SUOR: Sectioned Undirectional Optical Ring for Chip Multiprocessor

XIAOWEN WU, JIANG XU, YAOYAO YE, ZHEHUI WANG, MAHDI NIKDAST,
and XUAN WANG, The Hong Kong University of Science and Technology

Chip multiprocessor (CMP) is becoming an attractive platform for applications seeking both high performance and high energy efficiency. In large-scale CMPs, the communication efficiency among cores is crucial for the overall system performance and energy consumption. In this article, we propose a ring-based optical network-on-chip, called SUOR, to fulfill the communication requirement of CMPs. SUOR effectively explores the distinctive properties of optical signals and photonic devices, and dynamically partitions each data channel into multiple sections. Each section can be utilized independently to boost performance as well as reduce energy consumption. We develop a set of distributed control protocols and algorithms for SUOR, but physically allocate the corresponding cluster agents close to each other to benefit from the strengths of optical interconnects at long distances as well as electrical interconnects at short distances. Simulation results show that SUOR outperforms the alternative optical networks under a wide range of traffic patterns. For example, compared with MWSR design, SUOR achieves 2.58× throughput as well as saves 64% energy consumption on average in a 256-core CMP. Compared with MWMR design, SUOR achieves 1.52× throughput and reduces 73% energy consumption on average.

Categories and Subject Descriptors: C.1.2 [**Processor Architectures**]: Multiple Data Stream Architectures (Multiprocessors)—*Interconnection architectures*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network topology*

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Optical network-on-chip, silicon photonics, chip multiprocessor(CMP)

ACM Reference Format:
Xiaowen Wu, Jiang Xu, Yaoyao Ye, Zhehui Wang, Mahdi Nikdast, and Xuan Wang. 2014. SUOR: Sectioned undirectional optical ring for chip multiprocessor. ACM J. Emerg. Technol. Comput. Syst. 10, 4, Article 29 (May 2014), 25 pages.
DOI: http://dx.doi.org/10.1145/2600072

1. INTRODUCTION

As the number of available transistors on a single chip increases to billions or even larger, chip multiprocessor (CMP) is becoming an attractive platform delivering high performance with limited power budget. In a complex CMP system, the communication efficiency among the cores becomes crucial for the overall system performance and energy consumption. To cope with the growing communication requirements, on-chip communication architecture has gradually moved from ad-hoc or bus-based design to network-on-chip (NoC) design [Dally and Towles 2001; Hoskote et al. 2007; Owens et al. 2007]. However, the limitations of electrical interconnects such as limited bandwidth, high delay and energy consumption, have severely hindered the further improvements of NoC in providing even higher performance.

This work is partially supported by RGC of the Hong Kong Special Administrative Region and the Nanoscience and Nanotechnology Program of HKUST.
Authors' addresses: X. Wu, J. Xu, Y. Ye, Z. Wang, M. Nikdast, and X. Wang, Department of Electronic and Computer Engineering, The Hong Kong University of Science of Technology, Hong Kong, China; email: {wxxaf, jiang.xu, yeyaoyao, zhehui, mnikdast, eexwang}@ust.hk.

DOI: http://dx.doi.org/10.1145/2600072

Based on demonstrated photonic devices including the VCSEL laser [Krishnamoorthy et al. 2011], modulator [Q. Xu B. Schmidt and Lipson 2005], photodetector [Masini et al. 2007] and waveguide [Dong et al. 2010], optical interconnects have been proposed as an alternative to traditional metallic interconnects. Optical interconnects promise ultra-high bandwidth and low energy consumption. Optical NoC (ONoC) using optical interconnects has been put forwarded to replace electronic NoC by many studies [O'Connor 2004; Shacham et al. 2008; Cianchetti et al. 2009; Vantrease et al. 2008; Pan et al. 2010b]. However, there is a major limitation of optical interconnects that optical signals cannot be buffered easily unless they are converted to electrical signals, and the conversion consumes significant amount of extra energy. To avoid extra energy consumption, ONoC designs try to reduce electrical-optical/optical-electrical (EO/OE) conversions, and often encounter other overheads. For instance, in Shacham et al. [2008], an end-to-end optical path is set up ahead by an overlapped electronic network. Although this approach has its merits, the optical network could be inefficiently utilized when the setup delay is high or the packet is small.

Optical crossbars can be implemented so that EO/OE conversions are only required at the ends of each transaction. Although the crossbar has high throughput, it requires a large volume of resources as well. Resource sharing is an effective solution to alleviate the requirements. For example, in Vantrease et al. [2008], a waveguide for payload transmission can be shared by multiple writers and a single reader (MWSR). In contrast, in Pan et al. [2009], a waveguide can be shared by a single writer and multiple readers (SWMR). Furthermore, a waveguide can be shared by multiple writers and multiple readers (MWMR) in Pan et al. [2010b]. With resource sharing, many senders or/and receivers are attached to the same waveguide and an optical signal bearing information may pass through these attachments. This would inevitably introduce large optical loss for the optical signal, which in turn increases the energy consumption at light sources to compensate the loss.

To address the aforementioned problems, we propose a sectioned undirectional optical ring, called SUOR, where resources are efficiently shared and energy consumption is also brought down. In both MWSR and MWMR channel designs, a waveguide is unidirectional, and it can only be used by one sender/receiver pair for a period, even though the waveguide is accessible to multiple readers and writers. In SUOR, by utilizing the propagation property of light, we divide one waveguide into multiple un-overlapped sections such that each section can be independently utilized. The single waveguide can thus support multiple transactions *simultaneously*, and bidirectional transmission is also supported. With such segmentation, the passed waveguide length and the number of optical components encountered by the optical signals is minimized. Since the waveguide and the other optical components would induce the power loss for the light propagating through, the segmentation would effectively reduce the loss and thus the energy consumption at laser sources.

To support the efficient sharing of the resources, we propose a control subsystem that takes both the advantages of optical interconnects at long distances and electrical interconnects at short distances. Each processing node is assigned with an agent for channel accessing. We physically allocate these agents close to each other in the chip center. Agents can communicate with processing nodes optically with low delay; they can also share the information with each other by short electrical wires with high connectivity. With the highly shared resources and efficient control scheme, SUOR supports high throughput with relatively low power consumption. Simulation results show that SUOR significantly outperforms the alternative optical networks-on-chips under a wide range of traffic patterns. For example, under synthetic traffics, compared with the MWSR design in Corona, SUOR achieves 2.58× throughput and saves 64% energy consumption on average in 256-core CMP. Compared with the MWMR

design in Flexishare, SUOR achieves 1.52× throughput and reduces 73% energy on average.

The remainder of this article is organized as follows. In Section 2, the related work is reviewed. Section 3 describes the details of the SUOR architecture and the control scheme. Performance and energy efficiency of our SUOR are evaluated and compared with alternative designs in Section 4. We conclude the article in Section 5.

2. RELATED WORK

Nanophotonics can enable efficient interchip communication networks with optical interconnects. Batten et al. [2008] proposed opto-electrical crossbar connecting small groups of cores and DRAM modules. Koka et al. [2010] proposed a silicon-photonic network to enable a scalable system with multiple chips. Cianchetti et al. [2010] also proposed a system-in-package design with nanophotonic interconnects. The design of optically interconnected multiple chips is also explored by Pan et al. [2010a]. All these designs show that optical interconnects outperform the electrical ones with much higher throughput and lower power consumption.

Based on the photonic devices recently demonstrated, different optical on-chip networks have been proposed. Kirman et al. [2006] presented an opto-electrical hierarchical bus for future CMPs with cache-coherence supported. Optical loop on top is to address global communications, while electrical wires are used for local interconnects. Morris et al. [2013] proposed an optical tree-based broadcast network to address the snoopy cache coherence in CMP. Xu et al. [2011] proposed a hierarchical optical network and a composite cache coherence protocol, trying to acquire both advantages in snoopy and directory-based protocols. Bartolini and Grani [2012] proposed a hybrid network with mesh electrical network and optical ring network for CMP. Optical ring network is to transfer short control messages to reduce the control delay in cache coherence protocol. Pasricha and Dutt [2008] proposed an optical ring waveguide to replace global pipelined electrical interconnects. O'Connor presented a full connected optical NoC based on the λ-router with WDM technology. Shacham et al. [2008] proposed a hybrid optical NoC which combines an optical circuit-switched network with an electrical packet-switched network. Electrical network is used for path set-up and short packet transmission. Li et al. [2009] proposed a hybrid network in which optical network is used to broadcast latency-critical messages and electrical network is used to transfer high bandwidth traffic. Joshi et al. [2009] presented a photonic clos network in which long distance communication between routers are replaced by optical interconnects, providing more uniform latency and throughput compared with the mesh network. Kao and Chao [2011] reduced the buffer requirements in optical clos network by proposed scheduling algorithm. Cianchetti et al. [2009] proposed a packet-switched optical network. Data packets may pass through multiple routers before being buffered through OE conversions. Bahirat and Pasricha [2009] proposed a hybrid photonic NoC which utilizes optical rings to enhance an electrical mesh NoC. Ye et al. [2009] proposed an optical NoC combining 3D stacking and silicon nanophotonic technologies. Ding et al. [2012] proposed synthesis tools for optical interconnects on chip considering thermal effect. Ouyang et al. [2011] proposed an optical NoC based on free-space optical interconnects to reduce power consumption. Psota et al. [2010] used WDM technology to build contention-free network which facilitated new programming model. Koohi et al. [2011] proposed hierarchical optical rings, where local rings are used for intra-node communication and global rings are to connect the nodes. Ye et al. [2012] studied optical torus network with proposed network protocols and floorplanning. Ramini et al. [2013] discussed the physical layouts of different wavelength-routed on-chip optical networks. Poddar et al. [2012] proposed a hybrid network and CDMA technology is used to boost the throughput of the optical ring network. In many

networks mentioned above, WDM technology is used to facilitate link sharing. A channel may support multiple concurrent transactions given that the transactions are with different wavelengths. This approach is at the cost of reduced bandwidth (fewer wavelengths) for each transaction. SUOR explores the concurrency in another direction: it wisely supports concurrent transactions as long as they would not collide with each other, and each transaction is still with full bandwidth.

By employing more resources, crossbars have been proposed to achieve high throughput. Vantrease et al. [2008] proposed Corona architecture which uses optical interconnects for both intercore communication and off-stack communication to memory. Cores are integrated as clusters which are fully interconnected with a photonic crossbar. A distributed optical token-based arbitration scheme is proposed for channel allocation. Pan et al. [2009] proposed Firefly architecture as a hybrid hierarchical on-chip network. It utilizes an electrical network for short distance transmission and an optical crossbar for long distance transmission. The crossbar is partitioned into smaller crossbars with localized arbitration. Flexishare proposed by Pan et al. [2010b] provides a flexible optical crossbar in which each data channel is accessible for all clusters to write and read. Special token stream arbitration protocol is proposed to cope with the flexibility. Xu et al. [2012] proposed a channel borrowing technology to improve the channel utilization and also reduce the power consumption. SUOR further increases the channel utilization by supporting multiple concurrent transactions and bidirectional transmission. Power consumption is also reduced by the special waveguide segmentation scheme.

Le Beux et al. [2011] presented an optical ring NoC for both 2D and 3D architectures. In their design, a wavelength can be reused in a waveguide such that it can also support multiple transactions to improve the performance as our SUOR. The wavelength is statically assigned based on the connectivity requirements. In the SUOR, a single waveguide supports multiple transactions dynamically based on the arbitration. The connection states of the waveguide can be changed by configuring the senders and receivers on the waveguide. Also, bidirectional transmission is supported for the same link. Morris et al. [2012] proposed an optical network with 3D stacking technology. A large crossbar is decomposed into multiple small crossbars on different layers to reduce the power. Reconfiguration is supported to boost the performance. The idea of decomposing a long link into some shorter links is also adopted in our SUOR but we need not physically break the channel and only one optical layer is required. The link length is much shorter in SUOR and thus the power consumption is also lower. Datta et al. [2012] proposed segmented optical bus. Buses are segmented to reduce power consumption and they are interconnected by electrical routers. SUOR segments the bus in more depth and the throughput is higher with efficient arbitration and more independent segments. No electrical switching is required which consumes large power and area. The optical power is also lower due to the light would only pass the active parts of the link.

3. ARCHITECTURE DESIGN

SUOR targets a scalable CMP system. To facilitate optical transmission among the cores, an optical layer is stacked with the electrical layer with 3D technology, as shown in Figure 1. The electrical layer encompasses multiple computing cores and the local electrical wires, while the optical layer on the top facilitates the global optical communication among them. In the optical layer, waveguides, optical switches, and photodetectors are fabricated. On-chip lasers, VCSELs, are bonded on the chip as light sources. The cores on the electrical layer can access these optical components with through-silicon-vias (TSVs). In this 3D chip, a complete optical transmission involves three steps. First, data is sent from electrical layer to the stacked optical layer with TSVs. Then, the electrical signals are converted to optical signals and transmitted out

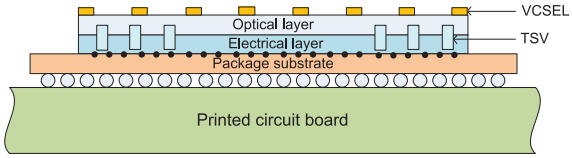

Fig. 1. Cross section of a 3D chip with an optical layer and an electrical layer.

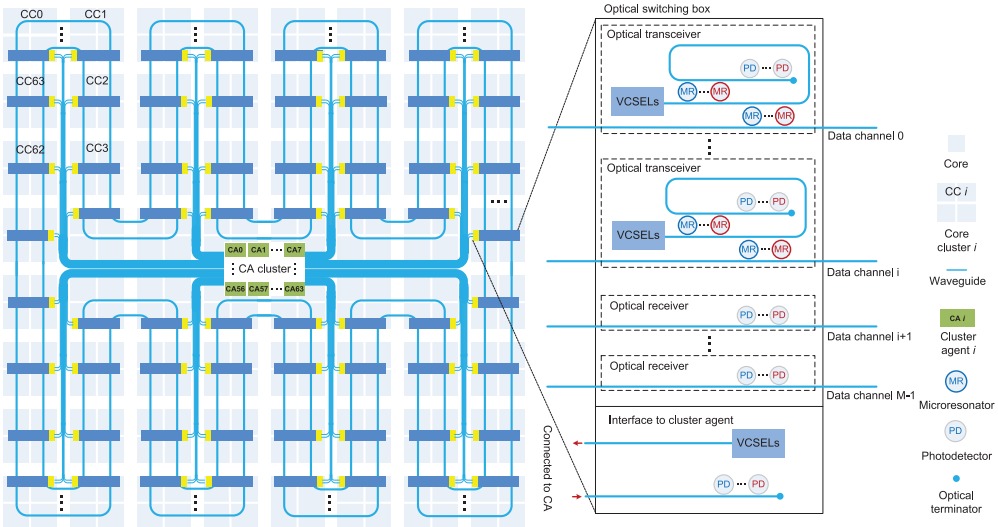

Fig. 2. The overview of SUOR architecture and its floorplan.

into the waveguides. Lastly, the photodetectors at the destination would convert the optical signals back to electrical ones and send them to the electrical layer.

The architecture overview of SUOR is shown in Figure 2. The chip includes multiple homogeneous cores. Each core is with a private L1 cache, and every four neighboring cores sharing an L2 cache form the core clusters (CC). The clusters are connected with optical waveguides which are aligned as closed-loops on the chip. The optical waveguides are parallel with each other, forming data channels of the system. Each cluster accesses the data channels with specially designed optical switching box shown in the right side of Figure 2. In SUOR, each data channel is accessible to multiple clusters, and there may be more than one concurrent transaction on the same channel. This would help to improve the utilization of the network resources. Control overhead is introduced by these features and it is handled by the control subsystem.

In the control subsystem, each cluster is assigned with a dedicated cluster agent (CA). An agent makes decisions in a relatively independent fashion. We moved all the agents to the center of chip, forming CA cluster as shown in Figure 2. Each CA is connected to the corresponding cluster with dedicated optical waveguides and interfaces. CAs communicate with each other using short local electrical wires. The agents are responsible for data channel set-up as well as flow control for all transactions. The architecture is designed in a way that the communication among the clusters are with nearly uniform latency and arbitration protocol is fair for all clusters.

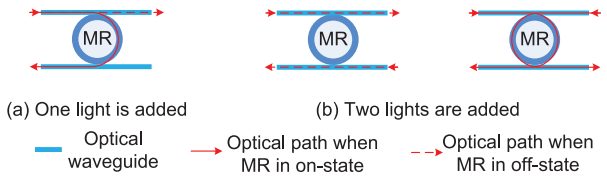

Fig. 3. The illustration of the functionality of MR.

3.1. Data Channel Design

Data channels are composed of multiple parallel waveguides which are aligned as closed loops and pass through all clusters on the chip. The cluster can send data to or receive data from the channels with optical transceivers as shown in the right side of Figure 2.

3.1.1. Optical Transceiver. The optical transceiver is composed of lasers, waveguides, photodetectors, and microresonators (MRs). VCSEL is chosen as the on-chip laser source. It can be directly modulated with a rate of 10 Gbps, and arrays of VCSELs can be bonded directly to CMOS VLSI chip [Krishnamoorthy et al. 2011; Gunn 2006]. Compared with off-chip laser source, the on-chip laser source owns the potential of substantially reducing the static power. The off-chip laser has to keep lasing once the system is booted, while the on-chip laser can be powered off when there is no data transfer. This would significantly reduce the power consumption if the application load is not high. The turn-on delay of VCSEL is below 1 ns [Bruensteiner and Papen 1999] and hidden by path-setup delay, and thus it would not impair the network performance. Another advantage of on-chip laser is that we can dynamically control the output power based on the optical power loss on the path. To transfer the data over a link successfully, the emission power from the laser should be larger than the summation of power loss along the path and the minimum optical power required at the photodetector for sufficient large SNR. If off-chip laser is adopted, in order to guarantee enough power for all possible transmission paths, the laser sources are set to provide the worst-case optical power for all packets. The adaptive power control mechanism we implement here uses routing information to calculate the optical power loss encountered on an optical path and control lasers to generate adequate optical power for transmission. This would require the information of routing, which can be easily satisfied by the control subsystem described later.

The disadvantage of on-chip laser is that it would introduce extra power dissipation for the chip compared to off-chip laser. However, by utilizing the good properties of on-chip laser, we bring down the overall power consumption of lasers to less than 4 W (shown in the Section 4), making the extra thermal problem caused by lasers be easily accommodated by current cooling technology. Another disadvantage of on-chip laser is that, the power efficiency of the laser would drop with high temperature. However, this overhead would be well compensated by saved power, which is verified in Section 4.

Besides lasers, MRs are also included in the transceiver. The MR is a switching element. It can divert the light with resonance wavelength from one waveguide to the opposite one. The resonance wavelength of the MR can be controlled by electrical voltage. As shown in Figure 3(a), by turning on the MR, the resonance wavelength of the MR is the same as the wavelength of the input light, resonance happens and the light would be diverted to another waveguide. By turning off the MR, the resonance wavelength of the MR changes, and thus resonance does not happen and the light would bypass the MR directly. Further, two light sources can be added into the two waveguides at the same time as shown in Figure 3(b). If resonance does not happen, two light waves would propagate along the original waveguides; if resonance happens,

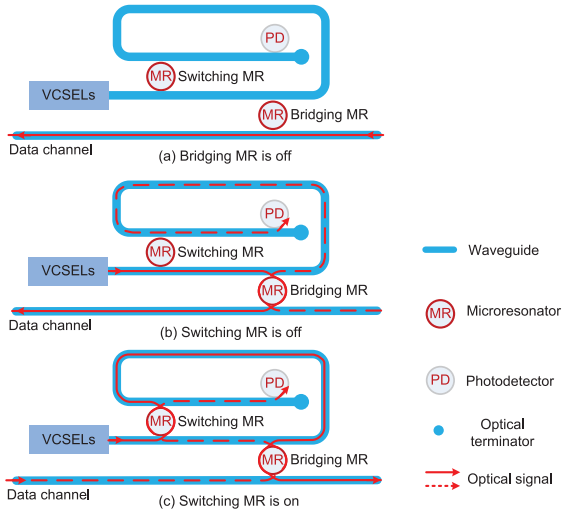

Fig. 4. The functionality of the transceiver. Only one wavelength is illustrated here. To distinguish two different optical signals with the same wavelength, one signal is shown in solid line and the other is in dashed line.

both light waves would be diverted to opposite waveguides as long as the two lights are not coherent. This property is discussed in [Simos et al. 2009], and illustrated with experiment in Geng et al. [2009]. The MR can also serve as a wavelength-selective photodetector by germanium doping. This design of the detector would reduce the capacitance and remove the trans-impedance amplifiers [Vantrease et al. 2008]. MR is wavelength-sensitive and each kind of MR can control corresponding light signals while not affecting the light in other wavelengths. In SUOR, we pack W wavelengths into the waveguide for each transaction, so W MRs are implemented at each switching stage and receiver, and all W MRs at each switching stage are powered on/off at the same time.

The functionality of the optical transceiver for one data channel is illustrated in Figure 4. The lower right MR in the transceiver is called *bridging MR* which is used to connect the cluster transceiver with data channel; and the upper left MR is called *switching MR* which is used to control the direction of lights. When the bridging MR is powered off, light in data channel would pass through this cluster without being disturbed as shown in Figure 4(a). In this way, the signal may bypass many clusters before being buffered by the destination cluster. When the bridging MR is powered on, light in data channel would be diverted into the transceiver and later received by photodetector. When the bridging MR is on, the cluster can also inject data into the data channel. The direction of the light in data channel can be controlled by switching MR at the upper left. When the switching MR is off as shown in Figure 4(b), the cluster can send optical signals (shown in solid line) to the left part of data channel and receive signals (shown in dashed line) from right part of channel. When the MR is on as shown in Figure 4(c), the cluster can send signals (shown in solid line) to the right part of data channel and receive signals (shown in dashed line) from left part of channel. Therefore, both directions of transmission in the waveguide are supported.

3.1.2. Bidirectional Transmission. In conventional design, each link is single directional and two links are required to support the communication between the two communicating nodes. Due to the imbalance property of the real traffics, it is often the case that one unidirectional link is busy with heavy traffic burden while the opposite link is idle

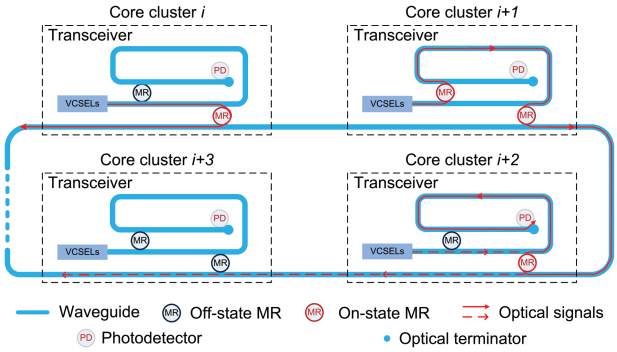

Fig. 5. A data channel with optical transceivers. Only one wavelength is illustrated. Both solid and dashed lines are used to distinguish the different transactions at the same wavelength.

with no data transmission, wasting the network resources. The profiling in Qian et al. [2012] shows that this case happens in around 67.5% of time on an 8×8 mesh NoC architecture, even when the injection rate is 80% of the saturation rate.

Motivated by this observation, we design the channel such that each link supports bidirectional transmission, as shown in Figure 5. For example, core cluster CC_{i+1} can send data to CC_{i+2} as illustrated by powering on the upper left switching MR in CC_{i+1} and turning off the switching MR in CC_{i+2}. On the other hand, by powering on the switching MR in CC_{i+2} and powering off the corresponding MR in CC_{i+1}, CC_{i+2} would be able to send data to CC_{i+1} with the same optical link between them. This flexibility in direction can well handle the heterogeneous real traffics.

3.1.3. Channel Segmentation. Another feature of the channel is that, the channel is *virtually* segmented into multiple sections, and these sections can work independently and concurrently. This can help improve the utilization of links effectively. For simplicity, we use $S[i, j]$ to denote the channel section from cluster CC_i to cluster CC_j, in clockwise direction. As illustrated in Figure 5, when CC_i sends data out to the CC_{i-1}(not shown in the figure), the cluster CC_{i+1} can send data to CC_{i+2} simultaneously. That is, $S[i-1, i]$ and $S[i+1, i+2]$ can work independently although they are in the same channel.

The segmentation feature can be further facilitated by the bidirectional feature in improving the resource utilization. Since the channel is a circle, a long link used by a transaction can be replaced by a short one in opposite direction such that the unused long link can be utilized by other transactions. For example, in Figure 5, CC_i sends data to cluster CC_{i-1} (not shown in the figure) in counterclockwise direction occupying the $S[i-1, i]$, leaving most of sections free to work. If only single direction (e.g., clockwise) is allowed, $S[i, i-1]$ would be occupied, leaving no sections for other clusters. It is also possible that we can choose an opposite direction for a transaction to prevent collisions. For example, if CC_i wants to send data to CC_{i+3} given that CC_{i+1} is sending data to CC_{i+2}, we can let CC_i choose the counter-clockwise direction such that no collision happens. This can not be achieved if bidirectional transmission is not supported.

In Figure 6, we show the abstract view of SUOR and the other two alternative designs. In MWSR channel design [Vantrease et al. 2008], multiple writers and single reader are attached on the channel. In MWMR channel design [Pan et al. 2010b], each channel is flexible for all writers and readers. In both designs, one transaction can be supported at a time and the channel is unidirectional. In contrast, each channel of SUOR can support multiple concurrent transactions and each section is bidirectional. As shown in the figure, the original long path $S[1, n]$ is replaced by a shorter path

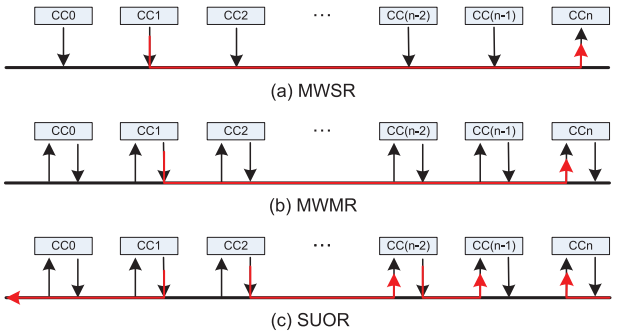

Fig. 6. Illustration of different optical channel designs. The red lines with arrows show the current transactions on the channel.

$S[n, 1]$, leaving other links available for other transactions, that is, CC_2 to CC_{n-2} and CC_{n-2} to CC_{n-1}. In best case, SUOR can support N (N is the number of clusters) concurrent transactions, improving the throughput by N times.

3.1.4. Power Reduction. On the data channel, multiple senders and receivers are attached to the waveguide, introducing power loss for the light passing through them. The waveguide itself would also introduce some loss. To compensate the optical loss, larger optical power at the laser is required. In both MWSR and MWMR channel designs, the light has to go through all clusters on the chip, disregarding the effective distance of the transaction. By utilizing bidirectional and segmentation properties, in SUOR, the light would only pass necessary links and thus avoid unnecessary power loss. For illustration here, we assume the 64 ($W = 64$) wavelengths are used for each transaction, 64 ($N = 64$) clusters are attached to the channel, the MR passing loss is 0.001 dB, and the waveguide loss is 1 dB/cm. If the light has to go through all clusters (covering $S[0, 63]$), it would experience 12.1 dB loss. The shortest path, for instance, $S[62, 63]$ would only cause 0.25 dB loss, which corresponds to 93.5% less optical power requirement. By segmentation, the light bearing signals from CC_{62} to CC_{63} would only go through $S[62, 63]$ instead of $S[0, 63]$, and thus the power consumption is effectively reduced. Bidirectional transmission would also benefit the power reduction here. Since the channel is a ring, an original long path can be replaced with a shorter one in opposite direction, for instance, replacing the link $S[0, 63]$ by $S[63, 0]$, reducing the optical loss effectively.

Since each sender may send data to multiple receivers, the optical loss of each transfer would vary from time to time. To decide the optical loss for each transaction, each sender has a table storing the losses to all receivers. The losses are pre-calculated, which are based on the losses introduced by all components on the path. Once the receiver is known, the laser can emit adequate power based on the loss. The output power of laser is tuned by changing the driving current. The overhead of this adaption is a table (with size N, N is the number of receivers) for each sender and also the control delay. However, the control delay can overlap with the path set-up delay and thus it would not affect the performance.

3.2. Control Subsystem

We implement a distributed control protocol for SUOR. Each cluster is assigned with a dedicated cluster agent which is responsible for processing the requests of accessing the data channels. The arbitrations are made in a relative independent fashion, except of sharing some limited information including the memory states and the channel states. We moved all the agents to the center of chip so that they can share limited information

with short latency by using the very-short local electrical wires. The relatively large distance between the agent and cluster is offset by the dedicated optical links which provide low communication delay. The connection between each agent and the corresponding cluster is composed of two unidirectional waveguides: one for transmitting requests and the buffer information from the cluster to the agent, and the other one for grant information from the agent to the cluster. The latency between cluster and agent is within one clock cycle.

3.2.1. Arbitration Protocol. Although optical signals can be transferred much faster than electrical signals, they can not be easily buffered or processed in optical domain. Converting them back to electrical signals would inevitably introduce significant energy overhead. Therefore, circuit-switching is chosen for SUOR, which means that the path establishment from source to destination is required before payload data transfer. The MRs on each channel are turned off by default, and thus the path set-up only involves the configuration of the receivers. The protocol is as follow. Before accessing the data channel, the cluster would send a request to its agent (called *source agent*) with the information including destination ID, request ID, and packet size. Destination ID is used to identify the receiver cluster. After receiving the request, the source agent would check the channel states, try to reserve a channel section for this request and lastly send the grant packet containing the channel ID back to the cluster. At the same time, the destination cluster's agent (called *destination agent*) would also send the grant information to the destination cluster. After receiving the grants, the source cluster would send data out on the assigned channel (identified by the channel ID) while the destination cluster would open the receivers to detect the data.

The request ID is attached for each request so that the cluster can send multiple requests out before receiving grant information. This will help to boost the throughput of the control system through pipelining, especially when the round trip delay is large. In SUOR, we support variable packet size for each transaction. Comparing to fixed packet size design, supporting variable packet size can prevent a large packet from being truncated into multiple small packets, which would alleviate the arbitration burden and potentially improve the performance.

3.2.2. Channel Grouping. In SUOR, each cluster can access multiple channels while each channel can be accessed by multiple senders independently. If the packet size of the transmissions is constant, the off-line scheduling problem can be formulated as *interval partitioning problem* which can be optimally solved [Kleinberg and Tardos 2005]. However, the solution is unapplicable to on-line scheduling, given that the traffic pattern is always unpredictable before the execution. In addition, we want to eliminate the constraint that packet size is fixed.

To cope with the complexity of the arbitration in cluster agent, we set some access rules in data channel at first place. This would eliminate some traffic patterns on a specific channel such that these patterns do not need to be considered for that channel at all. For instance, if we assume that a cluster can only communicate with two neighboring clusters on a specific channel, then the potential collision on that channel only happens between neighboring clusters. This will significantly reduce the complexity of the arbitration. Although imposing access rules may sacrifice some flexibility of the network, it effectively reduces the arbitration overhead and thus the processing delay, potentially achieving even higher performance in reality.

To impose the access rule, we classify the data channels into groups according to the allowed patterns on the channel. In each group, only special traffic patterns are allowed. Here, the pattern is referred to the transaction distance since the data channel is a symmetric un-directional ring. The distance in turn is measured as the *minimum* number of hops between two clusters, and a hop is the distance between two

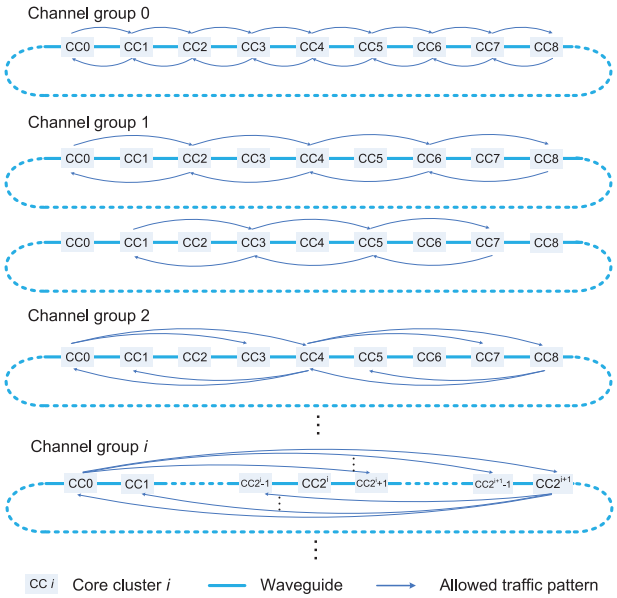

Fig. 7. Channel grouping. Channels are classified into multiple groups based on the allowed transmission patterns.

neighboring clusters. For example, the distance between cluster CC_0 and CC_{63} in Figure 2 is one hop instead of 63 hops since both clockwise and counter-clockwise directions are supported.

As shown in Figure 7, in the *i-th* group, only allowed transactions are with distances $\in (2^{i-1}, 2^i]$ hops. With this classification, intergroup collisions are skipped in arbitration. For example, the traffic with 8 hops will be assigned to the group 3, and will not be interfered by transactions with distance larger than 8 hops or smaller than 4 hops. Within a group, we further classify each data channel based on the allowed intervals on the channel. For example, as shown in Figure 7, on the first channel in group 1, the permitted senders are the clusters with even labels. And on the second channel, only odd labeled clusters are allowed. Formally, on the *j-th* channel in group i, the allowed senders are the clusters with labels $(j + k \times 2^i)\%N$ where k here is any nonnegative integer and N is the total number of clusters. In this way, each channel in group i is divided into $N/2^i$ sections with length 2^i hops. All transactions are confined within the sections such that no cross-section collisions exist. The arbitration left is relatively simple since only neighboring senders may conflict with each other. It is also necessary to mention that, due to the accessing rules, many senders/receivers can be omitted on the data channel. In Figure 7, take the first waveguide in group 2 for example, there is no sender or receiver attached to the waveguide at cluster 2. The senders in cluster 1 and 3 are also omitted in this waveguide.

3.2.3. Flow Control. Credit-based flow control is used in SUOR, which is facilitated by the cluster agents. For each transaction pair, the source agent has the initial number of tokens corresponding to the number of buffer slots of the receiver. It counts down the tokens each time a packet is sent. On the other hand, the receiver cluster would send the new tokens back to the destination agent via the optical link if the buffer slots are emptied. The destination agent would in turn send the tokens to the source agent via short electrical wires. If no token is left, the requests would not be processed by the source agent.

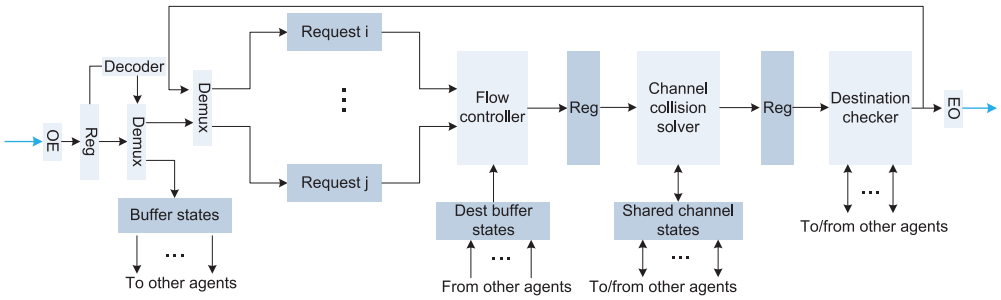

Fig. 8. Structure overview of one cluster agent.

3.2.4. Cluster Agent Design. Each cluster agent is optically connected to the corresponding cluster. Since the cluster would send both requests and buffer tokens to the agent, two types of packets are required. One bit is enough to identify the difference. From the agent to the cluster, there are also two types of information: the grants answering the request and the grants informing the receiver that new packets are coming.

The cluster agent would perform the arbitration according to the Algorithm 1, and the corresponding structure of agent is shown in Figure 8. With accessing rules in channel grouping, the cluster agent would only need negotiate with neighboring agents for each transaction, the complexity of the arbitration algorithm is reduced to $O(1)$. The agent receives the packet from cluster and then decodes it. If it is a packet containing requests, the requests would be stored in the request pool. If it is a packet with buffer tokens, the tokens would be sent to the related sender agents. For each request in the request pool, it has to undergo three steps before being granted. The first step is to check whether the destination buffer is full, which is implemented in flow controller unit shown in Figure 8. It is followed by the channel collision solver which checks whether the channel segment is available. Lastly, the state of the destination agent is checked in the destination checker. It is to make sure that the destination agent is able to inform the destination cluster to open the detector on time. If the three steps are passed successfully, the request is granted and the grant information is sent back to the cluster. Otherwise, the request would be stored back into the request pool. Multiple

ALGORITHM 1: Request arbitration algorithm

Input: Received data I.
Output: Granted request
if $IsBufferToken(I) == true$ **then**
  $SendToOtherAgents(I)$;
else if $IsRequests(I) == true$ **then**
  $RequestPool.Add(I)$;
end
$k <= RequestPool.SelectRequest()$;
if $(CheckDestinationBuffer(k) == success) AND$
$(CheckChannelState(k) == success) AND$
$(CheckDestinationAgent(k) == success)$ **then**
  $Grant(k)$;
else
  $RequestPool.Add(k)$;
end

requests are processed in parallel and the process of each request is well pipelined to improve the throughput of the controller.

The cluster agents share some information but the negotiation is limited such that delay is well confined. First of all, buffer information is shared among the agents. For example, if cluster i is allowed to send data to cluster j on channel k, the agent j would send the corresponding buffer information to agent i. The fabric for buffer information transfer is simple. Given the credit-based flow control is implemented, one single wire is enough: '1' indicates one new buffer slot is available and '0' means null. The source agent would accumulate the credits and consume n credits if a packet with n-slot length is granted. Buffer information transfer is unidirectional and there is no arbitration or broadcasting required. Furthermore, variable packet size is fully supported by this design.

The second shared information is channel states. The sharing is confined between two neighboring senders, given that a section can only be occupied by two senders at the two endpoints. The arbitration is as the following. If the channel section is occupied by one sender, no request is allowed. If channel is free and only one request is pending, this request is granted. However, if both requests are waiting for this requests, round-robin is adopted to decide the winner. The arbitration for each channel section only involves two agents and thus it consumes little time, thanks to the access rules set previously. In addition, the fairness among the clusters are achieved.

The last shared information is the state of the cluster agent. In our protocol, the destination agent is required to inform the receiver cluster to listen on a specific data channel at the right time. The link bandwidth between the cluster agent and the cluster is limited, and therefore it is possible that some informing messages can not be transmitted out on time. Therefore, if cluster i is to send data to cluster j, cluster agent i should check the state of cluster agent j. The checking protocol is NAK-based: a bit is sent from cluster agent i to agent j, if agent j is busy, a negative signal is sent back to agent i, otherwise no signal is transmitted back. This would save power since the case of busy is not common based on sufficient link bandwidth allocated.

4. QUANTITATIVE ANALYSIS AND SIMULATION RESULTS

In this section, we evaluate the performance and power efficiency of SUOR and compare it with alternative architectures including the MWSR design in Corona [Vantrease et al. 2008] and MWMR design in Flexishare [Pan et al. 2010b]. Both MWSR and MWMR designs are illustrated in Figure 6. In MWSR design of Corona, there are N (N equals the number of clusters) data channels. Each channel is destined to a specific reader (destination cluster) but is accessible to all other writers (source clusters). In MWMR design of Flexishare, each channel is accessible to all readers and writers. The number of data channel is flexible, but we set the number of waveguides to be the same as MWSR design for comparison. The control schemes for Corona and Flexishare designs are the same as proposed in the original papers [Vantrease et al. 2008; Pan et al. 2010b]. Specifically, in Corona, an optical token loops among the senders, and the sender which grabs the token is granted to send data out. In Flexishare, token streams are used to reduce the loop time and broadcasting is used to inform the readers. To show the scalability of SUOR, we consider different network sizes. The number of cores in CMP varies from 64, 128 to 256. In all configurations, every four cores share an L2 cache, forming a cluster. Therefore, the number of clusters in the networks are 16, 32 and 64 respectively. The clock frequency of the targeted CMP is assumed to be 5 GHz. We assume the cluster structure is the same for all three designs, and thus the communication within the clusters are not considered here.

We have developed a cycle-accurate network simulator with system C. In Corona and Flexishare, the continuous light is modulated both at rising and falling edges and thus

Table I. Overall Network Resource of SUOR, Flexishare [Pan et al. 2010b] and Corona [Vantrease et al. 2008]

	SUOR		Flexishare		Corona	
No. of cores	No. of data Waveguides	No. of MRs	No. of data Waveguides	No. of MRs	No. of data Waveguides	No. of MRs
64	76	107,520	64	133,120	64	66,048
128	156	389,120	128	533,504	128	264,192
256	284	1,363,548	256	2,117,632	256	1,056,768

the data rate for each wavelength is 10 Gbps as assumed in Vantrease et al. [2008]. In SUOR, we directly modulate the laser at 10 Gbps, and the VCSELs with the data rate higher than 10 Gbps have already been demonstrated in Kromer et al. [2005] and Ji et al. [2009]. In all three designs, we assume that 64 wavelengths are multiplexed into a single waveguide. In Corona and Flexishare, each data channel is composed of four waveguides as assumed. Therefore, 512 bits can be transmitted out in single cycle. In SUOR, to better utilize the channel grouping scheme, each data channel is composed of only one waveguide. This would introduce serialization delay for the system and it is taken into account in our simulation. To consider the propagation delay of the optical signal, we assume the chip size is 1cm $\times$ 1cm, and the group refractive index of the silicon waveguide is 4.2 [Dulkeith et al. 2006]. In SUOR, the floorplan of the data channel waveguides is shown in Figure 2. For comparison, in Corona and Flexishare, the floorplans of data channel are adapted as the same as SUOR.

In SUOR, for different network sizes, the number of waveguide groups in data channel varies. For 256-core CMP, the numbers of groups from group 0 to 5 are six, five, five, five, five, and four respectively. For 128-core CMP, the group 5 is omitted since the longest distance of a transaction is 32 hops while the transactions allowed in group 5 is larger than 32. Similarly, both group 4 and 5 are omitted in 64-core CMP. In each group i, there are 2^i waveguides. And on each waveguide in group $i (i > 0)$, there are $N/2^i$ senders and $N - N/2^i$ receivers (N is the number of cluster), while there are N sender/receiver pairs on the group 0 waveguide, which is shown in Figure 7. The resources of SUOR, Corona and Flexishare are listed in Table I. For fair comparison, in Corona, the MRs implemented for connection to memory and broadcasting are not taken into account in the table. As shown in the table, both SUOR and Flexishare use more MRs than Corona. This overhead is well compensated by higher performance which will be discussed later. In 64-core CMP, SUOR saves 20% of MRs compared to Flexishare, but 19% more data waveguides are implemented. When it is scaled to the 256-core system, SUOR saves 36% of MRs while implementing 11% more data waveguides, showing good scalability.

Besides data waveguides, control waveguides are also implemented in three networks. In SUOR, the control waveguides are used to connect the clusters and their corresponding agents. The number of control waveguides is linearly proportional to the network size. For example, in a 256-core system, 128 waveguides are required. Please be noted that these waveguides serve as point-to-point links and are very short compare with data waveguides, which is illustrated in Figure 2. As a result, the total area of the control waveguides are only 3.8% of the area of the data waveguides in a 256-core system. In Flexishare,the control waveguides include two 2-round token stream waveguides, two 2.5-round credit waveguides and 128 1-round reservation waveguides. The length of these control waveguides are equal or longer than data waveguides. The occupied area is 53.5% of the area of data links, which is much higher than SUOR. Corona requires only one control waveguide, therefore its control overhead is smaller than SUOR and Flexishare. However, this is at the cost of reduced

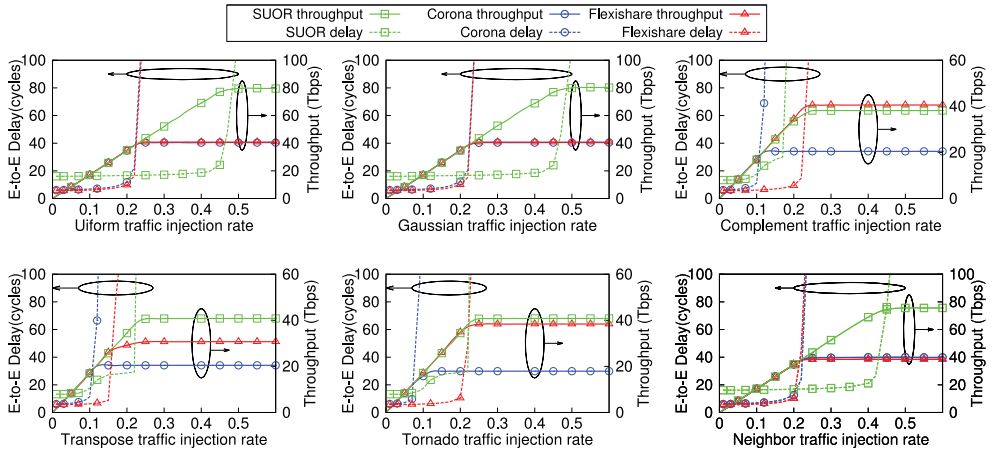

Fig. 9. The throughput and packet End-to-End delay comparisons of SUOR, Corona and Flexishare under six synthetic traffic patterns.

performance and lacking of flexibility of data channels, which would be more clear in the evaluation.

4.1. Performance Evaluation

We have conducted the evaluations under both synthetic traffics and realistic traffics. We first discuss the performance and power consumption results under synthetic traffics. The results under real applications will be presented in Section 4.3. Here, we consider six synthetic traffic patterns including uniform, Gaussian, transpose, tornado, bit complement, and neighbor traffics. In all traffics, the packet size is assumed to be constant and equal to 512 bits, simulating a cache line. Under uniform traffic, each cluster would send packets to all other clusters with the same probability. Bisectional bandwidth of the network is the critical factor under this traffic. Under Gaussian traffic, the probability distribution of the destination follows a Gaussian distribution, simulating the locality feature of real traffic. Neighbor traffic only allows packets with 1 hop, simulating the well-mapped tasks with communications with high locality. The bit complement, transpose and tornado traffics are all permutation traffics. These traffics would stress the load balance of the network [Dally and Towles 2003].

The throughputs and delay of three designs for 64-core CMP are shown in Figure 9. For all three designs, the throughputs are increased linearly with the injection rate before the saturation points, and they will be stable later. Meanwhile, the delay would increase dramatically after the saturation. All these phenomenons are expected from the models.

As shown in Figure 9, under uniform, Gaussian, and neighbor traffics, SUOR shows nearly 2x throughput compared with Flexishare and Corona. The performance gain under uniform traffic shows that SUOR supports high bisectional bandwidth. This high bandwidth is mainly achieved by segmentation. For a single data waveguide, SUOR would divide it into multiple independent sections, and all these sections can potentially support transactions. In Corona and Flexishare, only one transaction is allowed at a time on a single data waveguide. The performance gain under Gaussian and neighbor traffics show that SUOR supports locality quite well. This is another benefit from segmentation. When the waveguides are divided, more resources are provided for local transmission. For example, in group 0, one waveguide is divided into N sections and each section can serve as a channel for the local communication.

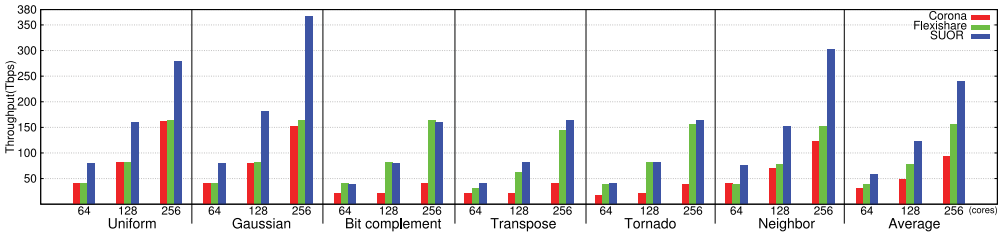

Fig. 10. Maximum throughputs of SUOR, Flexishare and Corona under synthetic traffic patterns for different network sizes.

When local transactions are intensive, the performance gain is even clear. In Corona and Flexishare, no special effort is paid for the local transaction.

Figure 9 also shows that the throughputs of SUOR under permutation traffics are only half of those under other traffic patterns. This is caused by static grouping of data channels. For instance, the channels in group 0 would only serve transactions between two neighboring clusters. On the other hand, a transaction from cluster 0 to cluster 31 would only use channels in group 5 but not other channels in other groups. Therefore, if a sender cluster only communicates with a fixed destination cluster, then most of channel resources designated to this sender would be wasted and they cannot be used by other clusters. The permutation traffic patterns are such kind of traffics that each cluster would only send packets to a single destination, and thus they can demonstrate the limitations of SUOR clearly. The design trade-off here is that large flexibility would lead to heavy arbitration burden, while lacking flexibility would cause the resources waste under adversarial traffics. However, under these adversarial traffics, SUOR still achieves similar throughput as Flexishare and higher throughput than Corona. SUOR partitions the data channel into multiple sections to increase the channel resources logically and make the waste affordable. These permutation traffics are also adversarial for Corona due to the special light-pulse token control protocol. Its performance is limited by the round-trip time of the token.

To show the scalability of SUOR, the throughput comparisons for different network sizes are shown in Figure 10. With larger network, higher throughputs are expected. For SUOR and Flexishare, the performance scales well with the network sizes for all traffic patterns. For Corona, the performance improvements under permutation traffics are not so significant as the other two designs. This is due to that, with larger network, the round trip time for the control token increases and it impairs the overall throughput. In Flexishare, token stream is implemented to avoid the large round trip time. The disadvantage of the scheme is that the packet size has to be fixed. In SUOR, the requests are forwarded to cluster agent in a pipelined manner, and therefore the control signal delay introduced by larger distance would not affect the performance. Also, different packet sizes are supported in our design. The simulation results also show that, under Gaussian and neighbor traffics, the performance ratio between SUOR and the other two designs increases as network size grows. With larger network size, there are more local traffics and this would favor SUOR with locality feature. In average, the performance gain of SUOR compared with Corona is increased approximately from $2\times$ to $2.58\times$ when the network size is increased from 64-core to 256-core. Comparing with Flexishare, SUOR achieves $1.52\times$ performance gain in average.

The delay of the networks is also shown in Figure 9. The delay climbs dramatically after saturation point for all designs. The network can only work stably with injection rate smaller than the saturation load. After the saturation load, the delay would go towards infinity with long enough simulation time. Figure 11 shows the saturation loads of three designs. SUOR has higher saturation loads comparing with Corona and

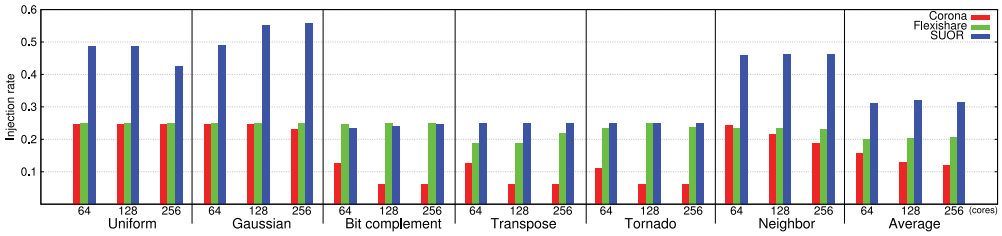

Fig. 11. The network saturation loads of SUOR, Flexishare and Corona under synthetic traffic patterns for different network sizes.

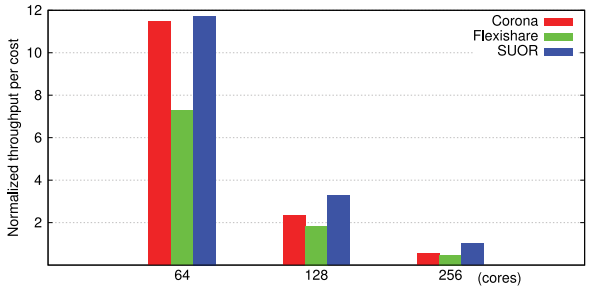

Fig. 12. The maximum throughputs per cost of three designs. All are normalized with the throughput per cost of SUOR in 256-core network.

Flexishare under uniform, Gaussian and neighbor traffics. Under permutation traffics, SUOR has similar saturation loads as Flexishare, but higher than Corona. These phenomenons are similar to the throughput comparison. Figure 11 can also show the network scalability in terms of the saturation loads. For example, when network size increases from 64-core to 256-core, the saturation load of Corona drops while that of Superb holds. It means that for SUOR, the same injection rate can be supported even with a larger network. SUOR has higher zero-load latency compared with Flexishare and Corona. This is due to the data channel serialization delay and also the arbitration delay of the cluster agent. However, with pipelining design of the control system, the higher delay affects the throughput little.

To better demonstrate the resource utilization efficiency of different designs, we show the average throughputs per cost in Figure 12. Here, the cost is defined as the product of the number of MRs and the number of waveguides. As shown in the figure, SUOR achieves highest utilization efficiency compared with the other two designs. The utilization efficiency decreases as the network size increases for all three designs. However, the efficiency gap between SUOR and the other two designs increases as the network size increases, showing the better scalability of SUOR.

4.2. Power Consumption

To show the power consumption in the architectures, we adopt the nanophotonic power model in Joshi et al. [2009]. There are various losses along the optical path and we list them in Table II. The bending loss is 0.005dB/90° [Xia et al. 2007]. In waste case, a packet may encounter 14 bending losses which would introduce a total of 0.07dB loss for the signals. In SUOR, there would be no waveguide crossings as shown in Figure 1, which helps reduce the crosstalk issue and also power consumption. The EO/OE conversion power is assumed to be 100 fJ/bit as projected in Krishnamoorthy et al. [2009]. The sensitivity of the photodetector is assumed to be 10 μW as in Pan et al. [2010b]. We assume the heating power is 1 μW per MR per Kelvin, with 20 K

Table II. Optical Loss

Component	Loss
Passing MR loss	0.001 dB
Filter Drop loss	1.5 dB
Waveguide Crossing loss	0.05dB
Waveguide Propagation Loss	1dB/cm
Splitter	0.2 dB
Coupler	1 dB
Bending	0.005dB/90°

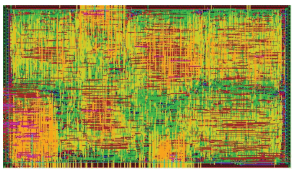

Fig. 13. The layout of one cluster agent.

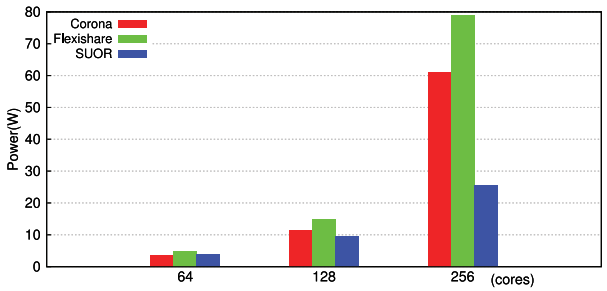

Fig. 14. Total energy consumption of SUOR, Corona and Flexishare with different network sizes.

tuning range, as in Joshi et al. [2009] and Pan et al. [2010b]. The power for switching MRs are assumed to be 50 μW per MR [Joshi et al. 2009]. The power efficiency of off-chip laser is assumed to be 30% as in [Joshi et al. 2009]. And it is assumed that lasing wavelength would be kept constant, implying that a temperature controller is required for off-chip laser. The power consumption of this controller in Corona and Flexishare is omitted in our power model. In SUOR, on-chip VCSELs are implemented. We tune the temperature of VCSELs to fix the lasing wavelength. The tuning power is assumed to be 1 μW per laser. The emission power efficiency of the laser would decrease with the increasing temperature [Syrbu et al. 2008]. We conservatively assume that the power efficiency of on-chip laser is only half of the off-chip laser. For the cluster agent of SUOR, we synthesis it with 45nm library and scale it to 17nm. The layout of one cluster agent is shown in Figure 13. It runs at 5 GHz, consuming 213 μW with switching rate of 15%. The area is $3517um^2$. The delay of processing one request is eight clock cycles.

We analyze the power consumption of all architectures under uniform traffic with injection rate of 0.1. Figure 14 shows the total power consumption of SUOR, Corona and Flexishare with different network sizes. When the network size is small (64-core CMP), the power consumption of SUOR is a little higher than Corona but smaller than Flexishare. When the network size increases, the power consumption of SUOR increases slow comparing to Corona and Flexishare. In the network with 256 cores, SUOR saves 64% and 73% of energy comparing to Corona and Flexishare respectively.

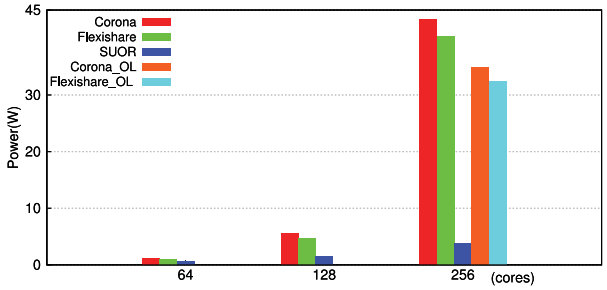

Fig. 15. Laser power for data channel in SUOR, Corona and Flexishare with different network sizes. Corona_OL and Flexishare_OL are the two designs that on-chip lasers are assumed for Corona and Flexishare respectively.

The high energy efficiency of SUOR is mainly contributed by the low power consumption of lasers, as shown in Figure 15. The power consumption of the laser is decided by the E/O conversion efficiency and the output power at laser source. The minimum output power required in turn is determined by the power loss on the path and the sensitivity of the photodetector. The sensitivity of the photodetector is assumed the same for all three designs, while the power loss on the path varies dramatically. For a data channel in crossbars like Corona and Flexishare, it passes through all clusters. The light has to propagate along the long waveguide and it also encounters a lot of MRs on the path. There is non-negligible propagation loss and passing ring loss on the path. When the network size becomes larger, these losses become significant. As a result, the power consumption of the lasers increases substantially. For example, the laser for data channel consumes around 1W in 64-core system, but the consumption increases dramatically to approximately 40W in system with 256 cores. This shows the scalability issue of conventional crossbar from the aspect of power consumption.

In SUOR, we classified the transactions based on the specific communication distance and divide the waveguide into multiple sections. In this case, the transmission distance and encountered MRs on the path are significantly reduced. Allowing bidirectional transmission also helps to reduce the distance and thus power loss. Shorter path and fewer MRs on the path means less propagation loss and passing MR loss, reducing the output power requirement at the laser source. The advantage of this scheme becomes more clear when we scale the network size up. For example, when the network size is scaled from 64-core to 256-core, the power consumption of lasers for data channel only increases from 0.66W to 3.8W. As a result, although we assume the power efficiency of on-chip laser is only half of the off-chip laser, the overall power consumption of the on-chip lasers in SUOR is still much smaller than that of off-chip lasers in Corona and Flexishare as shown in Figure 15. Static power can also be saved in SUOR. The on-chip laser can be powered off when no data is transferred. This cannot be done if off-chip laser is adopted as the case in Flexishare and Corona. When the application load is low, the static power can be significantly reduced. In Figure 15, we have also shown the on-chip laser power for data channels in Flexishare and Corona by assuming that on-chip lasers instead of off-chip lasers are used for the two designs. It is shown that, on-chip laser can help reduce the power consumption for two designs but it will still be much higher than SUOR due to higher power loss on the path.

In Figure 16, the power breakdown in percentage is shown for all three designs. For both Corona and Flexishare, a large portion of power is consumed by lasers for data channel, while this part is relatively small in SUOR. Corona consumes more power for data channel than Flexishare due to that the data channel waveguide in Corona is longer than that in Flexishare. Broadcasting is implemented in the control fabric

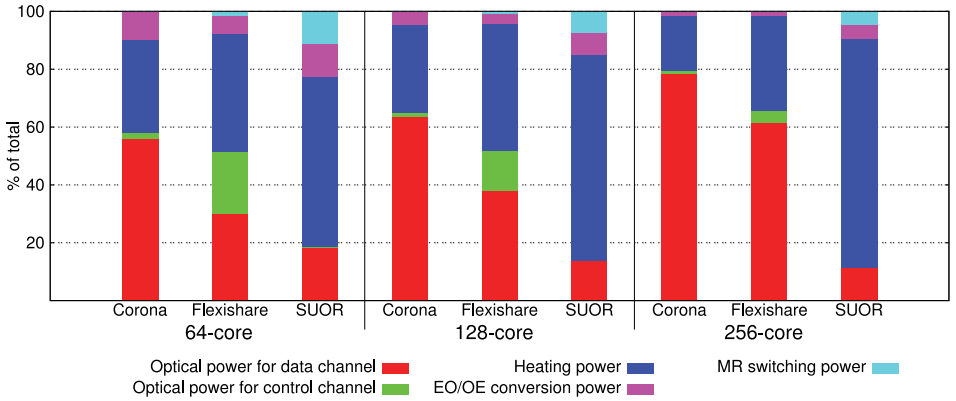

Fig. 16. Power consumption breakdown in percentage of SUOR, Corona and Flexishare with different number of cores.

of Flexishare so that there is a relatively large portion of power consumed by control channels in Flexishare. The control power for Corona is small because the simple token-ring is used. The control power for SUOR includes the optical communication power between the cluster and cluster agent and also the power consumed by the cluster agents. As shown in the Figure 16, the control power for SUOR is only 0.18W, thanks to the simple optical link between the clusters and agents and also the low cost design of the agents.

Besides power consumed by the lasers for data channel, another large portion of power is consumed by heating power for all three designs. Flexishare has the largest number of MRs, and it consumes largest heating power. SUOR and Corona has fewer MRs and thus consumes less heating power. However, the heating power is the most significant part in SUOR since the laser power is relatively small. More than 70% of power is consumed for MR heating in 256-core system. Therefore, after bringing down the laser power consumption, reducing tuning power would be very desirable. Many works have been done to analysis the thermal power [Ye et al. 2011; Nitta et al. 2011], and there are many technologies proposed to reduce the MR tuning power including channel re-mapping, adding spare rings and reducing the MRs on path [Zheng et al. 2012; Nitta et al. 2011]. Since the low-power consumption of SUOR is achieved by segmentation which is not related with MR designs, those approaches reducing MR tuning power can be easily applied in our design. It is also worth to note that the exact power model of MR would not affect the conclusion that SUOR has significantly reduced the power consumption, since it is the laser power but not MR tuning power reduced by our proposed segmentation technology.

The power efficiency is shown in Figure 17. In SUOR, the power consumption per bit increases slowly with the increasing network size, showing good scalability. In Corona and Flexishare, the power efficiency decreases quickly. The good scalability of SUOR comes from the efficient decomposition of the data channel.

4.3. Evaluation Results for Real Applications

Besides synthetic traffics, real applications are also used in the evaluation. We adopt the MCSL NoC benchmark suits [Liu et al. 2011], and the included applications in the evaluation are Fast Fourier transform (FFT), Reed-Solomon code encoder and decoder(RS_enc, RS_dec), and SPEC95 Fpppp (FPPPP). Similar to the evaluation under synthetic traffics, we consider three different network sizes, that is, 64-core, 128-core and 256-core systems.

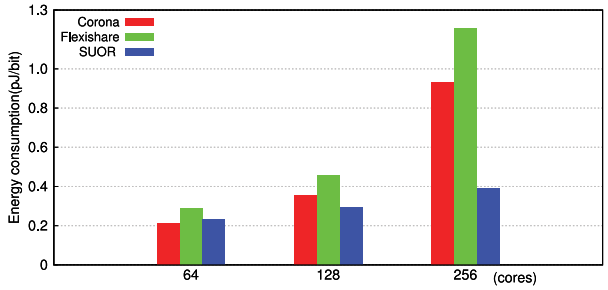

Fig. 17. The comparisons of energy consumption per bit for different network sizes.

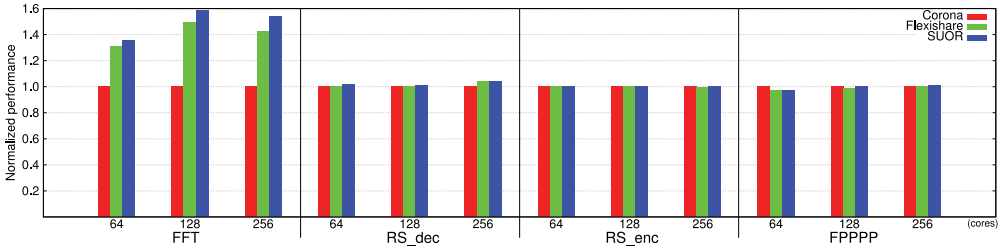

Fig. 18. Normalized performance results for real applications.

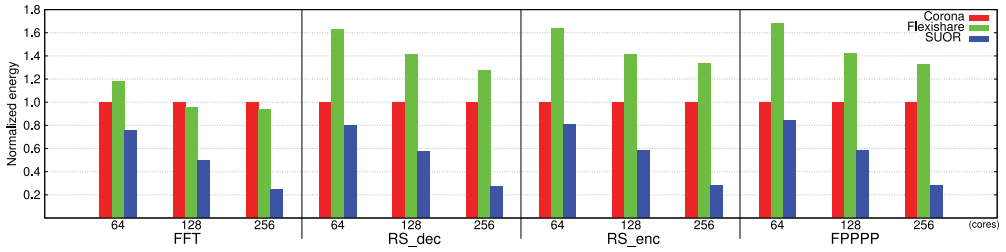

Fig. 19. Normalized energy consumption for real applications.

The performance results are shown in Figure 18. Under FFT traffic, SUOR improves the performance of Corona by around 49% in average; the performance is also 6% higher than that of Flexishare. Under the other three traffics, three networks achieve very similar performances. A closer look at the applications reveals that these traffics are with very low injection rates (less than 0.02 flit/cycle), while the network saturation points are much higher (larger than 0.1 flit/cycle).

The energy consumptions of three networks are shown in Figure 19. SUOR achieves very high energy efficiency under all traffic patterns. In 64-core system, on average SUOR saves 20% and 47% of energy compared with Corona and Flexishare respectively. When the system scales to 256-core, SUOR saves more than 70% of energy compared with the other two designs. The high energy efficiency of SUOR is achieved by both low dynamic-power and low static-power consumptions. The channel segmentation significantly reduces the power loss on the path and thus the laser power when transmitting data, saving dynamic power effectively. On the other hand, when the sender is idle, the on-chip laser can be turned off to significantly reduce the static power. To show the combinational effects of both energy and the performance, the comparisons of energy delay product (EDP) is given in Figure 20. Although the high-throughput merit

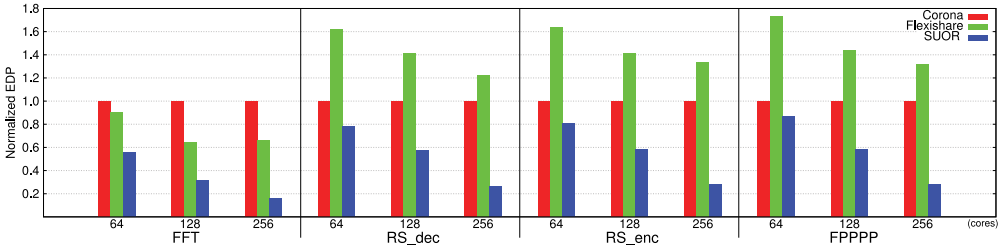

Fig. 20. Normalized energy delay product(EDP) for real applications.

of SUOR fails to be shown when the traffic load is very low, the high energy-efficiency is demonstrated clearly under all traffics. As a result, SUOR achieves lowest EDP for all applications. And the gap between SUOR and the other two becomes larger as the network size grows. In the 256-core system, the EDP of SUOR is less than 25% of EDPs of the other two networks.

5. CONCLUSIONS

The advances in nanophotonics have motivated us to exploit the benefits of optical interconnects for future CMP with a large number of cores. In this article, we propose a new ring-based ONoC, called SUOR. Resources are effectively shared by all clusters. Efficient arbitration scheme has been proposed to cope with the sharing overhead. Power loss is also minimized by shortening and simplifying the path. Compared with the alternatives, SUOR achieves much higher throughput and also higher energy efficiency.

REFERENCES

Shirish Bahirat and Sudeep Pasricha. 2009. Exploring hybrid photonic networks-on-chip for emerging chip multiprocessors. In *Proceedings of the 7th IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis*. ACM, New York, 129–136.

S. Bartolini and P. Grani. 2012. A simple on-chip optical interconnection for improving performance of coherency traffic in CMPs. In *Proceedings of the 15th Euromicro Conference on Digital System Design (DSD)*. 312–318.

Christopher Batten, Ajay Joshi, Jason Orcutt, Anatoly Khilo, Benjamin Moss, Charles Holzwarth, Milos Popovic, Hanqing Li, Henry Smith, Judy Hoyt, Franz Kartner, Rajeev Ram, Vladimir Stojanovic, and Krste Asanovic. 2008. Building many core processor-to-DRAM networks with monolithic silicon photonics. In *Proceedings of the 16th IEEE Symposium on High Performance Interconnects*. 21–30.

M. Bruensteiner and G. C. Papen. 1999. Extraction of VCSEL rate-equation parameters for low-bias system simulation. *IEEE J. Sel. Top. Quantum Electron.* 5, 3, 487–494. DOI:http://dx.doi.org/10.1109/2944.788410

Mark Cianchetti, Nicolás Sherwood-Droz, and Christopher Batten. 2010. Implementing system-in-package with nanophotonic interconnect. In *Proceedings of the Workshop on the Interaction between Nanophotonic Devices and Systems*.

Mark J. Cianchetti, Joseph C. Kerekes, and David H. Albonesi. 2009. Phastlane: a rapid transit optical routing network. In *Proceedings of the 36th Annual International Symposium on Computer Architecture*. ACM, New York, 441–450.

William Dally and Brian Towles. 2003. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann.

W. J. Dally and B. Towles. 2001. Route packets, not wires: on-chip interconnection networks. In *Proceedings of the Design Automation Conference*. 684–689.

I. Datta, D. Datta, and P. P. Pande. 2012. BER-based power budget evaluation for optical interconnect topologies in NoCs. In *Proceedings of the IEEE International Symposium on Circuits and Systems,*. 2429–2432.

Duo Ding, Bei Yu, and D. Z. Pan. 2012. GLOW: A global router for low-power thermal-reliable interconnect synthesis using photonic wavelength multiplexing. In *Proceedings of the 17th Asia and South Pacific Design Automation Conference*. 621–626.

Po Dong, Wei Qian, Shirong Liao, Hong Liang, Cheng-Chih Kung, Ning-Ning Feng, R. Shafiiha, J. Fong, Dazeng Feng, Ashok V. Krishnamoorthy, and M. Asghari. 2010. Low loss silicon waveguides for application of optical interconnects. In *Proceedings of the IEEE Photonics Society Summer Topical Meeting Series*. 191–192.

Eric Dulkeith, Fengnian Xia, Laurent Schares, William M. J. Green, and Yurii A. Vlasov. 2006. Group index and group velocity dispersion in silicon-on-insulator photonic wires. *Opt. Express* 14, 9, 3853–3863. DOI:http://dx.doi.org/10.1364/OE.14.003853

Minming Geng, Lianxi Jia, Lei Zhang, Lin Yang, Ping Chen, Tong Wang, and Yuliang Liu. 2009. Four-channel reconfigurable optical add-drop multiplexer based on photonic wire waveguide. *Opt. Express* 17, 7, 5502–5516. DOI:http://dx.doi.org/10.1364/OE.17.005502

Cary Gunn. 2006. CMOS Photonics for high-speed interconnects. *IEEE Micro* 26, 2, 58–66. DOI:http://dx.doi.org/10.1109/MM.2006.32

Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar. 2007. A 5-GHz mesh interconnect for a teraflops processor. *IEEE Micro* 27, 5), 51–61. DOI:http://dx.doi.org/10.1109/MM.2007.4378783

Chen Ji, Jingyi Wang, David Söderström, and Laura Giovane. 2009. High data rate 850 nm oxide VCSEL for 20 Gb/s application and beyond. In *Proceedings of the Asia Communications and Photonics Conference and Exhibition*. Optical Society of America.

A. Joshi, C. Batten, Yong-Jin Kwon, S. Beamer, I. Shamim, K. Asanovic, and V. Stojanovic. 2009. Siliconphotonic clos networks for global on-chip communication. In *Proceedings of the 3rd ACM/IEEE International Symposium on Networks-on-Chip*. 124–133.

Yu-Hsiang Kao and H. J. Chao. 2011. BLOCON: A Bufferless Photonic Clos network-on-chip architecture. In *Proceedings of the 5th IEEE/ACM International Symposium on Networks on Chip*. 81–88.

Nevin Kirman, Meyrem Kirman, Rajeev K. Dokania, Jose F. Martinez, Alyssa B. Apsel, Matthew A. Watkins, and David H. Albonesi. 2006. Leveraging Optical Technology in Future Bus-based Chip Multiprocessors. In *Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE, 492–503.

J. Kleinberg and E. Tardos. 2005. *Algorithm Design*. Addison-Wesley.

Pranay Koka, Michael O. McCracken, Herb Schwetman, Xuezhe Zheng, Ron Ho, and Ashok V. Krishnamoorthy. 2010. Silicon-photonic network architectures for scalable, power-efficient multi-chip systems. In *Proceedings of the 37th Annual International Symposium on Computer Architecture*. ACM, New York, 117–128.

S. Koohi, M. Abdollahi, and S. Hessabi. 2011. All-optical wavelength-routed NoC based on a novel hierarchical topology. In *Proceedings of the 5th IEEE/ACM International Symposium on Networks on Chip*. 97–104.

Ashok V. Krishnamoorthy, K. W. Goossen, W. Jan, Xuezhe Zheng, R. Ho, Guoliang Li, R. Rozier, F. Liu, D. Patil, J. Lexau, H. Schwetman, Dazeng Feng, M. Asghari, T. Pinguet, and J. E. Cunningham. 2011. Progress in low-power switched optical interconnects. *IEEE J. Sel. Top. Quantum Electron.* 17, 2, 357–376. DOI:http://dx.doi.org/10.1109/JSTQE.2010.2081350

Ashok V. Krishnamoorthy, Ron Ho, Xuezhe Zheng, H. Schwetman, Jon Lexau, P. Koka, Guo Liang Li, I. Shubin, and J. E. Cunningham. 2009. Computer Systems based on silicon photonic interconnects. *Proc. IEEE* 97, 7, 1337–1361. DOI:http://dx.doi.org/10.1109/JPROC.2009.2020712

C. Kromer, G. Sialm, C. Berger, T. Morf, M. L. Schmatz, F. Ellinger, D. Erni, G.-L. Bona, and H. Jackel. 2005. A 100-mW 4X10 Gb/s transceiver in 80-nm CMOS for high-density optical interconnects. *IEEE J. Solid-State Circuits* 40, 12, 2667–2679. DOI:http://dx.doi.org/10.1109/JSSC.2005.856575

S. Le Beux, J. Trajkovic, I. O'Connor, G. Nicolescu, G. Bois, and P. Paulin. 2011. Optical ring network-on-chip (ORNoC): Architecture and design methodology. In *Design, Automation Test in Europe Conference Exhibition*. 1–6.

Zheng Li, Dan Fay, Alan Mickelson, Li Shang,Manish Vachharajani, Dejan Filipovic, Wounjhang Park, and Yihe Sun. 2009. Spectrum: a hybrid nanophotonic-electric on-chip network. In *Proceedings of the 46th Annual Design Automation Conference (DAC'09)*. ACM, New York, 575–580.

Weichen Liu, Jiang Xu, Xiaowen Wu, Yaoyao Ye, Xuan Wang, Wei Zhang, M. Nikdast, and Zhehui Wang. 2011. A NoC traffic suite based on real applications. In *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*. 66–71. DOI:http://dx.doi.org/10.1109/ISVLSI.2011.49

G. Masini, G. Capellini, J. Witzens, and C. Gunn. 2007. A 1550nm, 10 Gbps monolithic optical receiver in 130nm CMOS with integrated Ge waveguide photodetector. In *Proceedings of the 4th IEEE International Conference on Group IV Photonics*. 1–3.

R. Morris, E. Jolley, and A. Karanth Kodi. 2013. Extending the performance and energy-efficiency of shared memory multicores with nanophotonic technology. *IEEE Trans. Parallel Distrib. Syst.* 99, 1. DOI:http://dx.doi.org/10.1109/TPDS.2013.26

R. Morris Jr., A. Kodi, A. Louri, and R. Whaley. 2012. 3D stacked nanophotonic network-on-chip architecture with minimal reconfiguration. *IEEE Trans Computers* 99, 1. DOI:http://dx.doi.org/10.1109/TC.2012.183

C. Nitta, M. Farrens, and V. Akella. 2011. Addressing system-level trimming issues in on-chip nanophotonic networks. In *Proceedings of the IEEE 17th International Symposium on High Performance Computer Architecture*. 122–131.

Ian O'Connor. 2004. Optical solutions for system-level interconnect. In *Proceedings of the International Workshop on System Level Interconnect Prediction*. ACM, New York, 79–88. DOI:http://dx.doi.org/10.1145/966747.966764

Jin Ouyang, Chuan Yang, Dimin Niu, Yuan Xie, and Zhiwen Liu. 2011. F2BFLY: an on-chip free-space optical network with wavelength-switching. In *Proceedings of the International Conference on Supercomputing (ICS'11)*. ACM, New York, 348–358.

J. D. Owens, W. J. Dally, R. Ho, D. N. Jayasimha, S. W. Keckler, and Li-Shiuan Peh. 2007. Research challenges for on-chip interconnection networks. *IEEE Micro* 27, 5, 96–108. DOI:http://dx.doi.org/10.1109/MM.2007.4378787

Yan Pan, Yigit Demir, Nikos Hardavellas, John Kim, and Gokhan Memik. 2010a. Exploring benefits and designs of optically-connected disintegrated processor architecture. In *Proceedings of the Workshop on the Interaction between Nanophotonic Devices and Systems*.

Yan Pan, J. Kim, and G. Memik. 2010b. FlexiShare: Channel sharing for an energy-efficient nanophotonic crossbar. In *Proceedings of the IEEE 16th International Symposium on High Performance Computer Architecture*. 1–12.

Yan Pan, Prabhat Kumar, John Kim, Gokhan Memik, Yu Zhang, and Alok Choudhary. 2009. Firefly: illuminating future network-on-chip with nanophotonics. In *Proceedings of the International Symposium on Computer Architecture*. 429–440. DOI:http://dx.doi.org/10.1145/1555815.1555808

S. Pasricha and N. Dutt. 2008. ORB: An on-chip optical ring bus communication architecture for multiprocessor systems-on-chip. In *Proceedings of the Asia and South Pacific Design Automation Conference*. 789–794. DOI:http://dx.doi.org/10.1109/ASPDAC.2008.4484059

S. Poddar, P. Ghosal, P. Mukherjee, S. Samui, and H. Rahaman. 2012. Design of an NoC with on-chip photonic interconnects using adaptive CDMA links. In *Proceedings of the IEEE International System On Chip Conference*. 352–357.

J. Psota, J. Miller, G. Kurian, H. Hoffman, N. Beckmann, J. Eastep, and A. Agarwal. 2010. ATAC: Improving performance and programmability with on-chip optical networks. In *Proceedings of the IEEE International Symposium on Circuits and Systems*. 3325–3328. DOI:http://dx.doi.org/10.1109/ISCAS.2010.5537892

S. Pradhan, Q. Xu, B. Schmidt, and M. Lipson. 2005. Micrometre-scale silicon electro-optic modulator. *Nature*.

Zhiliang Qian, Ying Fei Teh, and Chi-Ying Tsui. 2012. A flit-level speedup scheme for network-on-chips using self-reconfigurable bi-directional channels. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*. 1295–1300.

Luca Ramini, Paolo Grani, Sandro Bartolini, and Davide Bertozzi. 2013. Contrasting wavelength-routed optical NoC topologies for power-efficient 3d-stacked multicore processors using physical-layer analysis. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*. 1589–1594.

Assaf Shacham, Keren Bergman, and Luca P. Carloni. 2008. Photonic Networks-on-Chip for Future Generations of Chip Multiprocessors. *IEEE Trans. Comput.* 57, 9, 1246–1260. DOI:http://dx.doi.org/10.1109/TC.2008.78

Hercules Simos, Charis Mesaritakis, Dimitris Alexandropoulos, and Dimitris Syvridis. 2009. Dynamic analysis of crosstalk performance in microring-based add/drop filters. *J. Lightwave Technol*. 27, 12, 2027–2034.

A. Syrbu, A. Mereuta, V. Iakovlev, A. Caliman, P. Royo, and E. Kapon. 2008. 10 Gbps VCSELs with high single mode output in 1310nm and 1550 nm wavelength bands. In *Proceedings of the Optical Fiber communication/National Fiber Optic Engineers Conference.* 1–3. DOI:http://dx.doi.org/10.1109/OFC.2008.4528529

D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. P. Jouppi, M. Fiorentino, A. Davis, N. Binkert, R. G. Beausoleil, and J. H. Ahn. 2008. Corona: System implications of emerging nanophotonic technology. In *Proceedings of the 35th International Symposium on Computer Architecture*. 153–164. DOI:http://dx.doi.org/10.1109/ISCA.2008.35

Fengnian Xia, Lidija Sekaric, and Yurii Vlasov. 2007. Ultracompact optical buffers on a silicon chip. *Nat Photonics* 1, 1, 65–71.

Yi Xu, Yu Du, Youtao Zhang, and Jun Yang. 2011. A composite and scalable cache coherence protocol for large scale CMPs. In *Proceedings of the International Conference on Supercomputing (ICS'11)*. ACM, New York, 285–294.

Yi Xu, Jun Yang, and Rami Melhem. 2012. Channel borrowing: An energy-efficient nanophotonic crossbar architecture with light-weight arbitration. In *Proceedings of the 26th ACM International Conference on Supercomputing (ICS'12)*. ACM, New York, 133–142.

Yaoyao Ye, Lian Duan, Jiang Xu, Jin Ouyang, Mo Kwai Hung, and Yuan Xie. 2009. 3D optical networks-on-chip (NoC) for multiprocessor systems-on-chip (MPSoC). In *Proceedings of the IEEE International Conference on 3D System Integration*. 1–6. DOI:http://dx.doi.org/10.1109/3DIC.2009.5306588

Yaoyao Ye, Jiang Xu, Xiaowen Wu, Wei Zhang, Weichen Liu, and Mahdi Nikdast. 2012. A torus-based hierarchical optical-electronic network-on-chip for multiprocessor system-on-chip. *J. Emerg. Technol. Comput. Syst.* 8, 1, 5:1–5:26. DOI:http://dx.doi.org/10.1145/2093145.2093150

Yaoyao Ye, Jiang Xu, Xiaowen Wu, Wei Zhang, Xuan Wang, M. Nikdast, Zhehui Wang, and Weichen Liu. 2011. Modeling and analysis of thermal effects in optical networks-on-chip. In *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*. 254–259. DOI:http://dx.doi.org/10.1109/ISVLSI.2011.38

Yan Zheng, P. Lisherness, Ming Gao, J. Bovington, Kwang-Ting Cheng, Hong Wang, and Shiyuan Yang. 2012. Power-efficient calibration and reconfiguration for optical network-on-chip. *IEEE / OSA J. Opt. Commun. Networking* 4, 12, 955–966. DOI:http://dx.doi.org/10.1364/JOCN.4.000955

Received December 2012; revised April 2013, July 2013; accepted September 2013