all-optical on-chip interconnect using wavelength-based oblivious routing," in *Int. Conf. Architectural Support Programm. Languages Operat. Syst.*, Pittsburgh, PA, Mar. 2010, pp. 15–27.

[35] S. Koohi, M. Abdollahi, and S. Hessabi, "All-optical wavelength-routed NoC based on a novel hierarchical topology," in *Int. Symp. Networks-on-Chip*, Pittsburgh, PA, May 2011, pp. 97–104.

[36] D. Vantrease *et al.*, "Light speed arbitration and flow control for nanophotonic interconnects," in *Int. Symp. Microarchitecture*, New York, Dec. 2009, pp. 304–315.

[37] N. Sherwood-Droz *et al.*, "Optical 4 × 4 hitless silicon router for optical networks-on-chip," *Opt. Exp.*, vol. 16, no. 20, pp. 15 915–15 922, Sep. 2008.

[38] A. W. Poon *et al.*, "Cascaded microresonator-based matrix switch for silicon on-chip optical interconnection," *Proc. IEEE*, vol. 97, no. 7, pp. 1216–1238, Jul. 2009.

[39] M. J. Cianchetti and D. H. Albonesi, "A low-latency, high-throughput on-chip optical router architecture for future chip multiprocessors," *J. Emerg. Technol. Comput. Syst.*, vol. 7, no. 2, p. 9, Jun. 2011.

[40] S. L. Beux *et al.*, "Optical ring network-on-chip (ORNoC): Architecture and design methodology," in *Design, Automat., Test Eur.*, Grenoble, France, Mar. 2011, pp. 1–6.

[41] L. Zhou *et al.*, "Design and evaluation of an arbitration-free passive optical crossbar for on-chip interconnection networks," *Appl. Phys. A: Mater. Sci. Process.*, vol. 95, no. 4, pp. 1111–1118, Jun. 2009.

[42] A. Biberman *et al.*, "Photonic network-on-chip architectures using multilayer deposited silicon materials for high-performance chip multi-processors," *J. Emerg. Technol. Comput. Syst.*, vol. 7, no. 2, p. 7, Jun. 2011.

Christopher Batten (S'05–M'09) received the B.S. degree in electrical engineering as a Jefferson Scholar at the University of Virginia, Charlottesville, in 1999, the M.Phil. degree in engineering as a Churchill Scholar at the University of Cambridge, Cambridge, U.K., in 2000, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 2010.

He is an Assistant Professor in the School of Electrical and Computer Engineering at Cornell University, Ithaca, NY. His research interests include energy-efficient parallel computer architecture for both high-performance and embedded applications.

Dr. Batten received the National Science Foundation CAREER Award and the Defense Advanced Research Projects Agency's Young Faculty Award in 2012.

Ajay Joshi (S'99–M'07) received the B.Eng. degree in computer engineering from the University of Mumbai, Mumbai, India, in 2001, and the M.S. and Ph.D. degrees in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 2003 and 2006, respectively.

He is an Assistant Professor in the Electrical and Computer Engineering Department at Boston University, Boston, MA. His research interests span across various aspects of VLSI design including circuits and systems for communication and computation, and emerging device technologies including silicon photonics and memristors. Prior to joining Boston University, he worked as a postdoctoral researcher in the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology, Cambridge.

Dr. Joshi received the National Science Foundation CAREER Award in 2012.

Vladimir Stojanović (S'96–M'04) received the Dipl. Ing. degree from the University of Belgrade, Serbia in 1998, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 2005.

He is the Emanuel E. Landsman Associate Professor of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology, Cambridge. His research interests include design, modeling, and optimization of integrated systems, from CMOS-based VLSI blocks and interfaces to system design with emerging devices like NEM relays and silicon-photonics. He is also interested in design and implementation of energy-efficient electrical and optical networks, and digital communication techniques in high-speed interfaces and high-speed mixed-signal IC design. He was also with Rambus, Inc., Los Altos, CA, from 2001 through 2004.

Dr. Stojanović received the 2006 IBM Faculty Partnership Award, and the 2009 National Science Foundation CAREER Award as well as the 2008 ICCAD William J. McCalla, 2008 IEEE TRANSACTIONS ON ADVANCED PACKAGING, and 2010 ISSCC Jack Raper best paper awards.

Krste Asanović (S'90–M'98–SM'12) received the B.A. degree in electrical and information sciences from Cambridge University, Cambridge, U.K., in 1987, and the Ph.D. degree in computer science from the University of California, Berkeley, in 1998.

He is an Associate Professor in the Electrical Engineering and Computer Sciences Department at the University of California at Berkeley. His research interests include computer architecture, VLSI design, and parallel programming and run-time systems. In 2007, he co-founded the Berkeley Parallel Computing Laboratory, which is taking a vertically integrated approach to develop a new software and hardware stack for parallel computing systems. He was an Assistant and Associate Professor of electrical engineering and computer science at the Massachusetts Institute of Technology, Cambridge, from 1998 to 2007.