Runtime Management of Laser Power in Silicon-Photonic Multibus NoC Architecture

Chao Chen, *Student Member, IEEE*, and Ajay Joshi, *Member, IEEE*

(Invited Paper)

***Abstract*—Silicon-photonic links have been proposed to replace electrical links for global on-chip communication in future many-core processors. Silicon-photonic links have the advantage of lower data-dependent power and higher bandwidth density, but the high laser power can more than offset these advantages. We propose a solution to manage laser power of silicon-photonic network-on-chip (NoC) in many-core system. We present a silicon-photonic multibus NoC architecture between private L1 caches and distributed L2 cache banks which uses weighted time-division multiplexing to distribute the laser power across multiple buses based on the runtime variations in the bandwidth requirements within and across applications to maximize energy efficiency. The multibus NoC architecture also harnesses the opportunities to switch OFF laser sources at runtime, during low-bandwidth requirements, to reduce laser power consumption. Using detailed system-level simulations, we evaluate the multibus NoC architecture and runtime laser power management technique on a 64-core system running NAS parallel benchmark suite. The silicon-photonic multibus NoC architecture provides more than two times better performance than silicon-photonic Clos and butterfly NoC architectures, while consuming the same laser power. Using runtime laser power management technique, the average laser power is reduced by more than 49% with minimal impact on the system performance.**

***Index Terms*—Laser power management, many-core, network-on-chip (NoC), silicon-photonics.**

I. Introduction

FUTURE high-performance computers (HPCs) and data centers (DCs) will use several many-core processors with each processor having dozens to hundreds of cores on a die. The performance of these many-core processors and in turn that of the HPCs and DCs will be driven by the energy-limited bandwidth of both processor-to-memory (interchip), and core-to-core/core-to-cache/cache-to-cache (intrachip) communication networks. Hence, high bandwidth density and low-power communication networks are needed to maximize the performance of these many-core processors. To this end, silicon-photonic links have been projected to supplant the electrical links, in both interchip and intrachip communication networks in future many-core processors. Silicon-photonic link technology projections indicate an order of magnitude higher bandwidth density and several times lower energy cost compared to the projected electrical link technology [7], [22]. This will significantly improve the throughput of both interchip and intrachip communication networks, and also improve the energy efficiency of the overall many-core processor system.

In this paper, we focus on intrachip communication network designed using silicon-photonic link technology. The use of silicon-photonic link technology for intrachip communication has been widely explored. Several different silicon-photonic intrachip communication network architectures, referred to as network-on-chip (NoC) architectures, have been investigated [12], [22]–[24], [27], [32], [38], [39], [44]. Though the silicon-photonic NoC provides bandwidth density and data-dependent link energy advantages, a nontrivial amount of power is required in the laser source that is used for driving the silicon-photonic NoC [22], [31]. In fact, the laser power could more than offset the bandwidth density and data-dependent energy advantages of the silicon-photonic links. To use silicon-photonic link technology for communication in future HPCs and DCs, it is imperative to explore techniques to reduce the power consumed in the laser sources.

Typically, the applications running on HPCs and DCs exhibit spatially variant and/or temporally variant behavior. The application characteristics including instructions committed per cycle, cache/memory accesses, and generated NoC traffic could vary spatially across applications as well as temporally within an application. This provides us with an opportunity to proactively reconfigure the NoC architecture to minimize the power consumed in the laser source while maintaining application performance. In other words, we provide the minimum NoC bandwidth (which is directly proportional to laser power) required for an application to achieve the maximum possible performance (number of instructions committed per cycle) at any given point of time.

In this paper, we propose a policy for runtime management of the power consumed by one or more laser sources that drive the silicon-photonic NoC of many-core processor. We explore the application of our policy on a multibus NoC architecture that connects the private L1 caches of each core and the distributed L2 cache banks of the many-core processor. For laser power management, we adopt a token-based weighted time-division multiplexed approach, where depending on the spatial and

Manuscript received June 8, 2012; revised September 11, 2012; accepted October 26, 2012. Date of publication January 11, 2013; date of current version April 3, 2013. This work was supported in part by the National Science Foundation CAREER Award CCF-1149549.

The authors are with the Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215 USA (e-mail: chen9810@bu.edu; joshi@bu.edu).

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

Digital Object Identifier 10.1109/JSTQE.2012.2228170

temporal variations in the NoC bandwidth requirements of an application, we distribute the entire available NoC bandwidth by adjusting the bandwidth multiplexing weights associated with each bus to maximize application performance. In addition, we also switch ON/OFF one or more laser sources if there is a significant increase/decrease in the NoC bandwidth requirements of an application. Our policy uses the average packet latency to determine the minimum bandwidth required to keep the NoC out of saturation for an application, and then reconfigures the NoC architecture accordingly at runtime. The ultimate goal here is to sustain the application performance (instructions committed per cycle), while minimizing the power consumed in the laser.

1) We propose a token-based weighted time-division multiplexed multibus NoC architecture implemented using silicon-photonic technology for a many-core processor. The time-division multiplexing approach is used to multiplex the laser output power, i.e., network bandwidth, across the different buses in the network. This NoC architecture is specifically geared toward application-aware dynamic laser power management.
2) We propose a policy for runtime management of power consumed in the laser source driving the multibus NoC connecting private L1 caches and the distributed L2 cache banks. Based on the NoC bandwidth requirement of the running applications, which is continuously determined at runtime over fixed reconfiguration time intervals, we adjust the bandwidth weights associated with each bus to redistribute the aggregate NoC bandwidth across all the buses at runtime. In addition, we switch ON/OFF one or more laser sources depending on the aggregate NoC bandwidth requirements. We consider a range of reconfiguration time intervals and laser switch ON/OFF times to identify the optimal control parameters for our laser power management policy.
3) As a case study, for a 64-core system, we use the Gem5 [10] full-system simulator interfaced with Booksim [13], to first compare our multibus NoC architecture with silicon-photonic Clos and butterfly NoC architectures when running the NAS parallel benchmark (NPB) suite [5]. The multibus NoC architecture provides better performance, while consuming the same laser power. We then evaluate the application of our proposed runtime laser power management policy for the multibus NoC architecture. The use of runtime laser power management technique further reduces average laser power by more than 49% with minimal impact on the system performance (<6%).

The rest of the paper is organized as follows: Section II describes our target system, which is followed by a discussion of its underlying silicon-photonic link technology in Section III. A detailed description of the multibus NoC architecture and a brief overview of Clos and butterfly NoC architectures are presented in Section IV. Section V explains our policy for dynamic management of laser power, and the evaluation of our proposed policy using a full-system simulator is presented in Section VI. In Section VII, a discussion about the application of our proposed policy to other NoC and many-core processor architectures is given. In Section VIII, we provide a description of the related work, followed by Section IX that discusses the key conclusions of our analysis.

TABLE I
MICROARCHITECTURAL PARAMETERS OF THE 64-CORE TARGET SYSTEM

Micro-architecture Configuration	
Core Frequency	2.5 GHz
Branch Predictor	Tournament predictor
Issue	2-way Out-of-order
Reorder Buffer	128 entries
Functional Units	2 IntALUs, 1 IntMult 1 FPALU, 1 FPMult
Physical Regs	128 Int, 128 FP
Instruction Queue	64 entries
L1 I-Cache	16 KB @ 2 ns
L1 D-Cache	16 KB @ 2 ns
Cache Coherence	Directory based
L2 Cache	4-way set-associative, 64 B block Distributed L2: 8 x 2 MB @ 6 ns
Memory	8 memory controllers 8 PIDRAM @ 50 ns

II. TARGET SYSTEM

We choose a 64-core processor that is manufactured using 22-nm CMOS technology [26] as our target system. Each core supports two-way issue out-of-order execution, and has two integer ALUs, one integer multiplication unit, one floating-point ALU, and one floating-point multiplication unit. The core architecture is configured based on the cores used in Intel's 48-core signal component control [21]. The microarchitectural parameters are listed in Table I. The cores operate at 2.5-GHz frequency and have a supply voltage of 0.9 V.

Each core has 16 kB L1 instruction cache and 16 kB L1 data cache. The system has a 16 MB distributed L2 cache (eight banks, 2 MB/bank) with each bank mapping to a unique set of memory addresses. The L2 caches are located at one edge of the chip. We use CACTI [42] to estimate the L1 and L2 cache access time. The many-core processor has eight memory controllers, a memory controller per L2 cache bank, with each memory controller located next to the corresponding L2 cache bank. Cache lines are interleaved across eight banks to improve the parallel accessibility. We use MESI directory-based protocol [33] for maintaining cache coherency between the L1 and L2 caches. The cache coherency directories are located next to the associated L2 cache banks. They maintain a copy of cache line status by monitoring the on-chip network transactions. For our analysis, we consider three different NoC architectures—multibus, Clos, and butterfly, that provide connectivity between L1 and L2 caches. A detailed discussion for these NoC architectures is presented in Section IV.

There is a separate off-chip photonic network that connects memory controllers to PIDRAM chips [8]. We assume an average time of 50 ns for the communication from the memory controllers to PIDRAMs and back. We ignore the variations

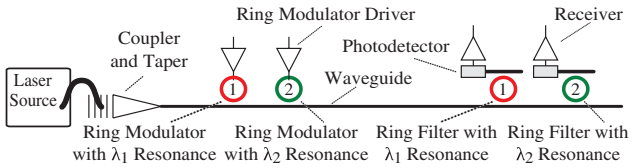


Fig. 1. Photonic link components—two point-to-point photonic links implemented with WDM.

in queuing latencies at the input of the memory controllers because the high off-chip bandwidth using PIDRAM significantly reduces the number of outstanding memory requests in the queue.

III. Photonic Technology

Currently, there are several efforts in place in both academia and industry to integrate photonic devices with VLSI chips. In particular, in 2010, Intel demonstrated a 50-Gb/s silicon-photonic link with integrated lasers using hybrid silicon laser technology [1]. Meanwhile, IBM created a CMOS-integrated nanophotonic technology that can integrate monolithically both the electrical and optical circuits on the same silicon chip on the front end of the standard CMOS line [2].

Our proposed technique for runtime management of laser power consumption in many-core processor is relatively agnostic of the exact underlying silicon-photonic technology. It is applicable to both, monolithic integration of photonic devices using bulk CMOS process [19], [28], [29] or silicon on insulator process [2]–[4], [11], [14], [17], [30], [40], [45], and 3-D integration of photonic devices by depositing SiN [6], [9], [15], [20] or polycrystalline silicon [35]–[37] on top of the metal stack, design approaches. Fig. 1 shows a generic wavelength-division multiplexed (WDM) photonic link used for intrachip communication. Light waves of wavelength λ_1 and λ_2 emitted by an off-chip laser source are coupled into the chip using vertical couplers. The vertical coupler guides the light waves into the waveguides. These light waves pass next to a series of ring modulators controlled by modulator drivers based on the data to be transmitted on the link. The modulators convert data from electrical medium to photonic medium. The modulated light waves travel along the waveguide and can pass through zero or more ring filters. At the receiver side, the ring filter "drops" the light wave having the filter's resonant wavelength onto a photodetector. The resulting photodetector current is sensed by an electrical receiver. At this stage, data are converted back into the electrical medium from the photonic medium.

For our analysis, we use the silicon-photonic link design described in [22]. We consider double-ring filters and a four THz free-spectral range, which enables up to 128λ modulated at 10 Gb/s on each waveguide (64λ in each direction, interleaved to alleviate filter roll-off requirements and crosstalk). A nonlinearity limit of 30 mW at 1 dB loss is assumed for the waveguides. The waveguides are single mode and have a pitch of 4 μm to minimize the crosstalk between neighboring waveguides. We assume modulator ring and filter ring diameters of $\approx$10 μm. The latency of a global photonic link is assumed to be 3 cycles (1 cycle in flight and 1 cycle each for E/O and O/E conversion). We assume a 5 μm separation between the photonic and electrical devices to maintain signal integrity. We use the projected silicon-photonic link energy cost (see Table II) and projected silicon-photonic device losses (see Table III) for our analysis [22].

TABLE II
Aggressive and Conservative Energy and Power Projections for Photonic Devices

Design	Tx (fJ/bt)		Rx (fJ/bt)		TT
	DDE	FE	DDE	FE	(fJ/bt/heater)
Aggressive	20	5	20	5	16
Conservative	80	10	40	20	32

Tx = Modulator driver circuits, Rx = Receiver circuits, TT = Thermal tuning circuits, fJ/bt = average energy per bit-time, DDE = Data-traffic dependent energy, FE = Fixed energy (clock, leakage) [22].

TABLE III
Optical Loss per Component [22]

Photonic device	Optical Loss (dB)
Optical Fiber (per cm)	0.5e-5
Coupler	1
Splitter	0.2
Non-linearity (at 30 mW)	1
Modulator Insertion	0
Waveguide (per cm)	1$\sim$3
Waveguide crossing	0.05
Filter through	1e-4
Filter drop	1.5
Photodetector	0.1

The basic premise of the proposed laser power management technique is to perform a weighted time-division multiplexing of the photonic network bandwidth, and if necessary the switching ON/OFF of the laser sources based on the application bandwidth requirements to improve energy efficiency. The weighted time-division multiplexing of the photonic network bandwidth involves tuning and detuning of the banks of filter rings associated with each bus. We assume this tuning and detuning is done through charge injection and the cost associated with tuning and detuning is the same as that for the modulator driver circuits.

For powering the waveguides in our many-core processor, we use a broadband off-chip laser source. When a laser source is switched ON, in addition to the need for stabilization against the interplay between the carrier and photon density, the laser source output also needs to stabilize against thermal variations. There is a large effort toward designing laser sources with high-energy efficiency and low switch ON/OFF times. In this paper, we explore the limits and opportunities for application of our power management technique for an aggressive range of laser source switch ON/OFF times (1 μs to 1 ms). For each data point in this range, we choose the sampling interval (i.e., time interval for adjusting bandwidth) to be 10$\times$ the switch ON/OFF times. One of the key goals of the analysis is to develop an architecture-driven roadmap for designing laser sources for many-core processors with silicon-photonic NoC.

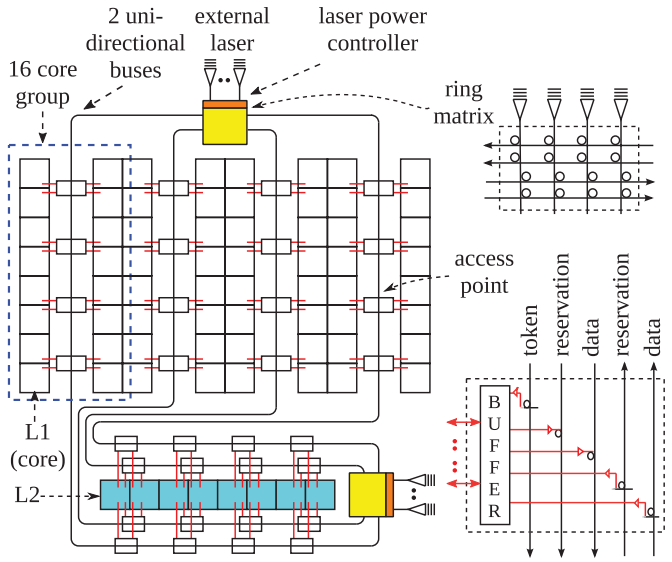


Fig. 2. Physical layout of the silicon-photonic multi-bus NoC architecture—The 64 cores are divided into four groups and each group communicates with L2 banks through two unidirectional buses. Any L1-to-L1 communication is through the cache coherence directory located at the L2 banks. The two laser power controllers dynamically guide light waves into the appropriate buses using time-division multiplexing through associated the ring matrices.

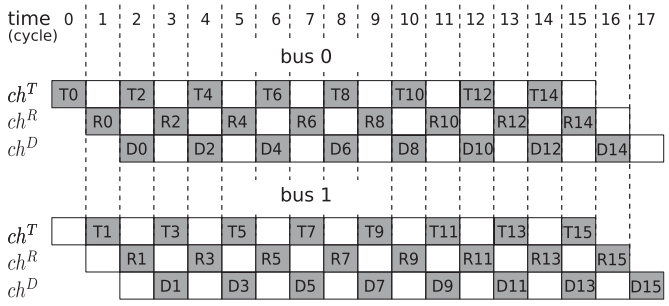


Fig. 3. Timing diagram for token stream arbitration—Here ch^T = token channel, ch^R = reservation channel, ch^D = data channel, Tx = Token for time slot "x," Rx = Reservation request for time slot "x" and Dx = Data in time slot "x." For L1-to-L2 communication, the L1 access point grabs a token and then reserves the destination L2 access point before modulating the data through data channel. The grabbing of token, transmission of reservation request, and transmission of data packet is performed in consecutive cycles. In the example, the two buses have 50% utilization each.

IV. Multibus NoC Architecture

In this section, we will provide a detailed description of our silicon-photonic multibus NoC architecture and an overview of silicon-photonic Clos and butterfly NoC architectures that are used as comparison points, for a many-core processor.

Fig. 2 shows the physical layout of the silicon-photonic multi-bus NoC architecture for our target 64-core system. We divide the 64 cores into four logical groups with 16 cores in two adjacent columns allocated to each group. Each group has two unidirectional silicon-photonic buses, one each for L1-to-L2 and L2-to-L1 communication. The light waves emitted by the off-chip laser sources are time-division multiplexed across all the buses using ring matrices controlled by two laser power controllers. Each unidirectional bus consists of three channels—token channel, reservation channel, and data channel. On each bus, there are 4 L1 access points—one each for a set of four cores, and four L2 access points—one each for a set of 2 L2 banks. The concentration does not require extra local electrical links considering the physical locations of the L1 and L2 caches. Moreover, the concentration helps to maximize the utilization of the multibus access points. A simple round-robin arbitration is used within each access point. Fixed priority arbitration is used across access points based on their physical locations on the bus. The access point that is closest to the laser source has the highest priority to grab the token and use the data channel.

In each bus, we divide the NoC packet into 4-bit sets and each set is mapped onto a wavelength to match the processor frequency of 2.5 GHz and photonic link bandwidth of 10 Gb/s. If the data channel uses ω wavelengths, we need a total of $\omega + 2$ wavelengths in each unidirectional bus (1 for token channel, 1 for reservation channel, and ω for data channel). In Fig. 2, each ring shown in the ring matrix represents $\omega + 2$ rings (1 for token, 1 for reservation, and ω for data). Thus, each ring matrix has $(\omega + 2) \times 16$ rings when using four laser sources and four bus channels. Each access point has $\omega + 2$ rings for transmitter (1 filter ring for token, 1 modulator ring for reservation, and ω modulator ring for data) and $\omega + 1$ filter rings for receiver (1 for reservation and ω for data). Considering the nonlinearity limit of 30 mW per waveguide and photonic device losses listed in Table III, we need ω waveguides for each bus channel with the conservative waveguide design (3 dB/cm loss) or $\omega/6$ waveguides for each bus channel with the aggressive waveguide design (1 dB/cm loss). Our analysis in Section VI uses 32 wavelengths per channel ($\omega = 32$) that can meet the worst case bandwidth demands of NAS benchmarks. In that case, the tuning power overhead of the two ring matrices is 0.5 W and the area occupied by all the silicon photonic devices is less than 1.54% of the total chip area of 400 mm^2.

During a L1 cache miss or while sending other cache coherency messages, the L1 access point uses a token stream protocol to arbitrate for the data channel access for L1-to-L2 communication. In this protocol, a token per TDM slot is issued by the laser source. Fig. 3 shows the timing of token distribution, reservation request and data transmission of two L1-to-L2 buses. In this example, tokens, and effectively the TDM slots, are multiplexed onto the two buses using time-division multiplexing. To access the data channel, a L1 access point needs to grab a token from the token channel two cycles prior to the actual slot of data transmission. We use single-round token channels in which the L1 access point near the laser source has the highest priority to obtain the photonic tokens. Fairness can be pursued by using alternate token-based arbitration protocols [31], [43]. After grabbing the token, the L1 access point sends a reservation request by setting one out of the 4 bits that can be mapped on the reservation channel wavelength. Here, each bit corresponds to the destination L2 access point. Each L2 access point can only filter its associated bit on the reservation channel wavelength. It uses that bit to tune its ring filters that will filter the data received on the data channel in the following cycle. Only the L1 access point that has a token can use the associated reservation slot on the reservation channel. This notification over the reservation channel ensures that the destination L2 access point is ready

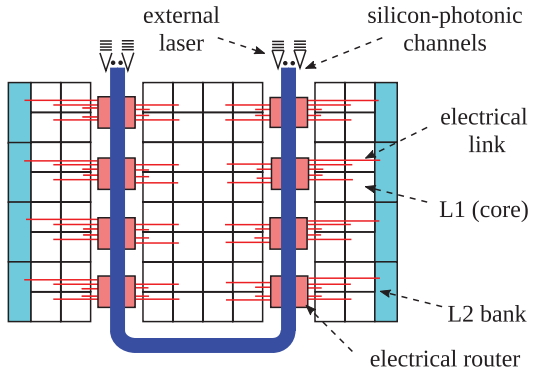


Fig. 4. Physical layout of the silicon-photonic Clos and butterfly NoC architecture. A concentration of eight cores (L1 caches) and 1 L2 bank is used at each router, and electrical links provide local communication between L1/L2 caches and routers. Each "electrical router" represents at least one injecting router and one ejecting router. In case of butterfly, the "silicon-photonic channels" represents 64 dedicated channels fully connecting the eight injecting routers to the eight ejecting routers. In case of Clos, κ of 8 "electrical router" represents extra κ middle routers. The "silicon-photonic channels" represents 64 dedicated channels fully connecting the eight injecting routers to the κ middle routers, and another 64 dedicated channels fully connecting the κ middle routers to the eight ejecting routers.

to receive data from the L1 access point in the following cycle (2 cycles after token is grabbed). In this following cycle, the L1 access point transmits the data to the L2 access point over the data channel. Each L2 bank has a dedicated access point for each bus. The laser power controller can decide the rate at which each bus receives photonic tokens. In Fig. 3, 16 photonic tokens are issued within a 16 cycle period, with a 50% of data channel utilization of each bus. We define the number of photonic tokens received by one bus in the 16 cycle period as bandwidth weight, which will be used in runtime laser power management (see in Section V). In this example, the bandwidth weight for each bus is 1/2. The same token-based weighted time-division multiplexed protocol is used for communication on the L2-to-L1 buses.

We compare our proposed multibus NoC architecture with silicon-photonic Clos and butterfly NoC architectures. Both these NoC architectures are well suited to be designed using silicon-photonic link technology. They both use smaller routers, but long global channels. They have less hop counts than mesh and at the same time do not need global arbitration like the crossbar. In addition, Clos provides extensive path diversity that can be used to minimize congestion in the NoC. We did not choose the mesh and crossbar NoC architecture for comparison as it has been shown previously that mesh and crossbar topologies are not suitable for silicon-photonic link technology [22]. For a fair comparison, we designed the physical layouts and chose the NoC parameters of the three topologies such that the total laser power consumption of the three topologies is the same. Fig. 4 shows the physical layout for the Clos and butterfly NoC architectures. A detailed comparison of the three topologies is presented in Section VI.

V. Laser Power Management

In this section, we describe our proposed runtime network reconfiguration methodology—joint application of weighted time-division multiplexing and switching ON/OFF laser sources, to reduce laser power while maintaining application performance.

A. Runtime Network Reconfiguration

For making network reconfiguration decisions on the L1-to-L2 network, each core group periodically sends its average packet latency calculated over a fixed time interval to the laser power controllers that are responsible for multiplexing the network bandwidth across the buses. This information about the average packet latency is used by the controller to decide on the new bandwidth weights (see more details in Section V-B) for each bus. The range and granularity of bandwidth weights depend on number of buses in the multibus NoC architecture and the desired level of control. For our target 64-core system, we have four buses and bandwidth weights range from 1/16 to 16/16, with a step size of 1/16. The precision of bandwidth weights could be increased for finer control. Depending on the bandwidth weights that are periodically calculated, the bandwidth is proportionally multiplexed across the buses at runtime. However, a simple round-robin arbitration does not work for laser power allocation if the four L1-to-L2 buses in our target system have different bandwidth weights. We use a proportional share resource algorithm similar to earliest eligible virtual deadline first [41] to assign the laser sources to buses according to their newly calculated bandwidth weights.

Using the new bandwidth weights that are required for each bus to sustain the performance of the various applications, the laser power controller determines the total L1-to-L2 bandwidth required across all buses. This total required bandwidth is in turn used to determine the total number of laser sources required in the system. In our case study, each laser source supplies the bandwidth equivalent to the baseline bandwidth of an individual bus channel. It uses 16 multiplexing slots corresponding to 16 consecutive clock cycles. At each time slot (or cycle), the light waves of the laser source are guided to the targeted L1-to-L2 bus. So, if the bandwidth weight of a bus is 16/16, we need to dedicate an entire laser source to that bus. The total number of laser sources can be calculated using

$$\text{Laser Source Number} = \left\lceil \sum_{i=0}^{N-1} W_i \right\rceil \tag{1}$$

where W_i is the bandwidth weight for the ith bus and N is the number of L1-to-L2 buses. For example, if the total number of laser sources is computed to be 3.5, and if only three laser sources are currently in use, then the laser power controller decides to switch ON one more laser source. It should be noted that in this example, half the bandwidth of the fourth laser source remains unutilized and gets wasted. We could uniformly distribute this unutilized bandwidth across all the buses to further improve the application performance. However, we use only the calculated bandwidth to ensure that the calculated bandwidth weights track the application bandwidth requirements. For example, if the laser power controller decides to reduce the bandwidth weight of a bus from $\alpha/16$ to $(\alpha-1)/16$, and the unutilized bandwidth is allocated to that bus resulting in an actual bandwidth weight of $\alpha/16$ for the new time interval, then

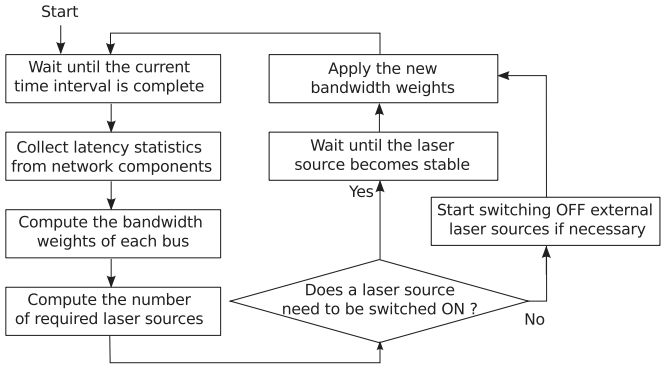


Fig. 5. Flowchart for runtime laser power management using weighted time-division multiplexing and switching ON/OFF laser sources.

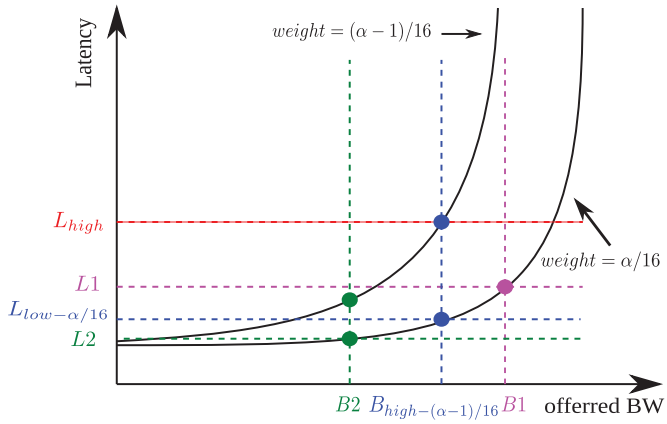


Fig. 6. Methodology to determine lower latency threshold for a bandwidth weight—The saturation bandwidth $B_{\text{high}-(\alpha-1)/16}$ for bandwidth weight $(\alpha-1)/16$ is projected on the latency-bandwidth plot for bandwidth weight of $\alpha/16$ to determine $L_{\text{low}-\alpha/16}$. We used uniform random traffic for generating this plot. For a bus with bandwidth weight of $\alpha/16$ if the calculated average latency is less than $L_{\text{low}-\alpha/16}$, then we can reduce bandwidth weight to $(\alpha-1)/16$ to save laser power without impacting performance.

the bandwidth weight will never go down to $(\alpha-1)/16$. As a result, the available network bandwidth may not necessarily track the required network bandwidth and can potentially result in unnecessary waste of laser power.

The ring matrices at the laser power controllers contain ring filters for each bus. These filters are tuned and detuned depending on the bandwidth weights associated with each bus. If the bandwidth weight for each bus needs to be updated but no new laser sources need to be switched ON, the entire network reconfiguration process takes less than 20 core clock cycles. On the other hand, if a new laser source needs to be switched ON, then we need to wait for a longer time for the laser source to switch ON and stabilize thermally. While the new laser source is stabilizing, the system does not stop execution and the laser power controllers use the older configuration to distribute the laser power across the bus channels. Fig. 5 shows the flowchart for the various steps involved in the network reconfiguration for laser power management. It should be noted that both bandwidth weight/laser number calculation block and ring filter tuning block can be implemented purely in hardware.

B. Bandwidth Weight Calculation for Weighted Time-Division Multiplexing

For runtime management, the bandwidth weight of each bus is determined and updated after every time interval. The bandwidth weight for each bus is calculated using the average packet latency for that bus over the previous time interval. We use a dual-threshold (L_{low} and L_{high}) approach for choosing the bandwidth weights. Here, if the average packet latency on a bus is greater than the upper threshold L_{high}, then the bandwidth weight of that bus is increased by $1/16$ to effectively increase the bus bandwidth and in turn reduce the average packet latency. The key idea here is to move the bus out of its saturation region by increasing the bus bandwidth and minimize the impact of the packet latency on the performance of the application running on the associated group of cores.

Similarly, if the average packet latency is smaller than the lower threshold L_{low}, then the associated bandwidth weight is reduced by 1/16 to decrease the bus bandwidth. This reduction in bandwidth saves laser power, with potentially minimal impact on the overall application performance. The value for L_{low} needs to be carefully chosen to ensure that after reducing the bandwidth weight, the bus does not become saturated in the next time interval. Fig. 6 shows our strategy for determining L_{low} based on the latency-bandwidth plots of our multibus NoC architecture for a uniform random traffic pattern. From the latency-bandwidth plot for a bus with bandwidth weight of $(\alpha-1)/16$, we can determine the bandwidth ($B_{\text{high}-(\alpha-1)/16}$) and the corresponding latency (L_{high}) beyond which the bus goes into saturation. By mapping this $B_{\text{high}-(\alpha-1)/16}$ onto the latency-bandwidth plot for bus with bandwidth weight of $\alpha/16$, we can determine the ideal lower threshold ($L_{\text{low}-\alpha/16}$) for bandwidth weight of $\alpha/16$. On a bus with bandwidth weight of $\alpha/16$ if the latency is less than $L_{\text{low}-\alpha/16}$, it would be safe to reduce bandwidth weight to $(\alpha-1)/16$ since we can guarantee the bus would not saturate if the traffic pattern does not change. This approach can be used to determine the lower threshold for each bandwidth weight. In Fig. 6, if the bus with bandwidth weight of $\alpha/16$ has an average packet latency of $L1$ ($>L_{\text{low}-\alpha/16}$), then the laser power controller should not reduce the bandwidth weight to $(\alpha-1)/16$, as the bus would get into the saturation region. If the bus with bandwidth weight of $\alpha/16$ has an average packet latency of $L2$ ($<L_{\text{low}-\alpha/16}$), then the laser power controller can safely reduce the bandwidth weight to $(\alpha-1)/16$ since the resulting increased latency would not exceed L_{high}.

We use the latency-bandwidth plot for random uniform traffic pattern to calculate the lower threshold L_{low} for runtime management. Table IV shows the calculated L_{low} for each bandwidth weight while setting L_{high} as 10, 15, 20, 30, and 50 cycles. For our analysis in Section VI, the laser power controller can choose L_{low} by mapping the current bandwidth weight and predefined L_{high} on the Table IV to find the corresponding L_{low} threshold. It should be noted that the latency threshold value is a function of the NoC architecture, and hence for each other NoC architecture, the latency thresholds need to be separately determined using synthetic traffic patterns.

TABLE IV
L_{Low} FOR ALL BANDWIDTH WEIGHT AT VARIOUS L_{High}

Weight	$\frac{2}{16}$	$\frac{3}{16}$	$\frac{4}{16}$	$\frac{5}{16}$	$\frac{6}{16}$	$\frac{7}{16}$	$\frac{8}{16}$	$\frac{9}{16}$	$\frac{10}{16}$	$\frac{11}{16}$	$\frac{12}{16}$	$\frac{13}{16}$	$\frac{14}{16}$	$\frac{15}{16}$	$\frac{16}{16}$
$L_{high} = 10$	X	X	X	X	X	X	X	X	X	X	X	9.4	9.4	9.3	9.5
$L_{high} = 15$	X	X	X	X	X	13.7	13.3	13.2	13.0	13.1	13.4	13.0	13.2	13.0	12.8
$L_{high} = 20$	X	X	X	X	16.5	16.7	16.5	16.4	16.3	16.4	15.8	16.1	16.0	15.6	15.8
$L_{high} = 30$	X	X	22.4	22.1	21.9	20.9	22.1	20.1	21.5	20.8	20.3	20.4	19.2	18.0	19.6
$L_{high} = 50$	X	29.3	29.4	29.1	28.7	27.8	27.4	26.4	29.1	27.3	25.9	25.4	27.8	28.4	24.7

X–means it is not possible to further reduce bandwidth weight, otherwise the serialization latency would be more than L_{high}.

VI. EVALUATION

In this section, we first provide an overview of our evaluation platform followed by a detailed discussion of the reduction in laser power of a 64-core system through communication-driven runtime laser power management. We compare our multibus NoC architecture with Clos and butterfly NoC architectures, and then investigate the impact of different reconfiguration thresholds and reconfiguration time intervals on the overall system performance and laser power consumption when using the laser power management policy.

A. Evaluation Platform

To evaluate the impact of our runtime laser power management approach, we integrated our network model into Gem5 full-system simulator [10]. Gem5 is an event-based manycore simulator that uses Alpha instruction set architecture (ISA). Our network model is cycle-accurate, and to ensure accurate simulation of the entire manycore system we added a packet exchange interface for handling interactions between the network model and the Gem5 simulator.

Our 64-core target system has a directory-based cache coherency protocol. Since the Gem5 simulator uses broadcast-based cache coherency protocol, we modified the simulator to trap all broadcast-based cache coherency operations and emulate the corresponding directory-based cache coherency operations. For example, if we trap a L1-to-L1 data response on a L2 cache miss in the broadcast-based cache coherency protocol, we translate it into a sequence of four consecutive network operations in directory-based cache coherency protocol. It starts with a request packet from the core with the L1 cache miss to the directory of the associated L2 bank, followed by a request packet from the directory to the core that has the missing cache line. After receiving the data response packet from the core having the missing cache line, the directory forwards the data to the original requester. This packet sequence is used to model the operations among the core with the L1 cache miss, the directory at the L2 bank and the L1 cache that has the missing cache line. Similarly, network operations corresponding to other directory-based operations such as upgrade request and data response from L2 are also trapped and emulated. Our emulation methodology approaches the real timing overhead of cache miss and the overall network traffic loads in a cache hierarchy with directory-based cache coherency protocol. In addition, our trap and emulation method is independent of the proposed multibus NoC architecture and laser power management technique as the variations in the network bandwidth requirements across benchmarks are maintained.

In the full system simulation using Gem5, we run NAS parallel benchmarks (`cg`, `ep`, `ft`, `is`, `lu`, `mg`, `sp` and `ua`) with class B problem sets [5]. We use a warm-up period of 2 billion instructions to get past the initialization phase and avoid cold-start effects. We execute each application for 100 ms with network reconfiguration at fixed time intervals (10 μs, 100 μs, 1 ms and 10 ms). We use application instruction committed per cycle (IPC) as the metric to prove that our runtime laser management has minimal impact on the system performance.

B. Evaluation Results

Fig. 7 shows a comparison of the power consumption and performance of various NoC architectures with the same laser power budget. We first determine the laser power consumption for a multibus having 32λ per channel and use that value as the laser power budget for other NoC architectures. The choice of 32λ is discussed later in this section. The channel width for other NoC architectures was determined using the same laser power budget of the entire network. For Clos, we consider two architectures—one with two middle routers (8λ per channel) and one with four middle routers (4λ per channel). The butterfly has a channel width of 4λ. Both Clos and butterfly use a concentration of 9 (8 L1 caches and 1 L2 cache bank), which requires local electrical links for the communications between L1/L2 caches and network access points. For Fig. 7, we assume the conservative transmitter/receiver circuits and thermal tuning circuits in Table II and the conservative waveguide design (3 dB/cm loss) in Table III. The laser power consumption is more than 24 W with these conservative assumptions. For the aggressive design (1 dB/cm loss), the laser power consumption is close to 5 W. In both cases, the the laser power consumption is dominant in the whole network and therefore the runtime laser power management is necessary. With the same laser power consumptions, the multibus topology achieves better performance than Clos and butterfly because it has the lowest serialization latencies due to the wider bus width. At the same time, it has better bandwidth utilization than other NoC architectures as it shares bandwidth among multiple network access points, while both Clos and butterfly use dedicated channels between routers. Among the two Clos architectures, the architecture with two intermediate routers exhibits better performance than the architecture with four intermediate routers due to lower serialization

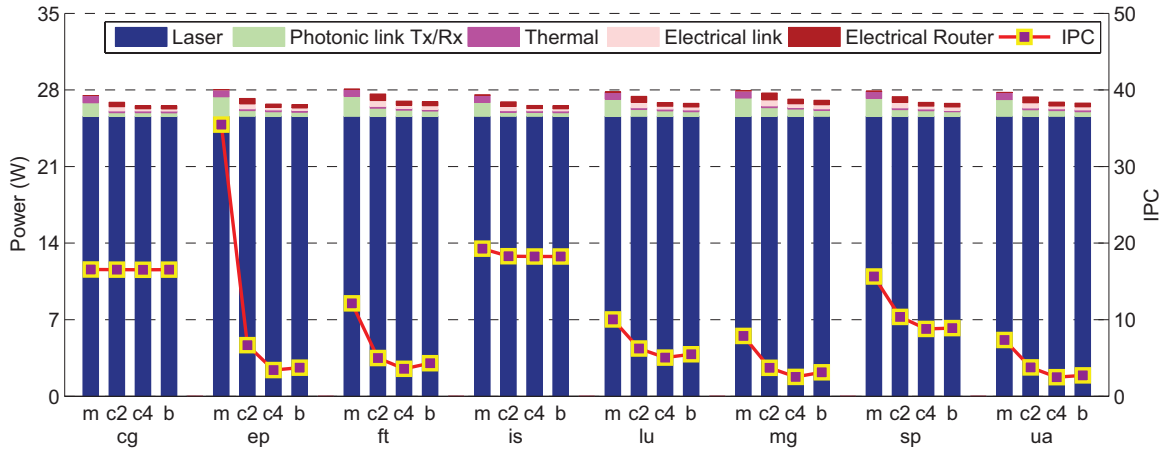


Fig. 7. Power and Performance (IPC) for various NoC architectures with the same laser power budget—Here "m" = multibus, "c2" = Clos with two middle routers, "c4" = Clos with four middle routers and "b" = butterfly. These four NoC architectures have 32λ, 8λ, 4λ, and 4λ per channel, respectively, to meet the same laser power budget. We assume the conservative design from Tables II and III.

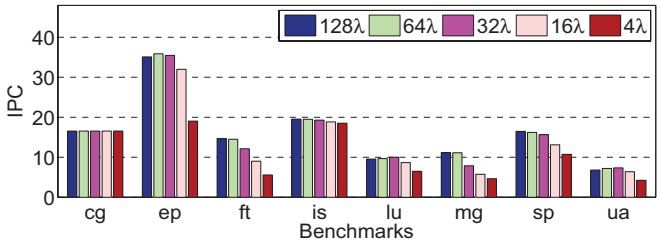


Fig. 8. Performance (IPC) for various baseline bandwidth—The ideal baseline bandwidth is between 32λ and 64λ. We choose 32λ conservatively to avoid the overprovisioning of the multibus NoC architecture.

latency. The butterfly and Clos with four intermediate routers exhibit similar performance due to same channel widths. The multibus achieves more than two times higher IPC on average than Clos and butterfly for NAS benchmarks.

To evaluate our laser power management policy using the multibus NoC architecture, we first determine its baseline bandwidth. Fig. 8 shows the impact of choice of baseline bandwidth on the overall system performance in terms of IPC. The baseline bandwidth corresponds to the case where all the laser sources required for the network are always ON, i.e., it corresponds to the bandwidth weights of 16/16. The baseline bandwidth should be chosen such that it provides the minimum bandwidth that would meet the worst case bandwidth demands of all applications, i.e., not limit the overall system performance. Fig. 8 shows that `ft` and `mg` benchmarks require the highest NoC baseline bandwidth to sustain the system performance. Their performance decreases when the baseline bandwidth reduces below 64λ per channel, which indicate that the ideal baseline bandwidth could be between 64λ and 32λ per channel. We use a conservative baseline bandwidth of 32λ per channel for the evaluation of our runtime management technique.

Fig. 9 shows the impact of reconfiguration threshold L_{high} and corresponding L_{low} on the system performance and laser power consumptions. The IPC values of each benchmark are normalized to the IPC of the same benchmark running on the same system without runtime management. The laser controller

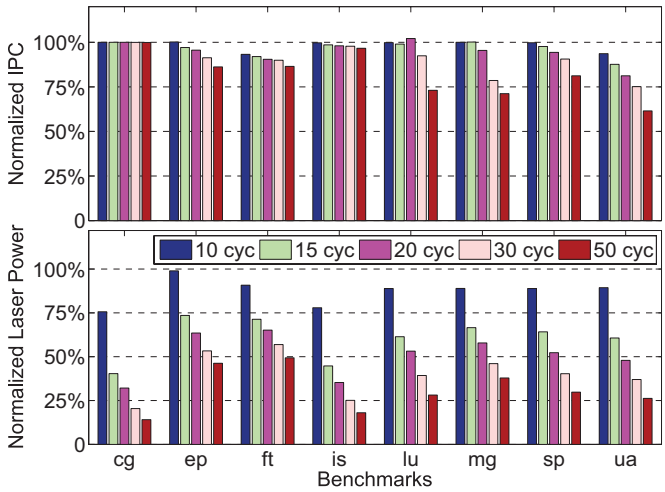


Fig. 9. Performance (IPC) and bandwidth weight for various reconfiguration threshold L_{high} in the multibus NoC architecture—Here, the baseline bandwidth is 32λ per channel and the reconfiguration time interval is 100 μs. The performance is normalized to the performance of each benchmark running on the system without runtime management (as shown in Figure 7). The laser power is normalized to the baseline design that has laser sources switched ON all the time. The L_{high} and the corresponding L_{low} from Table IV have a significant impact on the system performance and laser power consumption (proportional to the average bandwidth weight).

uses various L_{high} and L_{low} (see Table IV) to make the decision of increasing and reducing the bandwidth weight, respectively. Fig. 9 shows that when the L_{high} increases, the performance (IPC) decreases and laser power consumption (proportional to the average bandwidth weight) decreases. This behavior is observed because a higher L_{high} makes it difficult for the controller to make the decision of increasing the bandwidth weight. A higher L_{low} (due to higher L_{high}) is preferable from the perspective of saving laser power consumption. However, a L_{high} higher than 20 cycles starts having a strong impact of the system performance, especially on `mg` benchmark. Thus, we choose to use $L_{\text{high}} = 20$ cycles for the NAS benchmarks.

Fig. 10 shows the impact of the reconfiguration time interval on the system performance and laser power consumption. For

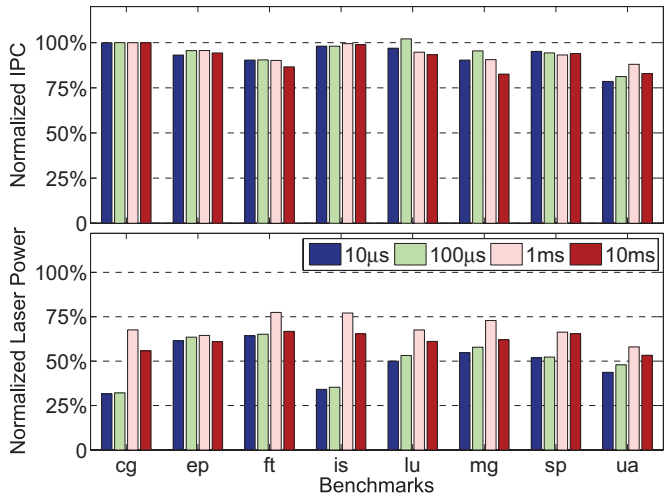


Fig. 10. Performance (IPC) and bandwidth weight for various reconfiguration time interval—The baseline bandwidth is 32λ per channel with 16/16 bandwidth weight and reconfiguration threshold L_{high} is 20 cycles. The performance is normalized to the performance of each benchmark running on the system without runtime management (as shown in Figure 7). The laser power is normalized to the baseline design that has laser sources switched ON all the time.

the projected laser source stabilization values ranging from 1 μs to 10 ms, we consider a reconfiguration time intervals that are ten times the stabilization times, i.e., from 10 μs to 100 ms to minimize the effect of laser source stabilization. Fig. 10 shows that with a shorter reconfiguration time interval (10 μs $\sim$ 100 μs), the average bandwidth weight (i.e., laser power) is much lower than for longer reconfiguration time interval (1 ms $\sim$ 10 ms). This is because a shorter reconfiguration time interval allows the laser power controller to quickly adjust the bandwidth weight based on the changes in the network traffic. As the reconfiguration time intervals increase, there is a slower response to changes in network traffic, which may result in the waste of laser power. This slower response to changes in network traffic can also impact system performance. For the 10-ms reconfiguration time interval, the average bandwidth weight does not match with the general trend of our simulation results as the initial bandwidth weight (during the start of the simulation) is chosen as 8/16, and our total simulation time of 100 ms does not provide the enough time to for warm up.

Fig. 11 shows the performance, network traffic, and bandwidth weight variations across time, with and without runtime management. Here, we use a reconfiguration threshold $L_{\mathrm{high}} =$ 20 cycles and a sampling interval of 100 μs. The left column corresponds to the performance and network traffic variations in the multibus with static bandwidth allocation (16/16), i.e., baseline bandwidth. The temporal variations of network traffic for each individual benchmarks and the spatial variations across benchmarks provide opportunities for reducing the laser power consumption through runtime management. The right column shows the performance and bandwidth weight variations after applying our runtime management technique. There is no significant change in the performance trace after applying the runtime management technique. The bandwidth weight tracks the variation of network traffic, and reduces the laser power consumption. The rate of bandwidth weight variation strongly depends on the variation of network traffic. For example, benchmarks like `mg`, `sp`, and `ua` benchmarks show frequent bandwidth weight changes as the network traffic changes frequently. On the other hand, benchmarks like `ep`, `ft`, and `lu` show much smooth changes in bandwidth weight since the network traffic remains constant for extended periods across the time. Benchmarks like `cg` and `is` do not show significant variations in bandwidth weight since the network traffic remains constant for most time.

The previous analysis shows that the baseline bandwidth of 32λ per channel can meet the worst case bandwidth demands of NPBs. The application of our laser power management technique provides more than 49% savings in laser power on an average without significant impact on system performance (less than 6% on average). Our approach saves more than 12.4 W given the conservative waveguide design (3 dB/cm loss), and more than 2.4 W given the aggressive waveguide design (1 dB/cm loss).

VII. Discussion

In this section, we qualitatively discuss the limits and opportunities for application of our proposed multibus NoC architecture and the laser power management technique.

A. Large Core Counts

For our analysis, we used a target system having 64 cores with 16 cores in adjacent columns sharing two unidirectional buses. Future many-core processors will have thousands of cores on a single die. Assuming corresponding progress in the area of parallel algorithms and programming, these large many-core processor systems could produce several times larger network traffic and require higher network bandwidth. Our multibus NoC architecture exhibits good scalability to larger core counts. We could either increase the number of buses, or increase the concentration at each bus access points and balance the energy consumed in the electrical and photonic links. For example, for a 1024-core processor, we could use 32 unidirectional bus channels with 64 cores in adjacent core columns sharing two unidirectional bus channels. The baseline bandwidth of each bus channel will need to be increased to match the increased network traffic. If we increase the network concentration, a local electrical network is needed to provide communication between cores and bus access points. The use of local electrical network could help increase the utilization of each bus access points, and in turn improve the energy efficiency of the entire network.

B. Simultaneously Executing Applications

Most of the legacy applications do not scale well to large core counts. Hence, the OS will need to have the capability to execute multiple applications simultaneously to support legacy applications. For example, if we have four applications and each application uses 16 threads with a thread per core, we could map each application onto a 16-core group. Our management policy can easily be applied to such situations. If these applications

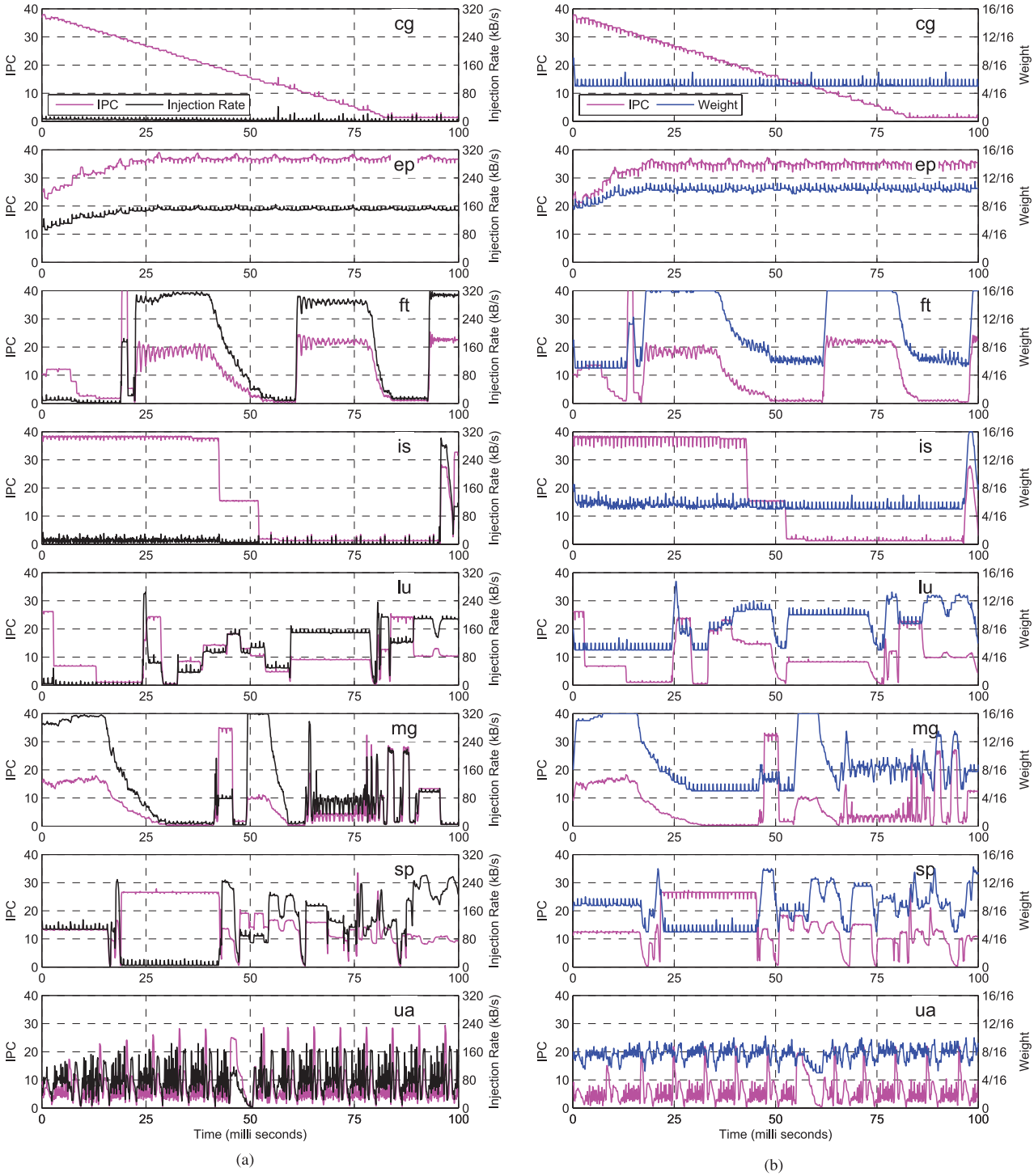


Fig. 11. Performance (IPC), network traffic, and bandwidth weight tracing for various NAS benchmarks in the multibus NoC architecture—The left column shows the performance and network traffic on a 64-core system with static bandwidth allocation, i.e., bandwidth weight maintains 16/16 and no runtime laser power management is used. The right column shows the performance and bandwidth weight while applying our runtime laser power management technique. Here, the baseline bandwidth is 32λ per channel, the reconfiguration threshold L_{high} is 20 cycles and the reconfiguration time interval is 100 μs. The initial bandwidth weight for each bus is 8/16. The runtime management technique allows the laser power controllers to dynamically allocate the bandwidth according to the temporal and spatial variations in the bandwidth demands. (a) Multibus without runtime reconfiguration. (b) Multibus with runtime reconfiguration.

exhibit any variations in the network bandwidth demands, our policy can easily increase/decrease the network bandwidth of each individual bus based on the bandwidth demands of each application. This would help maximize the energy efficiency of the on-chip network as well as the many-core system as a whole.

C. Distributed L2 Versus Private L2 Cache

The use distributed L2 cache versus private L2 cache for many-core architectures has been widely explored. Both approaches have their advantages and disadvantages. The

distributed L2 cache enables good cache line sharing across the large number of cores, but at the same time it has higher L1 miss penalty and requires large amount of intrachip communication for maintaining coherency. On the other hand, the private L2 cache architecture has low L2 hit latency, but the cache line sharing and synchronization is difficult. Moreover, the private L2 cache could have higher miss rates resulting in expensive off-chip memory accesses. Our multibus NoC architecture and laser management technique can be readily applied to both cache architectures. The multibus NoC architecture will provide L1 to L2 communication for distributed L2 cache architecture and L2 to memory controller communication for private L2 cache architecture. The key step is choosing the correct baseline design (bus bandwidth) for the multibus NoC such that the system can support the worst case network traffic. We could pursue an integrated solution where we jointly design the L2 cache architecture and silicon-photonic multibus NoC architecture to maximize the energy efficiency.

D. Alternate NoC Architectures

Our proposed laser power management policy is relatively agnostic of the underlying NoC architecture. It can be applied to NoC architectures like butterfly, Clos, and crossbar. In these NoC architectures, we would expect to see laser power savings at a similar scale as the multibus NoC architecture. It should, however, be noted that multibus NoC architecture provides better performance than these NoC architectures at same laser power consumption, and hence would have better energy efficiency. The laser power management policy could also be applied to silicon-photonic implementation of low-radix high-diameter NoC architectures like torus and mesh. The large channel count and distributed nature of these NoC architectures would however lead to significant overhead.

VIII. Related Work

For intrachip communication, researchers have explored silicon-photonic implementations of the entire spectrum of network topologies. The large number of global buses needed for the high-radix low-diameter crossbar that provide nonblocking connectivity can be efficiently implemented using silicon-photonics technology [23], [38], [44]. The silicon-photonic implementation of low-radix high-diameter networks like mesh and torus lying at the other end of the network spectrum have also been investigated [12], [24], [34], [39]. Silicon-photonic designs of intermediate network topologies like Clos and fat-tree that offer the same network guarantees like the global crossbar with potentially lower resource requirements have also been explored [16], [22], [32]. A general consensus among the various efforts so far is that silicon-photonic networks provide a bandwidth density and data-dependent energy advantage for NoC communication. However, the fixed amount of power consumed in the laser sources that drive these networks negates these advantages. Hence, to enable the use of silicon-photonic NoC in future many-core systems, we need to develop techniques to proactively manage laser power.

At the device level, standard design-time solutions to reduce optical loss in silicon-photonic devices and in turn reduce the laser power range from exploring different materials to process flows to device geometries. At the circuit level, we can explore the design of receivers that can operate with low-sensitivity photodetectors or use photonic devices that have lower losses but are more susceptible to noise, and use error detection/correction techniques to tackle any errors. At the architecture level, a nanophotonic crossbar architecture that uses optical channel sharing to manage static power dissipation is proposed in [31]. Here a token-stream mechanism is used for channel arbitration and credit distribution, to enable efficient global sharing of crossbar channels. Similarly, a reconfigurable photonic network for board-to-board communication is proposed in [25] for improving performance and reducing power. Here, depending on the network traffic, idle channels are reallocated to busy channels to improve performance, and bit rate and supply voltages of individual channels are regulated to manage power.

Our study specifically targets the on-chip silicon-photonic network between the private L1 cache and distributed L2 cache. We time-division multiplex the photonic bandwidth output from the laser source across all the channels based on weights that change at runtime to maximize the many-core system performance. At the same time, we also explore the opportunities to switch ON/OFF the laser source (i.e., reduce the net bandwidth of the network) to further reduce laser power. The ultimate goal is to maximize the overall execution efficiency of the many-core system. The decisions on the magnitude of change of the multiplexing weights and network bandwidth (through switching ON/OFF of laser sources) are made based on the average network packet latency for an application over fixed sampling intervals. Our technique ensures that the application runs at peak performance while consuming minimum amount of laser power. We have also proposed a multibus NoC architecture that is well suited to the proposed weighted time-division multiplexing technique and have presented a head-to-head comparison of this multibus NoC architecture with conventional Clos and butterfly NoC architecture. A time-division multiplexed arbitration technique for silicon-photonic mesh NoC is proposed in [18]. In contrast to our runtime approach, here the time division-multiplexed photonic paths between the various pairs of network access points are established statically during design time and do not change at runtime. The key idea is to provide complete network connectivity, with each pair of access points getting fair access to large network bandwidth.

IX. Conclusion

Silicon-photonic link technology is expected to replace electrical link technology in intrachip and interchip communication networks in future many-core processors. However, the large laser power consumption in these silicon-photonic networks is limiting their widespread adoption. In this paper, we propose a multibus NoC architecture for a many-core processor and a runtime technique that dynamically manages the laser power of this multibus NoC depending on the communication bandwidth

requirements of the various applications running on the many-core processor. For a silicon-photonic multibus NoC between the private L1 and distributed L2 caches, we propose a policy that uses weighted time-division multiplexing with token-stream flow control and switching ON/OFF laser sources as necessary, to maximize the total energy efficiency of the many-core processor. For a 64-core processor running the NPB suite, we get an average of more than 49% reduction in laser power, with a 6% reduction in application performance. The proposed technique can potentially pave the way for early adoption of silicon-photonic link technology in future many-core processors.

REFERENCES

[1] The 50G silicon photonics link. White Paper, Intel Labs Jul. 2010.

[2] S. Assefa, W. M. Green, A. Rylyakov, C. Schow, F. Horst, and Y. Vlasov, "CMOS integrated nanophotonics: Enabling technology for exascale computing systems," in *Opt. Fiber Commun. Conf.*, Optical Society of America, Mar. 2011, p. OMM6.

[3] S. Assefa, F. Xia, S. W. Bedell, Y. Zhang, T. Topuria, P. M. Rice, and Y. A. Vlasov, "CMOS-integrated high-speed MSM germanium waveguide photodetector," *Opt. Exp.*, vol. 18, no. 5, pp. 4986–4999, Mar. 2010.

[4] T. Baehr-Jones, M. Hochberg, C. Walker, and A. Scherer, "High-Q ring resonators in thin silicon-on-insulator," *Appl. Phys. Lett.*, vol. 85, no. 16, pp. 3346–3347, Oct. 2004.

[5] D. Bailey, E. Barszcz, J. Barton, D. Browning, R. Carter, L. Dagum, R. Fatoohi, S. Fineberg, P. Frederickson, T. Lasinski, R. Schreiber, H. Simon, V. Venkatakrishnan, and S. Weeratunga, "The NAS parallel benchmarks," Tech. Rep. RNR-94-007, Mar. 1994.

[6] T. Barwicz, H. Byun, F. Gan, C. W. Holzwarth, M. A. Popovic, P. T. Rakich, M. R. Watts, E. P. Ippen, F. X. Kärtner, H. I. Smith, J. S. Orcutt, R. J. Ram, V. Stojanovic, O. O. Olubuyide, J. L. Hoyt, S. Spector, M. Geis, M. Grein, T. Lyszczarz, and J. U. Yoon, "Silicon photonics for compact, energy-efficient interconnects (invited)," *J. Opt. Netw.*, vol. 6, no. 1, pp. 63–73, Jan. 2007.

[7] C. Batten, A. Joshi, J. Orcutt, C. Holzwarth, M. Popovic, J. Hoyt, F. Kartner, R. Ram, V. Stojanovic, and K. Asanovic, "Building manycore processor-to-DRAM networks with monolithic CMOS silicon photonics," *Micro, IEEE*, vol. PP, no. 99, p. 1, 2009.

[8] S. Beamer, C. Sun, Y.-J. Kwon, A. Joshi, C. Batten, V. Stojanović, and K. Asanović, "Re-architecting DRAM memory systems with monolithically integrated silicon photonics," in *Proc. 37th Annu. Int. Symp. Comput. Archit., ISCA'10*, New York, NY, Jun. 2010, pp. 129–140.

[9] A. Biberman, N. Sherwood-Droz, X. Zhu, K. Preston, G. Hendry, J. Levy, J. Chan, H. Wang, M. Lipson, and K. Bergman, "Photonic network-on-chip architecture using 3D integration," in *Proc. SPIE* vol. 7942, p. 79420M, 2011.

[10] N. Binkert, R. Dreslinski, L. Hsu, K. Lim, A. Saidi, and S. Reinhardt, "The M5 simulator: Modeling networked systems," *Micro, IEEE*, vol. 26, no. 4, pp. 52–60, Jul./Aug. 2006.

[11] L. Chen, K. Preston, S. Manipatruni, and M. Lipson, "Integrated GHz silicon photonic interconnect with micrometer-scale modulators and detectors," *Opt. Exp.*, vol. 17, no. 17, pp. 15248–15256, Aug. 2009.

[12] M. J. Cianchetti, J. C. Kerekes, and D. H. Albonesi, "Phastlane: a rapid transit optical routing network," in *Proc. 36th Annu. Int. Symp. Comput. Archit., ISCA'09*, New York, NY, Jun. 2009, pp. 441–450.

[13] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. San Francisco, CA: Morgan Kaufmann Publishers Inc, 2003.

[14] M. Gnan, S. Thorns, D. Macintyre, R. De La Rue, and M. Sorel, "Fabrication of low-loss photonic wires in silicon-on-insulator using hydrogen silsesquioxane electron-beam resist," *Electron. Lett.*, vol. 44, no. 2, pp. 115–116, Jan. 2008.

[15] A. Gorin, A. Jaouad, E. Grondin, V. Aimez, and P. Charette, "Fabrication of silicon nitride waveguides for visible-light using PECVD: A study of the effect of plasma frequency on optical properties," *Opt. Exp.*, vol. 16, no. 18, pp. 13509–13516, Sep. 2008.

[16] H. Gu, J. Xu, and W. Zhang, "A low-power fat tree-based optical networkon-chip for multiprocessor system-on-chip," in *Proc. Des. Autom. Test Eur. Conf. Exhibition, 2009. DATE'09*, Apr., pp. 3–8.

[17] C. Gunn, "CMOS photonics for high-speed interconnects," *Micro. IEEE*, vol. 26, no. 2, pp. 58–66, Mar./Apr. 2006.

[18] G. Hendry, E. Robinson, V. Gleyzer, J. Chan, L. P. Carloni, N. Bliss, and K. Bergman, "Time-division-multiplexed arbitration in silicon nanophotonic networks-on-chip for high-performance chip multiprocessors," *J. Parallel Distrib. Comput.*, vol. 71, no. 5, pp. 641–650, May 2011.

[19] C. W. Holzwarth, J. S. Orcutt, H. Li, M. A. Popovic, V. Stojanovic, J. L. Hoyt, R. J. Ram, and H. I. Smith, "Localized substrate removal technique enabling strong-confinement microphotonics in bulk Si CMOS processes," in *Proc. Lasers Electro-Optics/Quantum Electron. Laser Sci. Conf. Photonic Appl. Syst. Technol.*, Optical Society of America, May 2008, p. CThKK5.

[20] E. S. Hosseini, S. Yegnanarayanan, A. H. Atabaki, M. Soltani, and A. Adibi, "High quality planar silicon nitride microdisk resonators for integratedphotonics in the visible wavelength range," *Opt. Exp.*, vol. 17, no. 17, pp. 14543–14551, Aug. 2009.

[21] J. Howard, S. Dighe, Y. Hoskote, S. Vangal, D. Finan, G. Ruhl, D. Jenkins, H. Wilson, N. Borkar, G. Schrom, F. Pailet, S. Jain, T. Jacob, S. Yada, S. Marella, P. Salihundam, V. Erraguntla, M. Konow, M. Riepen, G. Droege, J. Lindemann, M. Gries, T. Apel, K. Henriss, T. Lund-Larsen, S. Steibl, S. Borkar, V. De, R. Van Der Wijngaart, and T. Mattson, "A 48-core IA-32 message-passing processor with DVFS in 45nm CMOS," in *Proc. IEEE Int. Solid-State Circuits Conf. Digest Tech. Papers*, Feb. 2010, pp. 108–109.

[22] 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 *Proc. 3rd ACM/IEEE Int. Symp. Netw.-on-Chip, NOCS'09*, Washington, DC, USA, May 2009, pp. 124–133.

[23] N. Kirman, M. Kirman, R. Dokania, J. Martinez, A. Apsel, M. Watkins, and D. Albonesi, "Leveraging optical technology in future bus-based chip multiprocessors," in *Proc. 39th Annu. IEEE/ACM Int. Symp. Microarchit.*, Dec. 2006, pp. 492–503.

[24] N. Kirman and J. F. Martínez, "A power-efficient all-optical on-chip interconnect using wavelength-based oblivious routing," in *Proc. Fifteenth Edition ASPLOS Archit. Support Program. Lang. Operating Syst., ASPLOS '10*, New York, NY, Mar. 2010, pp. 15–28.

[25] A. Kodi and A. Louri, "Energy-efficient and bandwidth-reconfigurable photonic networks for high-performance computing (HPC) systems," *IEEE J. Sel. Top. Quantum Electron.*, vol. 17, no. 2, pp. 384–395, Mar./Apr. 2011.

[26] K. Kuhn, M. Liu, and H. Kennel, "Technology options for 22nm and beyond," in *Proc. Int. Workshop Junction Technol. (IWJT)*, May 2010, pp. 1–6.

[27] R. Morris and A. Kodi, "Power-efficient and high-performance multilevel hybrid nanophotonic interconnect for multicores," in *Proc. 4th ACM/IEEE Int. Symp. Netw.-on-Chip, NOCS'10*, May 2010, pp. 207–214.

[28] J. Orcutt, A. Khilo, M. Popovic, C. Holzwarth, B. Moss, H. Li, M. Dahlem, T. Bonifield, F. Kartner, E. Ippen, J. Hoyt, R. Ram, and V. Stojanovic, "Demonstration of an electronic photonic integrated circuit in a commercial scaled bulk CMOS process," in *Proc. Lasers Electro-Optics/Quantum Electron. Laser Sci. Conf. CLEO/QELS 2008*, May 2008, pp. 1–2.

[29] J. S. Orcutt, A. Khilo, C. W. Holzwarth, M. A. Popovíc, H. Li, J. Sun, T. Bonifield, R. Hollingsworth, F. X. Kärtner, H. I. Smith, V. Stojanovíc, and R. J. Ram, "Nanophotonic integration in state-of-the-art CMOS foundries," *Opt. Exp.*, vol. 19, no. 3, pp. 2335–2346, Jan. 2011.

[30] R. M. Osgood, Jr., R. L. Espinola, J. I. Dadap, S. J. McNab, Y. A. Vlasov, "Silicon nanowire active integrated optics," *Proc. SPIE*, vol. 5729, pp. 110–117, Jan. 2005.

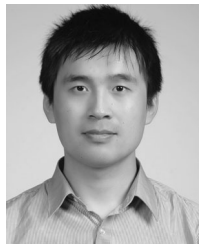
[31] Y. Pan, J. Kim, and G. Memik, "Flexishare: Channel sharing for an energy-efficient nanophotonic crossbar," in *Proc. 36th Annu. Int. High Perform. Comput. Archit. Symp.*, Jan. 2010, pp. 1–12.

[32] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A. Choudhary, "Firefly: illuminating future network-on-chip with nanophotonics," in *Proc. 36th Annu. Int. Symp. Comput. Archit., ISCA'09*, New York, NY, Jun. 2009, pp. 429–440.

[33] M. S. Papamarcos and J. H. Patel, "A low-overhead coherence solution for multiprocessors with private cache memories," in *Proc. 11th Annu. Int. Symp. Comput. Archit., ISCA'84*, New York, NY, 1984, pp. 348–354.

[34] M. Petracca, B. Lee, K. Bergman, and L. Carloni, "Design exploration of optical interconnection networks for chip multiprocessors," in *Proc. 16th IEEE Symp. High Perform. Interconnects., HOTI'08*, Aug. 2008, pp. 31–40.

[35] K. Preston and M. Lipson, "Slot waveguides with polycrystalline silicon for electrical injection," *Opt. Exp.*, vol. 17, no. 3, pp. 1527–1534, Feb. 2009.
[36] K. Preston, S. Manipatruni, C. Poitras, and M. Lipson, "2.5 Gbps electrooptic modulator in deposited silicon," in *Proc. Lasers Electro-Optics/Quantum Electron. Laser Sci. Conf. CLEO/QELS 2009. Conf.*, Jun., pp. 1–2.
[37] K. Preston, B. Schmidt, and M. Lipson, "Polysilicon photonic resonators for large-scale 3D integration of optical networks," *Opt. Exp.*, vol. 15, no. 25, pp. 17283–17290, Dec. 2007.
[38] J. Psota, J. Eastep, J. Miller, T. Konstantakopoulos, M. Watts, M. Beals, J. Michel, K. Kimerling, and A. Agarwal, "ATAC: On-chip optical networks for multicore processors," *Boston Area Archit. Workshop*, pp. 107–108, Jan. 2007.
[39] A. Shacham, K. Bergman, and L. Carloni, "On the design of a photonic network-on-chip," in *Proc. 1st ACM/IEEE Int. Symp. Netw.-on-Chip, NOCS'07*, May 2007, pp. 53–64.
[40] S. Sridaran and S. A. Bhave, "Nanophotonic devices on thin buried oxide silicon-on-insulator substrates," *Opt. Exp.*, vol. 18, no. 4, pp. 3850–3857, Feb. 2010.
[41] I. Stoica, H. Abdel-Wahab, K. Jeffay, S. Baruah, J. Gehrke, and C. Plaxton, "A proportional share resource allocation algorithm for realtime, time-shared systems," in *Proc. 17th IEEE Real-Time Syst. Symp.*, Dec. 1996, pp. 288–299.
[42] S. Thoziyoor, N. Muralimanohar, J.-H. Ahn, and N. P. Jouppi, "CACTI 5.1," Tech. Rep. HPL-2008-20, HP Labs, 2008.
[43] D. Vantrease, N. Binkert, R. Schreiber, and M. Lipasti, "Light speed arbitration and flow control for nanophotonic interconnects," in *Proc. 42nd Annu. IEEE/ACM Int. Symp. Microarchit.*, Dec. 2009, pp. 304–315.
[44] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. Jouppi, M. Fiorentino, A. Davis, N. Binkert, R. Beausoleil, and J. Ahn, "Corona: System implications of emerging nanophotonic technology," in *Proc. 35th Annu. Int. Symp. Comput. Archit., ISCA'08*, Jun. 2008, pp. 153–164.
[45] Q. Xu, B. Schmidt, S. Pradhan, and M. Lipson, "Micrometre-scale silicon electro-optic modulator," *Nature*, vol. 435, no. 7040, pp. 325–327, May 2005.



Chao Chen (S'10) received the B.Eng. degree in electronics engineering and B.Econ. degree in international trade and business from Shanghai Jiao Tong University, Shanghai, China, in 2005 and 2006, respectively, and the M.S. degree in electrical engineering from the Pohang University of Science and Engineering, Pohang, South Korea, in 2007. He has been working toward the Ph.D. degree in electrical and computer engineering at Boston University, Boston, MA, since 2009.

From 2007 to 2009, he was an SOC Engineer at Core Logic, Inc., Seoul, Korea, where he was mainly involved in designing multimedia processors for mobile applications. His research interests include reconfigurable network architectures and silicon-photonic technology for on-chip and off-chip communications. His research interests also include embedded system design and field-programmable gate array implementations for scientific applications.

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

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Boston University, Boston, MA. Prior to joining Boston University, he was a Postdoctoral Researcher with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology from 2006 to 2009. His research interests include various aspects of very large scale integration design including circuits and systems for communication and computation, and emerging device technologies including silicon photonics and memristors.

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