

Chapter 3

Designing Chip-Level Nanophotonic Interconnection Networks

Christopher Batten, Ajay Joshi, Vladimir Stojanović, and Krste Asanović

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 chapter discusses the approach we have used when designing such networks, and provides a foundation for designing new networks. We begin by reviewing the basic nanophotonic devices before briefly discussing our own silicon-photonics technology that enables monolithic integration in a standard CMOS process. We then outline design issues and categorize previous proposals in the literature 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 tile-to-tile network, processor-to-DRAM network, and DRAM memory channel to illustrate this design process.

C. Batten (✉)

School of Electrical and Computer Engineering,
College of Engineering, Cornell University, 323 Rhodes Hall, Ithaca, NY 14853, USA
e-mail: cbatten@cornell.edu

A. Joshi

Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215, USA

V. Stojanović

Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

K. Asanović

Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA 94720, USA

Keywords Nanophotonics • Optical interconnect • Multicore/manycore processors • Interconnection networks • Network architecture

Introduction

Today's graphics, network, embedded, and server processors already contain many cores on one chip, and this number will 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 network architectures that use nanophotonic devices to potentially improve performance and energy efficiency. These proposals explore different single-stage topologies from buses [9, 14, 29, 53, 74, 76] to crossbars [39, 44, 64, 65, 76] and different multistage topologies from quasi-butterflies [6, 7, 26, 32, 34, 41, 56, 63] to tori [18, 48, 69]. Note that we specifically focus on chip-level networks as opposed to cluster-level optical networks used in high-performance computing and data-centers. Most proposals use different routing algorithms, flow control mechanisms, optical wavelength organizations, and physical layouts. While this diversity makes for an exciting new research field, it also makes it difficult to see relationships between different proposals and to identify promising directions for future network design.

In previous work, we briefly described our approach for designing nanophotonic interconnection networks, which is based on thinking of the design at three levels: the architectural level, the microarchitectural level, and the physical level [8]. In this chapter, we expand on this earlier description, provide greater detail on design trade-offs at each level, and categorize previous proposals in the literature. *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. This level of design 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.

This chapter begins by briefly reviewing the underlying nanophotonic technology, before describing in more detail our three-level design process and surveying recent proposals in this area. The chapter then presents three case studies to illustrate this design process and to demonstrate the potential for nanophotonic interconnection networks, before concluding with several general design themes that can be applied when designing future nanophotonic interconnection networks.

Nanophotonic Technology

This section briefly reviews the basic devices used to implement nanophotonic interconnection networks, before discussing the opportunities and challenges involved with this emerging technology. See [10, 68] for a more detailed review of recent work on nanophotonic devices. This section also describes in more detail the specific nanophotonic technology that we assume for the case studies presented later in this chapter.

Overview of Nanophotonic Devices

Figure 3.1 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 chip A to chips B and C. 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 local photodetector. The photodetector turns absorbed light into current, which is sensed by the electrical amplifier. Ultimately, the example in Fig. 3.1 creates four point-to-point channels that connect the four inputs (I_1 – I_4) to the four outputs (O_1 – O_4), such that input I_1 sends data to output O_1 , input I_2 sends

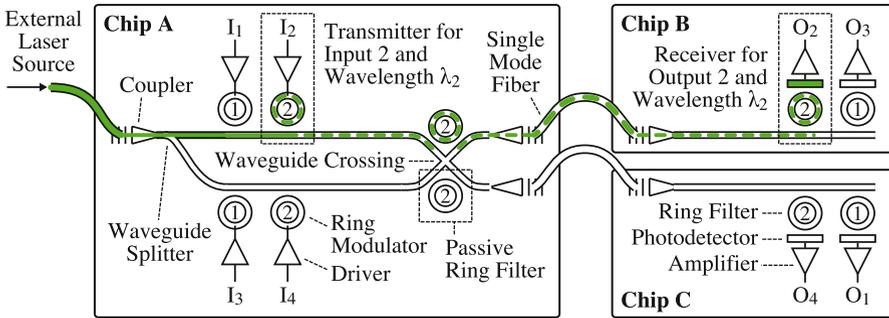


Fig. 3.1 Nanophotonic devices. Four point-to-point nanophotonic channels implemented with wavelength-division multiplexing. Such channels can be used for purely intra-chip communication or seamless intra-chip/inter-chip communication. Number inside ring indicates resonant wavelength; each input (I_1 – I_4) is passively connected to the output with the corresponding subscript (O_1 – O_4); link corresponding to $I_2 \rightarrow O_2$ on wavelength λ_2 is highlighted (from [8], courtesy of IEEE)

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. 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. 3.1, 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 μm radius) resulting in reduced area and power consumption. Although not shown in Fig. 3.1, 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. This actively steers the light to one of many possible outputs. 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. 3.1 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.

Most recent nanophotonic interconnection designs use the devices shown in Fig. 3.1, but some proposals also use alternative devices such as vertical cavity surface emitting lasers combined with free-space optical channels [1, 78] or planar waveguides [48]. This chapter focuses on the design of networks with the more common ring-based devices and linear waveguides, and we leave a more thorough treatment of interconnect network design using alternative devices for future work.

Nanophotonic Technology Opportunities and Challenges

Nanophotonic interconnect can potentially provide significant advantages in terms of bandwidth density and energy efficiency when compared to long electrical intra-chip and inter-chip interconnect [55]. The primary bandwidth density advantage comes from packing dozens of wavelengths into the same waveguide or fiber, with each wavelength projected to operate at 5–10 Gb/s for purely intra-chip communication and 10–40 Gb/s for purely inter-chip communication. With waveguide pitches on the order of a couple micron and fiber coupling pitches on the order of tens of microns, this can translate into a tremendous amount of intra- and inter-chip bandwidth. The primary energy-efficiency advantage comes from ring-modulator transceivers that are projected to require sub-150 fJ/bit of data-dependent electrical energy regardless of the link length and fanout. This improvement in bandwidth density and energy efficiency can potentially be achieved with comparable or improved latency, making nanophotonics a viable disruptive technology for chip-level communication.

Of course, there are many practical challenges to realizing this emerging technology including economic feasibility, effective system design, manufacturing issues, reliability concerns, and compromising various overheads [22]. We now briefly discuss three of the most pressing challenges: opto-electrical integration, temperature and process variability, and optical power overhead.

Opto-Electrical Integration

Tightly integrating optical and electrical devices is critical for achieving the potential bandwidth density and energy efficiency advantages of nanophotonic devices. There are three primary approaches for opto-electrical integration in intra-chip and inter-chip interconnection networks: hybrid integration, monolithic back-end-of-line (BEOL) integration, and monolithic front-end-of-line (FEOL) integration.

Hybrid Integration. The highest-performing optical devices are fabricated through dedicated processes customized for building such devices. These optical chips can then be attached to a micro-electronic chip fabricated with a standard electrical CMOS process through package-level integration [2], flip-chip bonding the two wafers/chips face-to-face [73, 81], or 3D integration with through-silicon vias [35]. Although this approach is feasible using integration technologies available currently or in the near future, it requires inter-die electrical interconnect (e.g., micro-bumps or through-silicon vias) to communicate between the micro-electronic and active optical devices. It can be challenging to engineer this inter-die interconnect to avoid mitigating the energy efficiency and bandwidth density advantages of chip-level nanophotonics.

Monolithic BEOL Integration. Nanophotonic devices can be deposited on top of the metal interconnect stack using amorphous silicon [38], poly-silicon [67], silicon nitride [5], germanium [52], and polymers [15, 33]. Ultimately, a combination of these materials can be used to create a complete nanophotonic link [79]. Compared to

hybrid integration, BEOL integration brings the optical devices closer to the microelectronics which can improve energy efficiency and bandwidth density. BEOL integration does not require changes to the front end, does not consume active area, and can provide multiple layers of optical devices (e.g., multi-layer waveguides). Although some specialized materials can be used in BEOL integration, the nanophotonic devices must be deposited within a strict thermal processing envelope and of course require modifications to the final layers of the metal interconnect stack. This means that BEOL devices often must trade-off bandwidth density for energy efficiency (e.g., electro-optic modulator devices [79] operate at relatively high drive voltages to achieve the desired bandwidth and silicon-nitride waveguides have large bending losses limiting the density of photonic devices). BEOL integration is suitable for use with both SOI and bulk CMOS processes, and can potentially also be used in other applications such as for depositing optics on DRAM or FLASH chips.

Monolithic FEOL Integration. Photonic devices without integrated electrical circuitry have been implemented in monocrystalline silicon-on-insulator (SOI) dies with a thick layer of buried oxide (BOX) [23, 49], and true monolithic FEOL integration of electrical and photonic devices have also been realized [25, 28]. Thin-BOX SOI is possible with localized substrate removal under the optical devices [31]. On the one hand, FEOL integration can support high-temperature process modifications and enables the tightest possible coupling to the electrical circuits, but also consumes valuable active area and requires modifications to the sensitive front-end processing. These modifications can include incorporating pure germanium or high-percentage silicon-germanium on the active layer, additional processing steps to reduce waveguide sidewall roughness, and improving optical cladding with either a custom thick buried-oxide or a post-processed air gap under optical devices. In addition, FEOL integration usually requires an SOI CMOS process, since the silicon waveguides are implemented in the same silicon film used for the SOI transistors. There has, however, been work on implementing FEOL polysilicon nanophotonic devices with localized substrate removal in a bulk process [58, 61].

Process and Temperature Variation

Ring-resonator devices have extremely high Q-factors, which enhance the electro-optical properties of modulators and active filters and enables dense wavelength division multiplexing. Unfortunately, this also means small unwanted changes in the resonance can quickly shift a device out of the required frequency operating range. Common sources of variation include process variation that can result in unwanted ring geometry variation within the same die, and thermal variation that can result in spatial and temporal variation in the refractive index of silicon-photonic devices. Several simulation-based and experimental studies have reported that a 1 nm variation in the ring width can shift a ring's resonance by approximately 0.5 nm [47, 70], and a single degree change in temperature can shift a ring's resonance by approximately 0.1 nm [22, 47, 51]. Many nanophotonic network proposals assume tens of wavelengths per waveguide [6, 32, 63, 74, 76], which results in a channel spacing of less than 1 nm (100 GHz). This means

ring diameter variation of 2 nm or temperature variation of 10 °C can cause a ring resonator to filter the incorrect neighboring wavelength.

Process Variation. A recent study of FEOL devices fabricated in 0.35 μm found that intra-die variation resulted in a 100 GHz change in ring resonance, and intra-wafer variation resulted in a 1 THz change in ring resonance across the 300 mm wafer [84]. A different study of FEOL devices in a much more advanced technology generation found a mean relative mismatch of 31 GHz within a multi-ring filter bank but a much more significant mean absolute mismatch of 600 GHz inter-die variation [58]. These results suggest that design-time frequency matching for rings in close proximity might be achievable at advanced technology nodes, but that frequency matching rings located far apart on the same die or on different dies might require some form of resonance frequency tuning.

Thermal Variation. Spatial and temporal temperature gradients are more troubling, since these can be difficult to predict; greater than 10 °C variation is common in modern high-performance microprocessors. Simulation-based chip-level models suggest maximum temperature differentials up to 17 °C in space [47, 72] and up to 28 °C in time across different benchmarks [47]. An experimental-based study measured various blocks in an AMD Athlon microprocessor increasing from an idle ambient temperature of 45 °C to a steady-state temperature of 70 °C in the L1 data cache and 80 °C in the integer instruction scheduler, and measured peak spatial variation at approximately 35 °C between the heavily used blocks and idle blocks in the chip [54].

There have been a variety of device-level proposals for addressing these challenges including injecting charge to use the electro-optic effect to compensate for variation [51] (can cause self-heating and thermal runaway), adding thermal “micro-heaters” to actively maintain a constant device temperature or compensate for process variation [3, 21, 77] (requires significant static tuning power), using athermal device structures [27] (adds extra area overhead), and using extra polymer materials for athermal devices [16, 83] (not necessarily CMOS compatible). There has been relatively less work studying variation in CMOS-compatible nanophotonic devices at the system-level. Some preliminary work has been done on integrating thermal modeling into system-level nanophotonic on-chip network simulators [57], and studying run-time thermal management techniques for a specific type of nanophotonic on-chip network [47]. Recent work has investigated the link-level implications of local thermal tuning circuitry and adding extra rings to be able to still receive wavelengths even after they have shifted due to thermal drift [24].

Optical Power Overhead

A nanophotonic link consumes several types of data-independent power: fixed power in the electrical portions of the transmitters and receivers (e.g., clock and static power), tuning power to compensate for process and thermal variation, and optical laser power. The laser power depends on the amount of optical loss that any given wavelength experiences as it travels from the laser, through the various devices

shown in Fig. 3.1, and eventually to the photodetector. In addition to the photonic device losses, there is also a limit to the total amount of optical power that can be transmitted through a waveguide without large non-linear losses. High optical losses per wavelength necessitate distributing those wavelengths across many waveguides (increasing the overall area) to stay within this non-linearity limit. Minimizing optical loss is a key device design objectives, and meaningful system-level design must take into account the total optical power overhead.

MIT Monolithic FEOL Nanophotonic Technology

In the case studies presented later in this chapter, we will be assuming a monolithic FEOL integration strategy. Our approach differs from other integration strategies, since we attempt to integrate nanophotonics into state-of-the-art bulk-CMOS micro-electronic chips with no changes to the standard CMOS fabrication process. In this section, we provide a brief overview of the specific technology we are developing with our colleagues at the Massachusetts Institute of Technology. We use our experiences with a 65 nm test chip [60], our feasibility studies for a prototype 32 nm process, predictive electrical device models [80], and interconnect projections [36] 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 chapter can be found in [30, 58–61], although the technology is rapidly evolving such that more recent device-level work uses more advanced device and circuit techniques [24, 25, 46]. Details about the specific technology assumptions for each case study can be found in our previous system-level publications [6, 7, 9, 32].

Waveguide. To avoid process changes, we design our waveguides in the polysilicon layer on top of the shallow-trench isolation in a standard bulk CMOS process (see Fig. 3.2a). Unfortunately, the shallow-trench oxide is too thin to form an effective cladding and shield the core from optical-mode leakage into the silicon substrate. We have developed a novel self-aligned post-processing procedure to etch away the silicon substrate underneath the waveguide forming an air gap. A reasonably deep air gap provides a very effective optical cladding. For our case studies, we assume eight-waveguide bundles can use the same air gap with a 4- μm waveguide pitch and an extra 5- μm of spacing on either side of the bundle. We estimate a time-of-flight latency of approximately 10.5 ps/mm which enables raw interconnect latencies for crossing a 400- mm^2 chip to be on the order of one to three cycles for a 5-GHz core clock frequency.

Transmitter. Our transmitter design is similar to past approaches that use minority charge-injection to change the resonant frequency of ring modulators [50]. Our racetrack modulator design is implemented by doping the edges of a polysilicon modulator structure creating a lateral PiN diode with undoped polysilicon as the intrinsic region (see Fig. 3.2b). Our device simulations indicate that with polysilicon carrier lifetimes of 0.1–1 ns it is possible to achieve sub-100 fJ per bit time (fJ/bt) modulator driver energy for random data at up to 10 Gb/s with advanced digital

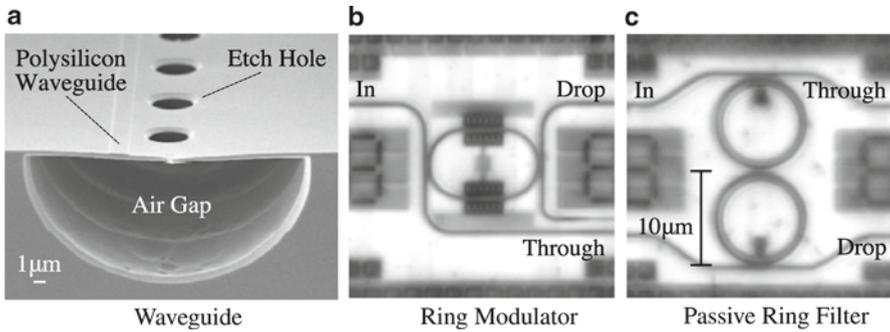


Fig. 3.2 MIT monolithic FEOL nanophotonic devices. (a) Polysilicon waveguide over SiO_2 film with an air gap etched into the silicon substrate to provide optical cladding; (b) polysilicon ring modulator that uses charge injection to modulate a single wavelength: without charge injection the resonant wavelength is filtered to the “drop” port while all other wavelengths continue to the “through” port; with charge injection, the resonant frequency changes such that no wavelengths are filtered to the “drop” port; (c) cascaded polysilicon rings that passively filter the resonant wavelength to the “drop” port while all other wavelengths continue to the “through” port (adapted from [7], courtesy of IEEE)

equalization circuits. To avoid robustness and power issues from distributing a multiple-GHz clock to hundreds of transmitters, we propose implementing an optical clock delivery scheme using a simple single-diode receiver with duty-cycle correction. We estimate the serialization and driver circuitry will consume less than single-cycle at a 5-GHz core clock frequency.

Passive Filter. We use polysilicon passive filters with two cascaded rings for increased frequency roll-off (see Fig. 3.2c). As mentioned earlier in this section, the ring’s resonance is sensitive to temperature and requires active thermal tuning. Fortunately, the etched air gap under the ring provides isolation from the thermally conductive substrate, and we add in-plane polysilicon heaters inside most rings to improve heating efficiency. Thermal simulations suggest that we will require 40–100 μW of static power for each double-ring filter assuming a temperature range of 20 K. These ring filters can also be designed to behave as active filters by using charge injection as in our transmitters, except at lower data rates.

Receiver. The lack of pure Ge presents a challenge for mainstream bulk CMOS processes. We use the embedded SiGe (20–30% Ge) in the p-MOSFET transistor source/drain regions to create a photodetector operating at around 1200 nm. Simulation results show good capacitance (less than 1 fF/ μm) and dark current (less than 10 fA/ μm) at near-zero bias conditions, but the sensitivity of the structure needs to be improved to meet our system specifications. In advanced process nodes, the responsivity and speed should improve through better coupling between the waveguide and the photodetector in scaled device dimensions, and an increased percentage of Ge for device strain. Our photonic receiver circuits would use the same optical clocking scheme as our transmitters, and we estimate that the entire receiver will consume less than 50 fJ/bt for random data. We estimate the deserialization and driver circuitry will consume less than single-cycle at a 5-GHz core clock frequency.

Based on our device simulations and experiments we project that it may be possible to multiplex 64 wavelengths per waveguide at a 60-GHz spacing, and that by interleaving wavelengths traveling in opposite directions (which helps mitigate interference) we can possibly have up to 128 wavelengths per waveguide. With a 4- μm waveguide pitch and 64–128 wavelengths per waveguide, we can achieve a bandwidth density of 160–320 Gb/s/ μm for intra-chip nanophotonic interconnect. With a 50- μm fiber coupler pitch, we can achieve a bandwidth density of 12–25 Gb/s/ μm for inter-chip nanophotonic interconnect. Total link latencies including serialization, modulation, time-of-flight, receiving, and deserialization could range from three to eight cycles depending on the link length. We also project that the total electrical and thermal on-chip energy for a complete 10 Gb/s nanophotonic intra-chip or inter-chip link (including a racetrack modulator and a double-ring filter at the receiver) can be as low as 100–250 fJ/bt for random data. These projections suggest that optical communication should support significantly higher bandwidth densities, improved energy efficiency, and competitive latency compared to both optimally repeated global intra-chip electrical interconnect (e.g., [36]) and projected inter-chip electrical interconnect.

Designing 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. Each level of design enables its own set of qualitative and quantitative analysis and helps motivate design decisions at both higher and lower levels. Although these levels can help focus our design effort, network design is inherently an iterative process with a designer moving between levels as necessary to meet the application requirements.

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 input terminals to a set of output terminals through a collection of buses and routers interconnected by point-to-point channels. Symmetric topologies have an equal number of input and output terminals, usually denoted as N . Figure 3.3 illustrates several topologies for a 64-terminal symmetric network ranging from single-stage global buses and crossbars to multi-stage butterfly and torus topologies (see [20] for

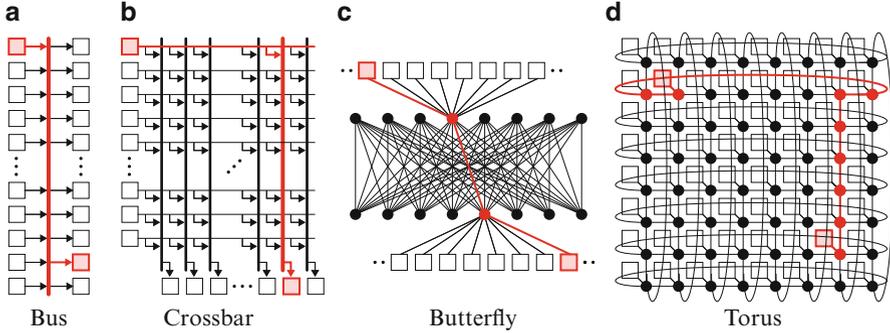


Fig. 3.3 Logical topologies for various 64 terminal networks. (a) 64-writer/64-reader single global bus; (b) 64×64 global non-blocking crossbar; (c) 8-ary 2-stage butterfly; (d) 8-ary 2-dimensional torus. Squares: input and/or output terminals; dots: routers; in (c) inter-dot lines: uni-directional channels; in (d) inter-dot lines: two channels in opposite directions (from [8], courtesy of IEEE)

a more extensive review of logical network topologies, and see [4] for a study specifically focused on intra-chip networks). 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 bandwidth and latency, but we can also begin to explore how more realistic routing algorithms might impact the network’s performance. When designing nanophotonic interconnection networks, it is also useful to begin by characterizing state-of-the-art electrical networks. Developing realistic electrical baseline architectures early in the design process can help motivate the best opportunities for leveraging nanophotonic devices. This subsection discusses a range of topologies used in nanophotonic interconnection networks.

A global bus is perhaps the simplest of logical topologies, and involves N input terminals arbitrating for a single shared medium so that they can communicate with one of $N-1$ output terminals (see Fig. 3.3a). Buses can make good use of scarce wiring resources, serialize messages which can be useful for some higher-level protocols, and enable one input terminal to easily broadcast a message to all output terminals. 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 DRAM memory channels [9, 29, 53, 74, 76] (discussed further in case study #3), although there have also been proposals for specialized nanophotonic broadcast buses to improve the performance of application barriers [14] and cache-coherence protocols [76]. Multiple global buses can be used to improve system throughput, and such topologies have also been designed using nanophotonic devices [62].

A global crossbar topology is made up of N buses with each bus dedicated to a single terminal (see Fig. 3.3b). Such topologies present a simple performance model

to software and can sustain high-performance owing to their strictly non-blocking connectivity. This comes at the cost, however, of many global buses crossing the network bisection and long global arbitration delays. Nanophotonic crossbar topologies have been particularly popular in the literature [39, 40, 44, 64, 65, 76], and we will see in the following sections that careful design at the microarchitectural and physical levels is required to help mitigate some of 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 where radix- k routers are arranged in n stages with N/k routers per stage (see Fig. 3.3c). 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 latencies and balancing channel load. For example, a butterfly topology lacks any form of path diversity resulting in poor performance on some traffic patterns. Nanophotonic topologies have been proposed that are similar in spirit to the butterfly topology for multichip-module networks [41], on-chip networks [56], and processor-to-DRAM networks [6, 7]. The later is discussed further as a case study in section "Case Study #2: Manycore Processor-to-DRAM Network." In these networks, the lack of path diversity may not be a problem if application requirements specify traffic patterns that are mostly uniform random. Adding additional stages to a butterfly topology increases path diversity, and adding $n-1$ stages results in an interesting class of network topologies known as Clos topologies [19] and fat-tree topologies [45]. Clos and fat-tree topologies can offer the same non-blocking guarantees as global crossbars with potentially lower resource requirements. Clos and fat-tree topologies have been proposed that use nanophotonic devices in low-radix [26] and high-radix [32, 34] configurations. The later is discussed further as a case study in section "Case Study #1: On-Chip Tile-to-Tile Network." Nanophotonic Clos-like topologies that implement high-radix routers using a subnetwork of low-radix routers have also been explored [63].

A k -ary n -dimensional torus topology is an alternative multi-stage topology where each terminal is associated with a router, and these routers are arranged in an n -dimensional logical grid with k routers in each dimension (see Fig. 3.3d). A mesh topology is similar to the torus topology except with the logically long "wrap-around" channels eliminated in each dimension. Two-dimensional torus and mesh topologies are particularly attractive in on-chip networks, since they naturally map to the planar chip substrate. Unfortunately, low-dimensional torus and mesh topologies have high hop counts resulting in longer latencies and possibly higher energy consumption. Moving from low-dimensional to high-dimensional torus or mesh topologies (e.g., a 4-ary 3-dimensional topology) reduces the network diameter, but requires long channels when mapped to a planar substrate. Also, higher-radix routers are required, potentially resulting in more area and higher router energy. Instead of adding network dimensions, we can use concentration to reduce network diameter [43]. Internal concentration multiplexes/demultiplexes multiple input/output

terminals across a single router port at the edge of the network, while external concentration integrates multiple terminals into a unified higher-radix router. There has been some work investigating how to best use nanophotonics in both two-dimensional torus [69] and mesh [18, 48] topologies.

While many nanophotonic interconnection networks can be loosely categorized as belonging to one of the four categories shown in Fig. 3.3, there are also more radical alternatives. For example, Koochi 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 [42].

Table 3.1 is an example of the first-order analysis that can be performed at the architectural level of design. In this example, we compare six logical topologies for a 64-terminal on-chip symmetric network. For the first-order latency metrics we assume a 22-nm technology, 5-GHz clock frequency, and a 400-mm² 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 start to 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. We can also see which topologies (e.g., crossbar and Clos) require more global bisection wiring resources, and which topologies 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 for both electrical and nanophotonic networks helps motivate the microarchitectural-level design discussed in the next section.

Microarchitectural-Level Design

For nanophotonic interconnection networks, microarchitectural-level design involves choosing which buses, channels, and routers to implement electrically and which to implement with nanophotonic devices. We must decide where nanophotonic transmitters and receivers will be used in the network, 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. At this level of design, we often use nanophotonic schematics to abstractly illustrate how the various components are integrated (see Fig. 3.4 for symbols that will be used in nanophotonic schematics and layouts). When working at the microarchitectural level, we want to focus on the higher-level operation of the nanophotonic devices, so it is often useful to assume we have as many wavelengths as necessary to meet our application requirements and to defer some practical issues related to mapping wavelengths to waveguides or waveguide layout until the final physical level of design. Although this means detailed analysis of area overheads or optical power requirements is not possible at this level of the design, we can still make many qualitative

Table 3.1 Architectural-level analysis for various 64 terminal networks

Topology		Buses and channels				Routers		Latency				
		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-stage	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-mm² 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 (from [8], courtesy of IEEE)

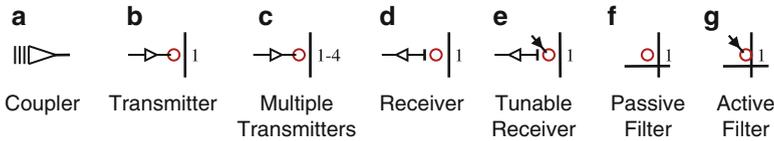


Fig. 3.4 Symbols used in nanophotonic schematics and layouts, For all ring-based devices, the number next to the ring indicates the resonant wavelength, and a range of numbers next to the ring indicates that the symbol actually represents multiple devices each tuned to a distinct wavelength in that range. The symbols shown include: (a) coupler for attaching a fiber to an on-chip waveguide; (b) transmitter including driver and ring modulator for λ_1 ; (c) multiple transmitters including drivers and ring modulators for each of $\lambda_1-\lambda_4$; (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 (from [8], courtesy of IEEE)

and quantitative comparisons between various network microarchitectures. For example, we can compare different microarchitectures based on the number of optoelectrical conversions along a given routing path, the total number of transmitters 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. This subsection discusses a range of microarchitectural design issues that arise when implementing the logical topologies described in the previous section.

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. In addition, an efficient global bus arbitration is 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

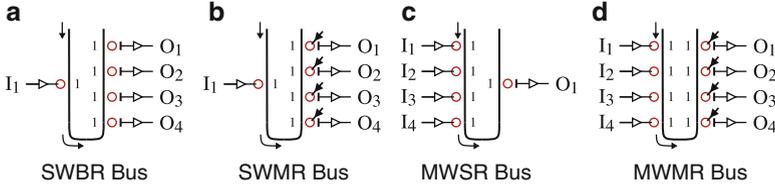


Fig. 3.5 Microarchitectural schematics for nanophotonic four terminal buses. The buses connect one or more input terminals (I_1 – I_4) to one or more output terminals (O_1 – O_4) via a single shared wavelength: (a) single-writer broadcast-reader (SWBR) bus; (b) single-writer multiple-reader bus; (c) multiple-writer single-reader bus; (d) multiple-writer multiple-reader bus (adapted from [8], courtesy of IEEE)

medium (see Fig. 3.5). Assuming a fixed modulation rate per wavelength, we can increase the bus bandwidth by using multiple parallel wavelengths. In the *single-writer broadcast-reader* (SWBR) bus shown in Fig. 3.5a, a single input terminal modulates the bus wavelength that is then broadcast to all four output terminals. This form of broadcast bus does not need any arbitration because there is only one input terminal. The primary disadvantage of a SWBR bus is simply the large amount of optical power required to broadcast packets to all output terminals. If we wish to send a packet to only one of many outputs, then we can significantly reduce the optical power by using active filters in each receiver. Figure 3.5b shows a *single-writer multiple-reader* (SWMR) bus where by default the ring filters in each receiver are detuned such that none of them 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 from the input terminal to the output terminals. Figure 3.5c illustrates a different bus network called 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. The most general bus network enables multiple input terminals to arbitrate for the shared bus and also allows a packet to be sent to one or more output terminals. Figure 3.5d illustrates a *multiple-writer multiple-reader* (MWMR) bus with four input terminals and four output terminals, but *multiple-writer broadcast-reader* (MWBR) buses are also possible. Here arbitration will be required at both the transmitter side and the receiver side. MWBR/MWMR buses will require $O(Nb_\lambda)$ transceivers where N is the number of terminals and b_λ is the number of shared wavelengths used to implement the bus.

There are several examples of nanophotonic buses in the literature. Several researchers have described similar techniques for using a combination of nanophotonic SWBR and MWSR buses to implement the command, write-data, and read-data buses in a DRAM memory channel [29, 53, 74, 76]. In this context the arbitration for the MWSR read-data bus is greatly simplified since the memory controller acts as a master and the DRAM banks act as slaves. We investigate various ways of implementing such nanophotonic DRAM memory channels as part of the

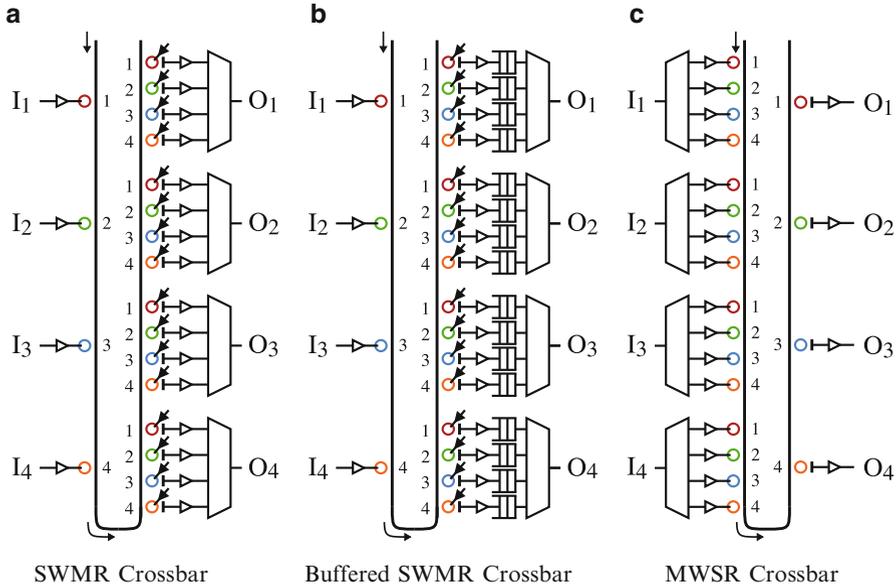


Fig. 3.6 Microarchitectural schematics for nanophotonic 4×4 crossbars. The crossbars connect all inputs (I_1 – I_4) to all outputs (O_1 – O_4) and are implemented with either: (a) four single-writer multiple-reader (SWMR) buses; (b) four SWMR buses with additional output buffering; or (c) four multiple-writer single-reader (MWSR) buses (adapted from [8], courtesy of IEEE)

case study in section “Case Study #3: DRAM Memory Channel”. Binkert et al. discuss both single-wavelength SWBR and SWMR bus designs for use in implementing efficient on-chip barrier networks, and the results suggest that a SWMR bus can significantly reduce the required optical laser power as compared to a SWBR bus [14]. Vantrease et al. also describe a nanophotonic MWBR bus used to broadcast invalidate messages as part of the cache-coherence protocol [76]. Arbitration for this bus is performed optically with tokens that are transferred between input terminals using a specialized arbitration network with a simple ring topology. Pan et al. proposed several techniques to help address scaling nanophotonic MWMR buses to larger numbers of terminals: multiple independent MWMR buses improve the total network bisection bandwidth while still enabling high utilization of all buses, a more optimized optical token scheme improves arbitration throughput, and concentrated bus ports shared by multiple terminals reduce the total number of transceivers [62].

Global crossbars have several attractive properties including high throughput and a short fixed latency. Nanophotonic crossbars use a dedicated nanophotonic bus per input or output terminal to enable every input terminal to send a packet to a different output terminal at the same time. Implementing such crossbars with nanophotonics have many of the same advantages and challenges as nanophotonic buses except at a larger scale. Figure 3.6 illustrates three types of nanophotonic crossbars. In the *SWMR crossbar* shown in Fig. 3.6a, there is one bus per input and every output can

read from any of these buses. 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. Although *SWBR crossbars* enable broadcast communication they use significantly more optical power than a *SWMR crossbar* for unicast communication. Note that even *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 $O(N^2 b_\lambda)$ receivers. Figure 3.6b illustrates an alternative called a buffered *SWMR crossbar* that avoids the need for any global or distributed arbitration. Every input terminal can send a packet to any output terminal at any time assuming it has space in the corresponding queue at the output. Each output locally arbitrates among these queues to determine which packet can access the output terminal. Buffered *SWBR/SWMR crossbars* simplify global arbitration at the expense of an additional $O(N^2)$ buffering. Buffered *SWMR crossbars* can still include a low-bandwidth *SWBR crossbar* to determine which receivers at the destination should be actively tuned. The *MWSR crossbar* shown in Fig. 3.6c is an alternative microarchitecture that 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 $O(N^2 b_\lambda)$ transmitters. For larger networks with wider channel bitwidths, the quadratic number of transmitters or receivers required to implement nanophotonic crossbars can significantly impact optical power, thermal tuning power, and area.

There have been several diverse proposals for implementing global crossbars with nanophotonics. Many of these proposals use global on-chip crossbars to implement L2-to-L2 cache-coherence protocols for single-socket manycore processors. Almost all of these proposals include some amount of concentration, so that a small number of terminals locally arbitrate for access to a shared crossbar port. This concentration helps leverage electrical interconnect to reduce the radix of the global crossbar, and can also enable purely electrical communication when sending a packet to a physically close output terminal. Kirman et al. describe three on-chip *SWBR nanophotonic crossbars* for addresses, snoop responses, and data for implementing a snoopy-based cache-coherence protocol [39]. The proposed design uses distributed redundant arbitration to determine which input port can write to which output port. A similar design was proposed by Pasrich et al. within the context of a multiprocessor system-on-chip [64]. Kirman et al. have recently described a more sophisticated *SWMR microarchitecture* with connection-based arbitration that is tightly coupled to the underlying physical layout [40]. 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 invalidation messages [44]. The proposed design requires several hundred thousand receivers for a 64×64 crossbar with each shared bus using 64 wavelengths modulated at 10 Gb/s. Vantrease et al. describe a MWSR nanophotonic crossbar for implementing a directory-based cache-coherence protocol, and a separate MWBR nanophotonic bus for invalidation messages [76]. The proposed design requires about a million transmitters for a 64×64 crossbar with each shared bus using 256 wavelengths modulated at 10 Gb/s. Arbitration in the MWSR nanophotonic crossbar is done with a specialized optical token scheme, where tokens circle around a ring topology. Although this scheme does enable round-robin fairness, later work by Vantrease et al. investigated techniques to improve the arbitration throughput for these token-based schemes under low utilization [75]. Petracca et al. proposed a completely different microarchitecture for a nanophotonic crossbar that uses optical switching inside the network and only $O(Nb_\lambda)$ transmitters and completely passive receivers [65]. The proposed design requires a thousand optical switches for a 64×64 crossbar with each shared bus using 96 wavelengths modulated at 10 Gb/s. Each switch requires around $O(8b_\lambda)$ actively tuned filters. The precise number of active filters depends on the exact switch microarchitecture and whether single-wavelength or multiple-wavelength active filters are used. Although such a microarchitecture has many fewer transmitters and receivers than the designs shown in Fig. 3.6, a separate multi-stage electrical network is required for arbitration and to setup the optical switches.

There are additional design decisions when implementing a multi-stage topology, since each network component can use either electrical or nanophotonic devices. Figure 3.7 illustrates various microarchitectural designs for a 2-ary 2-stage butterfly topology. In Fig. 3.7a, 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 is a natural approach, since we can potentially leverage the advantages of nanophotonics for implementing long global channels and use 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 in a microarchitectural schematic. Since a schematic is simply meant to capture the high-level interaction between electrical and nanophotonic devices, designers should simply use the simplest representation at this stage of the design. Similarly, the input and output terminals may be co-located in the physical design, but again the schematic is free to use a more abstract representation. In Fig. 3.7b, just the second stage of routers are implemented with nanophotonic devices and the channels are still implemented electrically. Since nanophotonic buffers are currently not feasible in intra-chip and inter-chip networks, the buffering is done electrically and the router's 2×2 crossbar is implemented with a nanophotonic SWMR microarchitecture. As with any nanophotonic crossbar, additional logic is required to manage arbitration for output ports. 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. In Fig. 3.7c, both the channels and the second stage of

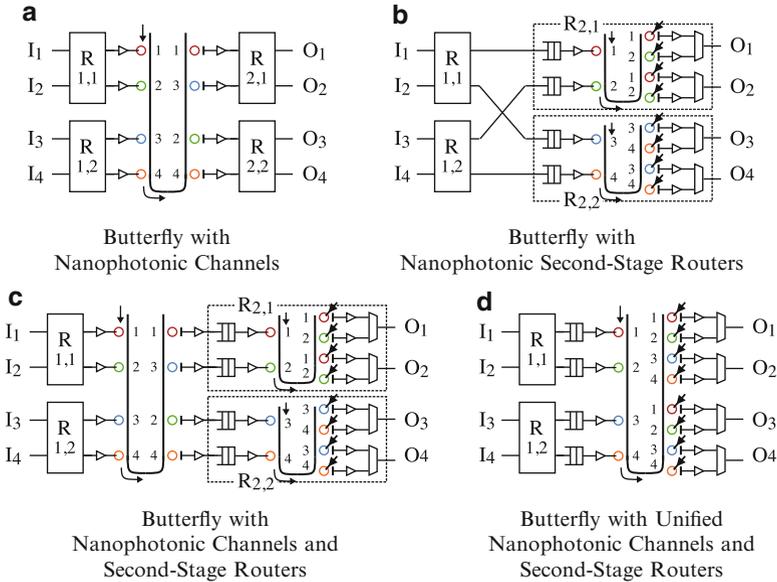


Fig. 3.7 Microarchitectural schematics for nanophotonic 2-ary 2-stage butterflies. Networks connect all inputs (I_1 – I_4) to all outputs (O_1 – O_4) with each network component implemented with either electrical or nanophotonic technology: (a) electrical routers and nanophotonic channels; (b) electrical first-stage routers, electrical channels, and nanophotonic second-stage routers; (c) electrical first-stage routers, nanophotonic channels, and nanophotonic second-stage routers; (d) similar to previous subfigure except that the channels and intra-router crossbars are unified into a single stage of nanophotonic interconnect (adapted from [8], courtesy of IEEE)

routers are implemented with nanophotonic devices. This requires opto-electrical conversions at two locations, and also needs electrical buffering to be inserted between the channels and the second-stage routers. Figure 3.7d illustrates a more promising microarchitecture where the nanophotonic channels and second-stage routers are unified and requires a single opto-electrical conversion. This does, however, force 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 two-stage butterfly-like topologies as the interconnect for large multichip modules [41]. Morris et al. proposed a two-stage butterfly-like topology for a purely on-chip network [56]. Both of these proposals are not true butterfly topologies since they incorporate some amount of flattening as in the flattened butterfly topology [37], or viewed differently some of the configurations resemble a generalized hypercube topology [12]. In addition,

some of the configurations include some amount of 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. 3.7a. The evaluation in both of these works suggests that only implementing the point-to-point channels using nanophotonic devices in a multi-stage topology might offer some advantages in terms of static power, scalability, and design complexity, when compared to more complicated topologies and microarchitectures. We will investigate a butterfly-like topology for processor-to-DRAM networks that only uses nanophotonic channels as a case study in section “Case Study #2: Manycore Processor-to-DRAM Network.” All of these butterfly networks have no path diversity, resulting in poor performance on adversarial traffic patterns when using simple routing algorithms. Pan et al. proposed a three-stage high-radix Clos-like topology for a on-chip network to enable much better load balancing [63]. In this design, the first and third stage of the topology effectively require radix-16 or radix-24 routers for a 64-terminal or 256-terminal network respectively. These high-radix routers are implemented with a mesh subnetwork, and the middle-stage routers connect corresponding mesh routers in each subnetwork. The middle-stage routers and the channels connecting the stages are all implemented with a unified nanophotonic microarchitecture similar in spirit to that shown Fig. 3.7d with buffered SWMR crossbars and a separate SWBR crossbar to determine which receivers should be actively tuned. 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 [26]. We will investigate a Clos topology for global on-chip communication as a case study in section “Case Study #1: On-Chip Tile-to-Tile Network.”

Designing nanophotonic torus topologies requires similar design decisions at the microarchitectural level as when designing butterfly topologies. Figure 3.8 illustrates two different microarchitectures for a 4-ary 1-dimensional torus (i.e., four node ring). In Fig. 3.8a, the four radix-2 routers are implemented electrically and the channels between each pair of routers are implemented with nanophotonic devices. In Fig. 3.8b, 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. 3.7d, 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 two-dimensional topologies. In addition, these proposals use fully optical microarchitectures in the spirit of Fig. 3.8b, since using electrical routers with short nanophotonic channels as in Fig. 3.8a yields little benefit. Shacham et al. proposed a fully optical two-dimensional torus with a combination of radix-4 blocking routers and specialized radix-2 injection and ejection routers [69]. A separate electrical control network is used to setup the control signals at each nanophotonic router. In this hybrid approach, the electrical control network

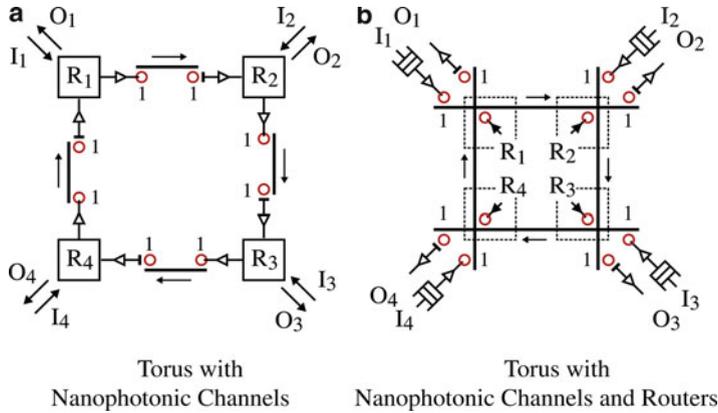


Fig. 3.8 Microarchitectural schematics for nanophotonic 4-ary 1-dim torus. Networks connect all inputs (I_1 – I_4) to all outputs (O_1 – O_4) with each network component implemented with either electrical or nanophotonic technology: (a) electrical routers and nanophotonic channels or (b) nanophotonic routers and channels. Note that this topology uses a single unidirectional channel to connect each of the routers (from [8], courtesy of IEEE)

uses packet-based flow control while the nanophotonic data network uses circuit-switched flow control. The radix-4 blocking routers require special consideration by the routing algorithm, but later work by Sherwood-Droz et al. fabricated alternative non-blocking optical router microarchitectures that can be used in this nanophotonic torus network [71]. Poon et al. survey a variety of designs for optical routers that can be used in on-chip multi-stage nanophotonic networks [66]. Li et al. propose a two-dimensional circuit-switched mesh topology with a second broadcast nanophotonic network based on planar waveguides for the control network [48]. Cianchetti et al. proposed a fully optical two-dimensional mesh topology with packet-based flow control [18]. This proposal sends control bits on dedicated wavelengths ahead of the packet payload. These control bits undergo an opto-electrical conversion at each router hop in order to quickly conduct electrical arbitration and flow control. If the packet wins arbitration, then the router control logic sets the active ring filters such that the packet payload proceeds through the router optically. If the packet loses arbitration, then the router control logic sets the active ring filters to direct the packet to local receivers so that it can be converted into the electrical domain and buffered. If the packet loses arbitration and no local buffering is available then the packet is dropped, and a nack is sent back to the source using dedicated optical channels. Later work by the same authors explored optimizing the optical router microarchitecture, arbitration, and flow control [17]. To realize significant advantages over electrical networks, fully optical low-dimensional torus networks need to carefully consider waveguide crossings, drop losses at each optical router, the total tuning cost for active ring filters in all routers, and the control network overhead.

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 often 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 diagram that specifies the exact placement of each device, and this layout is then used to calculate the area consumed by nanophotonic devices and the total optical power required for all wavelengths. This subsection discusses a range of physical design issues that arise when implementing the nanophotonic microarchitectures described in the previous section.

Figure 3.9 illustrates general approaches for the physical design of nanophotonic buses. These examples implement a four-wavelength SWMR bus, and they differ in how the wavelengths are mapped to each waveguide. Figure 3.9a illustrates the most basic approach where all four wavelengths are multiplexed onto the same waveguide. Although this produces the most compact layout, it also 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, $O(Nb_\lambda)$ through losses in the worst case, and a drop loss at the desired output terminal. As the number of wavelengths for this bus 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. Figure 3.9b 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 and thus the total optical power. Figure 3.9c–e illustrate *reader slicing*, where a subset of the bus readers are mapped to distinct waveguides. The example shown in Fig. 3.9c 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. Figure 3.9d illustrates a variation of reader slicing that uses optical power splitting. This *split nanophotonic bus* requires a single set of transmitters, but requires more optical power since this power must be split between the multiple bus branches. Figure 3.9e illustrates another variation of reader slicing that uses optical power guiding. This *guided nanophotonic bus* also only requires a single set of transmitters, but it uses active ring filters 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 physical design (e.g., redundant transmitters and optical power guiding)

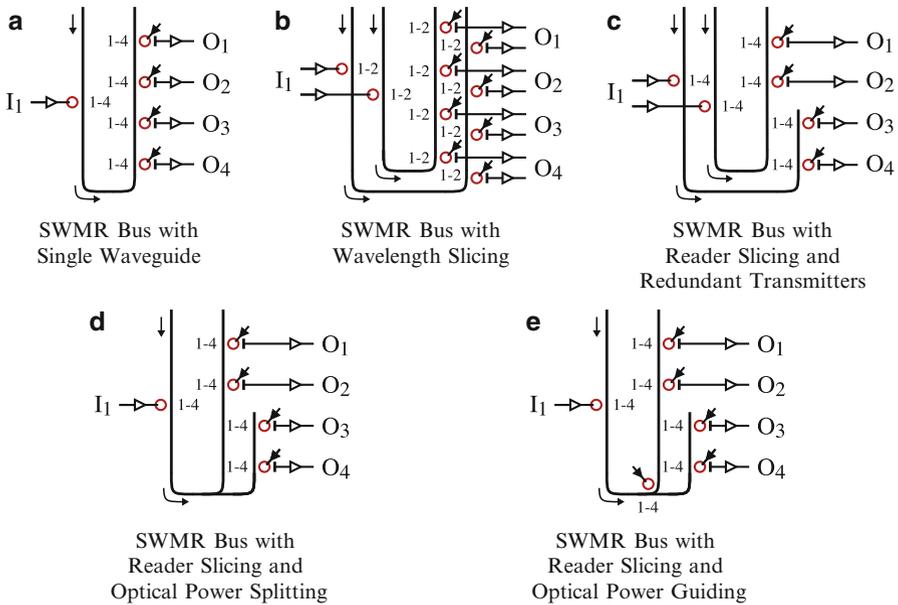


Fig. 3.9 Physical design of nanophotonic buses. The four wavelengths for an example four-output SWMR bus are mapped to waveguides in various ways: (a) all wavelengths mapped to one waveguide; (b) wavelength slicing with two wavelengths mapped to one waveguide; (c) reader slicing with two readers mapped to one waveguide and two redundant sets of transmitters; (d) reader slicing with a single transmitter and optical power passively split between two branches; (e) reader slicing with a single transmitter and optical power actively guided down one branch (adapted from [8], courtesy of IEEE)

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 [29, 74, 76] and there has been some exploration of the impact of using a split nanophotonic bus [14, 74]. We investigate the impact of using a guided nanophotonic bus in the context of a DRAM memory channel as part of the case study in section “Case Study #3: DRAM Memory Channel”.

Most nanophotonic crossbars use a set of shared buses, and thus wavelength slicing, reader slicing, and writer slicing are all applicable to the physical design of these crossbars. Figure 3.10a 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. 3.10a also illustrates a *serpentine layout*. Such layouts minimize waveguide crossings by “snaking” all waveguides

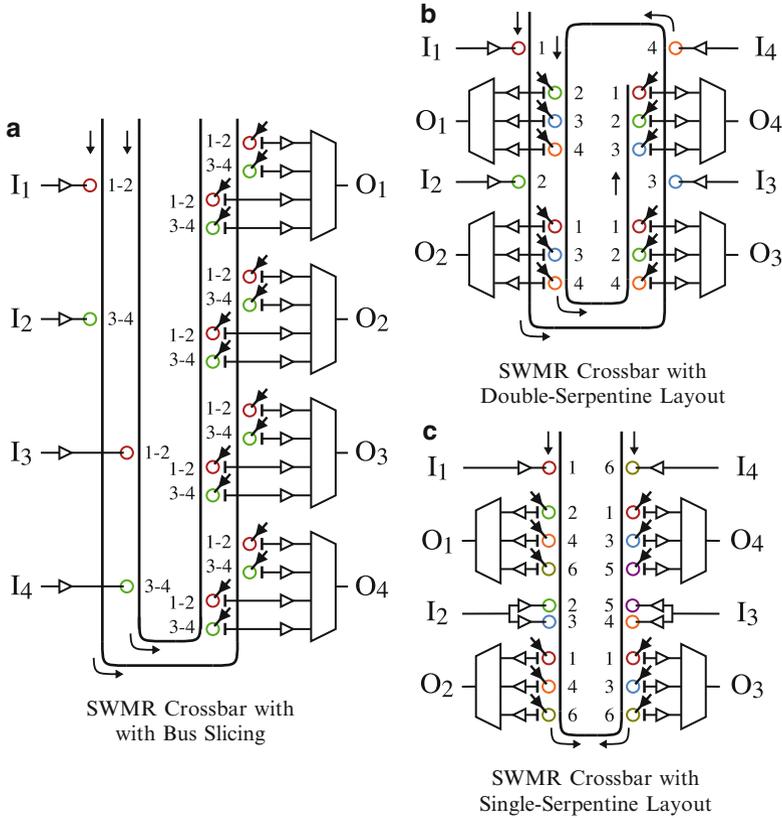


Fig. 3.10 Physical design of nanophotonic crossbars. In addition to the same techniques used with nanophotonic buses, crossbar designs can also use bus slicing: (a) illustrates a 4x4 SWMR crossbar with two wavelengths per bus and two buses per waveguide. Colocating input and output terminals can impact the physical layout. For example, a 4x4 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 (from [8], courtesy of IEEE)

around the chip, and they result in looped, U-shaped, and S-shaped waveguides. The example in Fig. 3.10a assumes that the input and output terminals are located on opposite sides of the crossbar, but it is also common to have pairs of input and output terminals co-located. Figure 3.10b illustrates a double-serpentine layout for a 4x4 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. Transmitters are on the first loop, and receivers are on the second loop. Figure 3.10c 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. For example, input I_2 uses λ_2 to

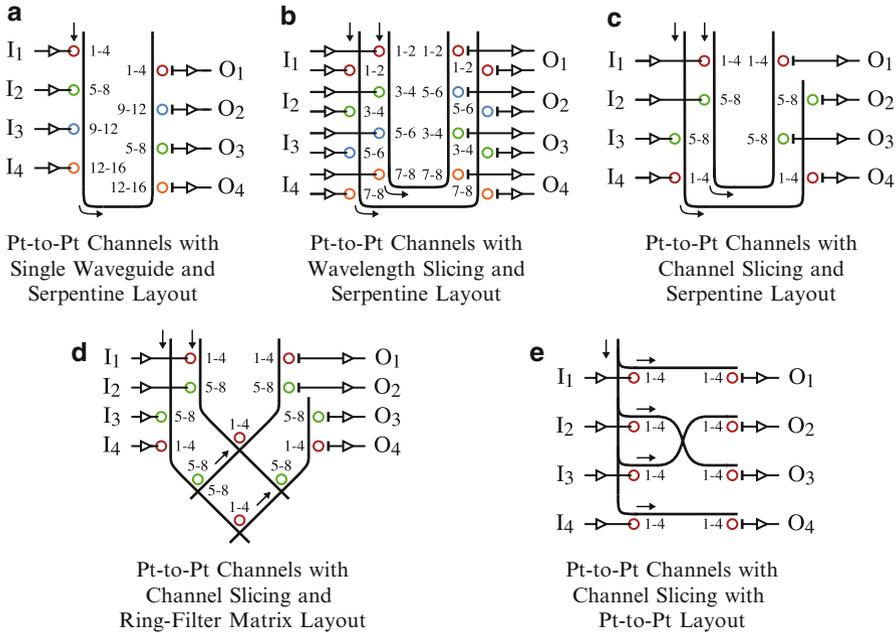


Fig. 3.11 Physical design of nanophotonic point-to-point channels. An example with four point-to-point channels each with four wavelengths can be implemented with either: (a) all wavelengths mapped to one waveguide; (b) wavelength slicing with two wavelengths from each channel mapped to one waveguide; (c) partial channel slicing with all wavelengths from two channels mapped to one waveguide and a serpentine layout; (d) partial channel slicing with a ring-filter matrix layout to passively shuffle wavelengths between waveguides; (e) full channel slicing with each channel mapped to its own waveguide and a point-to-point layout (adapted from [8], courtesy of IEEE)

send packets clockwise and λ_3 to send packets counter-clockwise. 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 [39, 44, 64], partially wavelength-sliced and bus-sliced MWSR/SWMR crossbars with a double-serpentine layout [63, 76], fully reader-sliced SWMR crossbars with multiple redundant transmitters and a serpentine layout [56], and a variant of a reader-sliced SWMR crossbar with a serpentine layout which distributes readers across waveguides and also across different wavelengths on the same waveguide [40]. Nanophotonic crossbars with optical switching distributed throughout the network have a significantly different microarchitecture and correspondingly a significantly different physical-level design [65].

Figure 3.11 illustrates general approaches for the physical design of point-to-point nanophotonic channels that can be used in butterfly and torus topologies. This particular 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. 3.7a. Figure 3.11a illustrates the most basic design where all sixteen wavelengths are mapped to a single waveguide with a serpentine layout. As with nanophotonic buses, wavelength slicing reduces the number of wavelengths per waveguide and total through losses by mapping a subset of each channel's wavelengths to different waveguides. In the example shown in Fig. 3.11b, two wavelengths from each channel are mapped to a single waveguide resulting in eight total wavelengths per waveguide. Figure 3.11c–e illustrate *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. 3.11c, maps two channels to each waveguide but still uses a serpentine layout. The example in Fig. 3.11d has the same organization on the transmitter side, but uses a passive *ring filter matrix layout* to shuffle wavelengths between waveguides. These passive ring filter matrices can be useful when a set of channels is mapped to one waveguide, but the physical layout requires a subset of those channels to also be passively mapped to a second waveguide elsewhere in the system. Ring filter matrices can shorten waveguides at the cost of increased waveguide crossings and one or more additional drop losses. Figure 3.11e 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 [39] or an iterative algorithm [11] 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 [41], and some experimental work on passive ring filter network components similar in spirit to the ring-filter matrix [82]. Point-to-point channels are an integral part of the case studies in sections “Case Study #1: On-Chip Tile-to-Tile Network” and “Case Study #2: Manycore Processor-to-DRAM Network.”

Much of the above discussion about physical-level design is applicable to micro-architectures that implement multiple-stages of nanophotonic buses, channels, and routers. 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 [26], torus networks [69], and mesh networks [18, 48] all use channel slicing and regular layouts that match the logical topology. Since unmodulated light will

need to be distributed across the chip to each injection port, these examples will most likely require more complicated optical power distribution, laser couplers located across the chip, or some form of hybrid laser integration.

Figures 3.12 and 3.13 illustrate several 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. Each tile is approximately $2.5 \times 2.5\text{mm}$ and includes a co-located network input and output terminal. The network bus and channel bandwidths are sized according to Table 3.1. The 64×64 crossbar topology in Fig. 3.12 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. Both layouts are able to co-locate the laser couplers in two locations along one edge of the chip to simplify packaging. Figure 3.12a uses a longer serpentine layout, while Fig. 3.12b 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. 3.13 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. Figure 3.13a 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 meaning that there is no need for any form of ring-filter matrix. Clever waveguide layout results in 16 waveguide crossings located in the middle of the chip. If we were to reduce the wavelengths per channel but maintain the total wavelengths per waveguide, then a ring-filter matrix might be necessary to shuffle channels between waveguides. Figure 3.13b uses a single-serpentine layout. The serpentine layout increases waveguide lengths but eliminates waveguide crossings in the middle of the chip. Notice that the serpentine layout requires co-located laser couplers in two locations along one edge of the chip, while the point-to-point layout requires laser couplers on both sides of the chip. The point-to-point layout could position all laser couplers together, but this would increase the length of the optical power distribution waveguides. Note that in all four layouts eight waveguides share the same post-processing air gap, and that some waveguide crossings may be necessary at the receivers to avoid positioning electrical circuitry over the air gap.

Figure 3.14 illustrates the kind of quantitative analysis that can be performed at the physical level of design. Detailed layouts corresponding to the abstract layouts in Figs. 3.12b and 3.13b are used to calculate the total optical power and area overhead as a function of optical device quality and the technology assumptions in the earlier section on nanophotonic technology.” Higher optical losses increase the power per waveguide which eventually necessitates distributing fewer wavelengths over more waveguides to stay within the waveguide’s total optical power limit. Thus higher optical losses can increase both the optical power and the 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

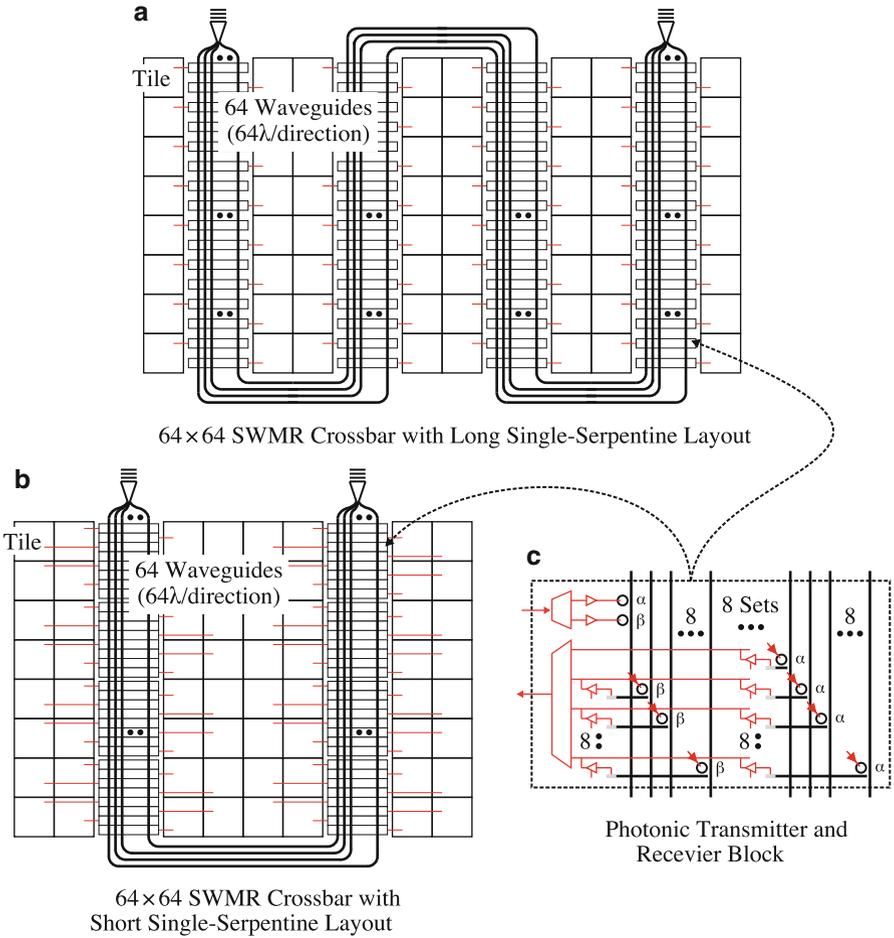


Fig. 3.12 Abstract physical layouts for 64x64 SWMR crossbar. In a SWMR crossbar each tile modulates a set wavelengths which then must reach every other tile. Two waveguide layouts are shown: (a) uses a long single-serpentine layout where all waveguides pass directly next to each tile; (b) uses a shorter single-serpentine layout to reduce waveguide loss at the cost of greater electrical energy for more distant tiles to reach their respective nanophotonic transmitter and receiver block. The nanophotonic transmitter and receiver block shown in (c) illustrates how bus slicing is used to map wavelengths to waveguides. One logical channel (128 b/cycle or 64 λ per channel) is mapped to each waveguide, but as required by a single-serpentine layout, the channel is split into 64 λ directed left to right and 64 λ directed right to left. Each ring actually represents 64 rings each tuned to a different wavelength; $\alpha = \lambda_1 - \lambda_{64}$; $\beta = \lambda_{64} - \lambda_{128}$; couplers indicate where laser light enters chip (from [8], courtesy of IEEE)

cost of providing $O(N^2 b_\lambda)$ 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 the technology assumptions in the earlier section on nanophotonic

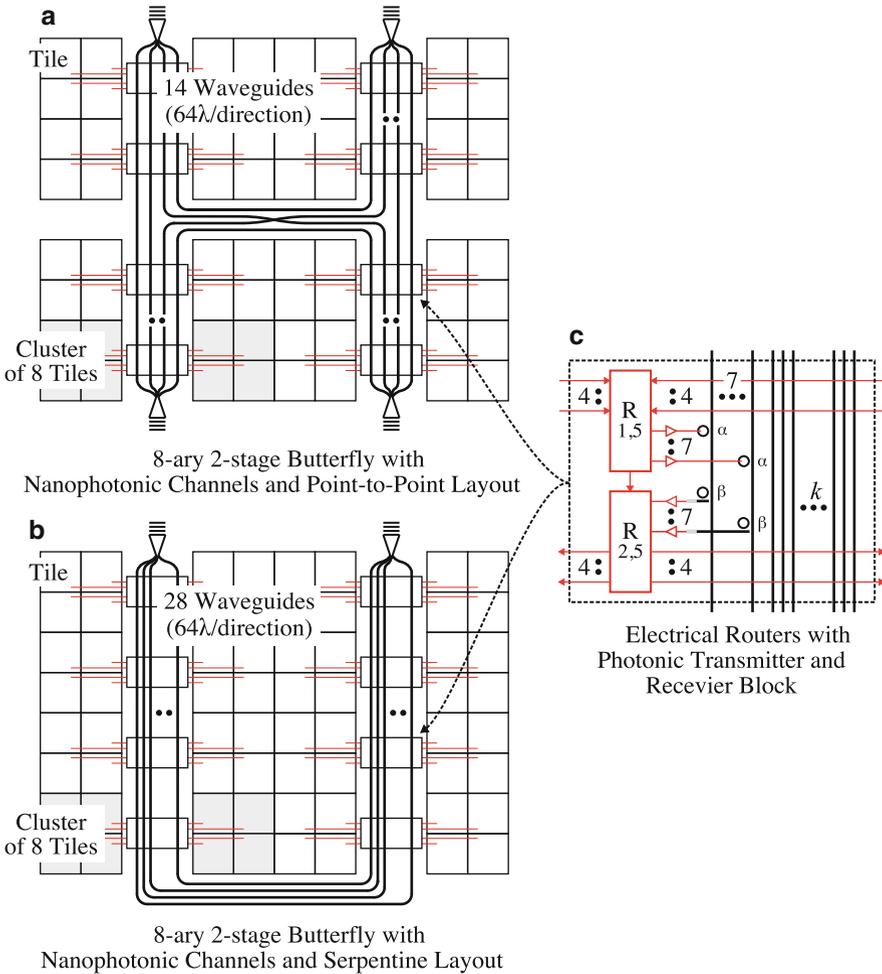


Fig. 3.13 Abstract physical layouts for 8-ary 2-stage butterfly with nanophotonic channels. In a butterfly with nanophotonic channels each logical channel is implemented with a set of wavelengths that interconnect two stages of electrical routers. Two waveguide layouts are shown: **(a)** uses a point-to-point layout; **(b)** uses a serpentine layout that results in longer waveguides but avoids waveguide crossings. The nanophotonic transmitter and receiver block shown in **(c)** illustrates how channel slicing is used to map wavelengths to waveguides. Two logical channels (128 b/cycle or 64λ per channel) are mapped to each waveguide, and by mapping channels connecting the same routers but in opposite directions we avoid the need for a ring-filter matrix. Each ring actually represents 64 rings each tuned to a different wavelength; $\alpha = \lambda_1 - \lambda_{64}$; $\beta = \lambda_{64} - \lambda_{128}$; k is seven for point-to-point layout and 21 for serpentine layout; couplers indicate where laser light enters chip (from [8], courtesy of IEEE)

technology” the crossbar network requires 500,000 rings and a fixed thermal tuning power of over 10W. 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 non-blocking topology. Since nanophotonics is still an emerging technology, evaluating a layout as a function of optical device quality is critical for a fair comparison.

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. The tiles might be homogeneous with each tile including both some number of cores and a slice of the on-chip memory, or the tiles might be heterogeneous with a mix of compute and memory tiles. The global on-chip network might be used to implement shared memory, message passing, or both. Our basic network design will be similar regardless of these specifics. We assume that software running on the tiles adhere to a dynamically partitioned application model; tiles within a partition communicate extensively, while tiles across partitions communicate rarely. This case study assumes a 22-nm technology, 5-GHz clock frequency, 512-bit packets, and 400-mm² chip. We examine networks sized for low (LTBw), medium (MTBw), and high (HTBw) target bandwidths which correspond to ideal throughputs of 64, 128, and 256b/cycle per tile under uniform random traffic. More details on this case study can be found in [32].

Network Design

Table 3.1 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 “Designing Nanophotonic Interconnection Networks.” 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. 3.3c except with three stages of routers. The associated configuration for the MTBw target is shown in Table 3.1. This topology is non-blocking 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 non-deterministic routing algorithm that efficiently balances load by always randomly picking a middle-stage router.

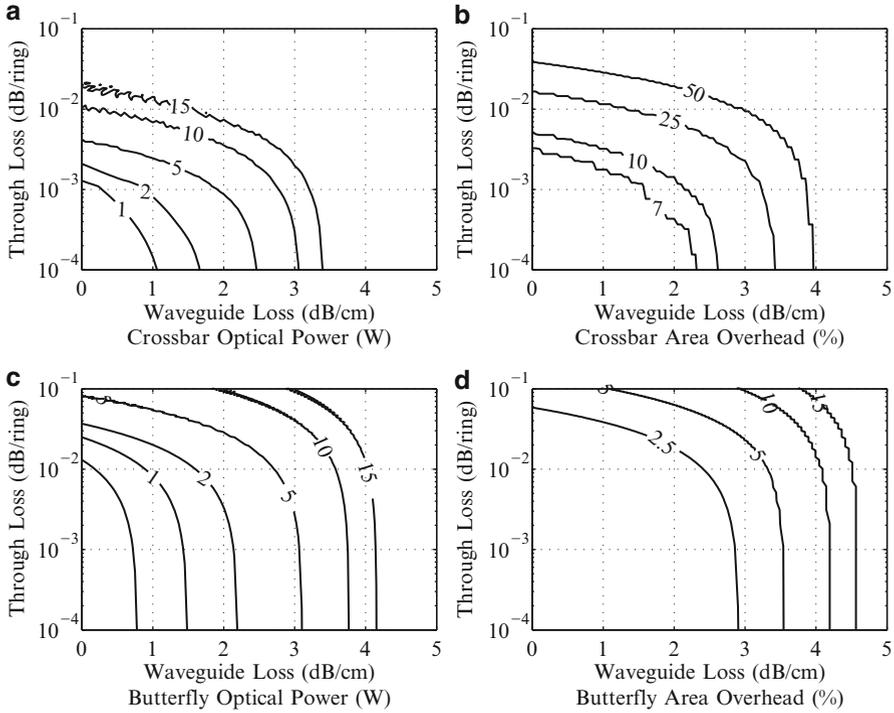


Fig. 3.14 Comparison of 64×64 crossbar and 8-ary 3-stage butterfly networks. Contour plots show optical laser power in Watts and area overhead as a percentage of the total chip area for the layouts in Figs. 3.12b and 3.13b. These metrics are plotted as a function of optical device quality (i.e., ring through loss and waveguide loss) (from [8], courtesy of IEEE)

The 8-ary 2-stage butterfly in Fig. 3.13b 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 the two microarchitectures illustrated in Fig. 3.15. For simplicity, these microarchitectural schematics are for a smaller (2,2,2) Clos topology. The microarchitecture in Fig. 3.15a 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. As an example, if input I_2 wants to communicate with output O_3 then it can use either middle router. If the routing algorithm chooses $R_{2,2}$, then the network will use wavelength λ_2 on the first waveguide to send the message to $R_{2,1}$ and wavelength λ_4 on the second waveguide to send the message to O_4 . The microarchitecture in Fig. 3.15b implements both the point-to-point channels and the middle stage of routers with nanophotonics. We chose to pursue 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

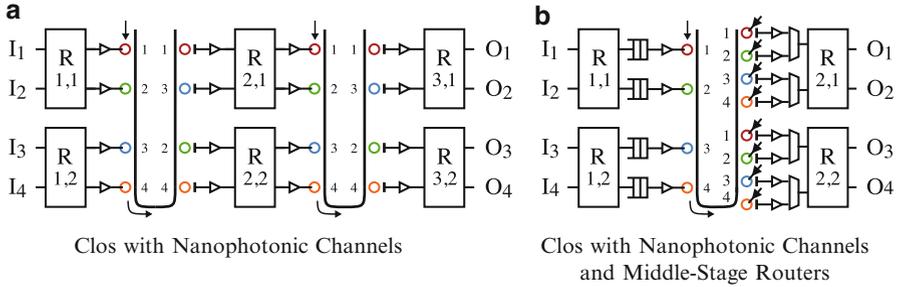


Fig. 3.15 Microarchitectural schematic for nanophotonic (2,2,2) Clos. Both networks have four inputs (I_{1-4}), four outputs (O_{1-4}), and six 22 routers ($R_{1-3;1-2}$) with each network component implemented with either electrical or nanophotonic technology: **(a)** electrical routers with four nanophotonic point-to-point channels; **(b)** electrical first- and third-stage routers with a unified stage of nanophotonic point-to-point channels and middle-stage routers (from [32], courtesy of IEEE)

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. 3.13b 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. 3.14c and 3.14d. Note that even with twice the number of bisection channels, the Clos network still uses 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.

Evaluation

Our evaluation uses a detailed cycle-level microarchitectural simulator to study the performance and power of various electrical and nanophotonic networks. For power calculations, important events (e.g., channel utilization, queue accesses, and arbitration) were counted during simulation and then multiplied by energy values derived from first-order gate-level models assuming a 22-nm technology. Our baseline includes three electrical networks: an 8-ary 2-dimensional mesh (*emesh*), a 4-ary 2-dimensional 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.

Figure 3.16 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 non-uniform 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. Figure 3.16 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. 3.16d) will have similar power consumption as a lower-capacity *ecmeshx2* configuration (e.g., Fig. 3.16a). This could enable a nanophotonic Clos network to have similar or better performance than an electrical network within a similar power constraint.

Figure 3.17 shows the power breakdowns for various topologies and traffic patterns. Figure 3.17a includes the least expensive configurations that can sustain an aggregate throughput of 2kb/cycle, while Fig. 3.17b includes the least expensive configurations that can sustain an aggregate throughput of 8kb/cycle. Compared to *emesh* and *ecmeshx2* at 8kb/cycle, the *pclos* network with aggressive technology projections provides comparable performance and low 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 non-trivial 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 meaning that every Watt of optical laser power requires three Watts of electrical power to generate. 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.

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

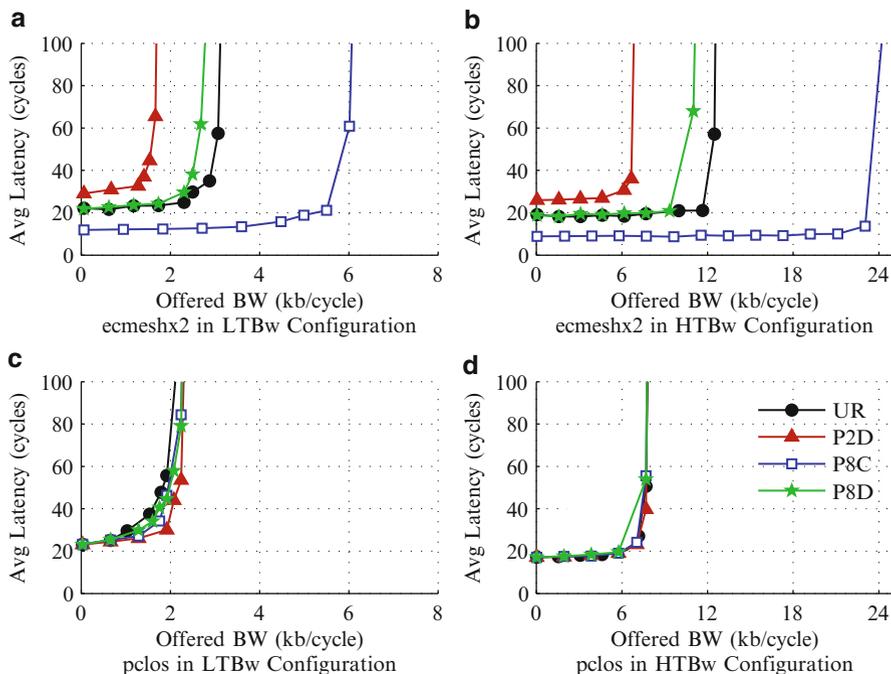


Fig. 3.16 Latency versus offered bandwidth for on-chip tile-to-tile networks. LTBw systems have a theoretical throughput of 64b/cycle per tile, while HTBw systems have a theoretical throughput of 256b/cycle both for the uniform random traffic pattern (adapted from [32], courtesy of IEEE)

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. This is especially true when we are considering networks that might be feasible in the near future. Third, it is important to use an iterative design process that considers all levels of the design. For example, Fig. 3.17 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 low-radix nanophotonic routers [34].

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

Off-chip main-memory bandwidth is likely to be a key bottleneck in future many-core 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. Each on-chip tile could contain one or more processor cores possibly with

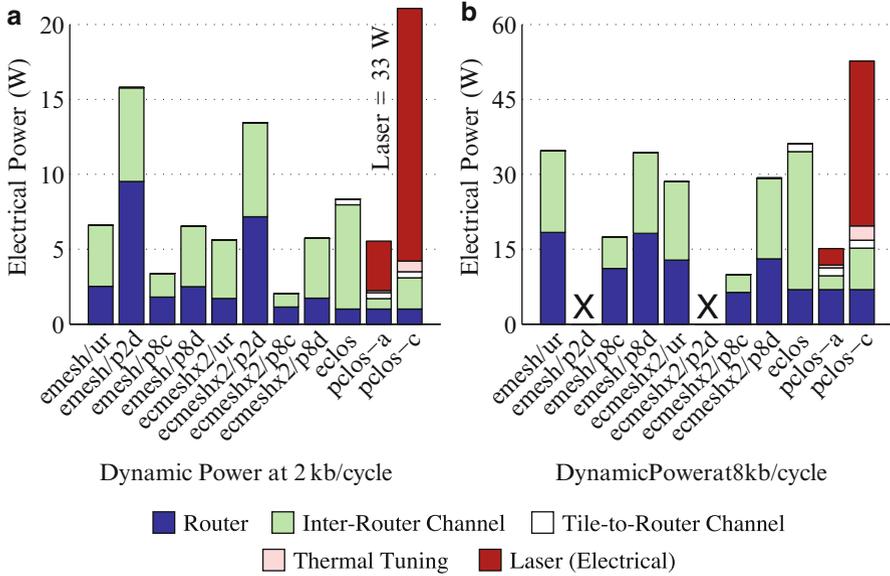


Fig. 3.17 Dynamic power breakdown for on-chip tile-to-tile networks. Power of *eclos* and *pclos* did not vary significantly across traffic patterns. (a) LTBw systems at 2 kb/cycle offered bandwidth (except for *emesh/p2d* and *ecmeshx2/p2d* which saturated before 2 kb/cycle, HTBw system shown instead); (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 technology projections (from [32], courtesy of IEEE)

shared cache, and each DRAM module includes multiple memory controllers and DRAM chips to provide large bandwidth with high capacity. 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. This case study assumes a 22-nm technology, 2.5-GHz clock frequency, 512-bit packets for transferring cache lines, and 400-mm² chip. We also assume that the total power of the processor chip is one of the key design constraints limiting achievable performance. More details on this case study can be found in [6, 7].

Network Design

We focus on high-radix, low-diameter topologies so that we can make use of simple point-to-point nanophotonic channels. Our hope is that this approach will provide a significant performance and energy-efficiency advantage while reducing risk by relying on simple devices. The lack of path diversity in the butterfly topology is less of an issue in this application, since we can expect address streams across cores to

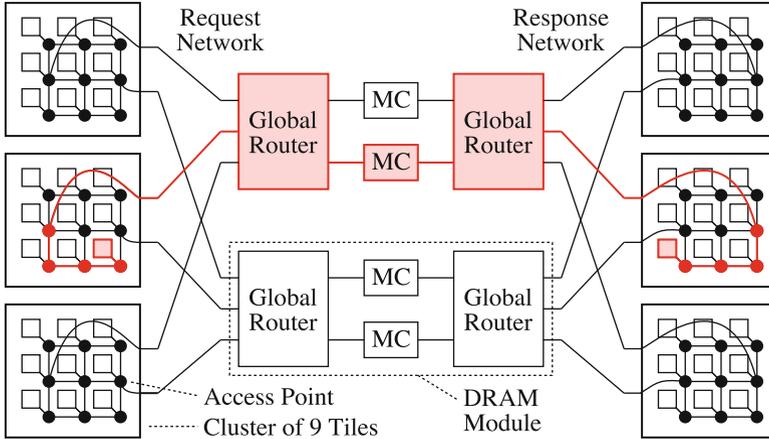


Fig. 3.18 Logical topology for processor-to-DRAM network. Two $(3,9,2,2)$ LMGs networks are shown: one for the memory request network and one for the memory response network. Each LMGs network includes three groups of nine tiles arranged in small 3-ary 2-dimensional mesh cluster and two global 3×2 routers 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 (from [8], courtesy of IEEE)

be less structured than in message passing networks. A two-stage symmetric 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 the complexity and risk associated with adopting nanophotonics. 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. 3.18 where each high-radix router is implemented with an electrical mesh sub-network 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 ($c \times r$). For simplicity, Fig. 3.18 illustrates a smaller $(3,9,2,2)$ LMGS topology supporting a total of 27 tiles. We assume dimension-ordered routing for the cluster mesh networks, although of course other routing algorithms are possible. Notice that 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. To avoid protocol deadlock, we use one LMGS network for memory requests from a tile to a specific DRAM module, and a separate LMGS network for memory responses from the DRAM module back to the original tile. In this study, we assume the request and response LMGS networks are separate physical networks, but they could also be two logical networks implemented with distinct virtual channels. The LMGS topology is particularly useful for preliminary design space exploration

since it decouples the number of tiles, clusters, and memory controllers. 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 by providing four electrical DRAM memory controllers per DRAM module. Note that high-bandwidth nanophotonic DRAM described as part of the case study in section “Case Study #3: DRAM Memory Channel” could potentially provide an equivalent amount of memory bandwidth with fewer memory controllers and lower power consumption.

As mentioned above, our design uses a hybrid opto-electrical microarchitecture that targets the advantages of each medium: nanophotonic interconnect for energy-efficient global communication, and electrical interconnect for fast switching, efficient buffering, and local communication. 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.

Figure 3.19 shows the abstract layout for our target system with 16 clusters. Since each cluster requires one dedicated global channel to each DRAM module, there are 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 still meet the 20 W power constraint. A ring-filter matrix layout is used to passively shuffle the 16λ channels on different horizontal waveguides destined for the same DRAM module onto the same set of four vertical waveguides. We assume that each DRAM module includes a custom *switch chip* containing the global router for both the request and response networks. The switch chip on the memory side arbitrates between the multiple requests coming in from the different clusters on the processor chip. This reduces the power density of the processor chip and could enable multi-socket configurations to easily share the same DRAM modules. A key feature of this layout is that the nanophotonic devices are not only used for inter-chip communication, but can also provide cross-chip transport to off-load intra-chip global electrical wiring. Figure 3.20 shows the laser power as a function of optical device quality for two different power constraints and thus 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.

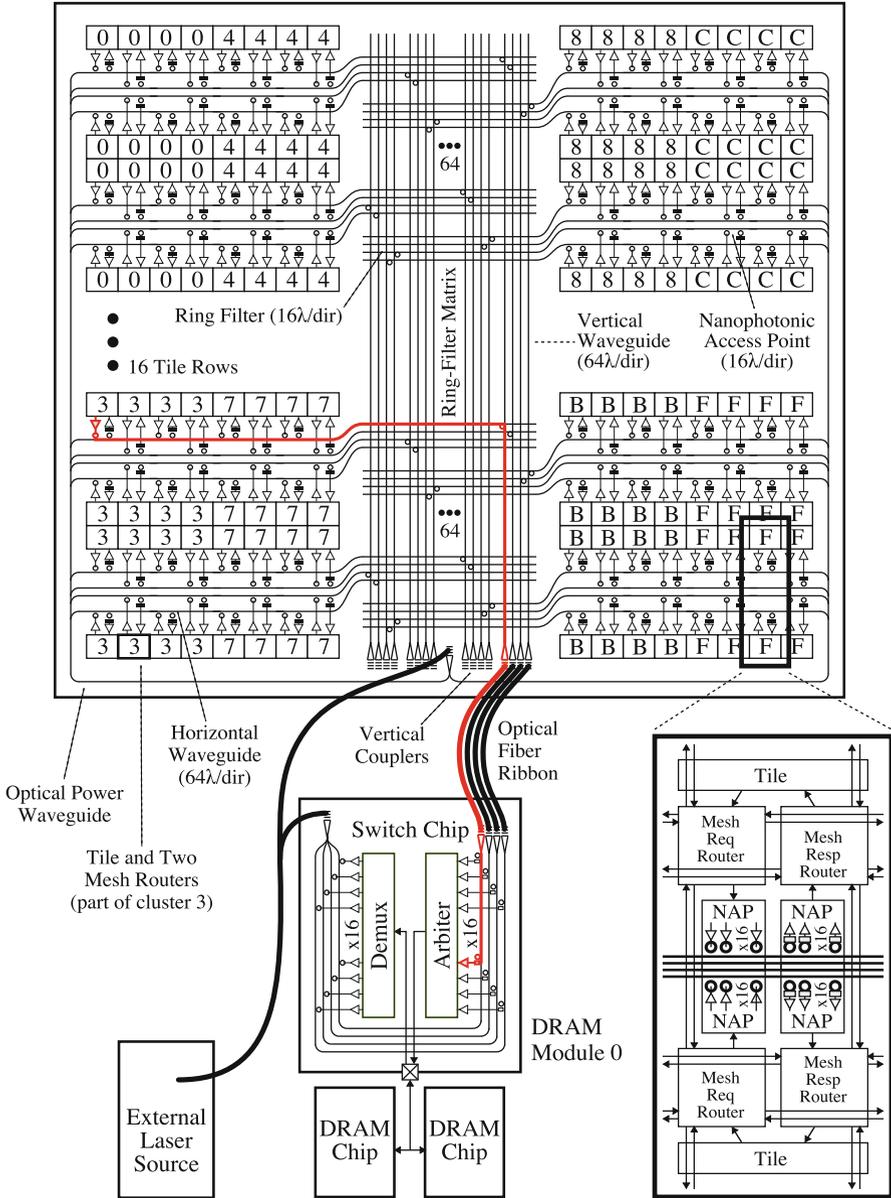


Fig. 3.19 Abstract physical layout for nanophotonic processor-to-dram network. Target (16,16,16,4) LMGS network with 256 tiles, 16 DRAM modules, and 16 clusters each with a 4-ary 2-dimensional electrical mesh. Each tile is labeled with a hexadecimal number indicating its cluster. For simplicity the electrical 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). *NAP*=nanophotonic access point; nanophotonic request channel from group 3 to DRAM module 0 is highlighted (adapted from [7], courtesy of IEEE)

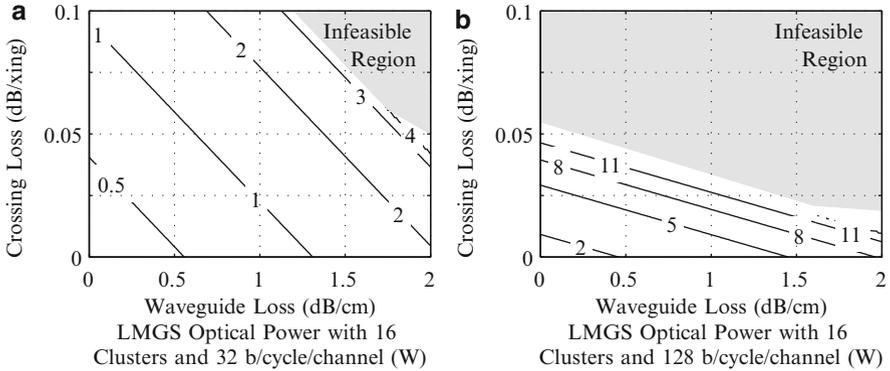


Fig. 3.20 Optical power for nanophotonic processor-to-DRAM networks. Results are for a (16,16,16,4) LMGs topology with a ring-filter matrix layout and two different power constraints: (a) low power constraint and thus low aggregate bandwidth and (b) high power constraint and thus high aggregate bandwidth (from [7], courtesy of IEEE)

Evaluation

Our evaluation uses a detailed cycle-level microarchitectural simulator to study the performance and power of various electrical and nanophotonic networks. We augment our simulator to count important events (e.g., channel utilization, queue accesses, and arbitration) which are then multiplied by energy values derived from our analytical models. 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.

Figure 3.21 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 = electrical, P = nanophotonics), the number of clusters ($1/4/16$), and the over-provisioning factor ($x1/x2/x4$). Overprovisioning improves the performance of the configurations with one and four clusters. $E1x4$ and $E4x2$ increase the throughput by 3–4 \times over the balanced configurations. Overprovisioning had minimal impact on the 16 cluster configurations since the local meshes are already quite small. 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 \times 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 $\approx 9\times$ improvement over the best electrical configuration ($E4x2$) at comparable latency. The best optical configurations consume ≈ 16 W near saturation.

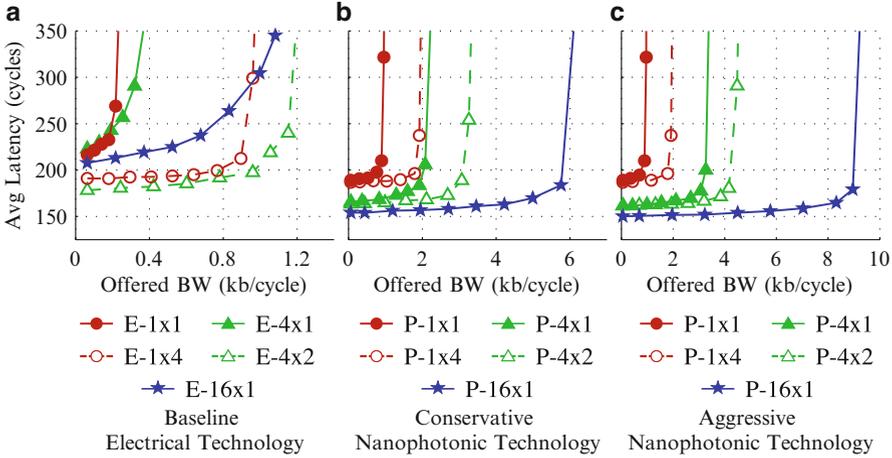


Fig. 3.21 Latency versus offered bandwidth for processor-to-DRAM networks. *E* electrical, *P* nanophotonics, *1/4/16* number of clusters, *x1/x2/x4* over-provisioning factor (adapted from [7], courtesy of IEEE)

Table 3.2 Power Breakdown for Processor-to-DRAM Networks

Configuration	Throughput (kb/cycle)	Component power (W)				Total power (W)
		Mesh routers	Mesh channels	Global hannels	Thermal tuning	
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

Table 3.2 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. Note that the laser power is not included here as it is highly dependent on the physical layout and photonic device design as shown in Fig. 3.20. The photonic configurations consume close to 15W leaving 5W for on-chip optical power dissipation as heat. Ultimately, photonics enables an 8–10× improvement in throughput at similar power consumption.

Design Themes

This case study suggests it is much easier to show a compelling advantage for implementing a 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 as deep into each chip as possible (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. 3.20, this layout puts significant constraints on the maximum tolerable losses in waveguides and crossings. We are currently considering alternate serpentine layouts that can reduce the losses in crossings and waveguides. However, the serpentine layout needs couplers at multiple locations on the chip, which could increase packaging costs. An alternative would be to leverage the multiple nanophotonic devices layers available in monolithic BEOL integration approach. Work by Biberman et al. has shown how multilayer deposited devices can significantly impact the feasibility of various network architectures [13], and this illustrates the need for a design process that iterates across the architecture, microarchitecture, and physical design levels.

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 photonicly integrated DRAM (PIDRAM) which involves re-architecting the DRAM channel, chip, and bank to make best use of the nanophotonic technology for improved performance and energy efficiency. As in the previous case study, we assume the address space is interleaved across DRAM channels at a fine granularity, and that this effectively results in approximately uniform random address streams. 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 [9].

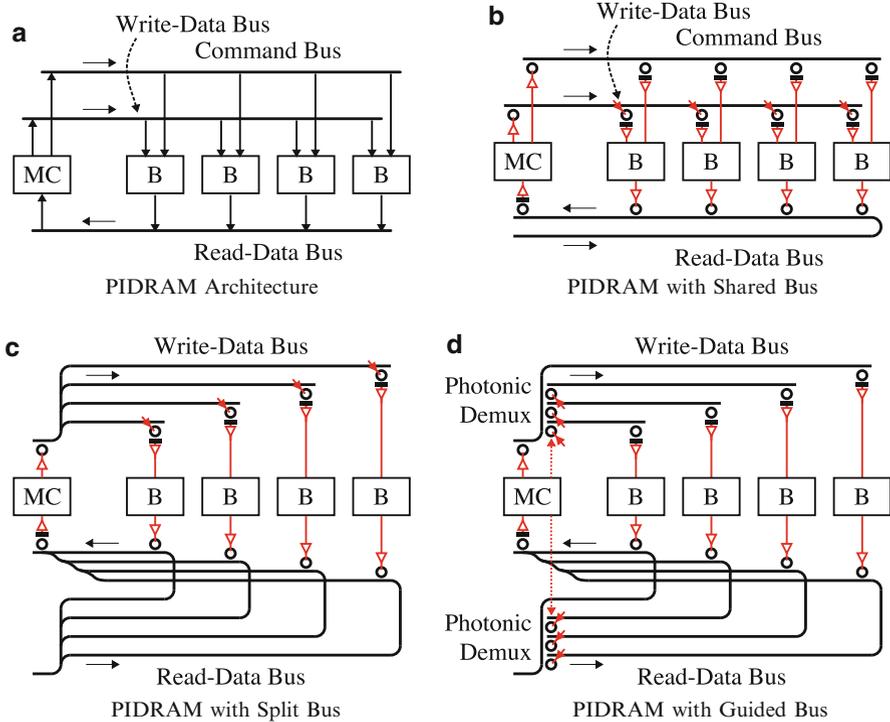


Fig. 3.22 PIDRAM designs. Subfigures illustrate a single DRAM memory channel (MC) with four DRAM banks (B) at various levels of design: (a) logical topology for DRAM memory channel; (b) shared nanophotonic buses where optical power is broadcast to all banks along a shared physical medium; (c) split nanophotonic buses where optical power is split between multiple direct connections to each bank; (d) guided nanophotonic buses where optical power is actively guided to a single bank. For clarity, command bus is not shown in (c) and (d), but it can be implemented in a similar fashion as the corresponding write-data bus or as a SWBR bus (adapted from [9], courtesy of IEEE)

Network Design

Figure 3.22a 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. Figure 3.22b illustrates a straight-forward nanophotonic microarchitecture for a DRAM memory channel with a combination of SWBR, SWMR, and MWSR buses.

The microarchitecture in Fig. 3.22b can also map to a similar layout that we call a shared nanophotonic bus. In this layout, 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. The memory controller places the write data on this bus; the target bank will receive the data and then perform the corresponding write operation. 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. If all of the banks are on the same PIDRAM chip, then the losses can be manageable. However, to scale to larger capacities, we will need to “daisy-chain” the shared nanophotonic bus through multiple PIDRAM chips. Large coupler losses and the exponential scaling of laser power combine to make the shared nanophotonic bus feasible only for connecting banks within a PIDRAM chip as opposed to connecting banks across PIDRAM chips.

Figure 3.22c shows the alternative reader-/writer-sliced split nanophotonic bus layout, which divides the long shared bus into multiple branches. In the command and write-data bus, modulated laser power is still sent to all receivers, and in the read-data bus, laser power is still sent to all modulators. The split nature of the bus, however, means that the total laser power is roughly a linear function of the number of banks. If each bank was on its own PIDRAM chip, then we would use a couple of fibers per chip (one for modulated data and one for laser power) to connect the memory controller to each of the PIDRAM chips. Each optical path in the write-data bus would only traverse one optical coupler to leave the processor chip and one optical coupler to enter the PIDRAM chip regardless of the total number of banks. This implementation reduces the extra optical laser power as compared to a shared nanophotonic bus 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 effective bandwidth.

To further reduce the required optical power, we can use a reader-/writer-sliced guided nanophotonic bus layout, shown in Fig. 3.22d. Each nanophotonic demultiplexer uses an array of either active ring or comb filters. For the command and write-data bus, the nanophotonic demultiplexer is placed after the modulator to direct the modulated light to the target bank. For the read-data bus, the nanophotonic demultiplexer is placed before the modulators to allow the memory controller to manage when to guide the light to the target bank for modulation. 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.

Figure 3.23 illustrates in more detail 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. The memory scheduler within the memory controller orchestrates access to each bus to

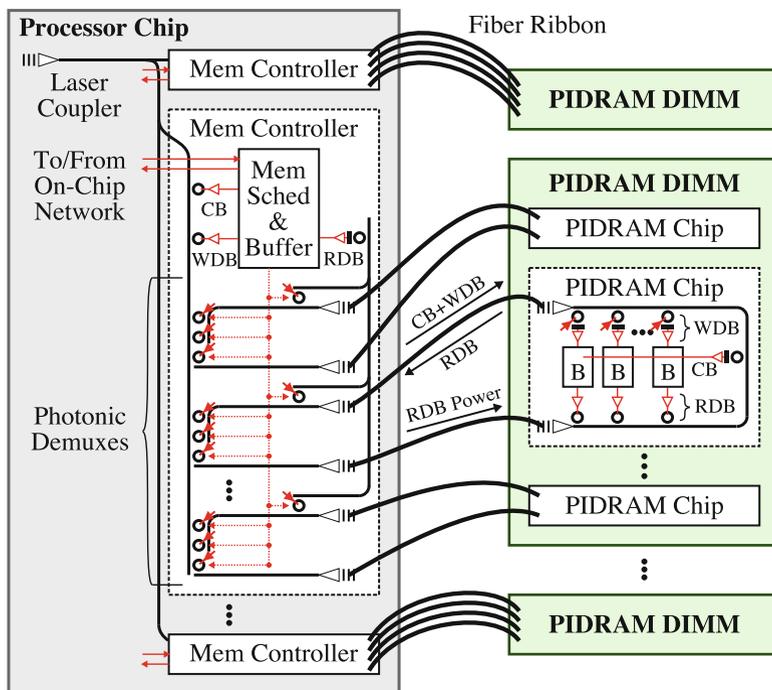


Fig. 3.23 PIDRAM memory system organization. Each PIDRAM memory channel connects to a PIDRAM DIMM via a fiber ribbon. The memory controller manages the command bus (CB), write-data bus (WDB), and read-data bus (RDB), which are wavelength division multiplexed onto the same fiber. Nanophotonic demuxes guide power to only the active PIDRAM chip. B =PIDRAM B =PIDRAM bank; each ring represents multiple rings for multi-wavelength buses (from [9], courtesy of IEEE)

avoid conflicts. The command bus is implemented with a single wavelength on a guided nanophotonic bus. The command wavelength is actively guided to the PIDRAM chip containing the target bank. Once on the PIDRAM chip, a single receiver converts the command into the electrical domain and then electrically broadcasts the command to all banks in the chip. 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 PIDRAM DIMM, and then they are implemented with a shared nanophotonic bus to distribute the data within the PIDRAM chip.

Figure 3.24 illustrates two abstract layouts for a PIDRAM chip. In the *P1* layout shown in Fig. 3.24a, 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. 3.24b, 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,

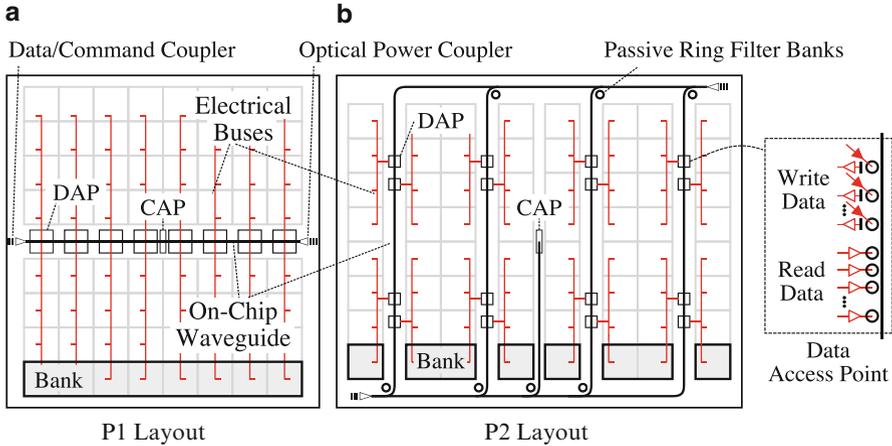


Fig. 3.24 Abstract physical layout for PIDRAM chip. Two layouts are shown for an example PIDRAM chip with eight banks and eight array blocks per bank. For both layouts, the nanophotonic command bus ends at the command access point (CAP), and an electrical H-tree implementation efficiently broadcasts control bits from the command access point to all array blocks. For clarity, the on-chip electrical command bus is not shown. The difference between the two layouts is how far nanophotonics is extended into the PIDRAM chip: (a) *P1* uses nanophotonic chip I/O for the data buses but fully electrical on-chip data bus implementations, and (b) *P2* uses seamless on-chip/off-chip nanophotonics to distribute the data bus to a group of four banks. CAP = command access point; DAP = data access point (adapted from [9], courtesy of IEEE)

and the optically passive ring filter banks at the bottom and top of the waterfall ensure that each of these vertical waveguides only contains a subset of the channel's wavelengths. Each of these vertical waveguides is analogous to the electrical vertical buses in *P1*, so a bank can still be striped across the chip horizontally to allow easy access to the on-chip nanophotonic interconnect. Various layouts are possible that correspond to more or less nanophotonic access points. For a P_n layout, n indicates the number of partitions along each vertical electrical data bus. All of the nanophotonic circuits have to be replicated at each data access point for each bus partition. This increases the fixed link power due to link transceiver circuits and ring heaters. It can also potentially lead to higher optical losses, due to the increased number of rings on the optical path. Our nanophotonic layouts all use the same on-chip command bus implementation as traditional electrical DRAM: a command access point is positioned in the middle of the chip and an electrical H-tree command bus broadcasts the control and address information to all array blocks.

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. Since nanophotonics is an emerging technology, 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. We simulate a range of different designs with each configuration name indicating the layout (Pn), the number of banks ($b8/b64$), and the number of I/Os per array core ($io4/io32$). We use the events and statistics from the simulator to animate our DRAM and nanophotonic device models to compute the energy per bit.

Figure 3.25 shows the energy-efficiency breakdown for various layouts implementing three representative PIDRAM configurations. Each design is subjected to a random traffic pattern at peak utilization and the results are shown for the aggressive and conservative photonic technology projections. Across all designs it is clear that replacing the off-chip links with photonics is advantageous, as $E1$ towers above the rest of the designs. How far photonics is taken on chip, however, 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. The data-independent energy includes: electrical laser power for the write bus, electrical laser power for the read bus, fixed circuit energy including clock and leakage, and thermal tuning energy. As shown in Fig. 3.25a, $P1$ 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. This is due to the fixed energy cost of photonic transceiver circuits and the energy spent on ring thermal tuning. By sharing the photonic access points across eight banks, the optimal design is $P8$. This design 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 $P8$ layout for the $b64-io4$ configuration), the activation energy becomes dominant. Figure 3.25b shows the results for the $b64-io32$ configuration which increases the number of bits we read or write from each array core to 32. This further reduces the activate energy cost, and overall this optimized design is 10 \times more energy efficient than the baseline electrical design. Figure 3.25c shows similar trade-offs for the low-bandwidth $b8-io32$ configuration.

In addition to these results, we also examined the energy as a function of utilization and the area overhead. Figure 3.26 illustrates this trade-off for configurations with 64 banks and four I/Os per array core. As expected, the energy per bit increases as utilization goes down due to the data-independent power components. The large fixed power in electrical DRAM interfaces helps mitigate the fixed power overhead in a nanophotonic DRAM interface at low utilization; these results suggest the potential for PIDRAM to be an energy efficient alternative regardless of utilization. Although not shown, 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.

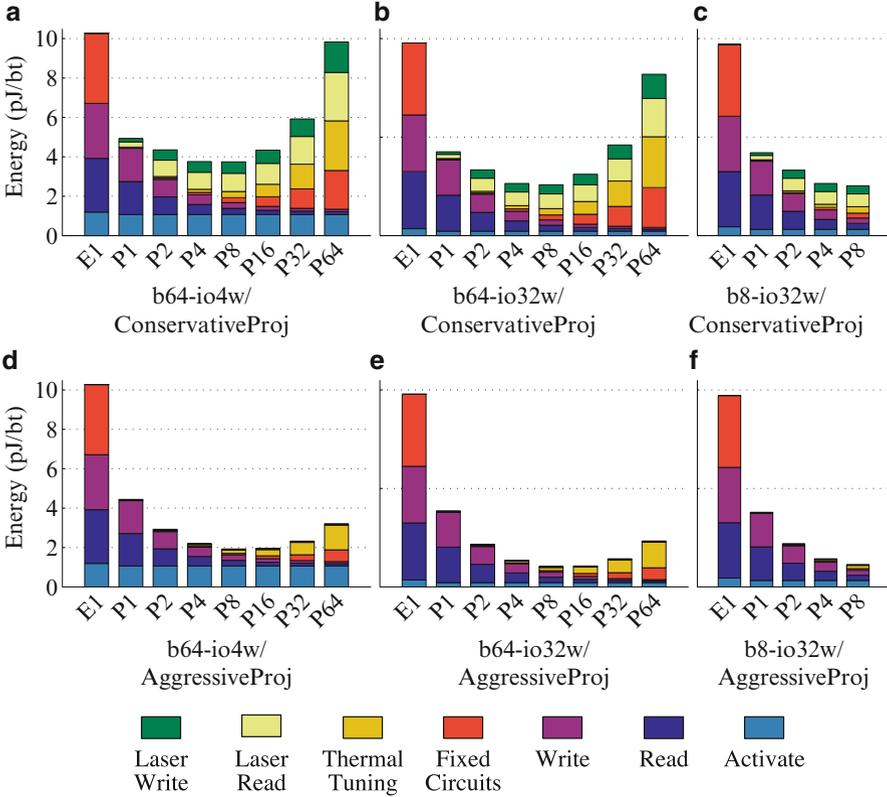


Fig. 3.25 Energy breakdown for DRAM memory channels. Energy results are for uniform random traffic with enough in-flight requests to saturate the DRAM memory channel. (a–c) Assume conservative nanophotonic device projections, while (d–f) assume more aggressive nanophotonic projections. Results for (a), (b), (d), and (e) are at a peak bandwidth of ≈ 500 Gb/s and (c) and (f) are at a peak bandwidth of ≈ 60 Gb/s with random traffic. 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 (from [9], courtesy of IEEE)

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 is 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

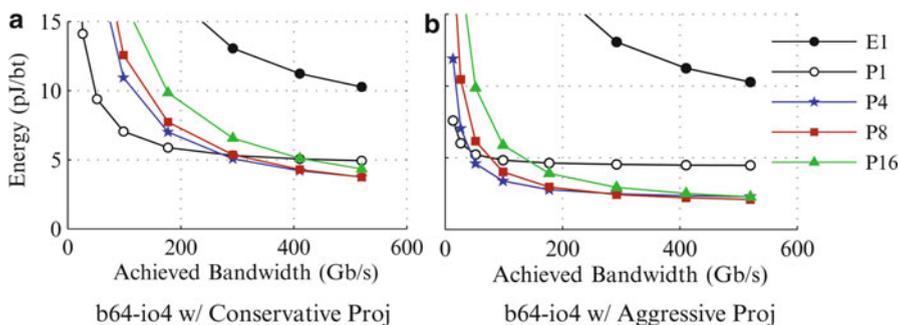


Fig. 3.26 Energy versus utilization. Energy results are for uniform random traffic with varying numbers of in-flight messages. To reduce clutter, we only plot the three most energy efficient waterfall floorplans (*P4*, *P8*, *P16*) (adapted from [9], courtesy of IEEE)

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.

Conclusions

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, Figs. 3.12b and 3.13b 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 for standard topologies.

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. Only exploring a single topology, microarchitecture, or layout ignores some of the trade-offs involved in alternative approaches. 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 only considering bus/channel slicing artificially constrains bus and channel bandwidths as opposed to using a combination of wavelength and bus/channel slicing. Iterating through the three levels of design can enable a much richer exploration of the design space. For example, as discussed in section “Case Study #2: Manycore Processor-to-DRAM Network,” an honest evaluation of our final 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 “Case Study #1: On-Chip Tile-to-Tile Network,” 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 “Case Study #2: Manycore Processor-to-DRAM Network” and “Case Study #3: DRAM Memory Channel”.

Assume a Broad Range of Nanophotonic 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 “Case Study #2: Manycore Processor-to-DRAM Network” helped not only motivate alternative layouts but also an interest in very low-loss waveguide crossings.

Carefully Consider Nanophotonic Fixed-Power Overheads. One of the primary disadvantages of nanophotonic devices are the many forms of fixed power including fixed transceiver circuit power, static thermal tuning power, and optical laser power. These overheads can impact the energy efficiency, on-chip power density, and system-level power. Generating a specific amount of optical laser power can require significant off-chip electrical power, and this optical laser power ultimately ends up as heat dissipation in various nanophotonic devices. Ignoring these overheads or only evaluating designs at high utilization rates can lead to overly optimistic results.

For example, section “Case Study #1: On-Chip Tile-to-Tile Network” 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 “Case Study #3: DRAM Memory Channel”, 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 packing issues. If we can achieve the same benefits with a much simpler network design, then ultimately 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.

Acknowledgements 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. Research 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.

We 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. We would like to acknowledge the MIT nanophotonic device and circuits team, including J. S. Orcutt, A. Khilo, M. A. Popović, 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. Abousamra A, Melhem R, Jones A (2011) Two-hop free-space based optical interconnects for chip multiprocessors. In: International symposium on networks-on-chip (NOCS), May 2011. <http://dx.doi.org/10.1145/1999946.1999961> Pittsburgh, PA
2. Alduino A, Liao L, Jones R, Morse M, Kim B, Lo W, Basak J, Koch B, Liu H, Rong H, Sysak M, Krause C, Saba R, Lazar D, Horwitz L, Bar R, Litski S, Liu A, Sullivan K, Dosunmu O, Na N, Yin T, Haubensack F, Hsieh I, Heck J, Beatty R, Park H, Bovington J, Lee S, Nguyen H, Au H, Nguyen K, Merani P, Hakami M, Paniccia MJ (2010) Demonstration of a high-speed 4-channel integrated silicon photonics WDM link with silicon lasers. In: Integrated photonics research, silicon, and nanophotonics (IPRSN), July 2010. <http://www.opticsinfobase.org/abstract.cfm?URI=iprsn-2010-pdiwi5> Monterey, CA
3. Amatya R, Holzwarth CW, Popović MA, Gan F, Smith HI, Kärtner F, Ram RJ (2007) Low-power thermal tuning of second-order microring resonators. In: Conference on lasers and electro-Optics (CLEO), May 2007. <http://www.opticsinfobase.org/abstract.cfm?URI=CLEO-2007-CFQ5> Baltimore, MA

4. Balfour J, Dally W (2006) Design tradeoffs for tiled CMP on-chip networks. In: International symposium on supercomputing (ICS), June 2006. <http://dx.doi.org/10.1145/1183401.1183430> Queensland, Australia
5. Barwicz T, Byun H, Gan F, Holzwarth CW, Popović MA, Rakich PT, Watts MR, Ippen EP, Kärtner F, Smith HI, Orcutt JS, Ram RJ, Stojanović V, Olubuyide OO, Hoyt JL, Spector S, Geis M, Grein M, Lyszczarz T, Yoon JU (2007) Silicon photonics for compact, energy-efficient interconnects. *J Opt Networks* 6(1):63–73
6. Batten C, Joshi A, Orcutt JS, Khilo A, Moss B, Holzwarth CW, Popović MA, Li H, Smith HI, Hoyt JL, Kärtner FX, Ram RJ, Stojanović V, Asanović K (2008) Building manycore processor-to-DRAM networks with monolithic silicon photonics. In: Symposium on high-performance interconnects (hot interconnects), August 2008 <http://dx.doi.org/10.1109/HOTI.2008.11> Stanford, CA
7. Batten C, Joshi A, Orcutt JS, Khilo A, Moss B, Holzwarth CW, Popović MA, Li H, Smith HI, Hoyt JL, Kärtner FX, Ram RJ, Stojanović V, Asanović K (2009) Building manycore processor-to-DRAM networks with monolithic CMOS silicon photonics. *IEEE Micro* 29(4):8–21
8. Batten C, Joshi A, Stojanović V, Asnaović K (2012) Designing chip-level nanophotonic interconnection networks. *IEEE J Emerg Sel Top Circuits Syst.* <http://dx.doi.org/10.1109/JETCAS.2012.2193932>
9. Beamer S, Sun C, Kwon Y-J, Joshi A, Batten C, Stojanović V, Asanović K (2010) Rearchitecting DRAM memory systems with monolithically integrated silicon photonics. In: International symposium on computer architecture (ISCA), June 2010. <http://dx.doi.org/10.1145/1815961.1815978> Saint-Malo, France
10. Beausoleil RG (2011) Large-scale integrated photonics for high-performance interconnects. *ACM J Emerg Technol Comput Syst* 7(2):6
11. Beux SL, Trajkovic J, O'Connor I, Nicolescu G, Bois G, Paulin P (2011) Optical ring network-on-chip (ORNoC): architecture and design methodology. In: Design, automation, and test in Europe (DATE), March 2011. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5763134> \bibitem{beux-photo-ornoc-date2011} Grenoble, France
12. Bhuyan LN, Agrawal DP (2007) Generalized hypercube and hyperbus structures for a computer network. *IEEE Trans Comput* 33(4):323–333
13. Biberman A, Preston K, Hendry G, Sherwood-Droz N, Chan J, Levy JS, Lipson M, Bergman K (2011) Photonic network-on-chip architectures using multilayer deposited silicon materials for high-performance chip multiprocessors. *ACM J Emerg Technol Comput Syst* 7(2):7
14. Binkert N, Davis A, Lipasti M, Schreiber R, Vantrease D (2009) Nanophotonic barriers. In: Workshop on photonic interconnects and computer architecture, December 2009. Atlanta, GA
15. Block BA, Younkin TR, Davids PS, Reshotko MR, Chang BMPP, Huang S, Luo J, Jen AKY (2008) Electro-optic polymer cladding ring resonator modulators. *Opt Express* 16(22):18326–18333
16. Christiaens I, Thourhout DV, Baets R (2004) Low-power thermo-optic tuning of vertically coupled microring resonators. *Electron Lett* 40(9):560–561
17. Cianchetti MJ, Albonesi DH (2011) A low-latency, high-throughput on-chip optical router architecture for future chip multiprocessors. *ACM J Emerg Technol Comput Syst* 7(2):9
18. Cianchetti MJ, Kerekes JC, Albonesi DH (2009) Phastlane: a rapid transit optical routing network. In: International symposium on computer architecture (ISCA), June 2009. <http://dx.doi.org/10.1145/1555754.1555809> Austin, TX
19. Clos C (1953) A study of non-blocking switching networks. *Bell Syst Techn J* 32:406–424
20. Dally WJ, Towles B (2004) Principles and practices of interconnection networks. Morgan Kaufmann. <http://www.amazon.com/dp/0122007514>
21. DeRose CT, Watts MR, Trotter DC, Luck DL, Nielson GN, Young RW (2010) Silicon microring modulator with integrated heater and temperature sensor for thermal control. In: Conference on lasers and electro-optics (CLEO), May 2010. <http://www.opticsinfobase.org/abstract.cfm?URI=CLEO-2010-CThJ3> San Jose, CA
22. Dokania RK, Apsel A (2009) Analysis of challenges for on-chip optical interconnects. In: Great Lakes symposium on VLSI, May 2009. <http://dx.doi.org/10.1145/1531542.1531607> Paris, France

23. Dumon P, Bogaerts W, Baets R, Fedeli J-M, Fulbert L (2009) Towards foundry approach for silicon photonics: silicon photonics platform ePIXfab. *Electron Lett* 45(12):581–582
24. Georgas M, Leu JC, Moss B, Sun C, Stojanović V (2011) Addressing link-level design tradeoffs for integrated photonic interconnects. In: Custom integrated circuits conference (CICC), September 2011 <http://dx.doi.org/10.1109/CICC.2011.6055363> San Jose, CA
25. Georgas M, Orcutt J, Ram RJ, Stojanović V (2011) A monolithically-integrated optical receiver in standard 45 nm SOI. In: European solid-state circuits conference (ESSCC), September 2011. <http://dx.doi.org/10.1109/ESSCIRC.2011.6044993> Helsinki, Finland
26. Gu H, Xu J, Zhang W (2009) A low-power fat-tree-based optical network-on-chip for multi-processor system-on-chip. In: Design, automation, and test in Europe (DATE), May 2009. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5090624> Nice, France
27. Guha B, Kyotoku BBC, Lipson M (2010) CMOS-compatible athermal silicon microring resonators. *Opt Express* 18(4):3487–3493
28. Gunn C (2006) CMOS photonics for high-speed interconnects. *IEEE Micro* 26(2):58–66
29. Hadke A, Benavides T, Yoo SJB, Amirtharajah R, Akella V (2008) OCDIMM: scaling the DRAM memory wall using WDM-based optical interconnects. In: Symposium on high-performance interconnects (hot interconnects), August 2008. <http://dx.doi.org/10.1109/HOTI.2008.25> Stanford, CA
30. Holzwarth CW, Orcutt JS, Li H, Popović MA, Stojanović V, Hoyt JL, Ram RJ, Smith HI (2008) Localized substrate removal technique enabling strong-confinement microphotonics in bulk-Si-CMOS processes. In: Conference on lasers and electro-optics (CLEO), May 2008. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4571716> San Jose, CA
31. Hwang E, Bhawe SA (2010) Nanophotonic devices on thin buried oxide silicon-on insulator substrates. *Opt Express* 18(4):3850–3857
32. Joshi A, Batten C, Kwon Y-J, Beamer S, Shamim I, Asanović K, Stojanović V (2009) Silicon-photonic Clos networks for global on-chip communication. In: International symposium on networks-on-chip (NOCS), May 2009 <http://dx.doi.org/10.1109/NOCS.2009.5071460> San Diego, CA
33. Kalluri S, Ziari M, Chen A, Chuyanov V, Steier WH, Chen D, Jalali B, Fetterman H, Dalton LR (1996) Monolithic integration of waveguide polymer electro-optic modulators on VLSI circuitry. *Photon Technol Lett* 8(5):644–646
34. Kao Y-H, Chao JJ (2011) BLOCON: a bufferless photonic Clos network-on-chip architecture. In: International symposium on networks-on-chip (NOCS), May 2011. <http://dx.doi.org/10.1145/1999946.1999960> Pittsburgh, PA
35. Kash JA (2008) Leveraging optical interconnects in future supercomputers and servers. In: Symposium on high-performance interconnects (hot interconnects), August 2008. <http://dx.doi.org/10.1109/HOTI.2008.29> Stanford, CA
36. Kim B, Stojanović V (2008) Characterization of equalized and repeated interconnects for NoC applications. *IEEE Design Test Comput* 25(5):430–439
37. Kim J, Balfour J, Dally WJ (2007) Flattened butterfly topology for on-chip networks. In: International symposium on microarchitecture (MICRO), December 2007 <http://dx.doi.org/10.1109/MICRO.2007.15> Chicago, IL
38. Kimerling LC, Ahn D, Apsel AB, Beals M, Carothers D, Chen Y-K, Conway T, Gill DM, Grove M, Hong C-Y, Lipson M, Liu J, Michel J, Pan D, Patel SS, Pomerene AT, Rasras M, Sparacin DK, Tu K-Y, White AE, Wong CW (2006) Electronic-photonic integrated circuits on the CMOS platform. In: Silicon Photonics, March 2006. <http://dx.doi.org/10.1117/12.654455> San Jose, CA
39. Kirman N, Kirman M, Dokania RK, Martínez JF, Apsel AB, Watkins MA, Albonesi DH (2006) Leveraging optical technology in future bus-based chip multiprocessors. In: International symposium on microarchitecture (MICRO), December 2006 <http://dx.doi.org/10.1109/MICRO.2006.28> Orlando, FL
40. Kirman N, Martínez JF (2010) A power-efficient all-optical on-chip interconnect using wavelength-based oblivious routing. In: International conference on architectural support for

- programming languages and operating systems (ASPLOS), March 2010 <http://dx.doi.org/10.1145/1736020.1736024> Pittsburgh, PA
41. Koka P, McCracken MO, Schwetman H, Zheng X, Ho R, Krishnamoorthy AV (2010) Silicon-photonics network architectures for scalable, power-efficient multi-chip systems. In: International symposium on computer architecture (ISCA), June 2010 <http://dx.doi.org/10.1145/1815961.1815977> Saint-Malo, France
 42. Koochi S, Abdollahi M, Hessabi S (2011) All-optical wavelength-routed NoC based on a novel hierarchical topology. In: International symposium on networks-on-chip (NOCS), May 2011 <http://dx.doi.org/10.1145/1999946.1999962> Pittsburgh, PA
 43. Kumar P, Pan Y, Kim J, Memik G, Choudhary A (2009) Exploring concentration and channel slicing in on-chip network router. In: International symposium on networks-on-chip (NOCS), May 2009 <http://dx.doi.org/10.1109/NOCS.2009.5071477> San Diego, CA
 44. Kurian G, Miller J, Psota J, Eastep J, Liu J, Michel J, Kimerling L, Agarwal A (2010) ATAC: a 1000-core cache-coherent processor with on-chip optical network. In: International conference on parallel architectures and compilation techniques (PACT), September 2010. <http://dx.doi.org/10.1145/1854273.1854332> Minneapolis, MN
 45. Leiserson CE (1985) Fat-trees: universal networks for hardware-efficient supercomputing. *IEEE Trans Comput C-34*(10):892–901
 46. Leu JC, Stojanović V (2011) Injection-locked clock receiver for monolithic optical link in 45 nm. In: Asian solid-state circuits conference (ASSCC), November 2011. <http://dx.doi.org/10.1109/ASSCC.2011.6123624> Jeju, Korea
 47. Li Z, Mohamed M, Chen X, Dudley E, Meng K, Shang L, Mickelson AR, Joseph R, Vachharajani M, Schwartz B, Sun Y (2010) Reliability modeling and management of nanophotonic on-chip networks. In: *IEEE Transactions on very large-scale integration systems (TVLSI)*, PP(99), December 2010
 48. Li Z, Mohamed M, Chen X, Zhou H, Michelson A, Shang L, Vachharajani M (2011) Iris: a hybrid nanophotonic network design for high-performance and low-power onchip communication. *ACM J Emerg Technol Comput Syst* 7(2):8
 49. Liow T-Y, Ang K-W, Fang Q, Song J-F, Xiong Y-Z, Yu M-B, Lo G-Q, Kwong D-L (2010) Silicon modulators and germanium photodetectors on SOI: monolithic integration, compatibility, and performance optimization. *J Sel Top Quantum Electron* 16(1):307–315
 50. Lipson M (2006) Compact electro-optic modulators on a silicon chip. *J Sel Top Quantum Electron* 12(6):1520–1526
 51. Manipatruni S, Dokania RK, Schmidt B, Sherwood-Droz N, Poitras CB, Apsel AB, Lipson M (2008) Wide temperature range operation of micrometer-scale silicon electro-optic modulators. *Opt Lett* 33(19):2185–2187
 52. Masini G, Colace L, Assanto G (2003) 2.5 Gbit/s polycrystalline germanium-on-silicon photodetector operating from 1.3 to 1.55 μ m. *Appl Phys Lett* 82(15):5118–5124
 53. Mejia PV, Amirtharajah R, Farrens MK, Akella V (2011) Performance evaluation of a multicore system with optically connected memory modules. In: International symposium on networks on-chip (NOCS), May 2011 <http://dx.doi.org/10.1109/NOCS.2010.31> Grenoble, France
 54. Mesa-Martinez FJ, Nayfach-Battilana J, Renau J (2007) Power model validation through thermal measurements. In: International symposium on computer architecture (ISCA), June 2007 <http://dx.doi.org/10.1145/1273440.1250700> San Diego, CA
 55. Miller DA (2009) Device requirements for optical interconnects to silicon chips. *Proc. IEEE* 97(7):1166–1185
 56. Morris R, Kodi A (2010) Exploring the design of 64 & 256 core power efficient nanophotonic interconnect. *J Sel Top Quantum Electron* 16(5):1386–1393
 57. Nitta C, Farrens M, Akella V (2011) Addressing system-level trimming issues in onchip nanophotonic networks. In: International symposium on high-performance computer architecture (HPCA), February 2011. <http://dx.doi.org/10.1109/HPCA.2011.5749722> San Antonio, TX
 58. Orcutt JS, Khilo A, Holzwarth CW, Popović MA, Li H, Sun J, Bonifield T, Hollingsworth R, Kaärtner FX, Smith HI, Stojanović V, Ram RJ (2011) Nanophotonic integration in state-of-the-art CMOS foundaries. *Opt Express* 19(3):2335–2346

59. Orcutt JS, Khilo A, Popovic MA, Holzwarth CW, Li H, Sun J, Moss B, Dahlem MS, Ippen EP, Hoyt JL, Stojanović V, Kaärtner FX, Smith HI, Ram RJ (2009) Photonic integration in a commercial scaled bulk-CMOS process. In: International conference on photonics in switching, September 2009. <http://dx.doi.org/10.1109/PS.2009.5307769> Pisa, Italy
60. Orcutt JS, Khilo A, Popovic MA, Holzwarth CW, Moss B, Li H, Dahlem MS, Bonifield TD, Kaärtner FX, Ippen EP, Hoyt JL, Ram RJ, Stojanović V (2008) Demonstration of an electronic photonic integrated circuit in a commercial scaled bulk-CMOS process. In: Conference on lasers and electro-optics (CLEO), May 2008. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4571838> San Jose, CA
61. Orcutt JS, Tang SD, Kramer S, Li H, Stojanović V, Ram RJ (2011) Low-loss polysilicon waveguides suitable for integration within a high-volume polysilicon process. In: Conference on lasers and electro-optics (CLEO), May 2011. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5950452> Baltimore, MD
62. Pan Y, Kim J, Memik G (2010) FlexiShare: energy-efficient nanophotonic crossbar architecture through channel sharing. In: International symposium on high-performance computer architecture (HPCA), January 2010. <http://dx.doi.org/10.1109/HPCA.2010.5416626> Bangalore, India
63. Pan Y, Kumar P, Kim J, Memik G, Zhang Y, Choudhary A (2009) Firefly: illuminating on-chip networks with nanophotonics. In: International symposium on computer architecture (ISCA), June 2009. <http://dx.doi.org/10.1145/1555754.1555808> Austin, TX
64. Pasricha S, Dutt N (2008) ORB: an on-chip optical ring bus communication architecture for multi-processor systems-on-chip. In: Asia and South Pacific design automation conference (ASP-DAC), January 2008 <http://dx.doi.org/10.1109/ASPDAC.2008.4484059> Seoul, Korea
65. Petracca M, Lee BG, Bergman K, Carloni LP (2009) Photonic NoCs: system-level design exploration. *IEEE Micro* 29(4):74–77
66. Poon AW, Luo X, Xu F, Chen H (2009) Cascaded microresonator-based matrix switch for silicon on-chip optical interconnection. *Proc IEEE* 97(7):1216–1238
67. Preston K, Manipatruni S, Gondarenko A, Poitras CB, Lipson M (2009) Deposited silicon high-speed integrated electro-optic modulator. *Opt Express* 17(7):5118–5124
68. Reed GT (2008) Silicon photonics: the state of the art. Wiley-Interscience. <http://www.amazon.com/dp/0470025794>
69. Shacham A, Bergman K, Carloni LP (2008) Photonic networks-on-chip for future generations of chip multiprocessors. *IEEE Trans Comput* 57(9):1246–1260
70. Sherwood-Droz N, Preston K, Levy JS, Lipson M (2010) Device guidelines for WDM interconnects using silicon microring resonators. In: Workshop on the interaction between nanophotonic devices and systems (WINDS), December 2010. Atlanta, GA
71. Sherwood-Droz N, Wang H, Chen L, Lee BG, Biberman A, Bergman K, Lipson M (2008) Optical 4x4 hitless silicon router for optical networks-on-chip. *Opt Express* 16(20):15915–15922
72. Skandron K, Stan MR, Huang W, Velusamy S, Sankaranarayanan K, Tarjan D (2003) Temperature-aware microarchitecture. In: International symposium on computer architecture (ISCA), June 2003. <http://dx.doi.org/10.1145/871656.859620> San Diego, CA
73. Thourhout DV, Campenhout JV, Rojo-Romeo P, Regreny P, Seassal C, Binetti P, Leijtens XJM, Notzel R, Smit MK, Cioccio LD, Lagahe C, Fedeli J-M, Baets R (2007) A photonic interconnect layer on CMOS. In: European conference on optical communication (ECOC), September 2007. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5758445 Berlin, Germany
74. Udipi AN, Muralimanohar N, Balasubramonian R, Davis A, Jouppi N (2011) Combining memory and a controller with photonics through 3D-stacking to enable scalable and energy-efficient systems. In: International symposium on computer architecture (ISCA), June 2011. <http://dx.doi.org/10.1145/2000064.2000115> San Jose, CA
75. Vantrease D, Binkert N, Schreiber R, Lipasti MH (2009) Light speed arbitration and flow control for nanophotonic interconnects. In: International symposium on microarchitecture (MICRO), December 2009. <http://dx.doi.org/10.1145/1669112.1669152> New York, NY

76. Vantrease D, Schreiber R, Monchiero M, McLaren M, Jouppi NP, Fiorentino M, Davis A, Binkert N, Beausoleil RG, Ahn JH (2008) Corona: system implications of emerging nanophotonic technology. In: International symposium on computer architecture (ISCA), June 2008. <http://dx.doi.org/10.1109/ISCA.2008.35> Beijing, China
77. Watts MR, Zortman WA, Trotter DC, Nielson GN, Luck DL, Young RW (2009) Adiabatic resonant microrings with directly integrated thermal microphotronics. In: Conference on lasers and electro-optics (CLEO), May 2009. <http://www.opticsinfobase.org/abstract.cfm?URI=CLEO-2009-CPDB10> Baltimore, MD
78. Xue J, Garg A, Çiftçio lu B, Hu J, Wang S, Savidis I, Jain M, Berman R, Liu P, Huang M, Wu H, Friedman E, Wicks G, Moore D (2010) An intra-chip free-space optical interconnect. In: International symposium on computer architecture (ISCA), June 2010 <http://dx.doi.org/10.1145/1815961.1815975> Saint-Malo, France
79. Young IA, Mohammed E, Liao JTS, Kern AM, Palermo S, Block BA, Reshotko MR, Chang PLD (2010) Optical I/O technology for tera-scale computing. *IEEE J Solid-State Circuits* 45(1):235–248
80. Zhao W, Cao Y (2006) New generation of predictive technology model for sub-45 nm early design exploration. *IEEE Trans Electron Dev* 53(11):2816–2823
81. Zheng X, Lexau J, Luo Y, Thacker H, Pinguet T, Mekis A, Li G, Shi J, Amberg P, Pinckney N, Raj K, Ho R, Cunningham JE, Krishamoorthy AV (2010) Ultra-low energy all-CMOS modulator integrated with driver. *Opt Express* 18(3):3059–3070
82. Zhou L, Djordjevic SS, Proietti R, Ding D, Yoo SJB, Amirharajah R, Akella V (2009) Design and evaluation of an arbitration-free passive optical crossbar for onchip interconnection networks. *Appl Phys A Mater Sci Process* 95(4):1111–1118
83. Zhou L, Okamoto K, Yoo SJB (2009) Athermalizing and trimming of slotted silicon microring resonators with UV-sensitive PMMA upper-cladding. *Photon Technol Lett* 21(17):1175–1177
84. Zortman WA, Trotter DC, Watts MR (2010) Silicon photonics manufacturing. *Opt Express* 18(23):23598–23607