Circuit-Switched Memory Access in Photonic Interconnection Networks for High-Performance Embedded Computing

Gilbert Hendry*, Eric Robinson†, Vitaliy Gleyzer†, Johnnie Chan*,
Luca P. Carloni‡, Nadya Bliss† and Keren Bergman*

*Lightwave Research Lab, Department of Electrical Engineering, Columbia University, New York, NY
†Lincoln Laboratories, MIT, Lexington, MA
‡Computer Science Department, Columbia University, New York, NY

***Abstract*— As advancements in CMOS technology trend toward ever increasing core counts in chip multiprocessors for high-performance embedded computing, the discrepancy between on- and off-chip communication bandwidth continues to widen due to the power and spatial constraints of electronic off-chip signaling. Silicon photonics-based communication offers many advantages over electronics for network-on-chip design, namely power consumption that is effectively agnostic to distance traveled at the chip- and board-scale, even across chip boundaries. In this work we develop a design for a photonic network-on-chip with integrated DRAM I/O interfaces and compare its performance to similar electronic solutions using a detailed network-on-chip simulation. When used in a circuit-switched network, silicon nanophotonic switches offer higher bandwidth density and low power transmission, adding up to over 10× better performance and 3-5× lower power over the baseline for projective transform, matrix multiply, and Fast Fourier Transform (FFT), all key algorithms in embedded real-time signal and image processing.**

I. Introduction

Many important classes of applications including personal mobile devices, image processing, avionics, and defense applications such as aerial surveillance require the design of high-performance embedded systems. These systems are characterized by a combination of real-time performance requirements, the need for fast streaming access to memory, and very stringent energy constraints [12], [46], [50]. While commodity general purpose processors offer a cheap and customizable solution, they typically do not meet the power and performance requirements for the systems in question. For this reason, specialized chip multiprocessors (CMPs) are used.

As the number of cores in CMPs scale to provide greater on-chip computational power, communication becomes an increasing contributor to power and performance. The gap between the available off-chip bandwidth and that which is required to appropriately feed the processors continues to widen under current memory access architectures. For many high-performance embedded computing applications, the bandwidth available for both on- and off-chip communications can play a vital role in efficient execution due to the use of data-parallel or data-centric algorithms.

Unfortunately, current electronic memory access architectures have the following characteristics that will impede performance scaling and energy efficiency for applications that require large memory bandwidths:

- **Distance-Dependant.** Electronic I/O wires must often be path-length matched to reduce clock skew. In addition, there are limitations on the length of these wires which constrains board layout and scalability [19].
- **Low I/O Density.** Electronic I/O wires are predicted to have pitches on the order of around 80 microns [18]. Increasing the available off-chip communication bandwidth will become difficult while staying within manageable pin counts.
- **Low I/O Frequencies.** Driving long I/O wires requires lower frequencies, currently up to 1600 MT/s with the most recent DDR3 implementation [32].

Recent advances in silicon nanophotonic devices and integration have made it possible to consider optical transmission on the chip- and board-scale [7], [28]. Microprocessor I/O signaling can directly benefit from photonics in the following ways:

- **Distance-Independent.** Optical transmission of data can be made agnostic to distance at the chip- and board-scale; photonic energy dissipation is effectively not a function of distance.
- **Data-rate Transparent.** Most photonic devices, including switches as well as on- and off-chip waveguides are not bitrate-dependent, providing a natural bandwidth match between compute cores and the memory subsystem.
- **High Bandwidth Density.** Waveguides crossing the chip boundary can have a similar pitch to that of electronics [41], which makes the bandwidth density of nanophotonics using wavelength division multiplexing (WDM) orders of magnitude higher than electronic wires.

Though photonics can offer significant physical-layer advantages, constructing a memory access architecture to realize them requires significant design space exploration. Trade-offs exist in the selection of specific components, architectures,

§This work is sponsored 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.

SC10 November 2010, New Orleans, Louisiana, USA 978-1- 4244-7558-2/10/$26.00

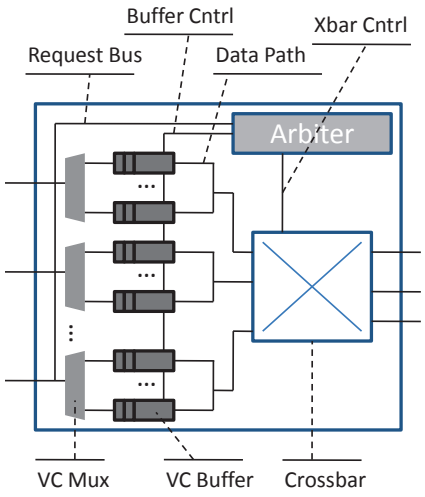


Fig. 1. **Packet-switching router.**

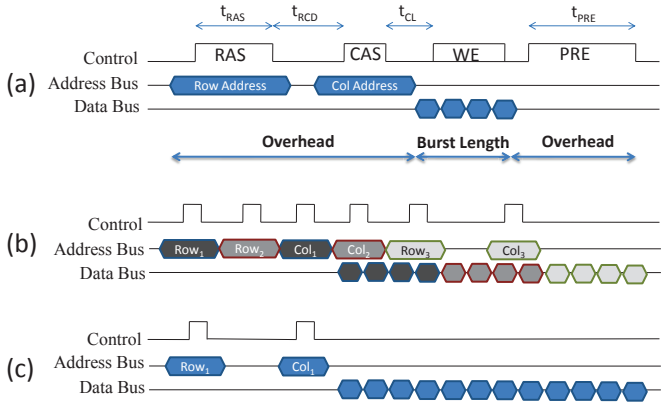


Fig. 2. **Control of a DRAM module (a) a single transaction (b) amortizing overhead by pipelining transactions (c) amortizing overhead through increased burst length.**

and protocols. Our approach to this problem employs a single circuit-switched photonic network-on-chip (NoC) design, enabling both core-to-core and core-to-DRAM communication which are necessary for efficiently implementing programming models such as PGAS [4].

In this work, we study the problem of designing a NoC architecture for an embedded computing platform that supports both on-chip communication and off-chip memory access in a power-efficient way. In particular, we propose the adoption of circuit-switched NoC architectures that rely on a simple mechanism to switch circuit paths off-chip to exchange data with the DRAM memory modules. While this method is presented independently of the particular transmission technology, we show the advantages offered by an implementation based on photonic communication over an electronic one.

We simulate this memory access architecture on a 256-core chip with a concentrated 64-node network using detailed traces of computation kernels widely used in signal and image processing high-performance embedded applications, specifically the projective transformation, matrix multiply, and Fast Fourier Transform (FFT). This work accomplishes the first complete detailed simulation of a nanophotonic NoC with physically-accurate photonic device models coupled with cycle-accurate DRAM device and control models. These simulations are used to determine the benefits of circuit-switching and silicon photonic technology in CMP memory access performance.

II. Packet-Switched Memory Access

Packet-switched NoCs use router buffers to store and forward small *packets* through the network, where a packet is a small number of flits (flow control units). Typically, purely electronic store-and-forward routers use multiple physical buffers to implement virtual channels, alleviating head-of-line blocking under congestion. An illustration of a pipelined router can be seen in Figure 1. If a core-to-DRAM or core-to-core application-level *message* is larger than the physical buffers themselves, or larger than the flow control mechanism can reasonably sustain without deadlock, these messages must be broken into several smaller packets.

The structure of a packet-switched NoC has important implications on how memory accesses are performed. Typically, multiple on-chip memory controllers distributed around the periphery of a CMP service requests from all the cores. If a memory controller receives packets from different cores (different messages), it must then schedule memory transactions with potentially disparate addresses. Indeed, the memory controller depends on this paradigm to optimize the utilization of the data and control buses using rank and bank concurrency.

Figure 2(a) shows the basic protocol of a single memory transaction. The row address is latched into the DRAM chip with the row address select (RAS) signal for the row access time (t_{RAS}) until the decoded row is driven into the sense amps. After the row-column delay time (t_{RCD}), the column address then selects the starting point in the array, using the column address select (CAS) signal. A write enable (WE) signal determines whether the I/O circuitry is accepting data from the bus or pushing data onto it. Data is then read or written after the column-access latency (t_{CL}), incrementing the initial column address in a burst. Once the transaction is complete, depending on the control policy, the row can be closed and must be precharged (PRE) for a time t_{PRE}.

Figure 2(b) shows how a contemporary DRAM memory controller schedules transactions concurrently across banks, chips, and ranks to maximize performance and hide the access latency. There exist different control policies to manage queued transactions for lower latency and higher throughput, both dynamic in the memory controller (*e.g.* page mode), and static at compile-time [29]. The burst length is usually fixed in this configuration, matching the on-chip cache-line size. Allowing a variable burst length would introduce significant complexity to the scheduling mechanism.

Typical DRAM subsystems implemented this way have been effective for providing short latencies for small, random accesses, as required by contemporary cache miss access patterns. However, providing the increasing bandwidth required by future embedded applications will come at the cost of power consumption in the on-chip interconnect, due partially

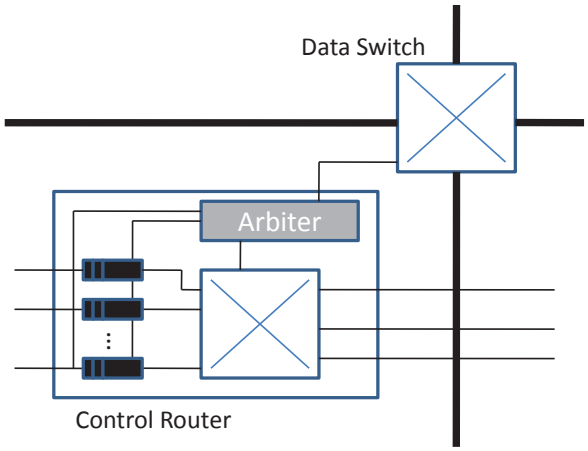


Fig. 3. **Typical circuit-switching router.**

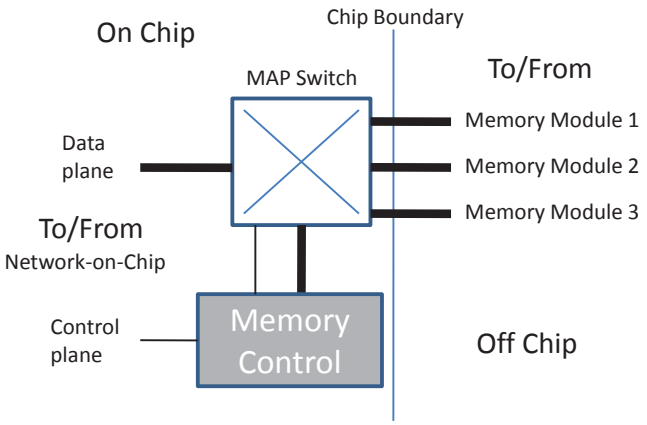


Fig. 4. **Circuit-Switched Memory Access Point.**

to the relationship of the amount of network buffering to performance.

III. Memory Access for Embedded Computing

Embedded processors are devices typically found in mobile or extreme environments, and their design is commonly driven by the needs of the application in question. They frequently require specialized hardware or software, or commonly, both to efficiently meet their performance, power, and reliablity requirements. Because of this, a hardware / software co-design approach is generally taken [31].

Of key consideration to this work are embedded applications that involve signal and image processing (SIP). These applications typically require the aggregation and processing of many data points collected from various locations over a period of time, originating from sensors or other continuous data streams. A typical example of this is a camera or other sensor placed on an unmanned air vehicle (UAV). Applications in this domain require signifcant computing power in the form of high bandwidth data access and streaming processing capabilities. In addition, they must achieve this using a low power budget.

In these applications, data is typically placed in contiguous blocks of an embedded computing system's memory space around a central CMP via direct memory access (DMA) or a similar mechanism by incoming data streams. The memory access system outlined in this section proposes to make use of the fact that these contiguous blocks of data can be accessed using long burst lengths. The application can exhibit very dynamic communication patterns between individual cores and banks of memory, all while making use of efficient memory access circuits.

A. Circuit-Switched Memory Access

In a circuit-switched network, a control network provides a mechanism for setting up and tearing down energy-efficient high-bandwidth end-to-end circuit paths. If a network node wishes to send data to another node, a PATH-SETUP message is sent to reserve the necessary network resources to allocate the path. A PATH-BLOCKED message is returned to the node if some parts of the path is currently reserved by another circuit. A PATH-ACK message is returned if the path successfully made it to the end node. After data is transmitted along the data plane, a PATH-TEARDOWN message is sent from the sending node to release network resources for other paths.

This method effectively relaxes the relationship between router buffer size, a large contributor ($> 30\%$) to NoC power [21], and performance because router buffers do not become directly congested as communication demands grow. Figure 3 shows the router architecture for a circuit-switched NoC. The control network uses smaller buffers and channels to transmit the small control messages, which reduces the total amount of buffering (and thus power) in the network. Because the higher-bandwidth data plane is circuit switched end-to-end, it suffers from higher latency due to the circuit-path setup overhead, which must be amortized through a combination of larger messages and well-scheduled or time-division multiplexed communication patterns.

Aside from the power savings advantage, we can also decrease considerably the complexity of the memory controller through circuit-switching. We propose to allow a circuit-switched on-chip network to directly access memory modules, giving a single core exclusive access to a memory module for the duration of the transaction it requested. Access overhead is amortized using increased burst lengths as shown in Figure 2(c). The memory controller complexity can be greatly reduced because a memory module must sustain only one transaction at a time. The key difference is that each transaction is an entire message using long burst lengths, as opposed to small packets that must be properly scheduled. In addition, variable burst lengths are inherently supported without introducing additional complexity.

To facilitate switching on-chip circuit paths off chip to memory modules, we place memory access points (MAPs) around the periphery of the chip connected to the network. These MAPs, shown in Figure 4, contain a memory controller that can service memory transactions and use the NoC to allow end-to-end communication between cores and DRAM modules. Figure 5 shows the logic behind this control.

Read transactions are first sent as small control messages to the memory controller. If another transaction is currently

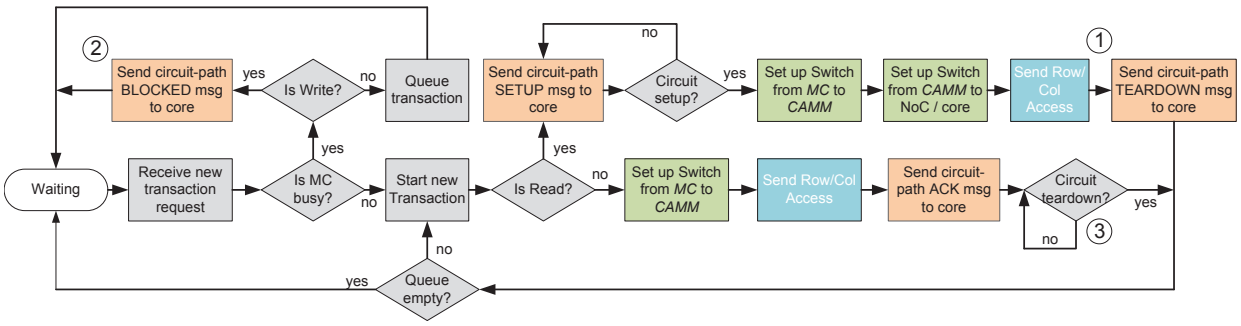


Fig. 5. **Flowchart of memory control logic.**

in progress at the MAP, this request is then queued up. Once a read is started, it first sets up the data switch for communication from the memory controller to the memory module (for DRAM commands) and from memory module back to the core (for returning read data). A circuit-path is then established back to the core via the NoC path-setup mechanism. The memory controller can then issue row and column access commands, allowing the memory module to freely send data back to the core. The memory controller is responsible for knowing the access time of the read, so that it can issue a PATH-TEARDOWN at the correct time (labeled 1 in Figure 5), which completes the transaction.

Writes begin by a core setting up a circuit-path to a MAP. By virtue of a PATH-SETUP message successfully arriving to the MAP, the core will have gained exclusive access to it. Writes that arrive to a MAP that is servicing a read return to the core as a blocked path (labeled 2) instead of queuing it, to release network resources for other transactions (including the potential read setup that is attempting). The memory controller then sets up the data switch from memory controller to memory, which allows the transmission of DRAM row/col access commands. The data switch is then set from core to memory module, and a PATH-ACK is sent back to the core, completing the path setup. Upon receiving the path acknowledgment, the core then begins transmitting write data directly to the memory module. The memory controller considers the transaction finished when it receives a PATH-TEARDOWN from the core (labeled 3). In this way, any core in the network can establish a direct, end-to-end circuit path with any memory module.

Livelock is avoided by using random backoff for path-setup requests. However, starvation for a core is possible, especially for writes in the presence of many reads. We leave the impact and effectiveness of the memory access mechanism on power and performance to Section V. Addressing memory access starvation through both network design and software/programming models remains a topic for future work.

B. Silicon Nanophotonic Technology

Circuit-switching photonic networks can be achieved using active broad-band ring-resonators whose diameter is manufactured such that its resonant modes directly align with all of the wavelengths injected into the nearby waveguide. For example,

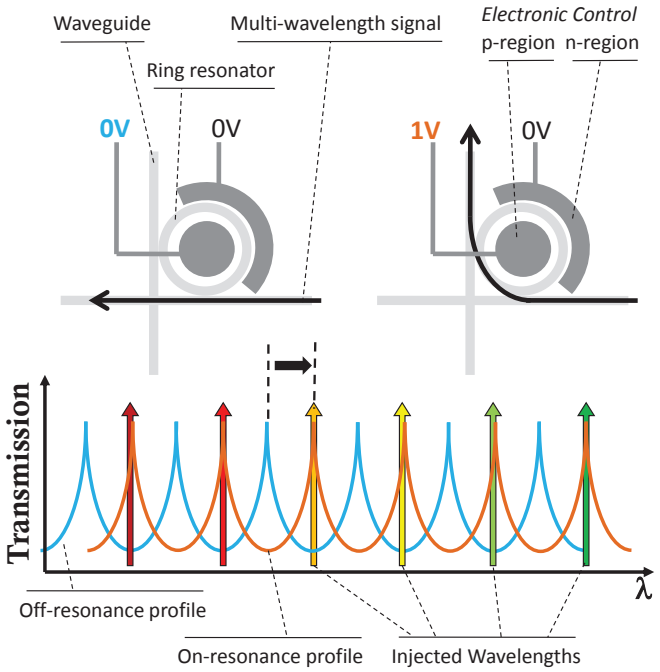


Fig. 6. **Operation of PSE. *Left* - PSE in off state. *Right* - PSE in on state. *Bottom* - Resonance profile of ring resonator, shifts from off to on.**

a $200um$ will have a wavelength channel spacing of 50 GHz. The ring resonator can be configured to be used as a photonic switching element (PSE), as shown in Figure 6. By electrically injecting carriers into the ring, the entire resonant profile is shifted, effectively creating a spatial switch between the ports of the device [27]. This process is analogous to setting the control signals of an electronic crossbar.

Given the operation of a single PSE, we can then construct higher order switches, and ultimately entire networks. Using ring-resonator devices in this way opens the possibility to explore different network topologies in much the same way as packet-switched electronic networks [36]. Different numbers and configurations of ring switches yield different amounts of energy, different path-blocking characteristics, as well as varying insertion loss.

We assume off-chip photonic signaling is achieved through lateral coupling [1] [30], where the optically encoded data is brought in and out of the chip through inverse-taper optical

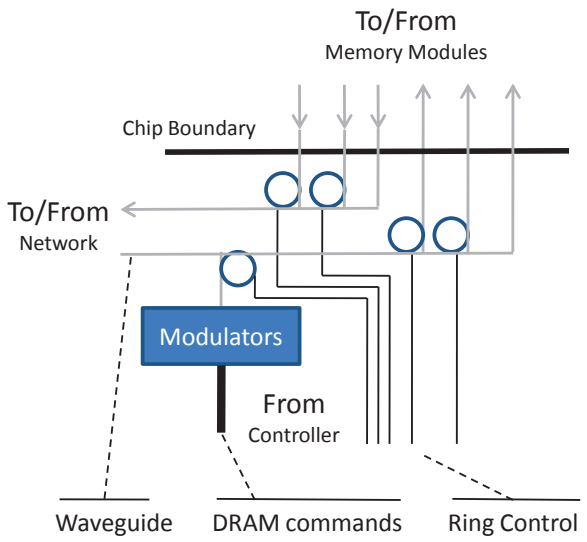


Fig. 7. **Photonic switch used in a MAP.**

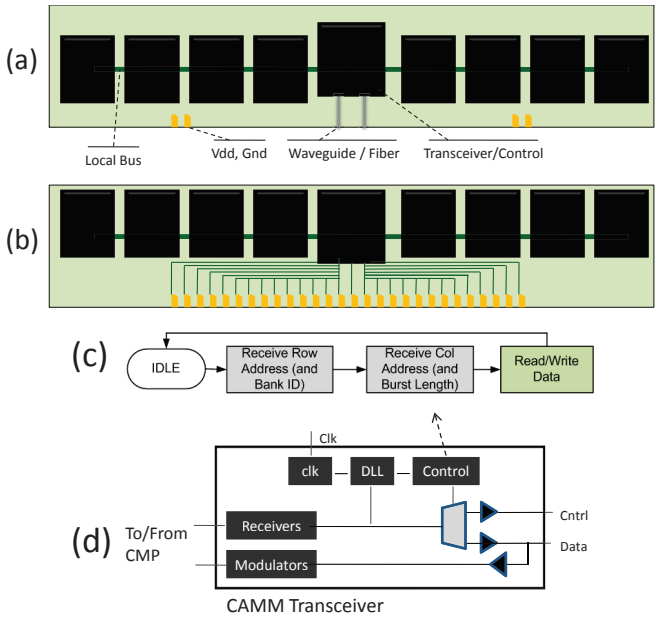


Fig. 8. **Circuit-Accessed Memory Module design (a) Photonic CAMM (b) Electronic CAMM (c) CAMM control logic (d) CAMM Transceiver.**

mode converters which expand the on-chip optical cross section to match the cross section of the external guiding medium. This method is employed due to its lower insertion loss, compared to vertical coupling [39] [14]. Waveguide pitch at the chip edge can easily be on the order of 60 μm interfacing to off-chip arrayed waveguides [41] or optical fiber. This photonic I/O pitch remains well below that of current electrical I/O pitch (*e.g.* 190 μm in the Sun ULTRASparc T2 [43]), illustrating the potential for vastly higher bandwidth density that is offered by using photonic waveguides when using WDM.

As shown in Figure 4, the MAP controls a switch that establishes circuit paths between individual memory modules and the network. The photonic version of this switch is illustrated in Figure 7, which uses broadband ring-resonators to allow access to multiple memory modules controlled by the same memory controller. Modulators convert electronic DRAM commands from the memory controller to the optical domain. Additional waveguides can be added to incorporate an arbitrary number of memory modules into one MAP, as shown in Figure 7 with three bidirectional memory module connections.

C. Circuit-Accessed Memory Module

Our proposed circuit-switched memory access architecture requires slightly different usage of DRAM modules. Figure 8(a) shows the Photonic Circuit-Accessed Memory Module (P-CAMM) design. Individual conventional DRAM chips are connected via a local electronic bus to a central optical controller/transceiver, shown in Figure 8(d). The controller (Figure 8(c)) is responsible for demultiplexing the single optical channel into the address and data bus much in the same way as Rambus RDRAM memory technology [38], using the simple control flowchart shown in Figure 5.

Figure 8(b) shows the anatomy of an Electronic Circuit-Accessed Memory Module (E-CAMM), similar to the P-CAMM in structure, but still requiring electronic pins as I/O.

This shift from electrical to photonic technology presents significant advantages for the physical design and implementation of off-chip signaling. One advantage is that the P-CAMM can be locally clocked, as shown, performing serialization and deserialization on the I/O bitrate, and synchronizing it to the DRAM clock rate. Coding or clock transmission can be used to recover the clock in the transceiver, and matched to the local DRAM clock after deserialization. Local clocking and the elimination of long printed circuit board (PCB) traces that the DRAM chips drove allow the P-CAMM to sustain higher clock frequencies than contemporary DRAM modules.

Although the P-CAMM shown in Figure 8(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 [11]. 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 [7].

IV. Experimental Setup

The main goal of this work is to evaluate how silicon photonic technology and circuit-switching affect power efficiency in transporting data to and from off-chip DRAM. We perform this analysis by investigating different network configurations using PhoenixSim, a simulation environment for physical-layer analysis of chip-scale photonic interconnection networks [6].

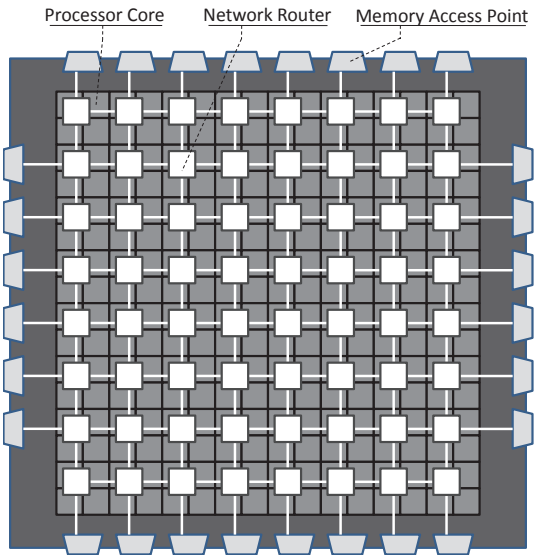


Fig. 9. **Abstract illustration of 8×8 mesh network-on-chip with peripheral memory access points**

A. On-chip Network Architectures

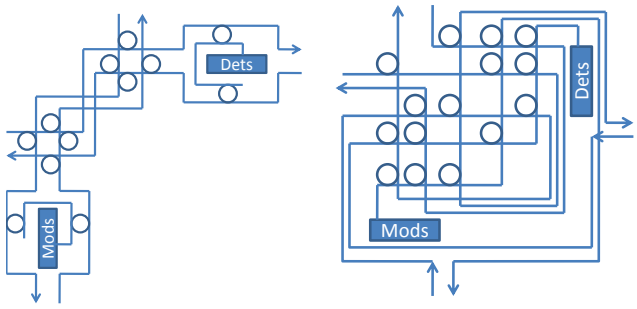
The 2D mesh topology has some attractive characteristics including a modular design, short interconnecting wires, and simple X-Y routing. For these reasons, the 2D mesh has been used in some of the first industry instantiations of tiled many-core networks-on-chip [16], [44]. The mesh also provides the simple and effective means of connecting peripheral memory access points at the ends of rows and columns, utilizing router ports that would have otherwise gone unused or required specialized routing.

We consider three different network architectures: Electronic packet-switched (Emesh), Electronic circuit-switched (EmeshCS), and Photonic circuit-switched (PmeshCS). All three use an 8×8 2D mesh topology to connect the grid of 64 network nodes with DRAM access points on the periphery. An abstract illustration of this setup is shown in Figure 9.

The Emesh and EmeshCS use the routers shown in Figures 1 and 3, respectively, to construct the on-chip 8×8 mesh. They also use *integrated* concentration [24] of 4 cores per network gateway, for a total core count of 256.

Similar to the electronic circuit-switched mesh, we replace the electronic data plane with nanophotonic waveguides and switches to achieve a hybrid photonic circuit-switched network. *External* concentration [24] is used because of the relative difficulty of designing high-radix photonic switches, and to reduce the number of modulator/detector banks. Designs of 4×4 photonic switches in the context of networks have been explored in [5], but because a mesh router requires 5 ports (4 directions + processor core), we must reconsider the design of the photonic switches to minimize power and insertion loss.

Figure 10 introduces two new designs for the photonic 5-port ring resonator-based broadband data switch used in the circuit-switching router for the PmeshCS, designated as PS-1 and PS-2. We designed the PS-1 starting with an optimized



(a) PS-1 (b) PS-2

Fig. 10. **Two designs for a 5-port photonic switch for the PmeshCS**

4×4 switch [5], and adding the modulator and detector banks between lanes. As a result, the switch has a small number of rings and low insertion loss, but exhibits blocking when certain ports are being used (*e.g.* when the detector bank is being used, the east-bound port is blocked). We designed the PS-2 switch from a full ring-matrix crossbar switch, taking out rings to account for no U-turns being allowed, and routing waveguides to eliminate terminations. The PS-2 switch uses more rings and has larger insertion loss, but is fully nonblocking. Because it is not obvious how the two switch designs will affect the network as a whole, we will consider separate photonic mesh instantiations using each switch.

B. Simulation Environment

The PhoenixSim simulation environment allows us to capture physical-layer details, such as physical dimensions and layout, of both electronic and nanophotonic devices to accurately execute various traffic models. We describe the relevant modeling and parameters below.

Photonic Devices. Modeling of optical components is built on a detailed physical-layer library that has been characterized and 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. The behavior of these devices are characterized and modeled at runtime by attributes such as insertion loss, crosstalk, delay, and power dissipation. Tables I and II show some of the most important optical parameters used.

Photonic Network Physical Layer Analysis. The number of available wavelengths is obtained through an insertion loss analysis, a key tool in our simulation environment [6]. Figure 11 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:

$$P_{tot} < P_{NT} \tag{1}$$

$$P_{inj} - P_{loss} > P_{det} \tag{2}$$

TABLE I
OPTICAL DEVICE ENERGY PARAMETERS

Parameter	Value
Data rate (per wavelength)	2.5 Gb/sec
PSE dynamic energy	375 fJ*
PSE static (OFF) energy	400 uJ/sec†
Modulation switching energy	25 fJ/bit‡
Modulation static energy	30 μW§
Detector energy	50 fJ/bit¶
Thermal Tuning energy	1uW/°K‖

TABLE II
OPTICAL DEVICE LOSS PARAMETERS

Device	Insertion Loss
Waveguide Propagation	1.5 dB/cm **
Waveguide Crossing	0.05††
Waveguide Bend	0.005 dB/90 °**
Passing by Ring (Off)	≈0‡‡
Insertion into Ring (On)	0.5‡‡
Optical Power Budget	**35 dB**

Equation 1 states that the total injected power at the first modulator must be below the threshold at which nonlinear effects are induced, thus corrupting the data (or introducing significantly more optical loss). A reasonable value for P_{NT} is around 10-20 mW [26]. Equation 2 states that the power received at the detectors must be greater than the detector sensitivity (usually about -20 dBm) to reliably distinguish between zeros and ones. To ensure this, every wavelength must inject at least enough power to overcome the worst-case optical loss through the network. From these relationships, we can see that the number of wavelengths that can be used in a network relies mainly on the worst-case insertion loss through it.

The two photonic switches that we consider here, labeled PS-1 and PS-2, have different insertion loss characteristics. We determine the worst case network-level insertion loss using each of the switches in the photonic mesh, and find that it equates to 13.5 dB and 18.41 dB for the PS-1 and PS-2, respectively. This means that the Pmesh can safely use approximately 128 wavelengths for the PS-1, and 45 for the PS-2. Despite the PS-1 having 2× more bandwidth than the PS-2, its blocking conditions may yield a lower total bandwidth for the network.

Simulation Parameters. The parameters for all networks have been chosen for power-efficient configurations, typically

*Dynamic energy calculation based on carrier density, 50-μm ring, 320×250-nm waveguide, 75% exposure, 1-V bias.

†Based on switching energy, including photon lifetime for re-injection.

‡Same as *, for a 3 μm ring modulator.

§Based on experimental measurements in [49]. Calculated for half a 10 GHz clock cycle, with 50% probability of a 1-bit.

¶Conservative approximation assuming femto-farad class receiverless SiGe detector with $C < 1fF$.

‖Same value as used in [20]. Average of 20 degrees thermal tuning required.

**From [51]

††Projections based on [13]

‡‡From [25]

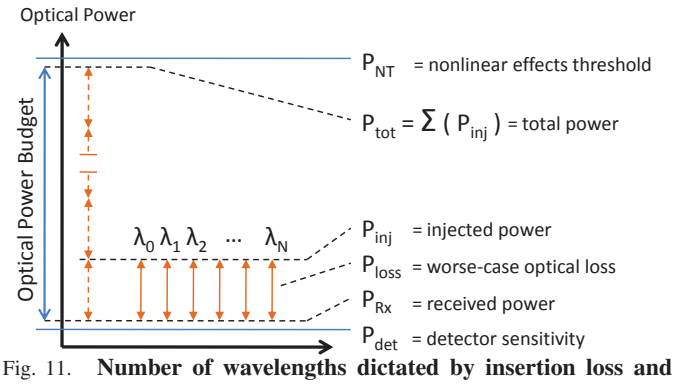


Fig. 11. **Number of wavelengths dictated by insertion loss and optical power budget.**

the most important concern for embedded systems. We consider the key limiting factor for our embedded system design to be ideal I/O bandwidth. For photonics, I/O bandwidth (which is the same as on-chip bandwidth due to bit-rate transparent devices) is limited by insertion loss as described above.

The electronic networks, however, are limited by pin count. Electronic off-chip signaling bandwidth is limited by packaging constraints at a total of 1792 I/O pins (64 pins per MAP), which is more than 2× that of today's CMPs (TILE64 [3]). Note that even though a real chip would require a significant number of additional I/O ports, we assume that all of these 1792 pins are dedicated to DRAM access. This places the total number of pins well over 4000, assuming a 50% total I/O-to-power/ground ratio. According to ITRS [18], attaining this pin count will require solutions to significant packaging challenges.

Table III shows the more important simulation parameters that will be used for simulations in Section V. For each network, we work backwards from the I/O bandwidth available across the chip boundary to the on-chip and DRAM parameters. We assume all cores run at 2.5 GHz.

The Emesh uses conventional DRAM bidirectional signalling with 2 DRAM channels for increased access concurrency running at 1.6 GT/s, using a conventional 8 arrays per chip and 8 chips per DIMM. The Emesh network runs at 1.6GHz to match this bandwidth. Our router model implements a fully pipelined router which can issue two grant requests per cycle (for different outputs) and uses dimension ordered routing for deadlock avoidance, and bubble flow control [37] for congestion management. One virtual channel (VC) is used for writes and core-to-core communication, and a separate VC is used for read responses for reduced read latency. For power dissipation modeling, the ORION 2.0 electronic router model [21] is integrated into the simulator, 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. Off-chip electronic I/O wires

TABLE III
ELECTRONIC SIMULATION PARAMETERS

Parameter	Emesh	EmeshCS	PmeshCS (PS1)	PmeshCS (PS2)
Chip IO Parameters				
Physical I/O per MAP	64	32 (diff pair)	2 (w/ 128 λ)	2 (w/ 45 λ)
I/O bit rate	1.6 GT/s	10 Gb/s	2.5 Gb/s	2.5 Gb/s
Ideal Bandwidth per I/O Link (Gb/s)	102	320	320	112
NoC Electronic Parameters				
Packet switched Clock Freq (GHz)	1.6	1.0	1.0	1.0
Data Plane Freq (GHz)	-	2.5	2.5	2.5
Buffer Size (b)	1024	128	128	128
Virtual Channels	2	1	1	1
Control Plane V_{DD}	0.8	0.8	0.8	0.8
Control Plane V_{th}	Norm	High	High	High
Data Plane V_{DD}	-	1.0	1.0	1.0
Data Plane V_{th}	-	Norm	Norm	Norm
Electronic Channel Width	64	32 (128 for data plane)	32	32
Bandwidth per On-chip Link (Gb/s)	102	320	320	112
DRAM Parameters				
Base DRAM Frequency (MHz)	1066	1066	1066	1066
Arrays Per Bank	8	32	32	16
Chips Per DIMM	8	10	10	8
DIMMs Per MAP	2	1	1	1
Total Memory Per MAP	2GB	2GB	2GB	2GB
Bandwidth per DIMM (Gb/s)	128	320	320	128

and transceivers are modeled as using 1 pJ/bit, a reasonable projection based on [18], [33].

The EmeshCS uses high speed (10Gb/s) bidirectional differential pairs for I/O signalling, requiring serialization and deserialization (SerDes) at the chip edge between the 2.5 GHz data plane. The path-setup electronic control plane runs at a slower 1.0 GHz to save power. The photonic networks use the exact same control plane as the EmeshCS, and the same 2.5 Gb/s bitrate per wavelength to avoid significant SerDes power consumption at the network gateways. SerDes power is modeled using ORION flip-flop models as shift registers running at the higher clock rate, bandwidth matching both sides with parallel wires. For all three circuit-switched configurations, we increase the number of DRAM arrays per chip by decreasing the row and column count to be able to continuously feed the I/O. 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.

DRAM Modeling. The cycle-accurate simulation of the DRAM memory subsystem along with the network on chip for the Emesh is accomplished by integrating DRAMsim [48] into our simulator. The Emesh behaves like a typical contemporary system in that the packetization of messages required by the packet-switched network yields small memory transaction sizes, analogous to today's cachelines. Therefore, a DRAM model which is based on typical DDR SDRAM components and control policies that might be seen in real systems, such as DRAMsim, is appropriate for this configuration.

The two circuit-switched networks, however, exhibit different memory access behavior than a packet-switched version, thus enabling a simplification of the memory control logic. For this reason, we use our own model for DRAM components and control. 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.

We base our model parameters around a Micron 1-Gb DDR3 chip [32], 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.

V. EMBEDDED APPLICATION SIMULATION

A. Evaluation Framework

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 [45], 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 interdependencies. 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.

Three applications are considered: projective transform, matrix multiply, and fast fourier transform (FFT). Results for power usage, performance (GOPS), and efficiency (GOPS/W) improvement are provided for each.

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 [22].

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.

MORE allows us to retain identical image and projections sizes while still inducing data movement in the projection process as well as investigating various transformation matrices. For this experiment, we consider this application on various image sizes where the image orientation is rotated by ninety degrees.

Matrix Multiply Matrix multiplication is a common operation in signal and image processing, where it can be used in filtering as well as to control hue, saturation and contrast in an image. It is a natural candidate for consideration on our architecture, given that multiple data points need to be accessed and then summed to form a single entry in the result.

While various algorithms for matrix multiplication can be considered for matrices of any dimension, we shall focus our analysis on an inner product algorithm over square matrices. Here, in an $N \times N$ matrix, each entry is generated by first multiplying together two vectors of size N (corresponding to a row and a column), and then summing the entries in the resulting vector to form a single entry in the result.

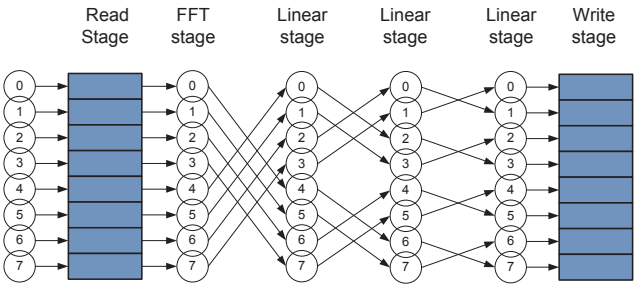


Fig. 12. **FFT computation per the Cooley-Tukey algorithm.**

The inner product algorithm requires time proportional to N^3. While the best known algorithm for matrix multiply is $O(N^{2.376})$, the constants in the algorithm make it infeasible for all but the largest of matrices. Even Strassen's algorithm [42], with a bound of $O(N^{2.806})$ is frequently considered too cumbersome and awkward to implement, especially in a parallel environment. Though more computationally expensive, the inner product algorithm also lends itself more naturally to a parallel implementation, making it our algorithm of choice.

Fast Fourier Transform Computing the Fast Fourier Transform (FFT) of a set of data points is an essential algorithm which underlies many signal processing and scientific applications. In addition to the widespread use of the FFT, the inherent data parallelism that can be exploited in its computation makes it a good match for measuring the performance of networks-on-chip. A typical way the FFT is computed in parallel, and which is employed in our execution model, is the Cooley-Tukey method [10]. The communication patterns and computation stages for 8 nodes are shown in Figure 12. We run the FFT where each core begins with 2^{10}, 2^{12}, 2^{14}, 2^{16}, and 2^{18} samples, and average the results.

B. Simulation Results

Table IV shows the averaged results for the different network configurations across the 3 applications, showing network-related power, total system performance (GOPS), and total system efficiency (GOPS/W) which is normalized to the Emesh for comparison. In all cases, the circuit switched networks achieve considerable improvements in both performance and power over the Emesh.

For the Projective Transform and Matrix Multiply, the EmeshCS consumes some additional power to achieve considerable gains in performance. The photonic networks also perform significantly better than the Emesh, though at much lower power than the EmeshCS. The PS-2 generally consumes less power because it has less modulators (but less bandwidth), and uses non-blocking switches which reduces path-setup block and retry on the electronic control plane, and therefore power. The FFT exhibits different communication and memory access behavior than the other applications, and gains are not

TABLE IV
RESULTS FOR PERFORMANCE, NETWORK POWER, AND IMPROVEMENT OVER ELECTRONIC MESH IN SIGNAL AND IMAGE PROCESSING APPLICATIONS

	Projective Transform			Matrix Multiply			FFT		
Network	*Power (Watts)*	*Perf. (GOPS)*	*Impr. (GOPS/W)*	*Net. Pow. (Watts)*	*Perf. (GOPS)*	*Impr. (GOPS/W)*	*Power (Watts)*	*Perf. (GOPS)*	*Impr. (GOPS/W)*
Emesh	11.2	1.04	1x	11.1	0.78	1x	11.4	1.75	1x
EmeshCS	19.0	47.3	26.9x	15.8	31.82	29.01x	11.2	4.74	2.82x
PS-1	4.37	27.80	68.6x	4.35	26.51	87.64x	4.28	4.32	6.72x
PS-2	2.21	17.76	86.7x	2.17	13.48	89.33x	2.15	3.12	9.67x

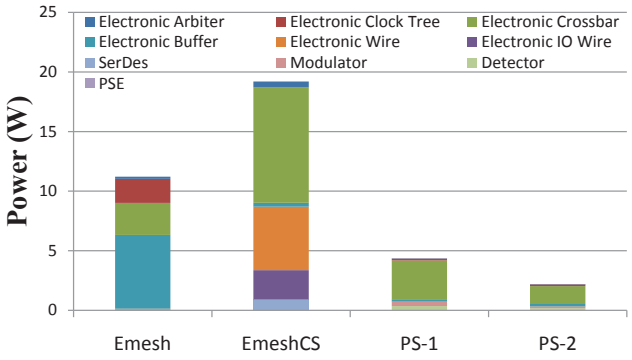


Fig. 13. **Projective Transform network power breakdown.**

as profound though still an order of magnitude in efficiency for PS-2.

The breakdown of power consumption for the various network components is shown in Figure 13 for the Projective Transform, one of the more network-active applications. We can see that the Emesh power is comprised mostly of buffer, crossbar, and clock power. EmeshCS alleviates buffer power as intended, but at the cost of crossbar and wire power in the higher-frequency data plane. Finally, the photonic networks achieve drastically lower power through distance-independent efficient modulation and detection.

VI. RELATED WORK

Networks-on-chip have entered the computer architecture arena to enable core-to-core and core-to-DRAM communication on contemporary processors. The Tilera TILE-Gx processors [44] and Intel Polaris [16] are examples of real packet-switched NoC implementations with up to 100 and 80 cores, respectively. The Cell BE [9] uses a circuit-switched network to connect heterogeneous cores and a single memory controller.

Next-generation NoC designs using silicon nanophotonic technology have also been proposed. 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 [47]. Similarly, wavelength-routed bus-based architectures have been proposed which take advantage of WDM for arbitration [23], [34]. Batten *et al.* proposed a wavelength-selective routed architecture 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 [2] on which source nodes transmit on the specific wavelength that is received by a single destination. Hadke *et al.* proposed OCDIMM, a WDM-based optical interconnect for FBDIMM memory banks, which uses wavelength-routing to achieve a memory system that scales while sustaining low latencies [15]. Phastlane was designed for a cache-coherent CMP, enabling snoop-broadcasts and cacheline transfers in the optical domain [8]. Finally, on-chip hybrid electronically circuit-switched photonic networks have been proposed by Shacham *et al.* [40] and Petracca *et al.* [35], and further investigated by Hendry *et al.* [17] and Chan *et al.* [5] .

The main contribution of this work over previous work is to explore circuit-switching as a memory access method in the context of a nanophotonic-enabled interconnect, using the same network resources which enable core-to-core communication. Uniquely, our simulation framework incorporates physically-accurate photonic device models, detailed electronic component models, and cycle-accurate DRAM device and control models into a full system simulation.

VII. CONCLUSION

By incorporating cycle-accurate DRAM control and device models into a network simulator with detailed physically-accurate models of both photonic and electronic components, we are able to investigate circuit-switched memory access in an embedded high-performance CMP computing node design. We run three signal and image processing applications on different network implementations normalized to topology, pin constraints, total memory, and CMOS technology to characterize the different networks with respect to bandwidth and latency. Accessing memory using a circuit-switched network was found to increase performance through long burst lengths and decrease power by eliminating performance-dependent buffers. Silicon nanophotonic technology adds to these benefits with low-energy transmission and higher bandwidth density which will enable future scaling. Additional benefits include reduced memory controller complexity, dramatically lower pin counts, and relaxed memory module and compute board design constraints, all of which are beneficial to the embedded computing world.

REFERENCES

[1] V. R. Almeida, R. R. Panepucci, and M. Lipson. Nanotaper for compact mode conversion. *Optics Letters*, 28(15):1302–1304, August 2003.

[2] C. Batten et al. Building manycore processor-to-DRAM networks with monolithic silicon photonics. In *IEEE Micro Special Issue: Micro's Top Picks from Hot Interconnects 16*, 2009.

[3] S. Bell, B. Edwards, J. Amann, et al. TILE64 Processor: A 64-Core SoC with Mesh Interconnect. In *Proceedings of the IEEE International Solid-State Circuits Conference*, pages 88–89, 598, 2008.

[4] W. Carlson, T. El-Ghazawi, B. Numrich, and K. Yelick. Programming in the partitioned global address space model. Presented at Supercomputing 2003. Online at http://www.gwu.edu/upc/tutorials.html.

[5] J. Chan, A. Biberman, B. G. Lee, and K. Bergman. Insertion loss analysis in a photonic interconnection network for on-chip and off-chip communications. In *IEEE Lasers and Electro-Optics Society (LEOS)*, Nov. 2008.

[6] J. Chan, G. Hendry, A. Biberman, K. Bergman, and L. P. Carloni. Phoenixsim: A simulator for physical-layer analysis of chip-scale photonic interconnection networks. In *DATE: Design, Automation, and Test in Europe.*, Mar. 2010.

[7] L. Chen, K. Preston, S. Manipatruni, and M. Lipson. Integrated GHz silicon photonic interconnect with micrometer-scale modulators and detectors. *Optics Express*, 17(17), August 2009.

[8] M. J. Cianchetti, J. C. Kerekes, and D. H. Albonesi. Phastlane: a rapid transit optical routing network. *SIGARCH Comput. Archit. News*, 37(3):441–450, 2009.

[9] S. Clark, K. Haselhorst, K. Imming, J. Irish, D. Krolak, and T. Ozguner. Cell broadband engine interconnect and memory interface. In *17th Annual Hot Chips*, Aug 2005.

[10] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of comples fourier series. *Mathematics of Computation*, 19:297–301, 1965.

[11] Corning Inc. Datasheet: Corning SMF-28e optical fiber product information. Online at http://www.princetel.com/datasheets/SMF28e.pdf.

[12] M. Duranton. The challenges for high performance embedded systems. In *DSD '06: Proceedings of the 9th EUROMICRO Conference on Digital System Design*, pages 3–7, Washington, DC, USA, 2006. IEEE Computer Society.

[13] T. Fukazawa, T. Hirano, F. Ohno, and T. Baba. Low loss intersection of Si photonic wire waveguides. *Japanese Journal of Applied Physics*, 43(2):646–647, 2004.

[14] C. Gunn. CMOS photonics for high-speed interconnects. *IEEE Micro*, 26(2):58–66, March 2006.

[15] A. Hadke et al. OCDIMM: Scaling the DRAM memory wall using WDM based optical interconnects. In *Proceedings of the 16th IEEE Symposium on High Performance Interconnects*, Aug. 2008.

[16] J. Held, J. Bautista, and S. Koehl. From a few cores to many: A tera-scale computing research overview, 2006. White paper. Online at http://download.intel.com/research/platform/terascale/.

[17] G. Hendry et al. Analysis of photonic networks for a chip-multiprocessor using scientific applications. In *The 3rd ACM/IEEE International Symposium on Networks-on-Chip*, May 2009.

[18] The international technology roadmap for semiconductors (ITRS). http://www.itrs.net.

[19] B. Jacob, S. W. Ng, and D. T. Wang. *Memory Systems: Cache, DRAM, Disk*. Morgan Kaufmann, 2007.

[20] A. Joshi, C. Batten, Y.-J. Kwon, S. Beamer, I. Shamim, K. Asanovic, and V. Stojanovic. Silicon-photonic Clos networks for global on-chip communication. In *3rd ACM/IEEE International Symposium on Networks-on-Chip*, May 2009.

[21] A. B. Kahng, B. Li, L.-S. Peh, and K. Samadi. Orion 2.0: A fast and accurate NoC power and area model for early-stage design space exploration. pages 423–428, April 2009.

[22] H. Kim, E. Rutledge, S. Sacco, S. Mohindra, M. Marzilli, J. Kepner, R. Haney, J. Daly, and 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*, pages 327–333, Washington, DC, USA, 2008. IEEE Computer Society.

[23] 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*, pages 492–503, Washington, DC, USA, 2006. IEEE Computer Society.

[24] P. Kumar, Y. Pan, J. Kim, G. Memik, and 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*, pages 276–285, Washington, DC, USA, 2009. IEEE Computer Society.

[25] B. G. Lee, A. Biberman, P. Dong, M. Lipson, and K. Bergman. All-optical comb switch for multiwavelength message routign in silicon photonic networks. *IEEE Photonics Technology Letters*, 20(10):767–769, May 2008.

[26] B. G. Lee, X. Chen, A. Biberman, X. Liu, I.-W. Hsieh, C.-Y. Chou, J. Dadap, R. M. Osgood, and K. Bergman. Ultra-high-bandwidth WDM signal integrity in silicon-on-insulator nanowire waveguides. *IEEE Photonics Technology Letters*, 20(6):398–400, May 2007.

[27] B. G. Lee et al. High-speed 2×2 switch for multi-wavelength message routing in on-chip silicon photonic networks. In *European Conference on Optical Communication (ECOC)*, Sept. 2008.

[28] B. G. Lee et al. High-speed 2×2 switch for multiwavelength silicon-photonic networks-on-chip. *Journal of Lightwave Technology*, 27(14):2900–2907, July 2009.

[29] D. Lee, S. S. Bhattacharyya, and W. Wolf. High-performance buffer mapping to exploit DRAM concurrency in multiprocessor DSP systems. In *IEEE/IFIP International Symposium on Rapid System Prototyping*, 2009.

[30] S. J. McNab, N. Moll, and Y. A. Vlasov. Ultra-low loss photonic integrated circuit with membrane-type photonic crystal waveguides. *Optics Express*, 11(22):2927–2938, November 2003.

[31] G. D. Micheli, R. Ernst, and W. Wolf. Readings in hardware/software co-design, 2001.

[32] Micron Technology Inc. Product specification. 1 Gb DDR3 SDRAM Chip. Online at http://www.micron.com/products/partdetail?part=MT41J256M4JP-125.

[33] D. A. B. Miller. Device requirements for optical interconnects to silicon chips. In *Proc.IEEE Special Issue on Silicon Photonics*, pages 1166 – 1185, 2009.

[34] Y. Pan et al. Firefly: Illuminating future network-on-chip with nanophotonics. In *Proceedings of the International Symposium on Computer Architecture*, 2009.

[35] M. Petracca, B. G. Lee, K. Bergman, and L. Carloni. Design exploration of optical interconnection networks for chip multiprocessors. In *16th IEEE Symposium on High Performance Interconnects*, Aug 2008.

[36] M. Petracca, B. G. Lee, K. Bergman, and L. P. Carloni. Photonic NoCs: System-level design exploration. *IEEE Micro*, 29:74–85, 2009.

[37] V. Puente et al. Adaptive bubble router: a design to improve performance in torus networks. In *Proc. Of International Conf. On Parallel Processing*, pages 58–67, 1999.

[38] Rambus. RDRAM memory technology. http://www.rambus.com/us/products/rdram/index.html.

[39] J. Schrauwen, F. V. Laere, D. V. Thourhout, and R. Baets. Focused-ion-beam fabrication of slanted grating couplers in silicon-on-insulator waveguides. *IEEE Photonics Technology Letters*, 19(11):816–818, June 2007.

[40] A. Shacham, K. Bergman, and L. P. Carloni. Photonic networks-on-chip for future generations of chip multiprocessors. *IEEE Transactions on Computers*, 57(9):1246–1260, 2008.

[41] L. Shares et al. Terabus: Terabit/second-class card-level optical interconnect technologies. *IEEE Journal of Selected Topics in Quantum Electronics*, 12(5), Sept. 2006.

[42] V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 14(3):354–356, 1969.

[43] Sun Microsystems, Inc. Sun Microsystems Ultrasparc T2 datasheet. available at http://www.sun.com/.

[44] Tilera Corporation. TILE-Gx processor family. Online at http://www.tilera.com/products/TILE-Gx.php.

[45] N. Travinin, H. Hoffmann, R. Bond, H. Chan, J. Kepner, and 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*, page 254, Washington, DC, USA, 2005. IEEE Computer Society.

[46] C. H. van Berkel. Multi-core for mobile phones. In *DATE*, pages 1260–1265, 2009.

[47] D. Vantrease et al. Corona: System implications of emerging nanophotonic technology. In *Proceedings of 35th International Symposium on Computer Architecture*, Aug 2008.

[48] D. Wang et al. DRAMsim: A memory-system simulator. *SIGARCH Computer Architecture News*, 33(4):100–107, Sept. 2005.

[49] M. R. Watts. Ultralow power silicon microdisk modulators and switches. In *5th Annual Conference on Group IV Photonics*, 2008.

[50] W. Wolf. Embedded computer architectures in the MPSoC age. In *WCAE '05: Proceedings of the 2005 workshop on Computer architecture education*, page 1, New York, NY, USA, 2005. ACM.

[51] F. Xia, L. Sekaric, and Y. Vlasov. Ultracompact optical buffers on a silicon chip. *Nature Photonics*, 1:65–71, Jan. 2007.