J. Parallel Distrib. Comput. 71 (2011) 641–650

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Time-division-multiplexed arbitration in silicon nanophotonic networks-on-chip for high-performance chip multiprocessors

Gilbert Hendry [a,*], Eric Robinson [b], Vitaliy Gleyzer [b], Johnnie Chan [a], Luca P. Carloni [c], Nadya Bliss [b], Keren Bergman [a]

[a] Lightwave Research Laboratory, Department of Electrical Engineering, Columbia University, 500 W 120th St, Mudd 1300, New York, NY 10027, United States

[b] Lincoln Laboratory, Massachusetts Institute of Technology, 244 Wood St, Lexington, MA 02420, United States

[c] Department of Computer Science, Columbia University, 450 Computer Science Building, 1214 Amsterdam Ave., Mailcode: 0401, New York, NY 10027, United States

ARTICLE INFO

Article history:
Received 12 April 2010
Received in revised form
15 August 2010
Accepted 14 September 2010
Available online 8 October 2010

Keywords:
Networks-on-chip
Photonic interconnection networks
Silicon photonics
Time division multiplexing
Memory systems

ABSTRACT

As the computational performance of microprocessors continues to grow through the integration of an increasing number of processing cores on a single die, the interconnection network has become the central subsystem for providing the communications infrastructure among the on-chip cores as well as to off-chip memory. Silicon nanophotonics as an interconnect technology offers several promising benefits for future networks-on-chip, including low end-to-end transmission energy and high bandwidth density of waveguides using wavelength division multiplexing. In this work, we propose the use of time-division-multiplexed distributed arbitration in a photonic mesh network composed of silicon micro-ring resonator based photonic switches, which provides round-robin fairness to setting up photonic circuit paths. Our design sustains over 10× more bandwidth and uses less power than the compared network designs. We also observe a 2× improvement in performance for memory-centric application traces using the MORE modeling system.

1. Introduction

Current trends in computer architecture indicate that the network-on-chip will play a critical role in determining future high-performance microprocessors. How this role is played out will impact many areas of computing, including the programming models, architecture designs and manufacturing.

It is becoming apparent that electronics may not be able to solve all the communications challenges in high-performance computing. Increased off-chip bandwidth means higher IO pin counts, a requirement that may become unrealistic if memory bandwidth is to be balanced with computational capabilities. Using electronic links to connect a microprocessor to memory on a board presents critical design trade-offs between the wire lengths, memory capacity, datarate, and power consumed by the IO.

Photonics offers key advantages in these areas, and deserves serious consideration as the leading communications technology of future high-performance chip multiprocessors. Using wavelength division multiplexing (WDM), a technique of transmitting many optical signals on different wavelengths simultaneously in the same transmission medium, photonics can achieve a bandwidth density orders of magnitude higher than electronics, which can greatly alleviate package IO pin constraints. Additionally, photonics provides an extremely energy efficient end-to-end transmission technology that is largely independent of the datarate and distance. Unlike electronics, the distance traveled in a waveguide or optical fiber is virtually independent of the energy spent, which is also decoupled from datarate.

Recent numerous advances in silicon photonic integration and the emerging field of CMOS photonics [3,10,21,36,23] allows us to consider practical designs for full-scale first generation interconnects in this technology platform. Many such novel photonic-enabled network architectures have been recently proposed that can deliver performance improvement over equivalent electronic interconnect designs [35,26,17,1,5,28,14].

In this work, we propose an improvement to an all-optical broadband network that uses time division multiplexing (TDM) to arbitrate setting up communication circuit-paths, first proposed in [11]. This network architecture is able to achieve high bandwidths between communicating pairs and better network resource utilization, while providing round-robin fairness to network requests through distributed control of photonic switches.

We evaluate an instantiation of the network connecting 256 cores with 128 GB of memory using both random network traffic and a detailed trace of an embedded computing application. We find that our design achieves over 10× higher total network

* Corresponding author.
E-mail addresses: gilbert@ee.columbia.edu, gilbert.hendry@gmail.com (G. Hendry), erobinson@ll.mit.edu (E. Robinson), vgleyzer@ll.mit.edu (V. Gleyzer), johnnie@ee.columbia.edu (J. Chan), luca@cs.columbia.edu (L.P. Carloni), nt@ll.mit.edu (N. Bliss), bergman@ee.columbia.edu (K. Bergman).

0743-7315/$ – see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2010.09.009

bandwidth over other solutions, including previously proposed circuit-switched and TDM-arbitrated solutions leading to lower latencies at high loads and lower power consumption. In addition, our design is 2× faster when running a projective transform, a key high-performance embedded computing kernel for signal and image processing.

2. Related work

Research into photonic circuit-switched networks-on-chip has progressed in the past few years, leading to a more complete understanding of the challenges both at the system level and the device level.

Many advances have been made towards the integration of silicon nanophotonic devices into the traditional CMOS production line. Ring resonators have become a prevalent building block for broadband spatial switches, wavelength filters, and modulators because of their low area and power consumption [38].

However, device temperature stability and manufacturing defects still remain as significant barriers to full-fledged integration. Currently, both of these problems are solved by heating the individual device, changing the effective index of refraction of the material and tuning the ring to the correct resonance [4]. Other solutions include manufacturing and design techniques to make more athermal devices [10].

System-level implications of photonics has made a large impact on the way architects are thinking of future CMPs. Reducing the cost of cross-chip, off-chip, and chip–chip communication allows a system designer to rethink programming models, memory hierarchy, and cost-performance optimization.

Next-generation NoC designs using silicon nanophotonic technology have been proposed in other works. The Corona network is an example of a network that uses optical arbitration via a wavelength-routed token ring to reserve access to a full serpentine crossbar made from redundant waveguides, modulators, and detectors [35]. Similarly, wavelength-routed bus based architectures have been proposed which take advantage of WDM for arbitration [26,17].

Batten et al. proposed an architecture using source routing and wavelength arbitration for off-chip communications which takes advantage of WDM to dedicate wavelengths to different DRAM banks, forming a large wavelength-tuned ring resonator matrix as a central crossbar [1]. Phastlane was designed for a cache-coherent CMP, enabling snoop broadcasts and cache line transfers in the optical domain [5].

3. Photonic circuit switching

On-chip hybrid circuit-switched photonic networks using an electronic control plane have been proposed by Shacham et al. [33] and Petracca [28]. The fundamental switching unit in these designs is the photonic switching element (PSE), which is a micro-ring resonator that is able to shift its periodic resonance to align with the optical signals present in the nearby waveguide by injecting carriers through a p–i–n junction. This operation is shown in Fig. 1.

These PSEs are strictly spatial switches, much like conventional electronic ones, which means paths from one port to another must be arbitrated before data can be sent through them. This issue is further complicated by the fact that no photonic equivalent of a buffer exists, making it a requirement that the path be completely set up from source to destination before data can be passed through the network. This has been solved in the past by using a conventional packet-switched electronic control network which circuit-switches a photonic data plane [33].

The idea behind this method is that once the optical path is set up between two nodes, the transmission of the data can amortize the setup latency with high bandwidth WDM. Also, PSEs are transparent to bit rate, meaning that energy is only dissipated at the modulators and detectors for each bit, making the end-to-end transmission energy practically distance independent.

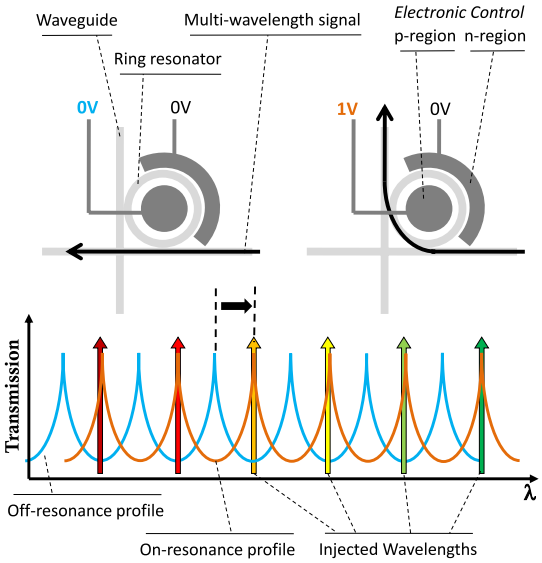


Fig. 1. PSE operation, switching from OFF to ON state, shifting wavelengths.

However, circuit switching in this way contains no implicit mechanism which ensures fairness, which can lead to degraded performance due to path blocking if messages are short or require the same photonic resources [12].

This work improves on these designs by removing the electronic control network responsible for allocating network resources, replacing it with a time-division-multiplexing distributed arbitration of photonic switches.

4. TDM arbitration

We propose using time division multiplexing (TDM) to arbitrate end-to-end photonic circuit paths in a network of ring resonator based photonic switches. The basic concept behind this is that during a specified amount of time, or time slot, switches in the network are configured to allow communication between one or more pairs of access points. Each time slot is of length

$$t_{\text{slot}} = t_{\text{setup}} + t_{\text{transmission}} + t_{\text{propagation}} \quad (1)$$

where t_{setup} is the time it takes to change the state of all PSEs at once, $t_{\text{transmission}}$ is the time each node is allowed to transmit data per time slot, and $t_{\text{propagation}}$ is the worst-case propagation latency between any two valid communicating pairs. If each switch is able to keep track of the current time slot using a global clock, this allows the control of the switches to be completely distributed in that they need not communicate with each other.

This concept should be distinguished from TDM mechanisms in other networks. Typically, requests to use network resources is arbitrated by sources or individual network nodes to dynamically allocate a temporal schedule for access to virtual channels, physical links, switches, or virtual circuits, thus providing fairness guarantees to latency and bandwidth [25,9,31,22,27].

Our method aims at providing the same fairness, but because there are no equivalent of buffers in photonic technology, we must apply TDM arbitration through the entire network creating end-to-end optical circuit paths. Here, the scheduling of all the nodes' accesses to network resources is done statically at design time. If there are N_{slot} time slots, each of duration t_{slot}, then the total TDM frame, T_{TDM} is

$$T_{\text{TDM}} = N_{\text{slot}} \times t_{\text{slot}}. \quad (2)$$

For the design proposed in this paper, we require that full network communication *coverage* is implemented, or that every network node is able to send messages to every other node within T_{TDM}.

During TDM arbitration, the network repeatedly cycles through every time slot. If a network node has data to send to another node, it waits for the correct time slot. If a node has multiple messages to different destinations queued up, it can send them out of order. Also, by statically selecting different values for $t_{\text{transmission}}$, we can vary the granularity of the arbitration. If, for instance, the system architecture specifies that only fixed-length messages may be sent on the network (i.e. cache lines), then we can adjust $t_{\text{transmission}}$ to exactly match that size.

The naive way to accomplish this is to assign a time slot to every possible communicating pair in the network. Thus, we would require

$$N_{\text{slot}} = N \times (N - 1) \tag{3}$$

time slots to implement full coverage, where N is the number of nodes in the network. A 64-node network would therefore require 4032 time slots. This naive scheduling of one path per time slot in the network achieves the worst-case network utilization. As we will see, it is easily possible to statically allocate the network to many transmissions during a single time slot.

4.1. Enhanced TDM arbitration

We can improve on the naive implementation by scheduling more than one transmission per time slot, thus reducing the total number of time slots, and the worst-case latency of a message waiting for its slot. In order to maintain correct operation we must adhere to the following constraints during a single time slot:

1. *Source contention*—A node can only send to one destination at a time, assuming a single set of modulators at an access point.
2. *Destination contention*—A node can only receive from one source at a time, assuming a single set of detectors at an access point.
3. *Topology contention*—Transmission cannot overlap in the same waveguide.

A method for statically scheduling end-to-end TDM-arbitrated optical transmissions was discussed previously by Hendry [11], using a genetic algorithm to search the solution space. In this work, we aim to improve on that implementation by decreasing the number of time slots required. Instead of searching the solution space, we will simplify the problem and describe a method we can apply manually to a mesh topology for scheduling which is scalable and results in significantly fewer time slots.

To simplify the problem, let us first concede that photonic transmission will no longer be entirely end-to-end for every node pair. Rather, the mesh X-dimension transmission is first completed, converted to the electronic domain, and stored in a buffer until the Y-dimension transmission can be completed. This means that we will pay optical to electrical conversion energy costs twice. This simplification reduces the energy per bit benefits that end-to-end photonic transmission technology provides, but we will leave this to our discussion of our results in Section 7.

We can first observe that two transmissions can always take place in a row during a time slot, for any size row, where the two sending nodes are on opposite sides of the row. This is illustrated in Fig. 2 for one row of four nodes, assuming bidirectional links connecting neighboring nodes consisting of two uni-directional waveguides. The red nodes are the sending nodes, and exhaust all possible combinations of destinations (green) in the row. The process repeats for all other nodes being designated as the sending nodes. Note that communications are shown to be symmetric across the midpoint of the row in Fig. 2, though this is not required.

We now make it our goal to schedule communications similar to Fig. 2 such that two transmissions occur in every row *and*

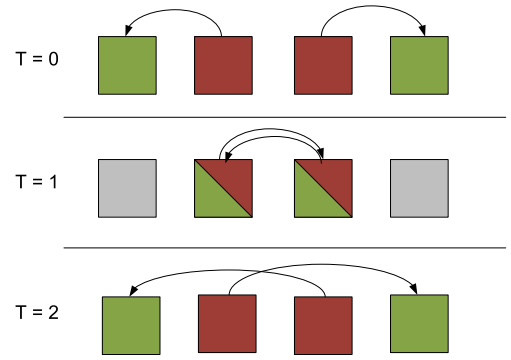


Fig. 2. Row communication TDM slot examples for four nodes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

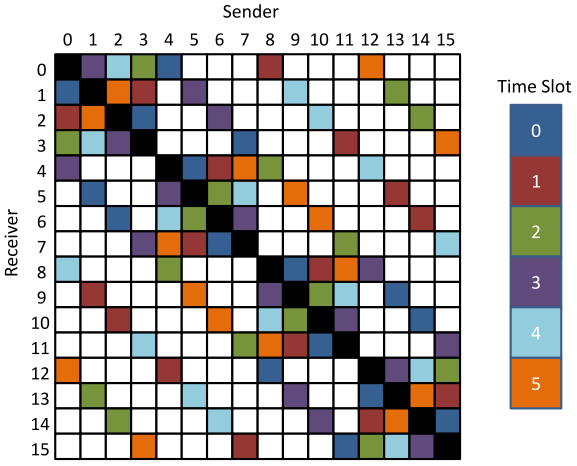


Fig. 3. Control matrix for a 4 × 4 network.

every column in each time slot. Since each node in a row must communicate with every other in its row ($R - 1$ of them), and two nodes are communicating at once per row, we would require

$$N_{\text{slot}} = (R - 1) \times \left(\frac{R}{2}\right) \tag{4}$$

time slots, where R is the number of nodes in a row (and column, assuming a square network), R is even, and $R \geq 4$. For an 8 × 8 64-node network, this is merely 28 time slots, a significant improvement over the previous end-to-end implementation with 142 time slots [11].

Fig. 3 illustrates an example of how to schedule a 4 × 4 TDM network, which requires 6 time slots. We represent the transmission possibilities as a 16 × 16 *control matrix*. Each entry in the matrix is color-coded to indicate which sender–receiver pair is enabled during a time slot. Note that a node may only send and receive once per time slot, which translates into the rule that a color may only appear once in a row and column in the control matrix. Also note that not all node combinations are necessary because we conceded that optical circuit paths only travel in one mesh dimension during a slot, which is why many control matrix entries are blank (white).

Some visual and numerical patterns are useful when specifying the control matrix for any size network. For instance, the 4 × 4 squares lying on the black diagonal indicate row communications. Other diagonal stripes represent column communication. First, all row communications are added, each block (row) utilizing every time slot exactly twice, as shown in Fig. 2. The block pattern shifts slightly to accommodate column communications, and is mirrored across the network bisection line (row $R/2$).

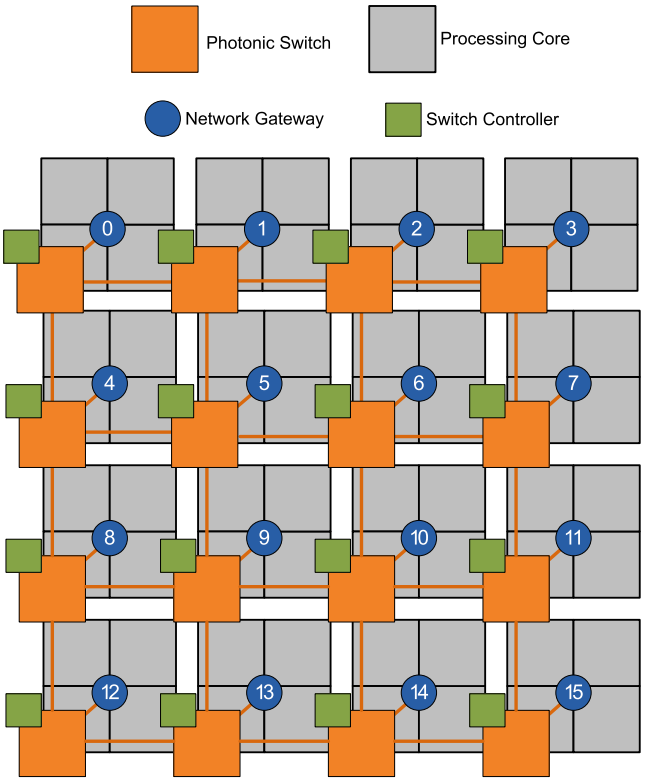


Fig. 4. Network architecture.

5. Network implementation

Fig. 4 shows a 64-core example of a CMP using a 4 × 4 instantiation of our photonic TDM network, consisting of three basic components: a photonic switch, switch controller, and network gateway. Switches are arranged in a mesh, each controlled by their controller. Each gateway connects four cores, known as gateway concentration. In addition, our network design has an added advantage that it is tiled, aligning with today's chip design flow and manufacturing techniques. In this section, we also describe the design for an optically attached DRAM memory module attached to each gateway.

5.1. Photonic switch

Fig. 5 shows the layout for the photonic switches in the network. It consists of waveguide paths and PSEs, operating as in Fig. 1. Ports are labeled as North, South, East, West, and Gateway.

Because we optimized our arbitration for fewer TDM slots at the cost of paying O–E–O energy by doing *X*-then-*Y* routing, the switch does not need to implement full connectivity between the ports. Table 1 shows the port combinations, and the PSE number that implements the path, referring to Fig. 5. For example, we can see in Fig. 5 that the PSE labeled as 1 can switch a signal from the gateway (modulator bank) to the North port. Note that the signal must pass through a ring only when coming from a gateway and entering a gateway, which saves on insertion loss when traveling in straight lines.

5.2. Switch controller

In the proposed network architecture, each switch is controlled by a local controller which is aware of the current TDM slot by tracking ticks of a global TDM clock, and is therefore aware of how the switch should be set. A global, synchronous TDM clock can be implemented with waterfall clock distribution, synchronous latency-insensitive design [7], or optical clock distribution [39].

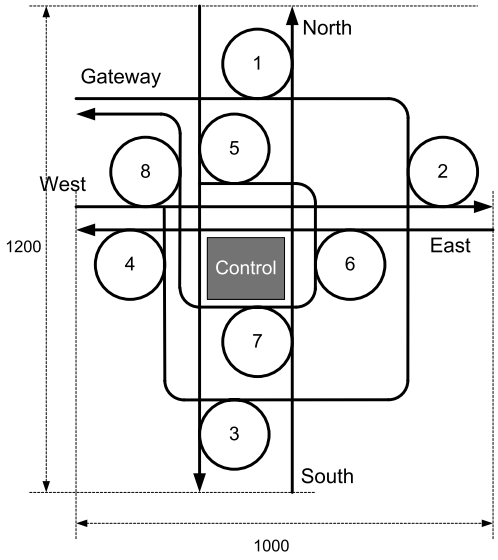


Fig. 5. Layout of photonic switch, showing waveguides and ring resonators. Units in microns.

Table 1
Switch functionality.

Inport	Outport	PSE
Mod	N	1
Mod	E	2
Mod	S	3
Mod	W	4
N	Det	5
E	Det	6
S	Det	7
W	Det	8
E/W	W/E	N/A
N/S	S/N	N/A

The period of this clock must be the TDM period, t_{slot}. As indicated later in Section 7, t_{slot} should be set to an expected average message transmission time, so that time slots are just big enough to allow end-to-end transmission. Taking into account the time slot overheads, this value could be at least ten nanoseconds equating to less than 100 MHz TDM clock frequency (depending on t_{slot}), a very feasible implementation by today's standards.

The output logic can be implemented as a single lookup table (LUT) which takes the switch ID register as an input, allowing identical ROM instantiation among network tiles. In practice, only the fraction of the table that is necessary to run the local switch would be instantiated to save area and power.

The size of the output logic is proportional to the number of TDM slots, which is dictated by the number of network nodes. Specifically, there is one bit per PSE per TDM slot, indicating whether the PSE is on or off. Since there are 8 PSEs per switch, this means that the ROM of each switch controller contains N_{slot} bytes of information. Referring to Eq. (4), a 64-node network needs a 28-byte LUT per switch.

5.3. Network gateway

Fig. 6 shows the microarchitecture of a network gateway, providing network and memory access to four cores. This is accomplished through the use of a main *TDM controller*, which arbitrates network and memory resources and acts as a memory controller by keeping a master schedule of events that occur during each time slot.

Each gateway has two vertically coupled [32] connections to a memory bank. Local reads and writes are serviced by scheduling row and column accesses during free slots in the master schedule.

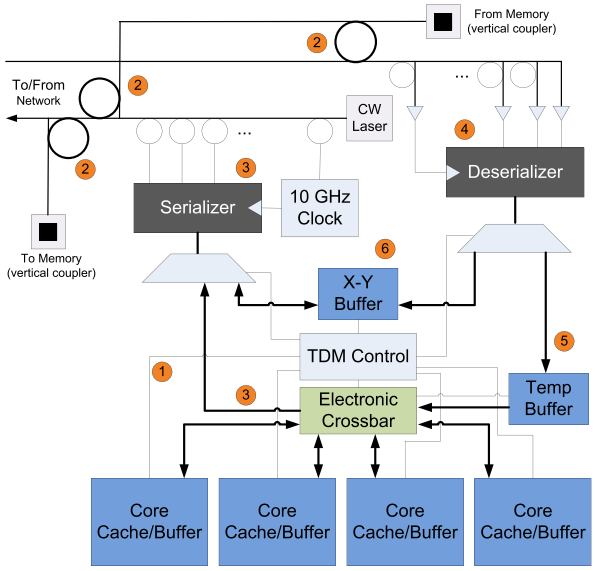


Fig. 6. Network gateway microarchitecture.

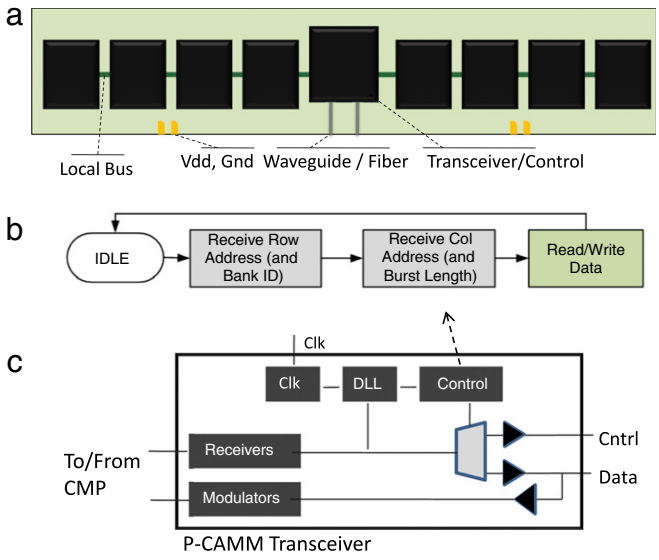


Fig. 7. Photonic Circuit-Accessed Memory Module design (a) Photonic CAMM (b) P-CAMM control logic (c) P-CAMM Transceiver.

Remote memory accesses are sent to the destination gateway, where they are then scheduled in a similar fashion. Remote reads are read directly from memory into the network to save on buffering power.

The following describes an example of the gateway operation, numbered in Fig. 6:

1. Communication requests are made to the TDM controller, which controls an electronic crossbar that connects the various gateway components.
2. When the network is in the correct TDM slot, depending on the type of communication (memory read, memory write, MPI-send, etc.), the TDM controller sets the broadband rings that control access to and from the modulators and detectors. This can also be done ahead of time when the time slot switches, if the transaction has been queued up.
3. The TDM controller also sets the crossbar from the requesting core to the serializer, which ramps the data up to 10 Gb/s modulation. The transmission clock is also transmitted on a separate wavelength.
4. When a signal is received, it is first deserialized, clocked by the received transmission clock.
5. If the data has reached its destination, it sits in a temporary buffer, waiting for access to the electronic crossbar. Access will be immediately available unless cores in the same gateway are communicating locally through the crossbar.
6. If the data is using the gateway as an intermediate point while switching dimensions, it sits in the X–Y buffer and notifies the TDM controller. It can then transmit during the correct TDM slot.

The sizes of the buffers can be exactly specified based on the size of the network. The X–Y buffer is used to hold transmissions that have arrived at this gateway to continue through the network in a different direction, and are waiting for their time slot. Therefore, they must hold a maximum of $2 \times (R - 1)$ transmissions, which is the number of time slots in one TDM frame in which a message could be received. A 64-node network will therefore require a buffer of size $14 \times S_{\text{transmission}}$, where $S_{\text{transmission}}$ is the maximum message size that can be transmitted in one time slot.

The temporary buffer is only used to store received transmissions that are destined for the cores in the gateway. The TDM controller gives priority to the temporary buffer over local core–core communication, therefore it needs to hold a maximum of 2 transmissions: one for receiving incoming transmissions, and one for sending the last received transmission on to the correct core.

A key characteristic of the gateway design is its ability to handle cores' requests out of order. For example, if core 0 requests that a message be sent but must wait for the correct time slot, the controller is free to grant access to subsequent requests from any of the other cores connected by the gateway. In a purely circuit-switched network with a dynamic path setup implementation, the first request must be handled first, potentially head-of-line blocking other cores.

5.4. Circuit-Accessed Memory Module

Our proposed memory access architecture uses a DRAM module in a less conventional way, which requires a redesign of the basic memory module discussed in the previous work [13]. Fig. 7(a) shows the Photonic Circuit-Accessed Memory Module (P-CAMM) design. Individual DRAM chips are connected via a local electronic bus to a central optical controller/transceiver, shown in Fig. 7(c). The controller (Fig. 7(b)) is responsible for demultiplexing the single optical channel into the address and data bus much in the same way as Rambus RDRAM memory technology [29], using the simple control flowchart shown. This shift from electrical to photonic technology presents significant advantages for the physical design and implementation of off-chip signaling.

Although the P-CAMM shown in Fig. 7(a) retains the contemporary SDRAM DIMM form factor, this is not required due to the alleviated pinning requirements. The memory module can then be designed for larger, smaller, or more dense configurations of DRAM chips. Furthermore, the memory module can be placed arbitrarily distant from the processor using low-loss optical fiber without incurring any additional power or optical loss. Latency is also minimal, paying 4.9 ns/m [6].

Additionally, the driver and receiver banks use much less power for photonics using ring resonator based modulators and SiGe detectors than for off-chip electronic I/O wires [3].

6. Experimental setup

We evaluate a 64-node enhanced TDM photonic network implementation (P-ETDM) using external concentration [18] for a total of 256 cores and compare it against a circuit-switched

Table 2
Optical device energy parameters.

Parameter	Value
Datarate (per wavelength)	2.5 Gb/s
PSE dynamic energy	375 fJ[a]
PSE static (OFF) energy	400 uJ/s[b]
Modulation switching energy	25 fJ/bit[c]
Modulation static energy	30 μW[d]
Detector energy	50 fJ/bit[e]
Thermal tuning energy	1 uW/K[f]

[a] Dynamic energy calculation based on carrier density, 50-μm ring, 320 × 250 nm waveguide, 75% exposure, 1-V bias.
[b] Based on switching energy, including photon lifetime for re-injection.
[c] Same as[a] that for a 3 μm ring modulator.
[d] Based on experimental measurements in [36]. Calculated for half a 10 GHz clock cycle, with 50% probability of a 1-bit.
[e] Conservative approximation assuming femto-farad class receiverless SiGe detector with $C < 1$ fF.
[f] Same value as used in [14]. Average of 20° thermal tuning required.

Table 3
Optical device loss parameters.

Device	Insertion loss
Waveguide propagation	1.5 dB/cm[a]
Waveguide crossing	0.05[b]
Waveguide bend	0.005 dB/90°[a]
Passing by ring (Off)	≈0[c]
Insertion into ring (On)	0.5[c]
Optical power budget	35 dB

[a] From [37].
[b] Projections based on [8].
[c] From [19].

photonic mesh (P-mesh) designated PS-1 in [13] and the original TDM network design (P-TDM) [11]. We describe the relevant modeling and parameters below.

6.1. Simulation environment

We use a simulation and CAD environment called PhoenixSim [2], developed for the analysis of electronic and photonic networks-on-chip. PhoenixSim includes a cycle-accurate network simulator which captures physical-layer details, such as physical dimensions and layout, of both electronic and nanophotonic devices to accurately execute various traffic models.

Photonic devices. Modeling of optical components is built on a detailed physical-layer library that has been validated through the physical measurement of fabricated devices. The modeled components are fabricated in silicon at the nano-scale, and include modulators, photodetectors, waveguides (straight, bending, crossing), filters, and PSEs. These devices are characterized and modeled at runtime by attributes such as insertion loss, crosstalk, delay, and power dissipation. Tables 2 and 3 show the most important optical parameters used.

Photonic network physical-layer analysis. The number of available wavelengths is obtained through an insertion loss analysis using PhoenixSim [2]. Fig. 8 shows the relationship between network insertion loss and the number of wavelengths that can be used. The following equations specify the constraints that must be met in order to achieve reliable optical communication:

$$P_{\text{tot}} < P_{NT} \tag{5}$$

$$P_{\text{inj}} - P_{\text{loss}} > P_{\text{det}}. \tag{6}$$

Eq. (5) states that the total injected power at the first modulator (P_{tot}) must be below the threshold at which nonlinear effects are induced (P_{NT}), which would corrupt the data (or introduce significantly more optical loss). A reasonable value for P_{NT} is around 10–15 dBm [20]. Eq. (6) states that the power received at the detectors (P_{det}) must be greater than the detector sensitivity

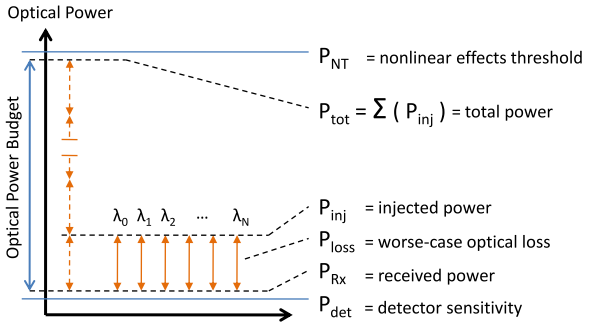


Fig. 8. Number of wavelengths dictated by insertion loss and optical power budget.

(usually about −20 dBm) to reliably distinguish between zeros and ones. To ensure this, every wavelength must inject at least enough power (P_{inj}) to overcome the worst-case optical loss through the network (P_{loss}). From these relationships, we can see that the number of wavelengths that can be used in a network can be limited by the worst-case insertion loss through it.

The three photonic networks that we consider have different insertion loss characteristics. We determine the worst-case P_{loss} for each network and find that it equates to 9.1, 10.1, and 6.3 dB for the P-mesh, P-TDM and P-ETDM, respectively. All networks can support a large number of wavelengths with a 35 dB optical power budget, though we limit this number at 128 because of modulator free spectral range (FSR) and inter-wavelength crosstalk limitations.

Simulation parameters. Each network uses 2.5 Gb/s signaling to reduce SerDes and driver power costs for an ideal link bandwidth of 320 Gb/s in and out of every gateway in each network. A bit-rate clock is sent with the data on a separate channel to lock on to the data at the receiver, and we allocate 16 clock cycles of overhead for each transmission for locking.

For power dissipation modeling, the ORION 2.0 electronic router model [15] is integrated into PhoenixSim, which provides detailed technology node-specific modeling of router components such as buffers, crossbars, arbiters, clock tree, and wires. The technology point is specified as 32 nm, and the V_{DD} and V_{th} ORION parameters are set according to frequency (lower voltage, higher threshold for lower frequencies). The ORION model also calculates the area of these components, which is used to determine the lengths of interconnecting wires for the P-mesh. The P-mesh uses a 1 GHz control plane with small (128-bit) buffers and narrow (32-bit) channels.

DRAM modeling. We employ the same DRAM subsystem modeling used in the previous work [13]. This model cycle accurately enforces all timing constraints of real DRAM chips, including row access time, row–column delay, column access latency, and precharge time. Because access to the memory modules is arbitrated by the on-chip path setup mechanism, only one transaction must be sustained by a MAP, which greatly simplifies the control logic as previously discussed. For the TDM networks, the gateway control logic handles memory transactions, scheduling them in empty time slots.

We base our model parameters around a Micron 1 Gb DDR3 chip [24], with (t_{RCD}–t_{RP}–t_{CL}) chosen as (12.5–12.5–12.5) (ns). To normalize the three different network architectures for experiment, we assign them the same amount of similarly configured DDR3 DRAM around the periphery.

7. Evaluation

7.1. Synthetic traffic

To test the network characteristics, we use PhoenixSim to run Uniform, Neighbor, Tornado, Bitreverse, and Hotspot random

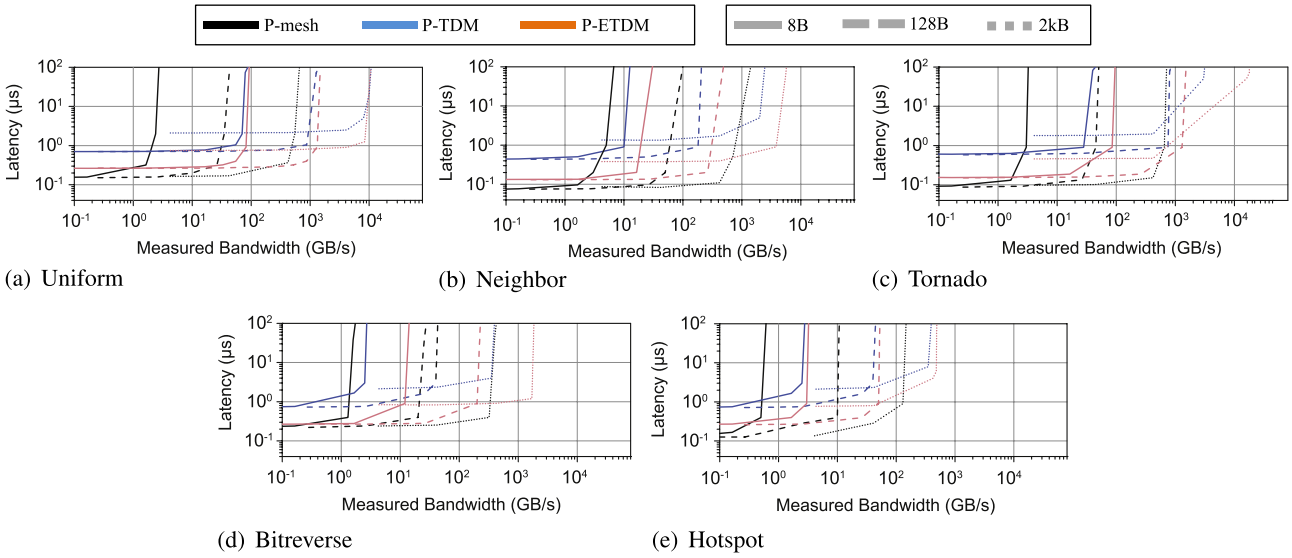


Fig. 9. Latency vs. bandwidth under synthetic traffic.

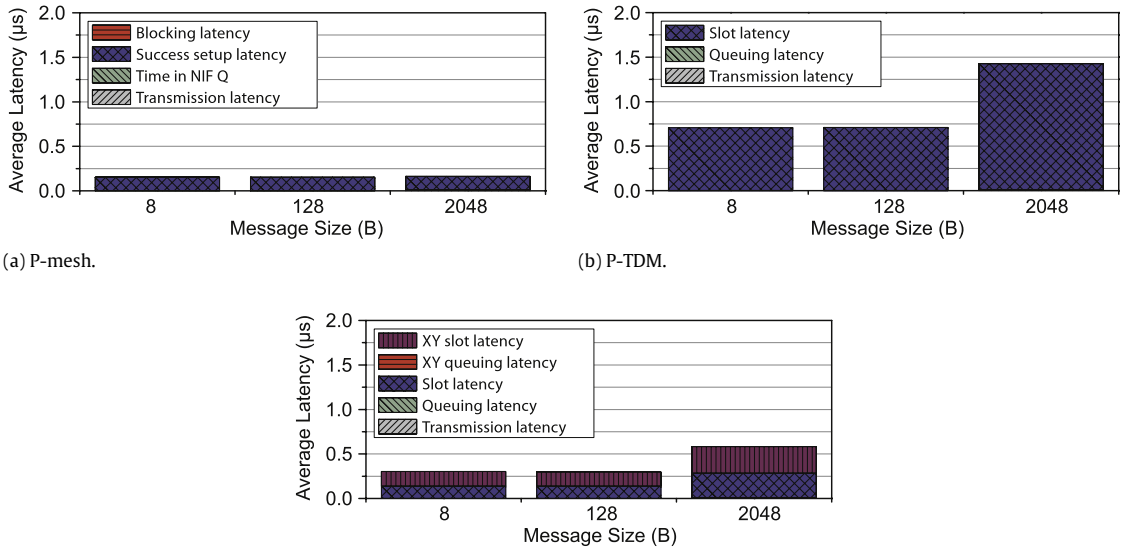


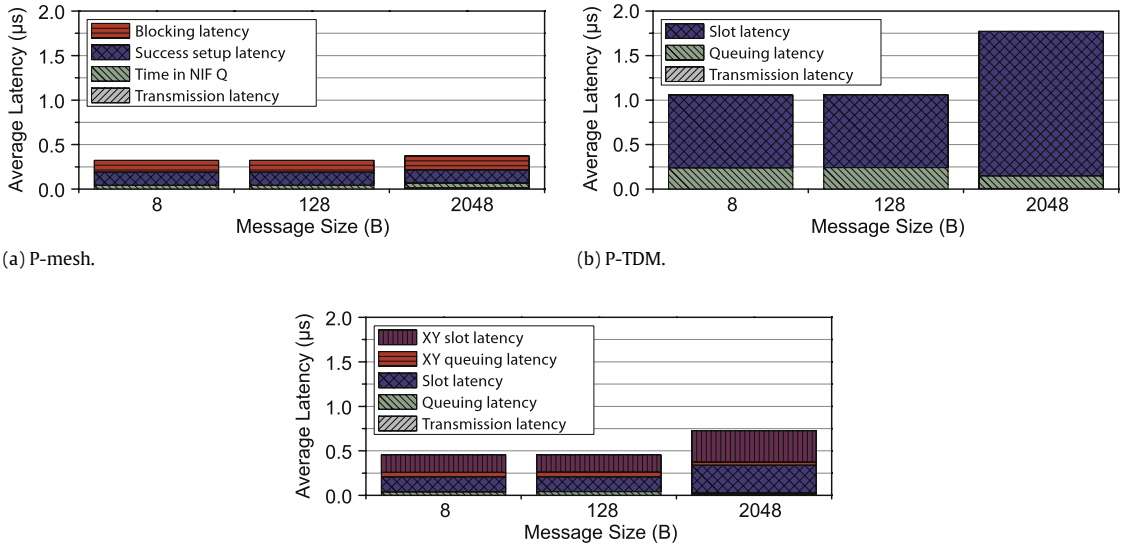
Fig. 10. Zero-load latency breakdown under Uniform traffic.

traffic in the network for 5 ms with 8, 128, and 2 kbyte messages, representing control, cache line, and application-level message sizes, respectively. We set t_{slot} at 10 ns, requiring 1 ns each for t_{setup} and $t_{propagation}$, making $S_{transmission}$ equal to 10 240 bits, or about 1.2 kB.

Fig. 9 shows the average read latency vs. total bandwidth in the network. The two TDM networks show higher zero-load latency than the P-mesh, as expected from the overhead of waiting for the correct slot. However, the enhanced TDM network shows significant zero-load latency improvement over the original TDM design. Both TDM networks also show higher throughput compared to the P-mesh for all message sizes, mostly due to their ability to service message requests that arrive at the gateway's controller out of order, thus increasing network utilization. Bandwidth gains are most profound in the traffic patterns with more chances of circuit-path blocking in the P-mesh, either from long communication (Uniform, Bitreverse) or predictably conflicting resources (Tornado).

Fig. 10 shows the sources of zero-load latency under Uniform traffic for each network as message size increases. The P-mesh is superior in this respect, as it is entirely dependent on the electronic router hop latency. The original TDM design's latency comes entirely from the slot latency, or when a message is next in line for a time slot, but is waiting for that slot. Again, our design improves the zero-load latency over the original TDM design by decreasing the time slot count, despite additional delay when changing dimensions (*XY*-buffer queuing and slot latency). The TDM networks also show a significant increase in latency for the larger 2 kB messages because the message must be sent in multiple slots. Though the slot period could have been changed to match the message size for the different simulations, we chose to keep a single slot period to illustrate the effects of its relationship to expected message size.

To illustrate the effects of contention on network latency, Fig. 11 shows the sources of latency while loaded at half capacity. For the P-mesh, blocking latency enters the picture, forcing queuing at the network gateways. The original TDM design is still dominated by slot latency, where queuing latency is dictated by the traffic pattern. The E-TDM method has a similar relationship, though much less severe because of the reduced slot count. An extra traffic-dependent queuing latency is introduced at the *XY*-buffer, though it is small compared to the total.



(a) P-mesh.

(b) P-TDM.

(c) P-ETDM.

Fig. 11. Half-load latency breakdown under Uniform traffic.

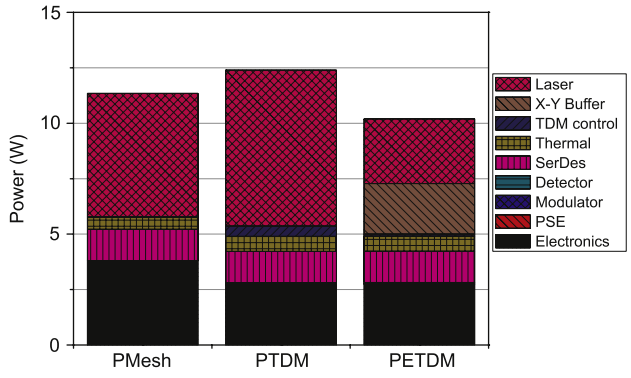


Fig. 12. Power breakdown.

Fig. 12 shows a coarse network power breakdown under Uniform traffic near saturation, assuming around 12% integrated laser efficiency [30]. Electronic power is still a large part of all the networks, mainly in the electronic crossbar necessary to implement external concentration, which must match the bandwidth of the photonic links using many parallel wires. The TDM control circuitry contributes minimal power overhead to the two TDM networks. An advantage of the E-TDM network is that is has less insertion loss, and therefore requires less laser power. Instead of laser power, the P-ETDM consumes power in the *XY*-buffer (∼2 W) which is necessary to implement dimension-only transmission. Regardless, the P-ETDM consumes the lowest total power.

7.2. Case study: embedded application

We evaluate the proposed network architectures using the application modeling framework, *Mapping and Optimization Runtime Environment* (MORE) to collect traces from the execution of high-performance embedded signal and image processing applications.

The MORE system, based on pMapper [34], is designed to project a user program written in Matlab onto a distributed or parallel architecture and provide performance results and analysis. The MORE framework translates application code into a *dependency based instruction trace*, which captures the individual operations performed as well as their interdependences. By creating an instruction trace interface for PhoenixSim, we were able to accurately model the execution of applications on the proposed architectures.

MORE consists of the following primary components:

- The *program analysis* component is responsible for converting the user program, taken as input, into a *parse graph*, a description of the high-level operations and their dependences on one another.
- The *data mapping* component is responsible for distributing the data of each variable specified in the user code across the processors in the architecture.
- The *operations analysis* component is responsible for taking the parse graph and data maps and forming the *dependency graph*, a description of the low-level operations and their dependences on one another.

PhoenixSim then reads the dependency graphs produced by MORE, generating computation and communication events. Combining PhoenixSim with MORE in this way allows us to characterize photonic networks on the physical level by generating traffic which exactly describes the communication, memory access, and computation of the given application.

7.2.1. Projective transform

When registering multiple images taken from various aerial surveillance platforms, it is frequently advantageous to change the perspective of these images so that they are all registered from a common angle and orientation (typically straight down with North being at the top of the image). In order to do this, a process known as *projective transform* is used [16].

Projective transform takes as input a two-dimensional image M as well as a transformation matrix t that expresses the transformational component between the angle and orientation of the image presented and the desired image. The projective transform algorithm outputs M', or the image M after projection through t. To populate a pixel p' in M', its x and y positions are back-projected through t to get their relative position in M, p. This position likely does not fall directly on a pixel in M, but rather somewhere between a set of four pixels. Using the distance from p to each of its corners as well as the corner values themselves, the value for p' can be obtained.

We consider this application on an image size of 256 × 256 pixels. We simulate a simple case, where the image orientation is rotated by ninety degrees. While the result of this transform is simply a corner turn on the matrix representing the image, it allows for identical image and projection sizes while still inducing data movement in the projection process. Also, with the use

Table 4
Performance and power results for projective transformation.

Network	Network power (W)	Performance (GOPS)	Efficiency (GOPS/W)
P-mesh	9.92	7.55	1×
P-TDM	10.95	3.7	0.44×
P-ETDM	8.84	15.7	2.35×

of MORE, analyzing different projections is simple. It requires only changing the transformation matrix in the source code and rerunning the simulations.

7.2.2. Simulation results

During the simulations, we collected average network power and system performance. These results are reported in Table 4. The P-ETDM is the superior solution, outperforming the P-mesh by around 2× while using slightly less power. Though the original TDM network can sustain higher total network bandwidth, it is inferior to the P-mesh for this particular application, directly illustrating the usefulness of the design improvement proposed in this paper.

8. Conclusions

TDM arbitration of photonic circuits proves to be an effective way to increase network utilization, which increases performance and energy efficiency for both random traffic and real applications. Key characteristics of the architecture are the ability to bypass head-of-line blocking at the gateways, and very low insertion loss due to single dimension transmission. In this paper, we presented an improvement over previous methods to decrease the number of time slots needed to implement full network coverage in one frame, which reduces zero-load latency and improves throughput, having a significant impact on the performance of a key embedded application kernel in real-time image processing, the projective transform.

Acknowledgments

This work is sponsored in part by Defense Advanced Research Projects Agency (DARPA) under Air Force contract FA8721-05-C-0002, DARPA MTO under grant ARL-W911NF-08-1-0127, the NSF (Award #: 0811012), and the FCRP Interconnect Focus Center (IFC). Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

References

[1] C. Batten, A. Joshi, J. Orcutt, A. Khilo, B. Moss, C. Holzwarth, M. Popovic, H. Li, H. Smith, J. Hoyt, F. Kartner, R. Ram, V. Stojanovic, K. Asanovic, Building manycore processor-to-DRAM networks with monolithic silicon photonics, in: HOTI'08: Proceedings of the 2008 16th IEEE Symposium on High Performance Interconnects, IEEE Computer Society, Washington, DC, USA, 2008, pp. 21–30.
[2] J. Chan, G. Hendry, A. Biberman, K. Bergman, L.P. Carloni, PhoenixSim: a simulator for physical-layer analysis of chip-scale photonic interconnection networks, in: DATE: Design, Automation, and Test in Europe, 2010.
[3] L. Chen, K. Preston, S. Manipatruni, M. Lipson, Integrated GHz silicon photonic interconnect with micrometer-scale modulators and detectors, Optics Express 17 (17) (2009).
[4] L. Chen, N. Sherwood-Droz, M. Lipson, Compact bandwidth-tunable microring resonators, Optics Letters 32 (22) (2007) 3361–3363.
[5] M.J. Cianchetti, J.C. Kerekes, D.H. Albonesi, Phastlane: a rapid transit optical routing network, SIGARCH Computer Architecture News 37 (3) (2009) 441–450.
[6] Corning Inc., Datasheet: Corning SMF-28e Optical Fiber Product Information. URL: http://www.princetel.com/datasheets/SMF28e.pdf (accessed 2010).
[7] A. Edman, C. Svensson, Timing closure through a globally synchronous, timing partitioned design methodology, in: DAC'04: Proceedings of the 41st Annual Design Automation Conference, ACM, New York, NY, USA, 2004, pp. 71–74.
[8] T. Fukazawa, T. Hirano, F. Ohno, T. Baba, Low loss intersection of Si photonic wire waveguides, Japanese Journal of Applied Physics 43 (2) (2004) 646–647.
[9] K. Goossens, J. Dielissen, A. Radulescu, Æthereal network on chip: concepts, architectures, and implementations, IEEE Design & Test 22 (5) (2005) 414–421.
[10] B. Guha, B.B.C. Kyotoku, M. Lipson, CMOS-compatible athermal silicon microring resonators, Optics Express 18 (4) (2010).
[11] G. Hendry, J. Chan, S. Kamil, L. Oliker, J. Shalf, L.P. Carloni, K. Bergman, Silicon nanophotonic network-on-chip using TDM arbitration, in: Proceedings of IEEE Symposium on High-Performance Interconnects, 2010.
[12] G. Hendry, S. Kamil, A. Biberman, J. Chan, B.G. Lee, M. Mohiyuddin, A. Jain, K. Bergman, L.P. Carloni, J. Kubiatowicz, L. Oliker, J. Shalf, Analysis of photonic networks for a chip multiprocessor using scientific applications, in: NOCS'09: Proceedings of the 2009 3rd ACM/IEEE International Symposium on Networks-on-Chip, IEEE Computer Society, Washington, DC, USA, 2009, pp. 104–113.
[13] G. Hendry, E. Robinson, V. Gleyzer, J. Chan, L.P. Carloni, N. Bliss, K. Bergman, Circuit-switched memory access in photonic interconnection networks for high-performance embedded computing, in: Proceedings of Supercomputing, 2010.
[14] A. Joshi, C. Batten, Y.-J. Kwon, S. Beamer, I. Shamim, K. Asanovic, V. Stojanovic, Silicon-photonic clos networks for global on-chip communication, in: NOCS'09: Proceedings of the 2009 3rd ACM/IEEE International Symposium on Networks-on-Chip, IEEE Computer Society, Washington, DC, USA, 2009, pp. 124–133.
[15] A.B. Kahng, B. Li, L.-S. Peh, K. Samadi, ORION 2.0: a fast and accurate NoC power and area model for early-stage design space exploration, 2009, pp. 423–428.
[16] H. Kim, E. Rutledge, S. Sacco, S. Mohindra, M. Marzilli, J. Kepner, R. Haney, J. Daly, N. Bliss, PVTOL: providing productivity, performance and portability to DoD signal processing applications on multicore processors, in: HPCMP-UGC'08: Proceedings of the 2008 DoD HPCMP Users Group Conference, IEEE Computer Society, Washington, DC, USA, ISBN: 978-0-7695-3515-9, 2008, pp. 327–333.
[17] N. Kirman, et al., Leveraging optical technology in future bus-based chip multiprocessors, in: MICRO 39: Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture, IEEE Computer Society, Washington, DC, USA, 2006, pp. 492–503.
[18] P. Kumar, Y. Pan, J. Kim, G. Memik, A. Choudhary, Exploring concentration and channel slicing in on-chip network router, in: NOCS'09: Proceedings of the 2009 3rd ACM/IEEE International Symposium on Networks-on-Chip, IEEE Computer Society, Washington, DC, USA, ISBN: 978-1-4244-4142-6, 2009, pp. 276–285.
[19] B.G. Lee, A. Biberman, P. Dong, M. Lipson, K. Bergman, All-optical comb switch for multiwavelength message routing in silicon photonic networks, IEEE Photonics Technology Letters 20 (10) (2008) 767–769.
[20] B.G. Lee, X. Chen, A. Biberman, X. Liu, I.-W. Hsieh, C.-Y. Chou, J. Dadap, R.M. Osgood, K. Bergman, Ultra-high-bandwidth WDM signal integrity in silicon-on-insulator nanowire waveguides, IEEE Photonics Technology Letters 20 (6) (2007) 398–400.
[21] H.L.R. Lira, S. Manipatruni, M. Lipson, Broadband hitless silicon electro-optic switch for on-chip optical networks, Optics Express 17 (25) (2009).
[22] Z. Lu, A. Jantsch, TDM virtual-circuit configuration for network-on-chip, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 16 (8) (2008) 1021–1034.
[23] A. Melloni, F. Morichetti, R. Costa, G.C. an dP Boffi, M. Martinelli, The ring-based optical resonant router, in: IEEE ICC, 2006, pp. 2799–2804.
[24] Micron Technology Inc., Product Specification. 1 Gb DDR3 SDRAM Chip, 2010. URL: http://www.micron.com/products/partdetail?part=MT41J256M4JP-125.
[25] M. Millberg, E. Nilsson, R. Thid, A. Jantsch, Guaranteed bandwidth using looped containers in temporally disjoint networks within the nostrum network on chip, in: DATE'04: Proceedings of the Conference on Design, Automation and Test in Europe, IEEE Computer Society, Washington, DC, USA, 2004, p. 20890.
[26] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, A. Choudhary, Firefly: illuminating future network-on-chip with nanophotonics, SIGARCH Computer Architecture News 37 (3) (2009) 429–440.
[27] C. Paukovits, H. Kopetz, Concepts of switching in the time-triggered network-on-chip, in: RTCSA'08: Proceedings of the 2008 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, IEEE Computer Society, Washington, DC, USA, 2008, pp. 120–129.
[28] M. Petracca, B.G. Lee, K. Bergman, L.P. Carloni, Design exploration of optical interconnection networks for chip multiprocessors, in: HOTI'08: Proceedings of the 2008 16th IEEE Symposium on High Performance Interconnects, IEEE Computer Society, Washington, DC, USA, 2008, pp. 31–40.
[29] Rambus, RDRAM Memory Technology. Online at: http://www.rambus.com/us/products/rdram/index.html (accessed 2010).
[30] G. Roelkens, D.V. Thourhout, R. Baets, Continuous-wave lasing from DVS-BCB heterogeneously integrated laser diodes, in: Integrated Photonics and Nanophotonics Research and Applications, Optical Society of America, 2007.
[31] M. Schoeberl, A time-triggered network-on-chip, in: International Conference on Field-Programmable Logic and its Applications, FPL 2007, Amsterdam, Netherlands, 2007, pp. 377–382.

[32] J. Schrauwen, F.V. Laere, D.V. Thourhout, R. Baets, Focused-ion-beam fabrication of slanted grating couplers in silicon-on-insulator waveguides, IEEE Photonics Technology Letters 19 (11) (2007) 816–818.
[33] A. Shacham, K. Bergman, L.P. Carloni, Photonic networks-on-chip for future generations of chip multiprocessors, IEEE Transactions on Computers 57 (9) (2008) 1246–1260.
[34] N. Travinin, H. Hoffmann, R. Bond, H. Chan, J. Kepner, E. Wong, PMapper: automatic mapping of parallel matlab programs, in: DOD_UGC'05: Proceedings of the 2005 Users Group Conference on 2005 Users Group Conference, IEEE Computer Society, Washington, DC, USA, ISBN: 0-7695-2496-6, 2005, p. 254.
[35] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N.P. Jouppi, M. Fiorentino, A. Davis, N. Binkert, R.G. Beausoleil, J.H. Ahn, Corona: system implications of emerging nanophotonic technology, in: Computer Architecture, International Symposium on, vol. 0, 2008, pp. 153–164.
[36] M.R. Watts, Ultralow power silicon microdisk modulators and switches, in: 5th Annual Conference on Group IV Photonics, 2008, pp. 4–6.
[37] F. Xia, L. Sekaric, Y. Vlasov, Ultracompact optical buffers on a silicon chip, Nature Photonics 1 (2007) 65–71.
[38] Q. Xu, B. Schmidt, J. Shakya, M. Lipson, Cascaded silicon micro-ring modulators for WDM optical interconnection, Optics Express 14 (20) (2006).
[39] J.-F. Zheng, et al., On-chip optical clock signal distribution, in: OSA Topical Meeting on Optics in Computing, 2003.

Gilbert Hendry received the B.S. and M.S. degrees in computer engineering from the Rochester Institute of Technology, Rochester, NY, in 2007. He is currently a Ph.D. candidate in the Department of Electrical Engineering at Columbia University, New York, NY. His research interests include the design of computing systems using silicon photonics, and the software tools used in this endeavor.

Eric Robinson graduated from Northeastern University with a Ph.D. in computer science. His doctoral work as part of the high-performance and distributed computing group focused on parallel disk based enumeration of terascale state spaces. He is now employed by MIT Lincoln Lab where he focuses on graph exploitation in real world graphs. He is working on developing effective high-performance algorithms and architectures for mining valuable intelligence from graphs generated from real world surveillance data. In addition, he continues to pursue interests in computational group theory, disk based computation, parallel computation, and software–hardware co-design.

Vitaliy Gleyzer has been a staff member in the Embedded Digital Systems group at MIT Lincoln Laboratory for two years. Prior to joining MIT Lincoln Laboratory, he received his Masters in Electrical and Computer Engineering from Carnegie Mellon University, with the research concentrated on network architecture and network modeling. His current work is primarily focused on high-performance computing systems and embedded systems engineering.

Johnnie Chan received the B.S. degree (with high distinction) in computer and electrical engineering and M.S. degree in electrical engineering from the University of Virginia, Charlottesville, in 2005 and 2007, respectively. He is currently a Ph.D. candidate in the Department of Electrical Engineering at Columbia University, New York, NY. His research interests include the design of photonic networks-on-chip in chip multiprocessor systems, and the modeling of the nanophotonic devices used to enable on- and off-chip communications.

Luca P. Carloni received the Laurea degree (summa cum laude) in electrical engineering from the Università di Bologna, Italy, in 1995, and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1997 and 2004, respectively.

He is currently an Associate Professor with the Department of Computer Science, Columbia University, New York, NY. He has authored over 70 publications and is the holder of one patent. His research interests are in the area of design tools and methodologies for integrated circuits and systems, distributed embedded systems design, and design of high-performance computer systems.

Dr. Carloni received the Faculty Early Career Development (CAREER) Award from the National Science Foundation in 2006 and was selected as an Alfred P. Sloan Research Fellow in 2008, and received the ONR Young Investigator Award in 2010. He is the recipient of the 2002 Demetri Angelakos Memorial Achievement Award "in recognition of altruistic attitude towards fellow graduate students". In 2002, one of his papers was selected for "The Best of ICCAD", a collection of the best IEEE International Conference on Computer-Aided Design papers of the past 20 years. He is a senior member of the ACM and IEEE.

Nadya Bliss is the Assistant Leader of the Embedded and High Performance Computing Group at MIT Lincoln Laboratory. She earned her bachelor and master degrees in Computer Science from Cornell University, where her research focus was on developing Bayesian techniques for natural language processing. After completing her degrees, she joined the Laboratory in 2002 and, as a member of technical staff, was one of the principal innovators and developers of the pMatlab: Parallel Matlab Toolbox and the pMapper automated parallelization system.

Her technical interests are in parallel and distributed computing, specifically program analysis and optimization, scalable intelligent/cognitive algorithms, representations for multi-INT data, and software/hardware co-design methodologies.

Keren Bergman is a Professor of Electrical Engineering at Columbia University where she also directs the Lightwave Research Laboratory (http://lightwave.ee.columbia.edu/). She leads multiple research programs on optical interconnection networks for advanced computing systems, data centers, optical packet-switched routers, and chip multiprocessor nanophotonic networks-on-chip. Dr. Bergman holds a Ph.D. from MIT and is a Fellow of the IEEE and of the OSA. She currently serves as the co-Editor-in-Chief of the IEEE/OSA Journal of Optical Communications and Networking.