NORTHWESTERN UNIVERSITY

High Performance and Energy Efficient Computer System Design Using Photonic Interconnects

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Electrical Engineering and Computer Science

By

YIGIT DEMIR

EVANSTON, ILLINOIS
August 2015

This work was partially supported by NSF awards CCF-1218768 and CCF-1453853.

Abstract

Silicon photonics have emerged as a promising solution to meet the growing demand for high-bandwidth, low-latency, and energy-efficient communication in manycore and multi-chip processors. Multi-chip designs can leverage nanophotonic interconnects to realize high performance "virtual chips" with the aggregate area and performance much higher than the single-chip multiprocessors. While much work has been done on the architecture, design, and analysis of optical interconnects, the laser is generally excluded from the analysis, and assumed to be lumped into a static off-chip power overhead that we should not be concerned much about. Unfortunately, the wall-plug laser consumption constitutes a major component of the overall power consumption of an optical interconnect. Another major contributor of the optical interconnect's power consumption is the ring-heating power. Photonic interconnects are primarily based on microrings, which are highly sensitive to temperature. As a result, current silicon-photonic interconnect designs expend a significant amount of energy heating the microrings to a designated narrow temperature range, only to have the majority of the thermal energy waste away and dissipate through the heat sink, and in the process of doing so heat up the logic layer, causing significant performance degradation to the cores and inducing thermal emergencies. As a result, the laser power and the ring heating power consumption remains a potential issue that needs to be addressed.

The scalability trends of modern semiconductor technology lead to increasingly dense multicore chips. Unfortunately, physical limitations in area, power, off-chip bandwidth, and yield constrain single-chip designs to a relatively small number of cores, beyond which scaling becomes

impractical. Multi-chip designs overcome these constraints, and can reach scales impossible to realize with conventional single-chip architectures. However, to deliver commensurate performance, multi-chip architectures require a cross-chip interconnect with bandwidth, latency, and energy consumption well beyond the reach of electrical signaling. We propose Galaxy, an architecture that enables the construction of a many-core "virtual chip" by connecting multiple smaller chiplets through optical fibers. The low optical loss of fibers allows the flexible placement of chiplets, and offers simpler packaging, power, and heat requirements. At the same time, the low latency and high bandwidth density of optical signaling maintain the tight coupling of cores, allowing the virtual chip to match the performance of a single chip that is not subject to area, power, and bandwidth limitations. Our results indicate that Galaxy is up to 3.4x faster (1.8-2.2x on average) over single-chip alternatives and achieves up to 6.8x smaller energy-delay product (2.6x on average), and scales to 4K cores while being 2.5x faster at 6x lower laser power than a waveguide-based design.

Photonic interconnects provide low cost signal modulation, and low latency communication, however, high optical loss of many nanophotonic components results in high power requirements for the laser source and thermal sensitivity of photonic devices force designers to have power hungry chip level ring-heating solutions. In this work, we propose EcoLaser, an adaptive laser control mechanism that saves laser power by turning the laser off when not needed, while at the same time meeting high bandwidth requirements by leaving the laser on longer. Our results indicate that EcoLaser saves up to 77% laser energy. Furthermore, we propose ProLaser, which is a laser control scheme that improves the EcoLaser scheme by keeping the majority of the data-bus inactive while sending small (dataless) messages, and anticipating upcoming messages to turn the

lasers on ahead of time. Our results indicate that ProLaser achieves even higher energy savings (up to 88%). On top of that, the power savings of ProLaser allow for providing a higher power budget to the cores, which enables them to run faster. Employing ProLaser on a topology with SWMR crossbars (Firefly [57]) allows the multicore to achieve 1.5-1.7x speedup (1.6x on average) and attain 35-52% lower energy consumption per instruction (40% on average).

Our results show that laser control is a powerful technique that improves the energy-efficiency of the photonic interconnects, so we extend our laser control techniques to Flattened Butterfly which is a scalable topology. We propose SLAC, a laser control scheme for flattened butterfly network which turns off majority of the network to save laser energy, while maintaining a fully connected network, which removes the laser turn-on latency from the critical path and causes minimal (next to nothing) performance decrease. SLAC turns off majority of the network when the utilization is low to save energy and activates additional stages when the utilization is high to provide better performance. From an on-chip interconnect to a datacenter network, any network with flattened butterfly topology can take advantage of SLAC. Our results show that, for on-chip and multi-chip applications, SLAC can save up to 67% laser energy while reducing the performance by only 2% while running real-world workloads. On a flattened butterfly datacenter network, SLAC saves 79% laser energy on average while running traces collected from university servers.

The nanophotonic devices are highly susceptible to temperature-induced changes, because their refraction index changes rapidly with temperature. In a multicore processor there is a potential for significant variation in temperature, so micro-ring resonators must be stabilized at a higher temperature using ring heaters which may consume significant amount of energy. We propose “Parka”, a nanophotonic NoC that encases the photonic die in a thermal insulator that keeps its

temperature stable with low energy expenditure, while minimizing the spatial and temporal thermal coupling between logic and silicon-photonic components.Our results indicate that Parka reduces the ring heating power by 3.8x on average across our workload suite. Moreover, the energy savings allow for providing a higher power budget to the cores, which enables them to run faster. Parka on a radix-16 crossbar allows the multicore to achieve 11-23% speedup (34% max) over a baseline scheme with no insulation, depending on the cooling solution used. All of the schemes we propose make the photonic interconnects, as well as the multicore processors more energy efficient, which makes the photonic interconnects a more attractive and feasible solution.

Acknowledgements

First of all, I would like to express my gratitude to my advisor Prof. Nikos Hardavellas. Thank you for teaching me how to do research, constantly encouraging me to think differently, challenging my ideas, and working with me side-by-side for countless hours. This work wouldn't existed without your guidance and help.

Special thanks to Prof. Gokhan Memik, Prof. John Kim and Prof. Russ Joseph for making time to serve in my thesis committee and review this work. Thank you for sharing your knowledge and insight, your comments and help have made this work successful.

I would like to thank my collaborators Pan Yan and Seukwoo Song.

I was very lucky to have George and Ali Murat as my labmates, so thank you guys for sharing your thoughts, ideas, food and drinks with me. I am thankful to my roommate Besim for being there for me whenever I needed help, and cooking all those tasty late night snacks.

Lastly, I would like to thank my family Yasemin, Mustafa, Tuna and my girlfriend for their unconditional love and support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Providing high bandwidth density and low latency communication over long distances, with low signal modulation cost nanophotonic interconnects promise to meet performance needs of the scaled-out single-chip multiprocessors and multi-chip designs. Multi-chip designs can leverage nanophotonic interconnects to realize high performance “virtual chips” with the aggregate area and performance much higher than the single-chip multiprocessors. Previous research has focused on designing photonic network topologies, that provide high performance while keeping the power consumption low. However, due to physical limitations of photonic devices and laser sources, the laser power and ring heating power remain the most significant sources of power consumption in photonic interconnects. Schemes that lowers the laser power and ring-heating power consumption improves the energy-efficiency of the photonic interconnects, which makes them even more attractive solution for high performance single-chip and multi-chip processors.

Multi-chip designs can reach scales impossible to realize with conventional single-chip architectures (Macrochip integration). However, to deliver commensurate performance, multi-chip architectures require a cross-chip interconnect with bandwidth, latency, and energy consumption well beyond the reach of electrical signaling. On top of that, the performance and the scalability of the single-chip designs are highly limited due to increased power density and limited off-chip pin counts, whereas multi-chip designs breaks free of these limitations. Motivated by these facts, we propose Galaxy, an architecture that enables

the construction of a many-core "virtual chip" by connecting multiple smaller chiplets through optical fibers. The low optical loss of fibers allows the flexible placement of chiplets, and offers simpler packaging, power, and heat requirements. At the same time, the low latency and high bandwidth density of optical signaling maintain the tight coupling of cores, allowing the virtual chip to match the performance of a single chip that is not subject to area, power, and bandwidth limitations. We evaluate the performance, power, energy, and thermal profile of Galaxy, and compare it against single-chip designs (*processor disintegration*) and multi-chip designs (*macrochip integration*). Galaxy is up to 3.4x faster (1.8-2.2x on average) over single-chip alternatives with electrical, photonic, or hybrid interconnects, achieves up to 6.8x smaller energy-delay product (2.6x on average), and scales to 4K cores while being 2.5x faster at 6x lower laser power than a waveguide-based design.

The high-speed and low-cost modulation of light make photonic interconnects an attractive solution for manycore processors' communication demands. However, high optical loss of many nanophotonic components results in high power requirements for the laser source and thermal sensitivity of photonic devices force designers to have power hungry chip level ring-heating solutions. As a result, the laser power and the ring heating power consumption remains a potential issue that needs to be addressed.

In order to address the high laser power consumption problem, we propose a laser control scheme EcoLaser, which opportunistically turns the laser off during periods of low activity to save energy, and leaves it on during periods of high activity in order to meet the high bandwidth demand. EcoLaser capitalizes on recent advancements in Ge lasers [42,47], which enable energy-efficient on-chip laser sources that can be turned on or off within nanoseconds. We propose a collection of static and dynamic laser control mechanisms and policies that approximate the maximum possible savings, and we present detailed designs of EcoLaser for both SWMR and MWSR optical crossbars. We evaluate the impact of EcoLaser on the

performance and energy of a multicore, and our results indicate that EcoLaser saves between 24-77% of the laser power for radix-16 and radix-64 SWMR and MWSR crossbars real-world workloads.

Improving upon EcoLaser, we propose ProLaser, which is a laser control scheme that achieves higher laser energy savings for all utilization levels while minimizing the additional laser turn-on delay overhead of the laser control, by keeping the majority of the data-bus inactive while sending small (data-less) messages, and anticipating upcoming messages to turn the lasers on ahead of time. We evaluated the impact of ProLaser on the performance and energy of a multicore running a range of synthetic and scientific workloads under realistic physical constraints, and show that it saves between 49-88% of the laser power, it outperforms the current state of the art by 2x on average, and closely tracks (within 2-6%) a perfect prediction scheme with full knowledge of future interconnect requests. On top of that, the power savings of ProLaser allow for providing a higher power budget to the cores, which enables them to run faster. Employing ProLaser on a topology with SWMR crossbars (Firefly [57]) allows the multicore to achieve 1.5-1.7x speedup (1.6x on average) and attain 35-52% lower energy consumption per instruction (40% on average).

Energy proportionality is desirable for not only the on-chip photonic interconnects, but also the multi-chip systems and the datacenters with photonic networks. Such scaled-out systems exploit scalable photonic network topologies such as “flattened butterfly” topology. Flattened butterfly topology provides path-diversity between source and destination pairs, so it can provide high throughput while keeping the hardware cost at bay. Laser power-gating is a promising technique to mitigate high laser power consumption of the photonic interconnects, however, it reduces the performance when messages have to wait for the laser turn-on. We propose SLAC, a laser control scheme for flattened butterfly network which turns off majority of the network to save laser energy, while maintaining a fully connected network which removes the laser turn-on latency from the critical path and causes minimal (next to nothing) performance decrease.

SLAC turns off majority of the network when the utilization is low to save energy and activates additional stages when the utilization is high to provide better performance. From an on-chip interconnect to a data-center network, any network with flattened butterfly topology can take advantage of SLAC. Our results show that, for on-chip and multi-chip applications, SLAC can save up to 67% laser energy while reducing the performance by only 2% while running real-world workloads. On a flattened butterfly datacenter network, SLAC saves 79% laser energy on average while running traces collected from university servers.

Ring heating power consumption remains a potential issue that needs to be addressed. The nano-photonic devices are highly susceptible to temperature-induced changes, because their refraction index changes rapidly with temperature. In a multicore processor there is a potential for significant variation in temperature, so micro-ring resonators must be stabilized at a higher temperature using ring heaters which may consume significant amount of energy. As current silicon-photonic designs are predominantly based on microring resonators which are highly temperature-sensitive devices, these thermal fluctuations in turn throw the microring resonators off-resonance and prevent the optical interconnect from functioning. To keep the microrings resonating at their appropriate wavelengths, the designers employ trimming, which is a technique that dynamically shifts the microring's resonant wavelength towards the red through heating, or shifts it towards the blue through current injection. However, trimming by current injection causes instability and thermal runaways [51], thus microrings are typically maintained at a constant temperature using the heaters only. The microrings are tuned to a temperature slightly over the maximum temperature that the microprocessor reaches, because only the heaters are employed. Unfortunately, this means that the heaters need to work continuously to keep the microrings at such high temperature, and at the same time the majority of the heating power is wasted as it dissipates through the package to the heat sink. As a result, it is common for microring heaters to consume upwards of 40W [51], mostly of which is wasted. To make matters worse, this thermal energy heats up the logic layer to temperatures very close to its operational limit, which

forces the system to throttle the cores, thereby reducing the performance. The runaway heat also increases the frequency and magnitude of the thermal emergencies, and accelerates the aging of the logic die.

The solution we propose is rather simple: thermally decouple the 3D-stacked logic die from the photonics die by introducing an insulating layer between them to maintain higher thermal stability and easier trimming. We propose “Parka”, a nanophotonic NoC that encases the photonic die in a thermal insulator that keeps its temperature stable with low energy expenditure, while minimizing the spatial and temporal thermal coupling between logic and silicon-photonic components. Our results indicate that Parka reduces the ring heating power by 3.8x on average across our workload suite. Moreover, the energy savings allow for providing a higher power budget to the cores, which enables them to run faster. Parka on a radix-16 crossbar allows the multicore to achieve 11-23% speedup (34% max) over a baseline scheme with no insulation, depending on the cooling solution used.

The rest of the document is organized as follows. In Chapter 2, we present the Galaxy Architecture and evaluate its performance and energy characteristics. In Chapter 3, we introduce EcoLaser and ProLaser, we present the SLAC laser control for Flattened Butterfly network in Chapter 4, and we evaluate their energy savings and performance characteristics. We present PARKA, which improves thermal stability of the photonic devices by introducing a 3D architecture with an insulator layer, in Chapter 5. We discuss the future work in Chapter 6, comment on related research in and conclude in Chapter 7.

Chapter 2

Energy-Efficient Disintegrated Processor Design

Advanced silicon fabrication allows for exponentially increasing transistor counts, which allows for increasingly dense multicore chips. However, physical limitations in area, yield, off-chip bandwidth, and power limit the scalability of single chip designs. Area and yield considerations push for small die sizes, and the latest ITRS models reflect the competitive requirements for afford ability by targeting flat chip-size trends for both high-performance and cost-performance processors (lowered to 260 mm^2 and 140 mm^2 respectively [23]). At the same time, while transistor counts grow exponentially, voltage scaling has slowed. This has lead to a dramatic increase in power density with decreasing feature size [31], creating chips that require a power budget beyond what is practical today to operate and leading to "dark silicon" [22,26,49]. Moreover, the limited pin count and low efficiency in off-chip communication severely limit the off-chip bandwidth [61], rendering it increasingly difficult to feed all cores with data fast enough to keep them busy. This bandwidth wall hampers the scalability of future CMPs and their performance, even for highly-parallel workloads [26].

As a result, multicore scalability is being rapidly pushed to an end. Physical constraints limit single chip designs to either a relatively small number of cores, beyond which scaling becomes impractical, or to designs that trade single-core performance for high aggregate instruction throughput, which can only be achieved if all cores are simultaneously employed by the executing workload. For example, a single core in Intel i7-3960X has a peak theoretical performance of 187 *GFLOPS*, but only 6 such cores fit in the chip's area and power budget. In contrast, Intel Phi 5110P features 60 cores, but at only 17 *GFLOPS* per core, and NVIDIA GTX-680 features 1536 CUDA cores but at a paltry 2 *GFLOPS* each.

Alternative designs can break free of some physical limitations, but not all. Aggregating together several discrete smaller dies instead of having a large one (*disintegration*) overcomes the area and yield limitations [13], as only few dies need to be replaced if they are faulty [5,13]. At the same time the total silicon area of the aggregate chip can scale beyond reticle size limits, allowing the aggregate chip to reach scales impossible to realize with a monolithic design (*macrochip integration*). 3D-die stacking can realize these benefits by vertically connecting several smaller dies in a package with through-silicon-vias (TSVs). However, 3D-die stacking incurs significant challenges in power delivery and heat removal, and is best employed when the additional dies implement low-power applications (e.g., DRAM). By contrast, high-power applications (e.g., high-performance processors), are ideally spread out as an array of chips, allowing for power delivery to and heat removal from each individual die directly. Unfortunately, connecting a large array of chips at high bandwidth presents unique challenges.

Limitations in the density of chip I/O and package routes dramatically constrain the number of links that can be routed across chips, and severely constrain bandwidth. A 580 mm^2 die can have 25600 pins to the package substrate at a pitch of 150 μm, but the substrate-to-board pitch is 0.8 *mm* which allows only 3844 pins to the board from a 5 *cm* x 5 *cm* package [23]. This forces the use of over-clocked and high power serial links for chip-to-chip communication. Thus, using electrical links (SerDes) [60] on an FR-4 board incurs significant energy consumption or long delays (20 *pJ/bit* typically, and at best 2.5 *pJ/bit* and 2.5 *ns* latency over 4 inches of electrical strip [60]) as the designers have to trade energy for performance or vise-versa. Silicon interposers (i.e., 2.5D integration) allow chips to connect laterally within the same package through "bridge" silicon chips, thus exploiting the high density of die-to-package and on-chip wires. However, this enables only modest-sized arrays of chips, and their scalability is further limited by the low speed of on-chip wires, especially over distances longer than 10 *mm* [39,41].

With the introduction of nanophotonics, systems can break free of all these limitations. The low latency and high bandwidth density of optical signaling can facilitate efficient off-chip communication and bring physically distant chips effectively close together. This makes it possible to build a physically large but logically dense many-core "virtual chip" by optically connecting several chiplets together [13,39,54].

To integrate chiplets into a larger system, NSiP [13] uses silicon-nitride waveguides across chiplets within a package, and the Oracle Macrochip [39] uses silicon waveguides etched on a wafer. While these proposals mitigate the area, yield, and memory bandwidth limitations of conventional designs, they do not address the power constraints. The high optical loss of silicon waveguides (typically 0.1-0.3 *dB/cm* [8]) makes routing long cross-chiplet optical channels impractical from a power standpoint. Thereby, designs utilizing waveguides are confined to a small physical space (e.g., a wafer [39] or a package [13]). This increases the thermal density to the point where liquid cooling is required to avoid thermal runaways [39,41], or confines the aggregate "virtual chip" to power limitations not much different from a monolithic design [13]. Aggressive technology can produce low-loss waveguides (0.05 *dB/cm* [41]) which enable the wide separation of discrete chiplets. However, these waveguides are 20x wider than conventional ones. Their high area occupancy forces the use of exceedingly narrow chiplet-to-chiplet links (e.g., 2-bit links for an 8x8 chiplet array [39,41]) which in turn imposes significant serialization that degrades performance. Thus, to design a large "virtual chip" using waveguides, one either has to suffer high optical loss which multiplies the power requirements, or employ narrow paths which impose serialization, hurt performance, and in turn increase energy consumption.

In contrast, Galaxy is designed to push back the power constraints, in addition to overcoming the area, yield, and bandwidth limitations, while matching the high performance of unconstrained tightly-coupled chips. Optical fibers have tremendously low optical loss that is measured in kilometers (0.2 *dB/km*), so very long channels can be drawn at very low power. Galaxy uses fibers for cross-chiplet communication, and

also guarantees that each optical path employs only a small fixed number of couplers, keeping the optical loss and the corresponding laser power low. These two design choices allow spreading discrete chiplets far apart in physical space to minimize heat transfer and lower the power density of the virtual chip, which in turn enables each chiplet to operate at a higher frequency than power-limited designs. At the same time, the propagation speed of light in fibers (0.676 *c)* is considerably higher than in silicon waveguides (0.286 *c*), or electrical lines on FR-4 boards (*0.5 c*), allowing for low-latency communication over long distances. Compared to electrical lines, fibers transmit at about 33x lower energy per bit [4].

Previous research [39] dismissed the use of optical fibers for cross-chiplet communication under the assumption that chips connect to fibers at a relatively large 250 μm core pitch, not the 20 μm pitch of optical proximity couplers that silicon waveguides use. Hence, the chip-to-chip bandwidth over fibers would not improve much over area solder balls connected to package routes. Galaxy overcomes this consideration by exploiting new tapered coupler technologies that couple an array of fibers at 250 μm pitch into an array of waveguides at 20 μm pitch at the edge of the chip [44]. Our results indicate that fibers can provide sufficient bandwidth for communication to chiplets and to memory, allowing for much wider data paths than low-loss but slow silicon waveguides, and in turn boost both the performance and the energy efficiency of the multi-chip system by several times.

In summary, optical fibers are faster, impose lower optical loss, and require lower energy than available alternatives for chiplet communication. They are also flexible, allowing for arbitrary placement of chiplets (e.g., across boards within a rack) without the need for additional coupling. Thus, fibers are especially suitable for long, inter-chiplet optical channels, as they are easy to route, and can even go off the plane or off the board. Galaxy utilizes optical fibers for cross-chiplet communication and offers simple packaging, power, and heat requirements, yet provides the performance advantages of a tightly-coupled system. While prior works have touched upon some of these issues in the context of multi-chip architectures [4, 5, 13, 39,

41, 54], to the best of our knowledge, this is the first work that quantifies the impact of disintegration and multi-chip integration on power constraints, and provides an analysis of the performance, power, energy, and thermal characteristics of several multi-chip architecture alternatives.

It is important to note that Galaxy is just one design that supports processor disintegration and macrochip integration. Other topologies and designs are possible. Our goal is not to perform a full design-space sweep and advocate Galaxy as the optimal solution. Rather, we aim to demonstrate that macrochip integration and processor disintegration can match the performance of designs that are not limited by power and off-chip bandwidth, effectively breaking free from the limitations of today's chips. More specifically, our contributions are:

1. We quantify the performance and energy impact of power and bandwidth constraints in monolithic single-chip designs, and the limitations of electrical links and SOI waveguides when used for chip communication.
2. We propose Galaxy, an architecture that allows both processor disintegration and macrochip integration. Galaxy builds a many-core "virtual chip" by connecting multiple smaller chiplets through optical fibers.
3. We evaluate the performance, power, energy, and thermal characteristics of Galaxy, and compare it against single-chip designs (processor disintegration) and multi-chip designs (macrochip integration). Galaxy is up to 3.4x faster (1.8-2.2x on average) over single-chip alternatives with electrical, photonic, or hybrid interconnects, achieves up to 6.8x smaller energy-delay product (2.6x on average), and scales to 4K cores while being 2.5x faster at 6x lower laser power than a waveguide-based design.

2.1 .The Galaxy Architecture

Galaxy builds a physically-large but logically-dense many-core “virtual chip” by optically connecting many discrete chiplets together. Each chiplet consists of a logic die with cores, caches, and support circuits, and a die with photonic devices and waveguides. The two dies are stacked in 3D: electrical signals from the logic die travel via TSVs to the photonic die, where they are converted to optical signals, and vise-versa.

Galaxy utilizes electrical signaling for nearest-neighbor communication within a chiplet, and silicon waveguides for long-distance communication within a chiplet. Silicon waveguides are compatible with CMOS processes [11] and they are more efficient for long-distance on-chip communication than electrical signaling [57], leaving global on-chip wires redundant. The on-chip photonic interconnect extends across chiplets by coupling light to an optical fiber at the edge of the chip [44]. A photonic link in Galaxy consists of an off-chip laser source, optical fibers, fiber to on-chip waveguide couplers, SOI waveguides on the chip, a laser splitter, ring modulators, drop filters, and Germanium-based photodetectors.

2.1.1 Network Topology

Galaxy employs a hybrid electrical/photonic interconnect. It extends Firefly [57] to support cross-chiplet communication at low power by minimizing coupler crossings and the number of sharers of each optical path. Figure 1(a), depicts a 5-chiplet Galaxy design. The colored squares within each chiplet represent routers. The routers within a chiplet are divided into local clusters. Each cluster contains exactly one router per remote chiplet. In our example, there are 4 clusters per chiplet, with 4 routers per cluster. A local cluster in Chiplet 3 consists of neighboring black, orange, blue, and green routers (red outline in Chiplet 3, Figure 1(a)). Each cluster supports a number of cores based on a concentration factor. The cores and routers in a cluster are electrically connected. In our example, we use concentration 1 and an electrical ring

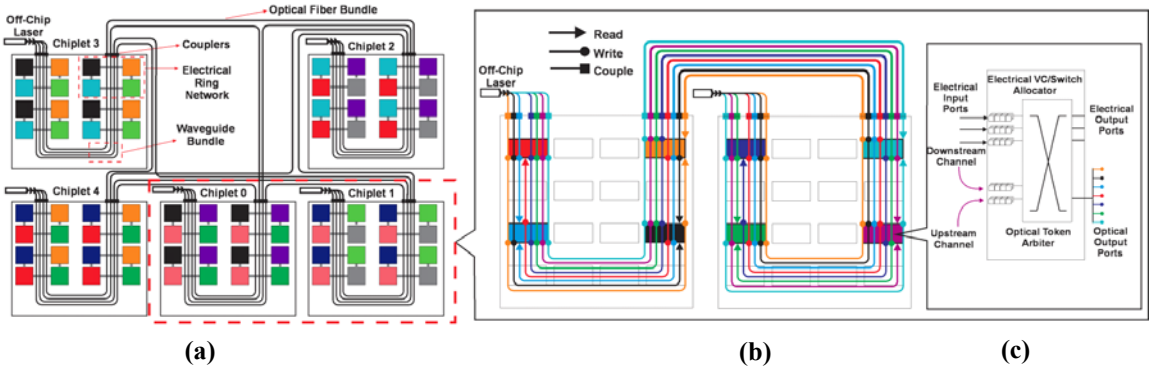


FIGURE 1: (a) Galaxy layout, (b) MWSR optical crossbar, and (c) router architecture.

within the cluster (other topologies are possible). A source-destination pair within the same cluster uses only electrical links.

Clusters communicate with each other through optical crossbars. Every optical crossbar is represented by coloring routers with the same color. For example, the pink routers in Chiplet 0 and the pink routers in Chiplet 1 belong to the same optical crossbar. Each optical crossbar extends across two chiplets. In our example, the crossbar between Chiplet 0 and Chiplet 1 consists of the pink routers in Chiplets 0 and 1, the U-shaped waveguides that connect these routers within each chiplet, and the fibers that connect the two chiplets. Figure 1(b) shows a close-up of that crossbar, where the pink routers have been re-colored to assist a detailed explanation later in the section.

Routing a packet from Chiplet 0 to Chiplet 1 is carried by traversing the corresponding optical crossbar. This is done in 3 steps: (1) Route electrically within the source cluster in Chiplet 0 to a pink router; (2) Take the optical link and arrive at the pink router of the destination cluster in Chiplet 1; (3) Route electrically within the destination cluster to the final destination. Communication between any two clusters is performed similarly. Source-destination clusters within the same chiplet use only the silicon waveguides in that chiplet. If the clusters are at different chiplets, the packet will traverse the waveguides within the source chiplet, the fiber connecting the two chiplets, and the waveguides in the destination chiplet.

In Galaxy, every cluster has as many routers as remote chiplets, and every router in a cluster is connected to a different optical crossbar. Thus Galaxy forms a point-to-point network between chiplets. Also, every crossbar extends across all clusters of the two chiplets it connects. Thus, each cluster has a direct connection to every cluster of every chiplet. A packet that traverses an optical link will directly reach a router in the destination cluster which is very close to the final destination, and every packet traverses the optical link only once. This minimizes coupler crossings and optical loss: every optical path is short because it extends across only two chiplets, and has at most 3 couplers (including the laser coupling).

In general, if each chiplet has X clusters, each with Y routers, and a concentration of c, the proposed Galaxy architecture can connect $(Y+1)$ chiplets, using radix-$(2X)$ optical crossbars, supporting a total of $c*Y*X*(Y+1)$ cores. The example in Figure 1 is a case with $X=Y=4$, $c=1$, for a total of 80 cores.

Firefly [57] uses Single Writer Multiple Reader (SWMR) optical crossbars, which use global broadcast channels to send messages or to reserve a channel, thereby increasing power consumption. Galaxy adopts a modified Firefly topology with Multiple Writer Single Reader (MWSR) optical crossbars. In MWSR crossbars, each router "listens" on a dedicated channel and sends flits on the listening channels of all the other routers in the crossbar. Figure 1(b) illustrates an MWSR crossbar that extends over chiplets 0 and 1, with 8 senders and 8 receivers. Every router is shown with a distinct color. Every router receives data from its own channel, which is shown with the same color as the receiver router, and writes 7 other channels which are the listening channels of the other routers in the crossbar. Galaxy adopts FairQuota [56] to guarantee that only a single router transmits on a channel at any moment, avoid starvation, and provide QoS support.

Figure 1(c) shows a hybrid electrical/optical router in Galaxy. Routers store the flits received from the electrical or optical networks in electrical buffers, after optical to electrical (O/E) conversion if needed. Two electrical input and output ports route packets on the electrical local cluster ring. The third electrical input

and output port is used for data injection. Each router has a pair of dedicated optical receiving channels, the upstream and downstream channels. The dark blue and green routers in Figure 1(b) send messages to the purple router through its upstream channel, while the rest send messages to the purple router through its downstream channel. Thus, 2 extra ports are added on the input side of the router to receive packets from the dedicated optical receiving channels from both directions. On the output side, 7 additional output ports switch outgoing packets to different optical channels.

2.1.2 Switch Arbitration and Flow Control

The electrical switch within each router is arbitrated using conventional electrical arbiters, and uses conventional credit-based flow control. The optical crossbars require arbitration of the optical channels and the buffers at the optical receiving ports. The optical channel arbitration is equivalent to a global switch allocation, and is achieved using a 1-pass optical token stream [73] that extends across two chiplets.

Because the optical links are traversed at most once, at most two Virtual Channels (VCs) are needed for the optical channels. The buffers of each optical VC channel are arbitrated using a separate optical VC token stream. Every router keeps a count of the available buffer space for each VC, and distributes an optical VC token every cycle as long as there is available space. A sender acquires a VC token of its intended VC before entering the arbitration for the data channel. An acquired VC token is held even if the sender fails the subsequent channel arbitration. To keep the balance of VC tokens, the tokens perform a double traversal. The receiver router of a channel first sends the VC tokens in the direction opposite to the data channel (*back-traversal*), all the way to the origin of the laser injection point, skipping all the senders on the way. Then, the VC token goes through O/E and E/O conversion, and is re-modulated onto a VC token stream in the same direction as the data channel (*forward-traversal*). The unused VC tokens eventually arrive back at the receiver and are re-collected to ensure that the receiver always knows how many VC

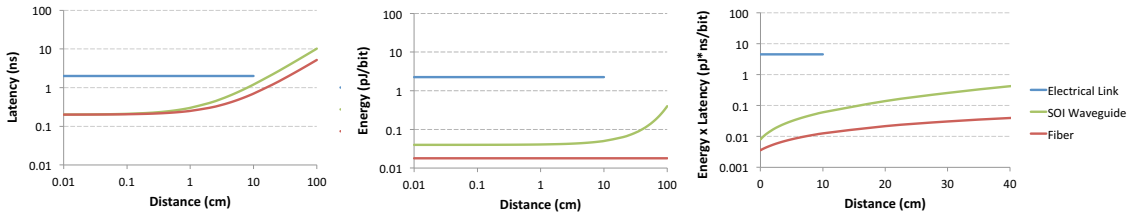


FIGURE 2: Latency, Energy / bit, and Energy x Delay product for electrical links, SOI waveguides, and fibers.

tokens are consumed by the senders. The extra OE/EO conversion at the origin of the data channel ensures that only short optical waveguides are used.

2.1.3 Inter-Chiplet Optical Connection

Galaxy employs optical fibers to connect chiplets, rather than silicon waveguides. Compared to electrical links, fibers offer lower latency, and two orders of magnitude lower energy/bit and energy-delay-product across the entire range of possible chiplet-to-chiplet distances (Figure 2). Similarly, fibers are almost 2x faster than SOI waveguides, and achieve between 2-10x lower energy/bit and energy-delay-product (Figure 2), mainly due to the high relative optical loss of typical silicon waveguides. Extremely low-loss waveguides (0.05 *dB/cm* [41]) reduce the difference in optical loss from 15000x [8,9] to 2500x, but they are much wider (20x) than conventional waveguides. This forces the design of narrow data-paths (e.g., 2-bit chiplet-to-chiplet links for an 8x8 chiplet array [39,41]) which impose serialization and degrade performance. Silicon interposers underperform waveguides (and by extension fibers) for inter-chiplet communication [41], and in addition confine designs to a single package, which in turn increases thermal density and allows only small-scale systems. Thus, fibers are especially suitable for long, inter-chiplet channels. Our findings in Figure 2 corroborate prior research [41].

Fibers connect to chiplets through a coupler that tapers an array of fibers at 250 μm pitch down to 20 μm pitch channels, and couples them into an array of SOI waveguides at the edge of the chip [44]. The mea-

sured coupling loss caused by the refraction index change from fibers to the waveguides including misalignment is 0.8 *dB*, and the internal loss of the coupler caused by tapering the channels is 3 *dB*. Misalignment within 0.7 μm, 0.4 μm, and 0.7 μm in the lateral, vertical, and optical axes produces losses under 1 *dB* [44]. The performance of the tapered coupler is comparable to that of an optical proximity coupler (3.5 *dB* coupler loss, plus 0.5 *dB* per 1 μm misalignment in the y-axis, plus less than 1 *dB* loss due to misalignment within 2.5 μm in the x- and z-axis [80]).

2.1.4 Nanophotonic Parameters and Power Budget

On-chip lasers dissipate a lot of power and heat up the chip, thus Galaxy adopts off-chip WDM-compatible lasers. The laser is brought on chip via optical fibers connected to tapered couplers [44], and a splitter distributes it to low-loss on-chip waveguides [8]. Tapered couplers [44] also transfer the laser from on-chip waveguides to the off-chip optical fibers and vise-versa. Galaxy uses the modulators, demodulators, drop filters, splitters, and detectors introduced in [2]. The modulation and demodulation energy is 150 *fJ/bit* at 10 *GHz* [2]. The optical parameters assumed in Galaxy are detailed in Table 1.

The example configuration of Galaxy which we evaluate in this paper consists of 10 radix-8 MWSR crossbars that transfer 64-bit flits. We assume a modest 16-way DWDM, thus Galaxy uses a total of 320 fibers (128 fibers attached to each chiplet) and 40960 ring resonators (8192 per chiplet). Because every optical channel requires a 1-token-pass arbitration mechanism, a total of 20 additional fibers and 3840 rings are used for arbitration. Another 80 rings and 10 fibers are used for forward clock signal distribution [41].

To calculate the total ring heating power we extend the method by Nitta *et al.* [51] by incorporating the heat generated by the cores. The cores heat up the photonic layer, and the ring heaters provide the remaining heat necessary to bring the photonic layer within the ring tuning range. As current injection may cause a

TABLE 1. Nanophotonic Parameters for Galaxy

	per Unit	Total
Splitters	0.2 *dB*	0.2 *dB*
Waveguide Loss	0.3 *dB/cm*	1.5 *dB*
Fiber Loss	0.2 *dB/Km*	~0 *dB*
Nonlinearity	1 *dB*	1 *dB*
Coupler Loss	3.8 *dB*	7.6 *dB*
Modulator Insertion	0.5 *dB*	0.5 *dB*
Ring Through	0.01 *dB*	1.28 *dB*
Filter Drop	1.5 *dB*	1.5 *dB*
Photodetector	0.1 *dB*	0.1 *dB*
Total Loss		**13.68 *dB***
Detector Sensitivity		**-20 *dBm***
Laser Power per Wavelength		**0.233 *mW***
Total Laser Power		**1.195 *W***

thermal runaway [51], we only consider trimming by heating. Section 2.2.2 details the model. While Galaxy may benefit from trimming power saving methods [51], they are out of the scope of this paper.

Figure 3 demonstrates the sensitivity of Galaxy's laser power to the nanophotonic parameters. The laser power is sensitive to the coupler loss, but relatively insensitive to the other parameters, indicating that our results will likely hold under a wide range of nanophotonic device technologies.

When evaluating electrical links for off-chip communication, existing literature typically omits inefficiencies in the generation and delivery of the electrical power. By analogy, and to ease comparisons with prior work, we didn't include the generation and delivery cost in the laser power calculations presented in the remainder of this paper. For completeness, however, here we calculate the laser power including all these overheads. The additional coupling loss increases the laser power to 2.9*W*. Assuming 10% efficiency for the WDM-compatible off-chip laser [82], the wall-socket laser power is 29*W*.

2.2 .Experimental Methodology

We evaluate the performance of an example 5-chiplet 80-core Galaxy on a full-system cycle-accurate simulation infrastructure using Flexus 4.0 [27,75] integrated with Booksim 2.0 [15] and DRAMSim 2.0 [62]. Table 2 details the architectural modeling parameters. We target a 16nm technology, and have updated our tool chain accordingly based on ITRS projections [23]. We follow the SimFlex sampling methodology [75] with 95% confidence intervals. We model performance as the number of user instructions committed per unit of time [75]. The simulated system executes a selection of SPLASH benchmarks and scientific workloads.

We compare Galaxy against three single-chip CMPs, all of which implement the architecture described in Table 2. The first CMP uses an all-electrical 2D-Concentrated Mesh on-chip interconnect with express links [15] and concentration of 4 (CMeshExp). Concentrated mesh is often chosen for on-chip networks as it maps well to a 2D-VLSI planar layout with low complexity. We evaluated a regular 2D-Mesh and a 2D Concentrated Mesh without express links, and found that CMeshExp outperforms the other designs on all metrics (performance, power, and energy). Thus, we only show results for CMeshExp. We model routers with 8 input and output ports and a 3-cycle routing delay. Routers are connected through 166-bit bi-directional links with a 1-cycle link delay.

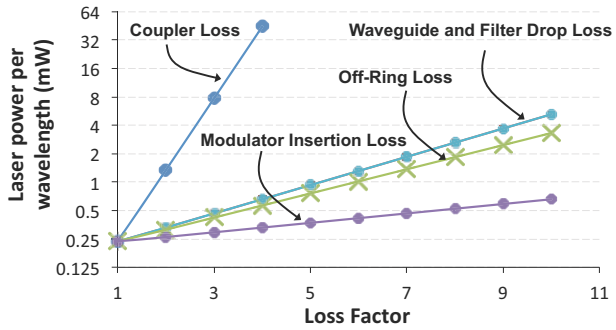


FIGURE 3: Laser power sensitivity to optical parameters.

TABLE 2. Architectural Parameters.

CMP Size	80-cores, 580 mm^2
Processing Cores	ULTRASPARC III ISA, max 5 *GHz*, OoO, 8-stage pipeline, 4-wide dispatch/retirement, 96-entry ROB
L1 Cache	split I/D, 64 *KB* 2-way, 2-cycle load-to-use, 2 ports, 64-byte blocks, 32 MSHRs, 16-entry victim cache
L2 Cache	shared, 512 *KB* per core, 16-way, 64-byte blocks, 14-cycle hit, 32 MSHRs, 16-entry victim cache
Memory Controllers	One per 4 cores, or 4 MCs per chip. 1 channel/MC Round-robin page interleaving;
Main Memory	DDR3, 80 *GB*, 8 *KB* pages, 20 *ns* access latency Interfaces: (a) Conventional pins, (b) Optically-connected memory (OCM) [2], (c) 3D-stacked [39]
Networks	CMesh, Corona, Firefly, Galaxy, Oracle Macrochip

The second CMP uses an all-optical MWSR crossbar (Corona [74]), implemented with 256-bit data channels creating 80 MWSR crossbars. We model global switch arbitration using an optical token ring. A token for each node, which represents the right to modulate on the node's wavelength, continuously passes around all nodes on a dedicated arbitration waveguide. A node grabs and absorbs a token to transmit a packet, and then releases the token to allow other nodes to obtain it. We estimate 16cm long waveguides for the Corona chip, resulting in 8 cycles token round-trip time at 5 *GHz*. The third CMP implements a hybrid interconnect where clusters of electrically-connected cores are connected through an SWMR optical crossbar implemented entirely on chip (Firefly [57]).

We model Galaxy with 1-cycle latency for processing an optical token request [73]. Each Galaxy router can initiate a maximum of 8 token requests per cycle, but can utilize at most 2 acquired tokens [73]. Galaxy uses 1-pass token stream arbitration for combined VC and channel arbitration. We estimate that the round-trip time of a token is also 8 cycles. The input buffers are implemented as a DAMQ [70], with packets queued separately based on their destination. A data packet contains 512 bits, which are divided into eight 64-bit flits.

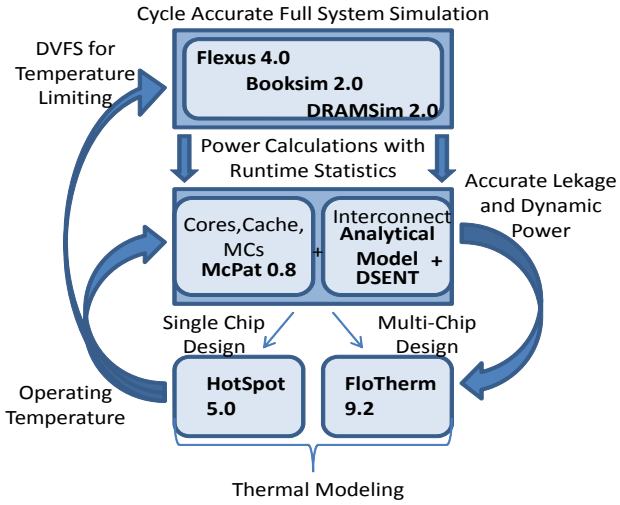


FIGURE 4: Simulation flow chart.

2.2.1 Power and Temperature Modeling

All systems we model employ Dynamic Voltage and Frequency Scaling (DVFS) to lower the voltage and frequency of a chip or chiplet when it reaches the limits of safe operational temperature (without loss of generality, we assume 90^oC). Figure 4 shows the flow diagram of our simulation tool chain. We collect runtime statistics from full-system simulations, and use them to calculate the power consumption of compute cores, caches, and memory controllers using McPAT [46], and the power consumption of the electrical and optical networks using DSENT [69] and the analytical model by Joshi *et al.* [33] respectively. Based on these power estimates, we calculate the temperature of the chip and chiplet assemblies using HotSpot 5.0 [67] and FloTherm [77], a computational fluid dynamics tool that models the heat transfer between chiplets through air flow and convection. The estimated temperature is then used to refine the leakage power estimate, and we iteratively calculate the power and temperature profiles until the system reaches a stable state. We use the stable-state power and temperature estimates to adjust DVFS, and repeat the process until we identify a DVFS setting for which the chip stays just below 90^oC, or operates at the maximum 5 *GHz*.

2.2.2 Resonant Ring Heater Modeling

To calculate the total ring heating power for Galaxy, Corona, and Firefly, we extend the method by Nitta *et al.* [51] by additionally accounting for the heating of the photonic die by the operation of the cores. We model the thermal characteristics of a 3D-stacked architecture where the photonic die sits underneath the logic die using the 3D-chip extension of HotSpot [67]. For each target architecture (Corona, Firefly, and Galaxy) we measure the maximum temperature of the logic die during the execution of each one of the workloads. Then, we tune the micro-rings to the maximum of all the observed temperatures that the logic layer reaches across all benchmarks executing on the target architecture, plus a small margin. When a workload executes, we calculate the ring heating power required to maintain the entire photonic die at the micro-ring trimming temperature during the entire execution.

2.2.3 Modeling Memory and Physical Constrains

To demonstrate the ability of disintegrated architectures to break free of power and bandwidth limitations, we evaluate Galaxy against all possible single-chip CMP combinations: power-constrained, off-chip bandwidth-constrained, fully constrained (i.e., power and bandwidth), and unconstrained.

We evaluate power-constrained CMPs by employing DVFS to keep the chips within $90^{o}C$. To evaluate CMPs that are not subject to power constraints, we allow the chips to run at the maximum speed allowed by the design (5 *GHz*), by disregarding power and thermal limits. We evaluate bandwidth-constrained single-chip CMPs by assuming a conventional DDR3 memory, and limit the total memory bandwidth by utilizing ITRS [23] pin projections for a 5 *cm* x 5 *cm* package, assuming 1/3 of the pins are used for power, 1/3 are used for I/O, and the remaining 1/3 are used for memory. The memory pins are distributed equally among four memory controllers (MCs). To evaluate designs that are not limited by memory bandwidth, we increase the number of pins well beyond ITRS projections and commensurately increase the number of

TABLE 3. Scalability of Galaxy.

# of Cores	Multi-Chip Architecture		Bandwidth per Chip (TB/s)	Laser Power (W)	Serialization Overhead (cycles)	Link Latency (cycles)
320	Galaxy	Fibers	10	4.0	1	2
		Waveguides	5	4.9	2	10
		Electrical links	0.320	3.9	32	12
1088		Fibers	20	27.0	2	10
		Waveguides	5	26.0	8	20
		Electrical links	0.640	26.8	64	12
4160		Fibers	40	47.6	4	10
		Waveguides	10	44.9	16	20
		Electrical links	0.320	47.9	512	12
4096	Oracle MacroChip		0.630	~40.0	64	20

MCs, until more pins or more MCs no longer increase performance. For our workloads, we reach this point when 5x more pins are distributed across 20 MCs. Fully constrained designs operate within the power, memory bandwidth, and thermal limits. Fully unconstrained designs operate beyond the power, thermal, and bandwidth limits and cannot realistically be built; however, they provide the highest performance that a particular architecture can achieve, limited only by the maximum speed allowed by the design (5 *GHz* in our evaluation). While we compare Galaxy to both constrained and unconstrained single-chip CMPs, Galaxy is always modeled to conform to realistic power, bandwidth, and temperature limits.

Emerging memory technologies such as optically-connected memory (OCM) [2] or 3D-stacked memory [39] are not pin-limited, and can remove the memory bandwidth bottleneck for all CMP designs. Thus, we separately evaluate the performance of Galaxy against single-chip CMPs with OCM and 3D-memory, where each CMP employs 20 MCs (additional MCs provide no benefit). We model a 10 *ns* access latency for OCM [2] and a 2 *ns* access latency for 3D-Memory [39].

2.2.4 Modeling Large-Scale Designs

Galaxy can scale up to 1088 cores with 17 chiplets (64 cores each with concentration 4), and 4160 cores with 65 chiplets. When increasing the number of chiplets, we decrease the width of chip-to-chip links to

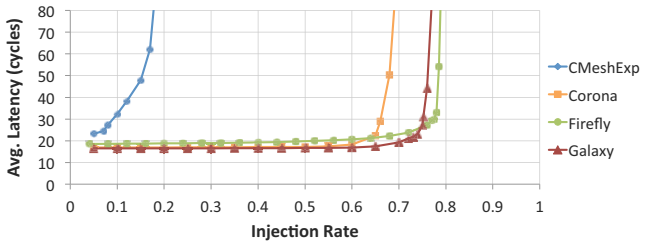


FIGURE 5: Load latency under uniform random traffic.

keep the network power consumption and component count within reasonable levels, and we faithfully model the serialization delay due to narrower datapaths, and increased link latency due to longer links. We evaluate the scalability of Galaxy by comparing it against (a) Galaxy with SOI waveguides and optical proximity (OPC) couplers [80], (b) Galaxy with electrical links (SerDes), and (c) the Oracle Macrochip [39]. For fairness, we adjust the datapath width of Galaxy alternatives so they fit into similar power envelopes, and then calculate the latency overhead. The Oracle Macrochip model closely follows [39,80]. Table 3 details the characteristics of each design. To keep the simulations tractable, we estimate the performance of the scaled-out designs by imposing the latency overheads of each scaled-out system from Table 3 on an 80-core 5-chiplet model. As we impose the scaling overheads onto same-size designs in all cases (80 cores, 5 chiplets), the higher core count of Galaxy compared to the Oracle Macrochip does not affect the results.

2.3 .Experimental Results

2.3.1 Network Performance

Figure 5 analyzes the load-latency of CMeshExp, Corona, Firefly, and Galaxy. CMeshExp saturates quickly, which is indicative of its relatively low bandwidth. Corona saturates at a little less than 0.7 injection rate, while Firefly reaches an injection rate of almost 0.8 before saturating. Galaxy trails Firefly closely, and falls only slightly short in performance. This is expected because Galaxy is similar to a 2-level

Firefly that creates a single datapath between two clusters, while packets in Firefly can take several alternate routes and utilize more of the available bandwidth. Nonetheless, the small difference indicates that Galaxy is a competitive interconnect.

2.3.2 Comparison to Single-Chip Designs

Figure 6 shows the speedup achieved by unconstrained single-chip designs (top of blue bar) with CMesh-Exp, Corona, and Firefly interconnects for memory-intensive and compute-intensive workloads. Submitting the CMPs running compute-intensive workloads (Figure 6 right) to realistic bandwidth constraints results in lower performance, but the loss is relatively small (top of green bar). Submitting them to power constraints, however, results in significant performance drop (top of orange bar). These CMPs employ DVFS to stay below $90^{o}C$, which slows down the compute-intensive workloads the most, as they have high core utilization which in turn dissipates more power. For example, Corona runs barnes at only 2.25 *GHz* from a nominal frequency of 5 *GHz*, and Firefly exhibits a similar slowdown. In comparison, Galaxy never exceeds $70^{o}C$, and thus it can run at the full 5 *GHz* and outperform all single-chip alternatives by 1.8x on average. CMPs running memory-intensive workloads also show degraded performance when power-constrained (Figure 6 left, top of orange bar), indicating that power limitations are always an important factor. However, they incur the highest performance loss mainly when limited in off-chip bandwidth (top of green bar), while the slowdown due to DVFS is secondary. For example, CMeshExp runs em3d at 4.25 *GHz*, but Galaxy still demonstrates 3x speedup. Because of this dual slowdown, Galaxy achieves the maximum speedup over fully-constrained single-chip CMPs (their performance is indicated by the top of the black bar) on memory-intensive workloads (2.3x on average, and up to 3.46x for ocean). More importantly, Galaxy manages to match or exceed the performance of designs that are entirely unconstrained. This demonstrates the ability of processor disintegration to break free of the power and bandwidth walls of

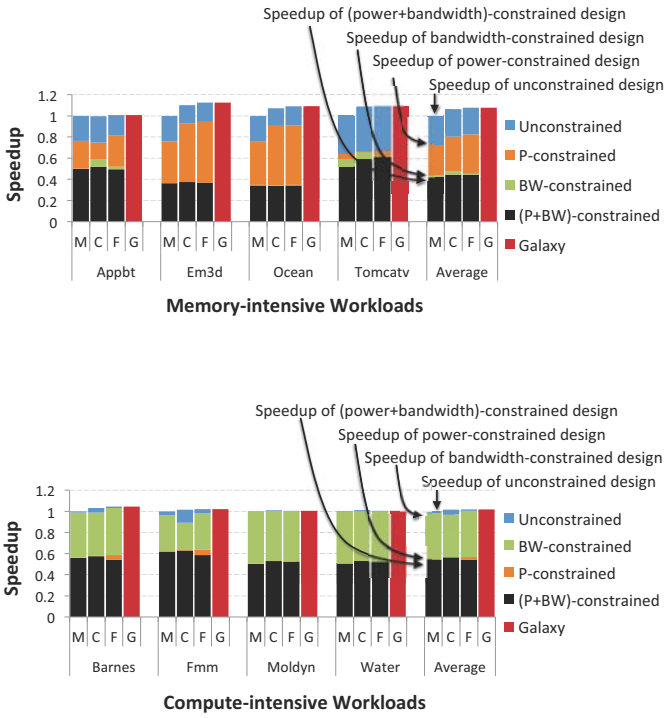


FIGURE 6: Speedup of constrained and unconstrained architectures: CMeshExp (M), Corona (C), Firefly (F), and Galaxy (G).

conventional monolithic designs. All the designs we evaluate in the remainder of this paper are subject to power constraints, while the bandwidth limitations depend on the assumed memory technology.

Optically-connected memory (OCM) [2] is able to overcome the bandwidth limitations and decrease the memory latency. Corona with OCM outperforms Corona with conventional DDR3 by 3-4x on memory intensive workloads (Figure 7). Firefly and CMeshExp show similar trends. Galaxy, however, still outperforms all alternatives by 1.8x on average, as it runs at the full 5 *GHz* while DVFS limits the single-chip designs (e.g., Corona with OCM runs em3d at 3.25 *GHz*). 3D-stacked memory has a similar effect on Galaxy, while Corona, Firefly, and CMeshExp do not get faster as they are still power limited. Overall, Galaxy outperforms alternative designs by up to 2.95x (2x on average). We conclude that Galaxy can leverage the

emerging memory technologies to the fullest extent, while single-chip CMPs are limited by the single-chip power envelope and fail to utilize fully the new memory technologies.

Figure 8 shows the breakdown of the normalized energy-delay product (EDP) and the average energy per instruction of CMeshExp, Corona, Firefly, and Galaxy with conventional memory. The dynamic energy consumption of cores and caches for Galaxy is higher as it achieves 2.3x speedup on average over single-chip designs. This effect is more pronounced for compute-intensive workloads (barnes, moldyn). However, the chiplets in Galaxy run at only $70^{o}C$ and dissipate $55W$ each, compared to $90^{o}C$ and $130W$ for CMesh-Exp-, Corona-, and Firefly-based chips. As a result, Galaxy lowers leakage to just over 10% of energy, while single-chip designs waste 36-40% of their energy on leakage. Overall, single-chip designs consume 1.12-1.2x more energy per instruction than Galaxy (Figure 8(b)). Galaxy reaches its highest energy efficiency increase on memory-bound workloads (2-2.3x), as it achieves over 3x speedup and the chiplets dissipate less power waiting for memory. Galaxy attains up to 6.8x lower EDP than single-chip CMPs (2.8x on average; Figure 8(a)).

Because Galaxy chiplets run cooler when running memory intensive workloads, the energy consumption of the photonic network (including laser power, modulation/demodulation, and ring heating) is higher, as the ring heaters dissipate more power to keep the photonics layer at the trimming temperature. The energy consumption of photonics is lower with compute intensive workloads, because cores dissipate more power and heat the photonic die, so ring heaters work less.

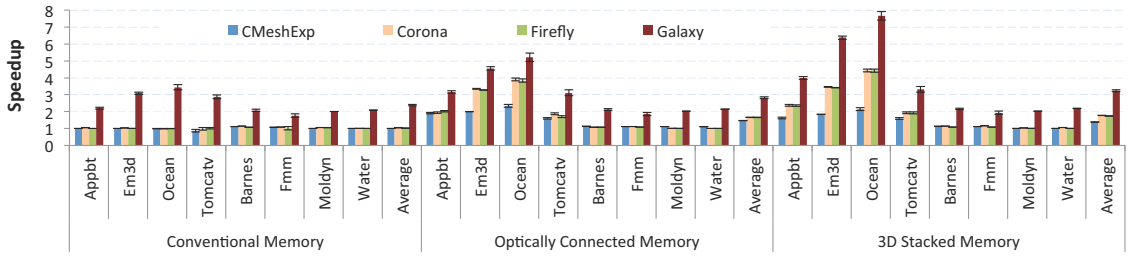


FIGURE 7: Speedup of power-constrained designs with various memory technologies (normalized to CMeshExp with DDR3).

2.3.3 Comparison to Multi-Chip Designs

Galaxy can scale up to 1088 cores with 17 chiplets, and 4160 cores with 65 chiplets (Section 2.2.4). Table 3 details the power, bandwidth, and latency characteristics of the scaled out designs, and compares Galaxy with fibers to designs that utilize SOI waveguides or electrical links for chiplet-to-chiplet communication, as well as the Oracle Macrochip. Figure 9 compares the performance of these alternatives by modeling the effect of link latency and serialization on performance following the methodology in Section 2.2.4.

The power-hungry electrical links cannot provide enough bandwidth within the power envelope, resulting in high serialization delay that increasingly hurts performance as the system scales up. Similarly, SOI waveguides require higher laser power than fibers, as the optical loss in SOI waveguides grows rapidly with increasing length, and at the same time they are 2.3x slower than fibers due to different light propagation speeds between the two materials. As a result, fibers increasingly outperform SOI waveguides as the system scales up. The performance gap is higher for memory-intensive workloads which stress the interconnect more. A 65-chiplet Galaxy with fibers outperforms Galaxy with SOI waveguides by up to 1.44x (1.24x on average), and Galaxy with electrical links by up to 9.53x (4.58x on average).

The Oracle Macrochip [39,41] uses SOI waveguides and OPCs [80] to create point-to-point photonic links across chips. Galaxy outperforms the Oracle Macrochip by 2.5x on average (Figure 9) because the Macrochip employs 2-bit-wide data channels which impose high serialization delay, and SOI waveguides are slower that optical fibers.

Because the coupler loss is the biggest contributor to the laser power consumption, we evaluate the sensitivity of laser power to the coupler loss for the Oracle Macrochip and Galaxy (Figure 10). In the figure we indicate the laser power consumption of the Oracle Macrochip with measured coupler losses for passive-aligned and active-aligned OPCs [80], as well as under aggressive OPC loss predictions [39,41]. For Galaxy, we indicate the laser power consumption under SION and SU8 tapered couplers using loss measure-

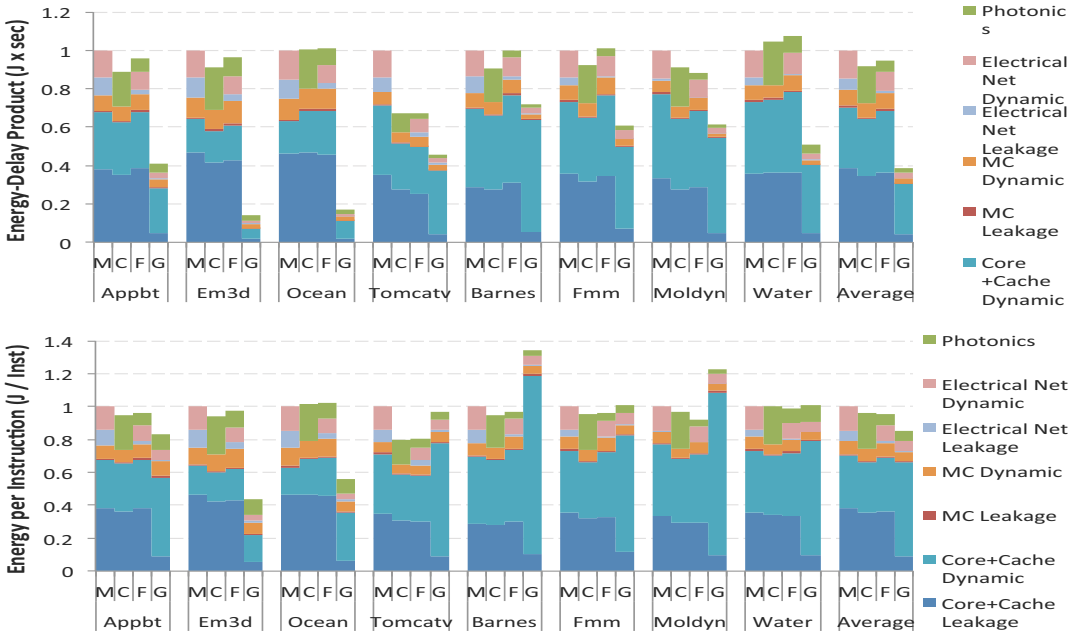


FIGURE 8: (a) Energy x Delay, and (b) Average energy / instruction for CMeshExp (M), Corona (C), Firefly (F), and Galaxy (G).

ments of existing prototypes [44]. Because macrochip links have to pass through 3 couplers to go from one chiplet to another (vs. 2 for Galaxy), the slope of the laser power is higher indicating that it is more sensitive to coupler loss. The Macrochip with actively-aligned OPCs requires 6x more laser power than Galaxy. Even if the predicted OPC loss is achieved, Galaxy with existing couplers would still require less laser power.

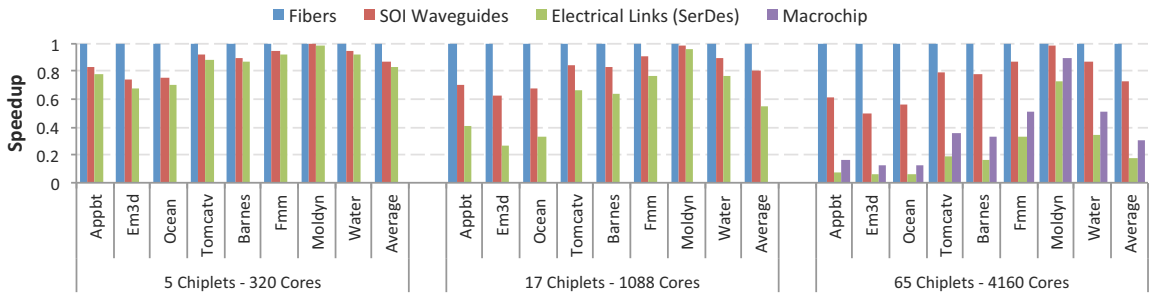


FIGURE 9: Comparison of Galaxy with different chiplet-to-chiplet interconnect technologies, and the Oracle Macrochip.

2.3.4 Thermal Evaluation

To effectively push back the power wall while still employing conventional forced air cooling solutions and cheap packaging appropriate for high-volume markets, a disintegrated design requires the chiplets to be physically far enough from each other to minimize heat transfer. Our thermal modeling using computational fluid dynamics tools [77] and HotSpot [67] indicates that a Galaxy architecture with active heatsinks on each chiplet allows the chiplets to operate at $66.2^{o}C$, sufficiently cool for most applications. In fact, even cheaper cooling solutions seem adequate. Figure 11(a) shows a Galaxy design with 5 chiplets. The chiplets use passive heatsinks and are spaced 8 *cm* apart, with a global fan blowing air horizontally in $45^{o}C$ ambient temperature in a box shell. The fanless (passive) heatsinks cool chiplets to $88.2^{o}C$, and deliver low packaging and cooling costs, and increased lifetime. Thus, even very simple and cheap cooling solutions (fanless heatsinks, a global fan) suffice for an 80-core 5-chiplet Galaxy.

Optical fibers allow Galaxy to spread chiplets far apart for better cooling, while SOI waveguides and electrical links can not. As the Oracle Macrochip utilizes SOI waveguides for intra-chiplet communication, it is confined to a single wafer [39] and requires specialized liquid cooling solutions, which are too expensive for most market segments. We compare the thermal characteristics of a Macrochip-like dense design to an equal-size Galaxy by modeling a 3x3 Macrochip architecture and a 9-chiplet Galaxy. Both designs use the same heatsinks. Based on the Macrochip architecture [39,41], we estimate that the heatsinks will almost touch each other resulting in the layout shown at Figure 11(b). We observe that the sites that are further away from the fan reach $249^{o}C$, and hence cannot be cooled with conventional forced air solutions.

In comparison, a 9-chiplet Galaxy design which dissipates the same amount of dynamic power as the Macrochip can be cooled with forced air and passive heatsinks. The thermal-aware placement of chiplets on a 2D-plane shown in Figure 11(c) increases the x-dimension of the board from 12 *cm* in the Macrochip lay-

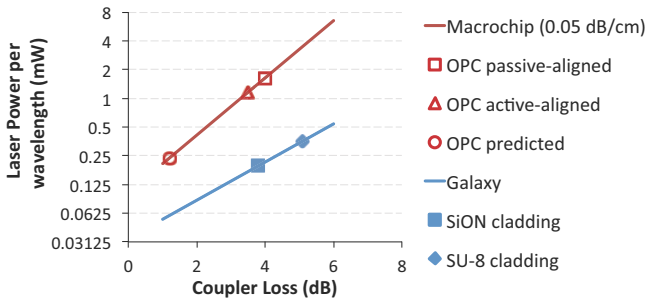


FIGURE 10: Laser power sensitivity to coupler loss.

out to 28 *cm*, while the y-dimension remains the same. In return for the larger board, the Galaxy design achieves a maximum temperature of 110^{o}C, which is a full 139^{o}C lower than Macrochip. Furthermore, using optical fibers for cross-chiplet communication allows Galaxy to utilize multiple boards. Figure 11(d) shows that Galaxy can bring a 9-chiplet design down to a cool 87^{o}C using only conventional forced air and a 3D layout. This freedom of placement gives a significant advantage to Galaxy compared to silicon-wave-guide-based designs, and allows it to spread the volume enough to cool even large-scale designs.

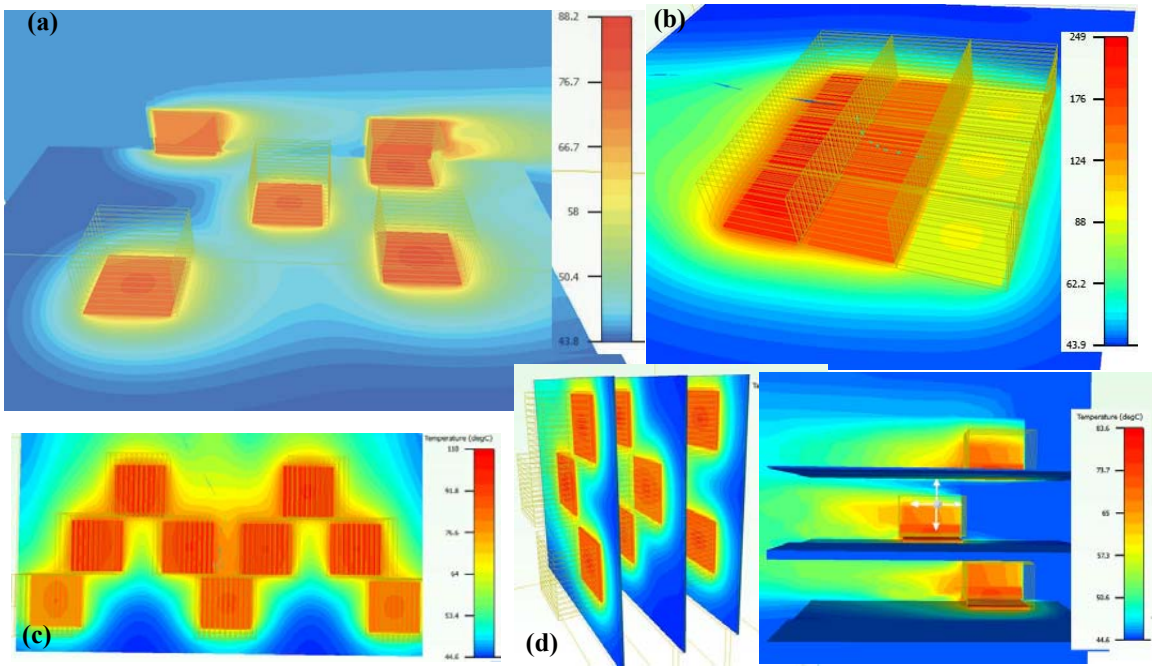


FIGURE 11: Thermal effects of chiplet placement.

2.4 Limitations and Challenges

2.4.1 Misalignment and Fiber Density Considerations

The use of fibers for chiplet-to-chiplet communication in Galaxy brings two new challenges: coupling the fibers on chip and attaching enough fibers to achieve the highest performance or lowest EDP, depending on the optimization target. Fibers connect to chiplets through a coupler that tapers an array of fibers at 250 μm pitch down to 20 μm pitch channels, and couples them into an array of SOI waveguides at the edge of the chip [44]. Characterization on fabricated tapered couplers has measured total coupling loss as low as 3.8 dB [44]. Part of this loss comes from misalignment. Misalignment within 0.7 μm, 0.4 μm, and 0.7 μm in the lateral, vertical, and optical axes produces losses under 1 dB [44]. In comparison, optical proximity couplers have been measured to achieve as low as 3.5 dB optical loss [80]. The optical loss of OPC couplers increases by 0.5 dB per 1 μm misalignment in the y dimension, plus less than 1 dB loss due to misalignment within 2.5 μm in the x and z dimensions [80].

Overall, the performance of the tapered coupler is comparable to that of an optical proximity coupler. OPC coupling is more forgiving of misalignment, allowing three times higher misalignment than tapered couplers in the x- and z-axis for similar loss. Without a large volume of characterization experiments, however, it is hard to distill statistically significant results for either technology. In addition, tapered couplers are more amenable to active alignment (albeit at a higher manufacturing cost), as each time only a subset of the fibers is aligned, while OPC couplers need to be aligned all together. Despite the misalignment hurdles, tapered couplers allow the use of fibers which exhibit simultaneously both negligible optical loss and high bandwidth density, more than making up for the higher misalignment loss (Figure 10).

Galaxy requires enough length along the periphery of a chiplet to attach the fibers. Galaxy's 116 mm^2 chiplets provide over 43 mm in total length along the edge of a chip, allowing up to 172 fibers at a 250 μm

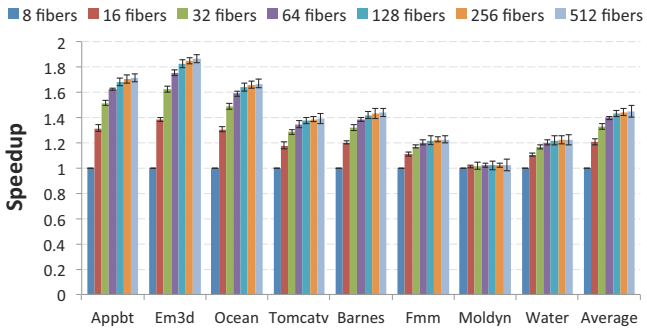


FIGURE 12: Sensitivity to fiber density per chiplet.

pitch. The design we have evaluated assumes 128 fibers per chiplet with 16 DWDM on 64-bit-wide data-paths. Figure 12 indicates that having 512 fibers (i.e., 4x the fiber density) increases the performance by only 3%, while dissipating 4x more laser power, so this is not a desirable design point. On the other hand, using 64 fibers reduces performance by only 2.4% over the Galaxy configuration we evaluated, and consume half the laser power, so this is also a viable design point. Employing a less dense fiber array, however, causes evident performance degradation. Galaxy with 32 fibers per chiplet is 7% slower, and Galaxy with 16 fibers is 15.5% slower than the design we evaluated in this paper. While these design points still provide a performance and EDP benefit over electrical links and SOI waveguides, the bandwidth limitations quickly reduce the performance of the system. Thus, applications that require significant chiplet-to-chiplet bandwidth, but allow only a few fibers per chiplet due to practical or economic considerations, may not benefit as much as the workloads we evaluated in this paper..

Finally, while fibers are edge-coupled, electrical links are face-coupled, so their density will scale better with decreasing chiplet size and increasing chip disintegration. However, the fibers still provide enough bandwidth density to support the smaller chiplets, while they remain more energy-efficient than electrical links or waveguides (Figure 2).

2.4.2 Board-Level Effects

Spreading the chiplets far apart to decrease the thermal density and allow forced-air cooling requires larger boards (Section 2.3.4). This may be an impediment to designs that target compute density. However, it is important to note here that any cooling solution applicable to multi-chip or multi-socket systems is readily applicable to Galaxy. The additional advantage that Galaxy offers is that the system designer can choose how close the chiplets should be to realize a given cooling solution. Thus, Galaxy allows higher design flexibility, and the ability to explore all cooling solutions and their economic trade-offs, from forced air to liquid cooling and beyond.

Similarly, by allowing the chiplets to run at full speed, Galaxy consumes more power at the board level which may stress the board-level power delivery. However, fibers allow the chiplets to spread in 3D-space and occupy multiple boards, while still behaving like a large virtual chip (Section 2.3.4). Thus, Galaxy can utilize as much power as can be delivered to each board and improve performance (Figure 7) while minimizing waste (Figure 8). Overall, the ability to spread the design over multiple boards allows the system designer greater flexibility in deciding how many boards to use and how much power to deliver to each one, based on the other physical, financial, and engineering constraints that the system must satisfy.

2.4.3 Yield, Cost and Lifetime Considerations

Galaxy relies on the manufacturing of a photonic die, 3D integration of the photonic and the logic dies, and the manufacturing and assembly of the tapered couplers and the fibers. Each one of these steps carries its own inefficiencies and costs, which are likely to be higher (at least initially) than the cost of the mature CMOS processes. Of all these components, fibers have been manufactured at high volumes and they have become very cheap (a few cents per foot). To assist in calculating the cost of the system, Section 2.1.4 provides component counts for the nanophotonic devices. While the absence of yield and manufacturing data

for nanophotonic systems do not allow us to make quantitative arguments, we expect that the additional manufacturing steps will increase the overall cost of the system.

However, processor disintegration allows Galaxy to recover the additional overhead or even achieve lower overall cost than conventional monolithic single-chip designs. By breaking a monolithic chip into multiple smaller chiplets, one can increase yield and lower non-recurring and marginal costs by a significant factor, especially for low and medium volume markets, as only the defective chiplets need to be replaced rather than an entire large chip ([13]).

In Figure 13, we compare the yield of a 5-chiplet Galaxy architecture against monolithic chip designs with Mesh and Corona. Yield of Mesh doesn't change with the yield for photonics, because it has an all electrical on-chip interconnect. On the other hand, the yield of Corona depends on the yield of on-chip photonics only, since it has an all photonic crossbar. Yield of Galaxy depends on both the yield of on-chip photonics and off-chip photonics. We observe that, with medium yield in both on-chip and off-chip photonic integration process, Galaxy can achieve better yield compared to single-chip designs. As the system scales, the yield improvement due to disintegration will become more significant. As technology matures, nanopho-

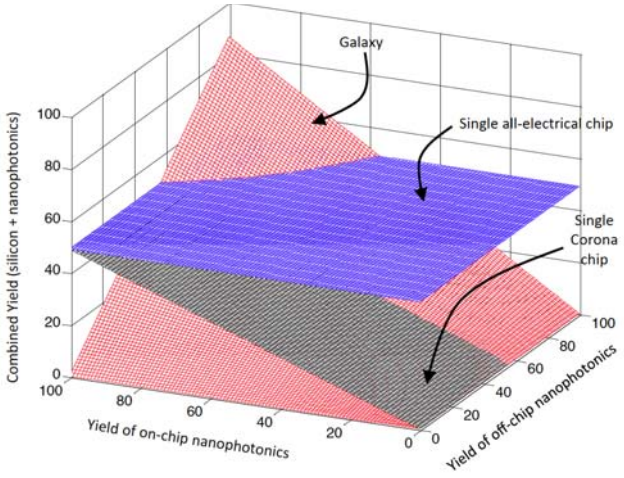


FIGURE 13: Impact of nanophotonics on overall yield.

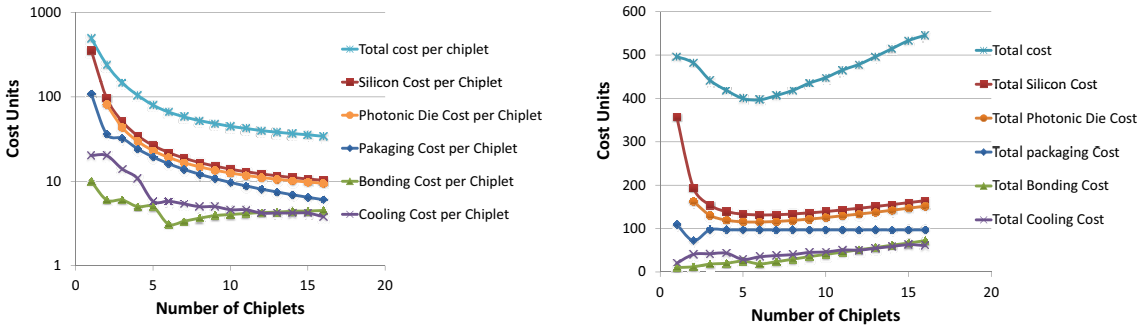


FIGURE 14: Impact of disintegration on the cost of each chiplet (right) and the total processor cost (left)

tonic devices and 3D integration are likely to enjoy higher yields and be competitive to CMOS processes, tilting the balance more in favor of disintegrated architectures.

We also investigated the impact of processor disintegration on the overall cost, using the 3D-processor cost models presented in [20,79]. The 3D-cost model [20,79] takes into account 4 things: the silicon wafer cost, the 3D bonding cost, the package cost and the cooling cost. Firstly, we noticed that the packaging cost can increase the overall cost rapidly, and it is dominated by the number of the pins placed on the package. The cost of disintegration increases rapidly when the processor connects to the memory using traditional electrical pins, therefore, using optical fibers to connect to the memory is the preferable option. In Figure 14, we present the impact of disintegration on the cost of multiprocessor, when we disintegrate a 580 mm^2 chip up to 16 chiplets. As the number of chiplets grow, the silicon die cost per chiplet decreases due to yield improvements, however, the improvements slow down as chiplets get smaller than 100 mm^2. The cooling cost per chiplet decreases, because smaller chiplets stay cooler. The total cost of bonding and cooling increase with disintegration, because each chiplet requires individual steps and equipment. The total cost of packaging stays the same, because the number of pins required for the power delivery doesn't increase. Overall, the silicon die cost dominates the total cost, and the best yield is achieved at 5-6 chiplet disintegration point, which is similar to Galaxy.

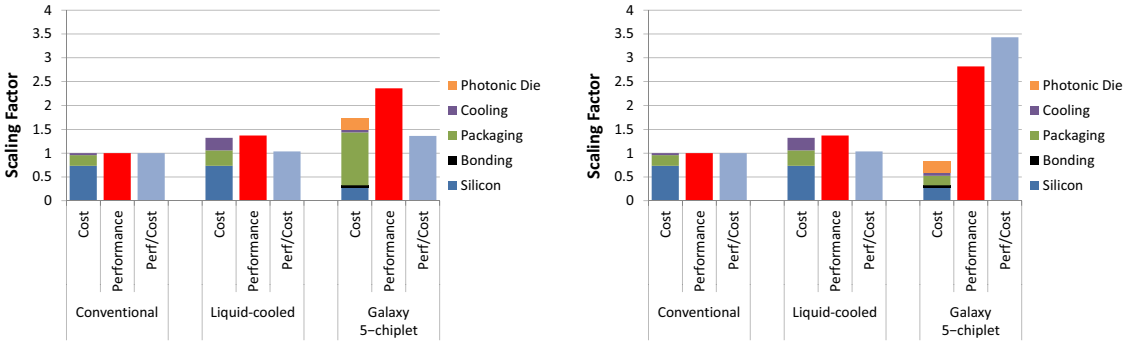


FIGURE 15: Cost breakdown for Galaxy with traditional memory connection (left) and optical memory connection (right) compared to Convetional and Liquid-cooled single-chip designs

We calculated the cost of the 5-chiplet Galaxy architecture and compared it against the conventional single-chip designs with forced-air cooling and advanced liquid-cooling solution (Figure 15). When Galaxy connects to memory using traditional electrical pins, the packaging cost increases Galaxy's overall cost, however, Galaxy provides 1.35x more performance per cost, simply because it is 2-2.2x faster than the conventional single-chip designsf.In this cost calculation, we don't take the cost of connecting optical fibers to the package into account, simply because there is no exact data. However, we calculate that, if adding a single fiber into the package costs less than adding 36 pins into the package, Galaxy will keep providing better performance per cost.

When it connects to memory using optical fibers, Galaxy costs 44-21% less than conventional single-chip designs, because of the yield improvements. Furthermore, Galaxy provides 3.4x higher performance for the same cost, because, it is already 2.2-2.8x faster. In this case, if adding a single fiber into the package costs less than adding 114 pins into the package, Galaxy will keep its performance per cost benefits.

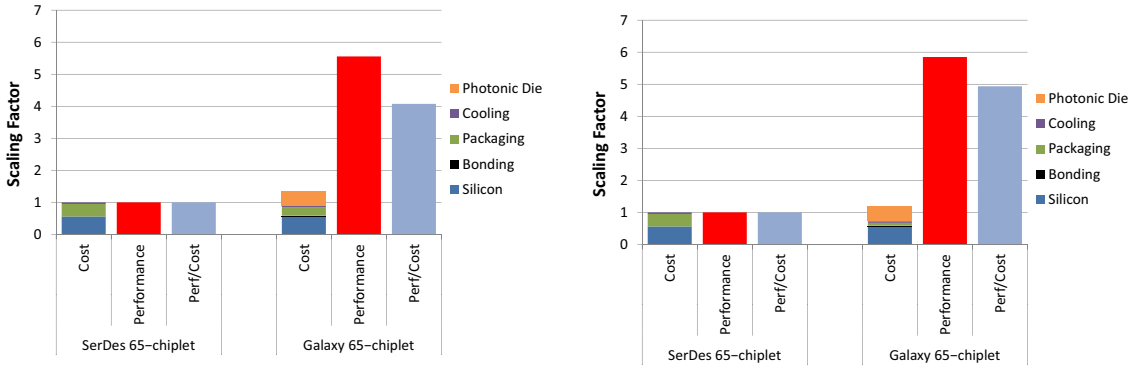


FIGURE 16: Cost breakdown for a scaled-out Galaxy with traditional memory connection (left) and optical memory connection (right) compared to a scaled-out design with electrical (SerDes) chip-to-chip cennections

Using the same methodology, we calculated the overall cost of a scaled-out Galaxy with 65 chiplets, and compared it with a multi-chip system which uses electrical links (SerDes) to connect chiplets together. We observed that the cost of Galaxy is 10-32% higher than the conventional design, however it provides 4-5x higher performance per cost, because, Galaxy is 5.55-5.85x faster than the conventional design. In this case, if connecting the fibers to the package cost less than adding 596 pins to the package, the Galaxy provides better performance per cost with electrically connected memory. This break-even point increases to 663 pins, if Galaxy uses optical fibers to connect to the memory.

In the 5-chiplet Galaxy design, each chiplet runs at 67 ^{0}C, which is 23 ^{0}C lower than the conventional single-chip designs. According to the data published in [000], this decrease in operating temperature will lead to at least 2x improvement in the lifetime, because it reduces the failure rate due to bulk silicon or oxide defects by a factor of 2 - 3x, failures due to assembly defects by a factor of 3 - 5x, and the electro-migration by a factor of 4 - 7x.

Chapter 3

Introducing Laser Control for Energy Proportional Photonic Interconnects

Silicon photonics have emerged as a promising solution to meet the growing demand for high-bandwidth, low-latency, and energy-efficient communication in manycore processors. Silicon waveguides can be manufactured alongside CMOS logic on the same die by adding a few new steps in the manufacturing process [11], and they are more efficient for long-distance on-chip communication than electrical signaling [41,57]. However, the high optical loss of typical silicon waveguides, optical couplers, and on-ring resonators, together with the low efficiency of WDM-compatible lasers, dramatically increase the laser power consumption.

Typical silicon waveguides exhibit optical loss between 0.1-0.3 *dB/cm* [8], resulting in modest optical loss over short distances. However, replacing global wires with silicon photonics often requires long optical channels that traverse the entire chip in a serpentine form (for example, Corona [74] on a 580 mm^2 chip would require a 16 *cm* waveguide, which increases the laser power by a factor of 1.5-3x). Aggressive technology can produce low-loss waveguides (0.05 *dB/cm* [41]) which allow the routing of long optical channels. However, these low-loss waveguides are much wider than conventional ones [41,45]. Their high area occupancy may force the use of narrow data paths (e.g., 2-bit links for an 8x8 array in the Oracle MacroChip [39,41]) which in turn impose significant serialization delays that degrade performance, and ultimately increase power consumption.

Additionally, WDM-compatible lasers are highly inefficient, with typical efficiencies in the range of 5-8% [71], and up to 10% [82]. Thus, the wall-plug laser power requirement is 10-20x higher than the required

laser output power. Process variations impose additional losses, forcing designers to increase the laser power even higher, in order to maintain a safety margin. Sharing the optical path with other senders or receivers may also increase the laser power. While sharing is commonly employed to keep the hardware overhead manageable, it requires additional components which accumulate optical loss. While some optical interconnect topologies strike a better balance between power and performance [14,57,55], most of these costs are hard to avoid, and the laser power remains a considerable fraction of the total power budget. As all these factors are multiplicative, and not additive, it is easy for the wall-plug laser power to grow by more than one order of magnitude when all the losses and inefficiencies are factored in.

Unfortunately, the majority of this power is typically wasted. While the full laser power is required to support periods of high interconnect activity, most of it is wasted when activity is low because photonic interconnects are always on. In a typical setting, light is continuously injected into the waveguides and coupled onto several optical devices, regardless of whether packets are actively sent or not. By comparison, electrical interconnects stay idle consuming only a small amount of leakage power, until a packet attempts to traverse them. It is often the case that the interconnect stays idle for long periods of time, both in scientific computing (compute-intensive execution phases under utilize the interconnect), and in server computing (servers in Google-scale datacenters have a typical utilization of less than 30% [1]).

Motivated by these observations, we propose EcoLaser, a collection of static and adaptive laser control mechanisms that react to the demands of the aggregate workload by opportunistically turning the laser off during periods of low activity to save energy, and leaving it on during periods of high activity in order to meet the high bandwidth demand. EcoLaser capitalizes on recent advancements in Ge lasers [42,47], which enable energy-efficient on-chip laser sources that can be turned on or off within nanoseconds.

More specifically, the contributions of this paper are:

1. We quantify the maximum opportunity of saving power through laser control.

2. We propose EcoLaser, a collection of static and dynamic laser control mechanisms and policies that approximate the maximum possible savings. EcoLaser is amenable to implementation in both SWMR and MWSR optical crossbars, and we present detailed designs for both.
3. We evaluate the impact of EcoLaser on the performance and energy of a multicore processor running a range of synthetic and scientific workloads, under realistic physical constraints, and across a range of optical crossbar sizes.

Our results indicate that EcoLaser saves between 24-77% of the laser power for radix-16 and radix-64 SWMR and MWSR crossbars real-world workloads. EcoLaser closely tracks (within 2-3% on average) a perfect controller with the knowledge of future interconnect requests. Thus, EcoLaser harvests the vast majority of the energy benefits that can be achieved by controlling the laser source. Moreover, the power savings of EcoLaser allow for providing a higher power budget to the cores, which enables them to run faster. Employing EcoLaser on a radix-16 and radix-64 crossbars allows the multicore chip to achieve 1.1x and 2x speedups over a baseline scheme with no control respectively.

Improving upon EcoLaser [16], we propose ProLaser, a novel laser control scheme, which achieves higher laser energy savings for all utilization levels, while minimizing the additional laser turn-on delay overhead of laser control. Different than EcoLaser, ProLaser extends the laser control for the off-chip laser sources to achieve higher overall energy efficiency. ProLaser achieves energy efficiency with high performance by keeping the majority of the data-bus inactive while sending small (dataless) messages, and anticipating upcoming messages to turn the lasers on ahead of time. ProLaser controller builds upon the laser controller microarchitecture proposed for EcoLaser [16] (Figure 22.a), but different than EcoLaser, it implements a bloom filter anticipate the messages generated by L2 accesses. Similar to EcoLaser, ProLaser keeps the lasers on until all of the messages queued in the injection buffers are sent out, this way achieves high throughput under heavy utilization.

We evaluate the impact of ProLaser on the performance and energy of a multicore running a range of synthetic and scientific workloads under realistic physical constraints, and show that it saves between 49-88% of the laser power, it outperforms the current state of the art by 2x on average, and closely tracks (within 2-6%) a perfect prediction scheme with full knowledge of future interconnect requests. Moreover, the power savings of ProLaser allow for providing a higher power budget to the cores, which enables them to run faster. Employing ProLaser on a topology with SWMR crossbars (Firefly [57]) allows the multicore to achieve 1.5-1.7x speedup (1.6x on average) and attain 35-52% lower energy consumption per instruction (40% on average).

3.1 Background

Previous works [2,3,36,42,57] typically use off-chip lasers because of their temperature stability, easy replacement, and energy efficiency (30% for gaussian comb lasers [21]). However, recent work [28] shows that output spectrum power variations and laser-to-fiber and fiber-to-chip coupling losses add 7-8 *dB* optical loss, thus off-chip lasers are in reality only 6% efficient. In comparison, on-chip laser sources [38] attain wall-plug efficiencies up to 15%, while enabling wavelength-division multiplexing (WDM). WDM can be implemented by feeding a set of wavelengths generated by an array of single-wavelength lasers into an optical bus. On-chip lasers offer energy efficiency and easy packaging, but their wall-plug power consumption counts against the processor's overall power budget. In either case, the laser power consumption remains a considerable overhead, especially when accounting for realistic optical loss parameters and laser efficiencies, emphasizing the need for power gating the laser source. Power gating on-chip lasers can increase the energy efficiency of a photonic interconnect by up to 4x [28].

Laser power-gating has been overlooked due to the high turn-on latency (0.1 *us* [28]) of the traditional distributed feedback comb lasers that are widely assumed in photonic interconnects [2,3,36, 42,57]. Comb

lasers use diffraction grating to form the optical cavity. Temperature affects the diffraction grating pitch and the active region's refractive index, which alter the diffraction grating's wavelength selection, and hence the laser's emission wavelength. Thus, when comb lasers turn on they need time to reach a set temperature and lock at the designated wavelength. This high delay hampers power gating. In contrast, Fabry-Perot (FB) lasers use two discrete mirrors to form the optical cavity, and their emission wavelength depends not on temperature but on the n-type doping level and the strain applied during the cavity development. Thus, when they are turned on (pumped to the lasing threshold), they lase at the designated wavelength without requiring time for temperature stabilization/locking, and, hence, are suitable for power gating.

ProLaser, and laser power gating in general, strongly depends on fast lasers. While such technology is still experimental, it is important to note that fast lasers with *ns*-scale turn-on times have been manufactured and their turn-on delay has been characterized on real hardware prototypes [40,58,24,7,47], and is in agreement with theoretically-derived results. To turn the laser on, a supply current is applied to the laser. When the carrier density exceeds the threshold density, laser oscillation starts and light output increases drastically (laser turn-on). The time it takes from the current injection to the laser turn-on is the "laser turn-on delay" which is governed by the carrier life time and is in the order of *ns* ([59], pp. 80-82). The turn-on delay of Fabry-Perot lasers is highly tunable by design parameters, and nanosecond or sub-nanosecond laser turn-on delays are both theoretically predicted [30,29,59 pp. 83] and achievable in real implementations [7,47,40,59].

For example, InP-based diode FB lasers [40] have been manufactured and shown to emit light with a 2 *ns* long electrical pulse excitation (so the laser turn-on latency is at most 2 *ns*). InP-lasers have high peak power, and their emission wavelength is tunable in a wide range and highly stable with temperature, which makes them WDM compatible. Moreover, InP-lasers can be integrated on Si [58, 24] so they can be used as an on-chip laser source.

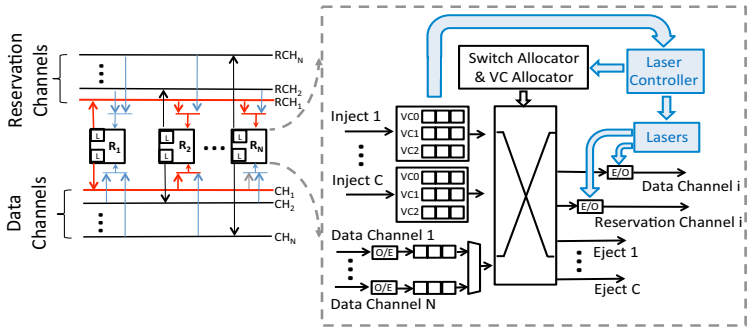


FIGURE 17: SWMR crossbar and router microarchitecture.

Similarly, Ge-based FB on-chip lasers have been manufactured [47] and the turn-on delay of real hardware prototypes was measured at 1.5 *ns* at most for both optically- and electrically-pumped implementations [47, 7]. We directly verified this claim for both optically- and electrically-pumped Ge-lasers with the lead author of [47] in personal communication, and with the director of MIT's Microphotonics Center ([35] slide.19).Besides their fast turn-on time, Ge-lasers [7] are suitable for on-chip photonic interconnects because they can be built within a standard-width (1 μm) waveguide at only 7.68×10^{-3} mm^2, operate in room temperature, and are WDM-compatible as they exhibit gain spectrum over 200 *nm* [7].

We want to emphasize that ProLaser does not depend on a singular laser technology. Any fast WDM-compatible continuous-wave laser that can be integrated on chip is suitable for laser power gating, including the InP and Ge lasers [40,58,24,7,47] we assume in this work (their 1.5-2 *ns* delay offers similar benefits; we show ProLaser's sensitivity to the turn-on delay in Section 3.7.6).

Both the theoretical estimations [29,59 pp. 83] and the measurements over the real implementations [47,40] show that, the injection current follows a logarithmic increase pattern (slowly increasing) when the laser is biased with a pulse signal. When the carrier density exceeds the threshold value, the laser starts emitting light (lasing), and the photon and the carrier density oscillate, but the injection current doesn't oscillate nor overshoot [29]. This means, during the "laser turn-on period" the laser power consumption

never exceeds the full laser power consumption (the laser power consumption during lasing). In order to simplify the calculations, we assumed that the laser consumes full power (as if it is lasing) during the laser turn-on period, even though it may potentially be consuming less.

It is important to offer the interested reader an additional perspective on laser turn-on times: VCSELs can turn-on with sub-100 *ps* delay [59], and thus can be directly modulated over 35 GHz [76]. However, VCSELs are unsuitable for on-chip applications with WDM because they emit significant heat, and their operating wavelengths are defined by the epitaxial growth [28] which challenges the implementation of a multi-wavelength VCSEL array on chip. Moreover, it is hard to protect the integrity of messages with direct laser modulation due to chirping and the pattern effect [59].

3.2 Nanophotonic Interconnect Topologies

In Single-Writer-Multiple-Reader (SWMR) [36] crossbars, each router has its own dedicated data channel which delivers messages to all other routers (Figure 22.a). R-SWMR [42,57] crossbars add a reservation channel to SWMR. A sender in R-SWMR first broadcasts on its reservation channel a flit with the receiver's ID (in Figure 22.a, router R_1 broadcasts on RCH_1 a flit with ID=2). Upon receiving a reservation flit, the receiver (R_2) turns on its demodulators to receive the message from the sender's data channel (CH_1), which is now dedicated to transfer data from the sender to the receiver. Reservation channels are narrow because reservation flits only carry the receiver ID and message type information. However, the laser power required to broadcast increases exponentially with the number of readers, making it impractical to broadcast at high-radix crossbars (e.g., radix-64). Instead of having a single broadcast link with many readers, slicing [3] spreads the readers across multiple waveguides and enables high-radix R-SWMR cross-bars.

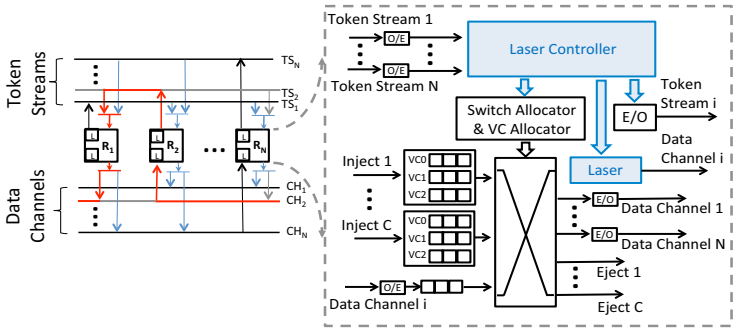


FIGURE 18: MWSR crossbar and router microarchitecture.

In Multiple-Writer-Single-Reader (MWSR) crossbars [74] (Figure 18), each router "listens" on a dedicated channel and can write data on the listening channels of all the other routers in the crossbar. A writer (e.g., R_1 in Figure 18) first arbitrates with other writers by grabbing a token from the receiver's (R_2) token stream (TS_2). Upon a successful token acquisition, the receiver's data channel (CH_2) is now dedicated to transfer data from the sender to the receiver.

3.3 EcoLaser's Laser Control Schemes

The objective of the laser control is to save laser energy by turning off the lasers whenever the bus (i.e., data channel) is idle. The laser should be turned back on when the bus will be used. The Ge-based laser [47] assumed in this work turns on in 1 *ns*, during which period it consumes the same power as when it is lasing. To control the lasers quickly, we place the laser for each router's dedicated channel within the router.

3.3.1 Laser Control for SWMR Crossbar

The shaded components in the router microarchitecture in Figure 17 correspond to components added by EcoLaser. The laser controller turns the laser on if there is a message at any of the injection buffers, and it does not turn it off unless (a) there is no message at the injection buffers, and (b) the laser has stayed on for

the minimum laser stay-on time *"K"*. The laser controller keeps the switch allocator waiting while the laser turns on. After the turn-on delay, the laser is ready and the switch allocator moves messages to the modulators.

3.3.2 Laser Control for MWSR Crossbar

In a MWSR crossbar (Figure 18) every router reads from its own bus, and writes on the other routers' busses. Contention occurs when two routers try to transmit at the same time to the same destination. Token-based arbitration [74,56] resolves the contention by using ring-shaped waveguides to move the tokens in the direction of data travel, and one cycle ahead of the data slot. In the ring-shaped crossbar, the reader node also snoops the returning tokens to control the input buffer utilization [56]. Because the reader collects back its tokens, we can use the token stream to send a "Laser turn-on request" from any writer node to the reader node. Each reader in EcoLaser holds the lasers for its own bus (Figure 18), and its laser controller controls the lasers by looking at the laser turn-on requests received through the reader's token stream.

We construct the tokens to perform three tasks: (a) maintain the time share on the bus, (b) indicate if there is light in the data bus so a writer will know if he can write, and (c) bring the laser turn-on requests back to the reader. Note that only the reader can inject tokens in his token stream, and any type of snooping of the token by writers is destructive. In order to meet all these needs, we design 3-bit tokens as shown at the top of Figure 19: the "*T*" bit provides mutual exclusion on the data bus, similar to the original MWSR tokens; the "*L*" bit indicates if the laser was on when this token was released from the reader node (i.e., the subsequent slot in the data bus has light that can be modulated); the "*S*" signals the reader to turn on the laser. Because the nanophotonic interconnect runs at double the processor frequency, EcoLaser can send *S* and *T* back-to-back on the same wavelength in a single processor cycle, thus requiring only 2 wavelengths for the token stream.

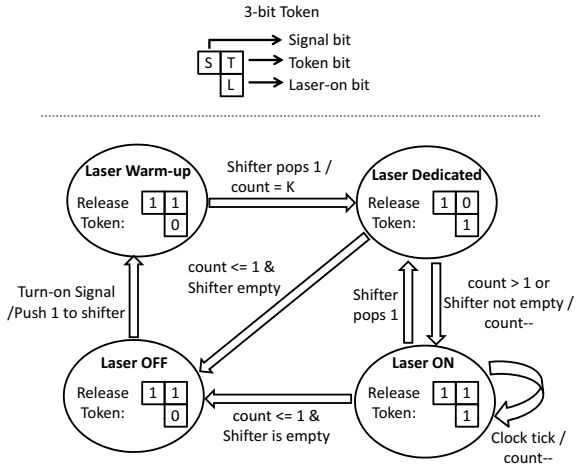


FIGURE 19: 3-bit Token and Laser Controller FSM.

Figure 19 shows the **Laser Controller Logic.** When the bus is idle, the data lasers stay at the "Laser OFF" state; they do not consume energy, and the released tokens indicate this with a clear *L* bit (note that the laser for the token stream is always on). The laser controller snoops the incoming tokens for laser turn-on signals. Writers send the laser turn-on signal by clearing the *S* bit of a token. When the laser controller receives the turn-on signal, the data channel lasers move to the "Laser Warm-up" state. During the warm-up the lasers consume full power preparing photons, but cannot emit any light yet. When the lasers are ready, they start emitting light into the data bus (emit data slots), moving to the "Laser Dedicated" state. The data slot injected first is dedicated to the writer who requested the laser turn-on; the corresponding token has a clear *T* bit, to prevent any other writer from grabbing the slot. The writer who sent out a laser turn-on signal expects to receive a dedicated data slot at the round-trip time plus the laser turn on time after sending out the signal (equal to the worst-case delay). This dedicated data slot ensures that the writer who turned the laser on will be serviced, preventing starvation. The laser controller sends out a dedicated data slot at a delay equal to the laser turn-on time after receiving the laser turn-on signal; this is ensured by pushing a 1 into a 5-bit barrel shifter, and sending out the dedicated token when this 1 pops from the other end (1 *ns* is 5 cycles at 5 *GHz*), and keeping the laser on for as long as there is a set bit in the shifter.

The laser controller keeps track of the duration the laser has stayed on through the counter "*count*". When the laser emits the first (dedicated) slot, the count is assigned the value *K*. *Count* decrements on every cycle, and the laser stays on and releases data slots which are available for any writer node to use (tokens indicate this availability with set *T* and *L* bits). When *count = 1* the laser turns off, unless there is another set bit in the shifter, which indicates a new pending laser turn-on request. If there is, the laser will remain on until the set bit pops out from the shifter, in order to service the new writer.

Figure 20 shows the **Writer Node Logic**. When a writer has a message to send, it moves to the "Request Slot" state, and looks for an available data slot. The writer reads the *T* and *L* bits of the first token, and if they are both set (i.e., the data slot has light and is available), it modulates the message into the data slot. If *T* and *L* are not both set, the writer sends a laser turn-on signal by reading (clearing) the *S* bit of the token, and sets the Estimated Time of Arrival (*ETA*) of the dedicated slot. If all of the token bits are clear, the token has been grabbed and used to send out a laser turn-on signal already, so the writer stays in the "Request Slot" state and re-tries. After sending a laser turn-on signal, the writer moves to "Slot Polling", in which the writer looks for an available slot (by reading both *T* and *L* bits) while waiting for its dedicated slot to arrive. The writer transmits when either an available slot with light arrives, or the writer's dedicated slot arrives, whichever happens first. Note that the writer sends at most one laser turn-on signal, which avoids wasting laser energy. Also, writers can send a laser turn-on signal using a token that has been through another "Slot Polling" writer, which improves performance. Once a writer sends out a laser turn-on signal, it is guaranteed to receive a data slot in ETA time.

Figure 21 shows an example that demonstrates how the laser control scheme which works on a ring-shaped MWSR crossbar. Here, R0 is the reader node, and R1, R2, and R3 are the writers which send messages to R0. The direction of data travel is shown on the left. At cycle 1, the laser is off, so R0 injects tokens with a

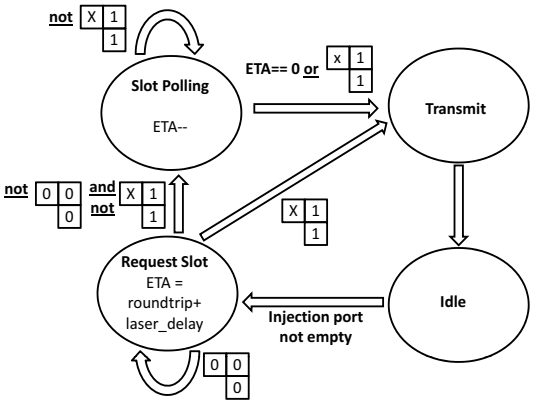


FIGURE 20: Writer Node FSM.

clear *L* bit. At the same cycle, R2 tries to send, but could only send a laser turn-on signal (all bits of token T2 are clear), and then starts polling for an available slot. At cycle 2, R3 tries to send, but comes across the token used by R2 before (T2), so it tries again in the next cycle, and sends a turn-on signal at cycle 3 (token T3). At cycle 3, R0 receives R2's turn-on request (T2), turns on the laser (at the end of the cycle), pushes a 1 into the shifter, and injects T2 back into the token stream. At cycle 4, R0 receives R3's turn-on signal, and pushes a 1 into the shifter; R1 tries to send, but only sends a turn-on signal and starts polling. At cycle 7, R1's request arrives at R0, and R0 pushes a 1 into the shifter. At cycle 8, the shifter pops a 1, so R0 injects a dedicated token, and the laser turns on at the end of the cycle. At cycle 9, R0 injects a dedicated token, so slot D0 is dedicated to R2 and slot D1 is dedicated to R3. At cycles 9 and 10, R1 polls two dedicated tokens and does not transmit. At cycles 10 and 11, R0 injects a "Laser ON" token, because the shifter is not empty

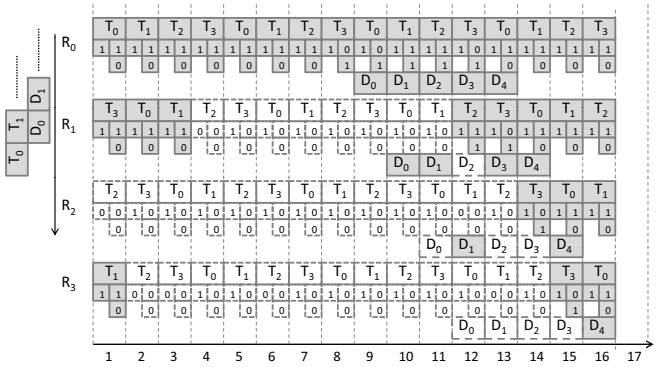


FIGURE 21: A Case Study: MWSR Laser Control Scheme=

so the laser stays on. At cycle 11, R1 finds an available token (transmits at cycle 12), and the ETA of R2 runs out and R2 uses its dedicated data slot (D0). At cycle 12, R0 injects a dedicated token for R1. At cycle 13, R2 tries to send a new message, finds an available token (transmits at cycle 14 on D3), and the ETA of R3 runs out and transmits on its dedicated slot D1.

3.3.3 Adaptive Laser Control

The static laser control schemes discussed thus far leave the laser on for at least "*K*" cycles, where "*K*" is a fixed value. With lower laser stay-on time "*K*", EcoLaser tends to turn off the laser quicker, which saves more laser energy when the crossbar is not heavily utilized. However, under heavier traffic, turning the laser off quickly results in lost opportunities to catch the laser on and send, and increases the number of times the laser has to be turned on anew. The frequent laser turn-on delays decrease performance. On the other hand, when *K* is high, the laser tends to stay on for longer, which increases performance under heavier traffic, but wastes more laser energy when the utilization is low. Thus, no static scheme is expected to perform best under all traffic conditions.

We propose an adaptive scheme that observes the amount of laser turn-on requests to adjust the laser stay-on time *K* at run time. Frequent laser turn-on requests hint to lost opportunities to transmit opportunistically, and the adaptive scheme increases *K* to keep the laser on for longer. A low number of laser turn-on requests hints at potentially wasted laser energy, so the adaptive scheme decreases *K* to save more laser energy, by turning the laser off more quickly.

To prevent oscillation or overshooting *K* from its ideal setting, we employ a hysteresis counter which robustly captures the laser turn-on request trends. The hysteresis counter decrements on every cycle on which there is no other counter activity. Upon sensing a laser turn-on signal, the counter increments by adding some value to it. Whenever the counter reaches its upper threshold, *K* increases by 1; whenever the

counter reaches its lower threshold, *K* decreases by 1. The hysteresis counter controls the value of *K* in a stable manner, because increasing *K* results in a reduction of laser turn-on requests, as the likelihood of a writer finding an available data slot with light increases, and vice versa. The threshold settings and the increment and decrement values of the hysteresis counter change its reactive behavior (making it more lazy or aggressive). Through a design space exploration, we identified the settings that provide the highest energy savings for our workloads, and use these settings for the remainder of our study. Other than adapting *K* at runtime, the rest of the design of the adaptive laser control is the same as the designs described earlier for SWMR and MWSR crossbars.

3.3.4 The Perfect Laser Control

A perfect laser control scheme has complete knowledge of future interconnect accesses. The perfect scheme saves the maximum laser energy without incurring any performance overhead by turning the laser on ahead of time, so the light reaches the writer at the exact time the writer attempts to transmit. After transmitting, the control deactivates the laser or leaves it on for an upcoming message, if deactivation could cause a delay. Thus, the perfect scheme presents the maximum energy savings for a given laser technology.

3.4 ProLaser's Laser Control Schemes

The laser control schemes aim to save laser energy by turning the lasers off whenever the bus (i.e., data channel) is not utilized. Energy savings come at the cost of potential increase in the message latency, because messages have to wait for the laser to turn on before transmission, when they find the laser off. Previously proposed adaptive EcoLaser scheme [16] controls on-chip lasers to achieve laser energy savings at low utilization levels while providing high performance under higher utilization. We propose ProLaser, a novel laser control scheme, which achieves higher laser energy savings for all utilization levels while minimizing the additional laser turn-on delay overhead of laser control. Furthermore, different than EcoLa-

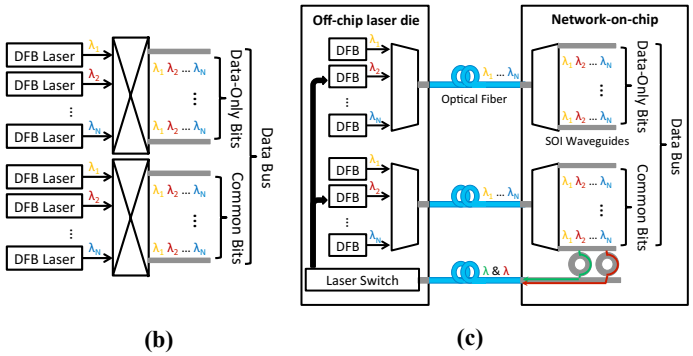


FIGURE 22: On-Chip and Off-Chip Laser Configurations.

ser, ProLaser extends the laser control for off-chip laser sources to achieve higher overall energy efficiency. ProLaser achieves energy efficiency with high performance by keeping the majority of the data-bus inactive while sending small (dataless) messages, and anticipating upcoming messages to turn the lasers on ahead of time. ProLaser controller builds upon the laser controller microarchitecture proposed for EcoLaser [16] (Figure 22.a), but different than EcoLaser, it implements a bloom filter anticipate the messages generated by L2 accesses. Similar to EcoLaser, ProLaser keeps the lasers on until all of the messages queued in the injection buffers are sent out, this way achieves high throughput under heavy utilization.

3.4.1 Segregating the Data from the Control Bits

EcoLaser saves laser energy by turning the optical bus off when it is idle. However, EcoLaser still wastes some laser energy, because it activates the whole optical bus to send small coherence messages (data-less) which don't occupy the whole bus. As photonic links provide high bandwidth, they offer wide busses which can send a data message in one cycle. A data message is 600-bits wide, and contains a 64 byte cache block and 64-bit address and 20-bit ID and 4-bit message type. However, an optical bus is 300-bit (or 300 wavelengths) wide, because the optical links runs at double the processor frequency. On the other hand, small coherence messages are transmitted in two 44-bit wide flits (64-bit address, 20-bit ID and 4-bit mes-

sage type), which means 256 bits of this optical bus are used only when sending data messages (data-only bits) and remain idle otherwise. Remaining 44 bits of the optical bus is activated for all messages (common bits). Figure 22.b illustrates the separation of the data bus into two independent sections as common bits and data-only bits.

In order to observe the effect of this optimization alone, we implemented a simple scheme (**Simple**) which activates the data-only portion of the to send data messages only. In other words, Simple behaves like the EcoLaser, expect that it keeps the data-only portion of the bus deactivated while sending smaller messages and activates it only for data messages.

Simple promises high laser energy savings, because it keeps a big fraction of the optical bus turned-off while servicing the majority of coherence messages. Although it requires independent control of the data-only portion (256 wavelengths) of the bus, that shouldn't increase the total laser power consumption, as the optical link loss, and the total number of wavelengths remain the same. Simple requires an additional WDM laser array, and may require an additional waveguide at high DWDM levels depending on how wavelength generation, splitting and waveguide assignment is done. However, this potential area overhead won't be significant, as waveguides have small pitch and lasers are built within the waveguides. When implementing off-chip lasers, separation of the data bus may require an additional optical fiber connection (Figure 22.c).

3.4.2 Proactive Laser Turn-On Mechanism

The Simple scheme activates the whole optical bus only for data messages, and keeps the data-only portion (256 wavelengths) of the bus switched off most of the time. This lowers the data messages' likelihood of finding the whole data bus turned on. Therefore, data messages suffer from higher message latencies,

which degrades performance. **ProLaser** turns the laser on proactively for most of the data and control messages to reduce the latency overhead.

ProLaser anticipates the early laser activation by correlating cache coherence request messages to replies. In a directory-based cache coherence protocol, every data message is generated upon receipt of a read, write, or directory-forwarded request. When a node receives a directory-forwarded request (meaning that the node is the owner or a sharer), ProLaser turns the whole data bus on anticipating that this node will send out a data reply. When a node receives a read or write request, a lookup in the local L2 cache slice decides the type of the reply: an L2 miss generates another read request (to the tile with the memory controller), while an L2 hit generates a data reply. ProLaser turns the lasers on proactively for both of these messages, but it turns on the data-only portion only if it anticipates an L2 hit. In order to turn the laser on proactively and accurately for both of these messages, we add a small Bloom filter [64] that monitors the requests to the L2 slice and predicts the L2 misses quickly and accurately. A lookup in the Bloom filter takes 1 cycle when an L2 hit takes up to 14 cycles. When the Bloom filter predicts an L2 miss, ProLaser does not turn on the whole bus, thereby avoiding energy waste, but turns on the common bits only. When the bloom filter predicts an L2 hit, the whole data bus is turned on 1ns before the L2 hit latency, so that the data bus will be ready when the data is ready to be sent out. ProLaser implements a 1KB counting Bloom filter for each L2 cache slice, which decrements on cache misses, and provides less than 2% false positives.

Note that, one can implement a ProLaser scheme which anticipates an L2 miss using the result of the L2 tag-lookup (which takes ~10-11 cycles) rather than using 1 cycle Bloom filter lookup. However in that case, the small time window between L2 tag lookup and L2 hit latency (3-4 cycles) wouldn't be able to completely hide the laser turn-on delay and incur higher message latency. Furthermore, such a scheme would be highly susceptible to higher laser turn-on delays. ProLaser aims to keep the design simple by keeping all hardware modifications on the interconnect side, whereas a tag-lookup scheme requires modi-

fying the cache. For these reasons, ProLaser implements Bloom Filters to anticipate L2 misses accurately and quickly.

ProLaser also sends out acknowledgement messages quickly, by turning on the common bits of the data bus right after receiving a reply or an invalidation. Proactive laser turn-on avoids the formation of longer queues at the output buffers, which improves the throughput of the network.

ProLaser doesn't predict the initial refill and writeback request from L1 slice to L2, however these request don't result in high latency overhead, as they only use the common bits of the data-bus which they can find active frequently. Other non-critical messages, such as write-backs, are also not predicted by ProLaser, because they don't have significant impact on the overall performance.

3.4.3 Controlling an Off-Chip Laser Source

Previously proposed designs [2, 3, 36, 42, 57] assume off-chip laser sources because of their temperature stability, easy replacement, manufacturing and packaging, and energy efficiency (30% efficiency for comb laser [21]). However, recent work [28] shows that comb lasers incur significant losses, because of output spectrum power variations, and laser-to-fiber and fiber-to-chip coupling losses. On the other hand, the on-chip laser sources introduced in [38] show wall-plug efficiencies up to 15%, while enabling wavelength-division multiplexing (WDM) and can be power-gated to lower their power consumption [28, 47]. The downside of on-chip lasers is that their total power consumption is dissipated within the processor die, which limits the power available to the other on-chip components (e.g., cores, caches), may cause over-heating, and degrade the overall system performance.

In order to achieve the best of the both worlds, [28] proposes to use a single-wavelength laser array with lasers introduced in [7, 38, 47] as an off-chip laser source. In this arrangement, a fiber-to-chip coupling is still required and the packaging costs are higher, but the losses native to comb lasers and thermal concerns

are avoided. With such an implementation, a feedback signal from the laser control in the processor die can be used to turn on and off the lasers in the array, thereby improving energy efficiency.

Based on the previous discussion, we extend ProLaser to control an off-chip WDM-laser-array source [47]. Our design sends a laser turn-on signal to the off-chip source by redirecting the laser signals back to the off-chip laser array (Figure 22.c). The green and red wavelengths shown in the Figure 22.c are dedicated for the laser control and they are always on. When the node wants to turn on the laser, it redirects these wavelengths back to the off-chip source using the microrings. We estimate that signaling the laser source takes 2 cycles (0.4 *ns* or 2 *cm* waveguide plus 4 *cm* fiber travel). We use two wavelengths to be able to control data-only and common bits separately for ProLaser. Once the laser source detects the laser turn-on signal, it activates the appropriate data bus portion. The Ge based lasers assumed turn on in 1 *ns*, after they start lasing, light travels back to the node within 2 cycles, and the message can be sent out.

Using light to signal the off-chip laser source requires minimal additional hardware (a waveguide, fiber and a few microrings), but it provides low-latency signaling which is essential to achieve high performance. Also, the proactive laser turn-on feature of ProLaser hides the majority of this additional signaling latency.

3.5 Experimental Methodology

3.5.1 Interconnect Performance and Energy Analysis

To evaluate the performance and energy consumption of EcoLaser in isolation from the interference of other system components or application characteristics, we employ a cycle-accurate network simulator based on Booksim 2.0 [15], which models radix-16 and radix-64 SWMR and MWSR crossbars servicing random uniform traffic (we refer to the crossbars using the notation <*type*>*_XBAR_*<*radix*>). The simulator models a single cycle router, with 1-cycle E/O and O/E conversions. We assume a 480 mm^2 chip, which employs a 10 *cm* waveguide with a round trip time of 5 cycles. The link latency (1-5 cycles) is calculated

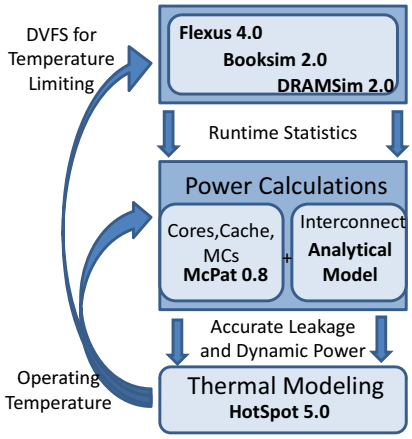


FIGURE 23: Simulation flow chart.

based on the traversed waveguide length. The buffers are 40-flits deep, with a flit size of 300 bits. The maximum core frequency is 5 *GHz*, and the optical interconnect runs at 10 *GHz*. Latency is measured as the time required for the network to process a sample of injected packets. We evaluate the load-latency and energy-per-flit of EcoLaser, and compare it against a baseline without laser control (*No-Ctrl*), and against a perfect control scheme with full knowledge of future interconnect requests (*Perfect*).

3.5.2 Multicore System Performance and Energy Analysis

To evaluate the impact of EcoLaser on a realistic multicore system, we model a 64-core processor on a full-system cycle-accurate simulator based on Flexus 4.0 [27,75] integrated with Booksim 2.0 [15] and DRAMSim 2.0 [62]. Table 2 details the architectural modeling parameters. We target a 16 *nm* technology, and have updated our tool chain accordingly based on ITRS projections [23]. The simulated system executes a selection of SPLASH-2 benchmarks and other scientific workloads. All systems we model employ a throttling mechanism to keep the chip within safe operational temperatures (below 90*C).* Without loss of generality, we employ Dynamic Voltage and Frequency Scaling (DVFS).

We collect runtime statistics from full-system simulations, and use them to calculate the power consumption of the system using McPAT [46], and the power consumption of the optical networks using

TABLE 4. Architectural Parameters.

CMP Size	64 cores, 480mm^2
Processing Cores	ULTRASPARC III ISA, max 5Ghz, OoO, 8-stage pipeline, 4-wide dispatch/retirement, 96-entry ROB
L1 Cache	split I/D, 64KB 2-way, 2-cycle load-to-use, 2 ports, 64-byte blocks, 32 MSHRs, 16-entry victim cache
L2 Cache	Shared, 512 KB per core, 16 way, 64-byte blocks, 14 cycle-hit, 32 MSHRs, 16-entry victim cache
Memory Controllers	One per 4 cores, 1 channel per Memory Controller Round-robin page interleaving
Main Memory	Optically connected memory[2], 10ns access
Networks	SWMR and MWSR crossbars, radix-16 and -64

the analytical power model by Joshi *et al.* [33]. We estimate the temperature of the chip using HotSpot 5.0 [67]. The estimated temperature is then used to refine the leakage power estimate. We adjust DVFS based on the stable-state power and temperature estimates (Figure 4).

The power savings of EcoLaser allow the processor chip to run cooler, thereby allowing the cores to run faster. We evaluate the impact of EcoLaser on two multicore processor designs, one employing an MWSR_XBAR_16 optical crossbar and one employing MWSR_XBAR_64. The design with the radix-16 crossbar has low laser power consumption and high concentration factor (4), creating heavier traffic, so EcoLaser will save only small amounts of power. The radix-64 crossbar consumes higher laser power, but has lower concentration factor (1), so EcoLaser will save more laser power. Thus, these two case studies examine the impact of EcoLaser across two opposite ends of the spectrum. We model all networks as described in Section 3.5.1.

To demonstrate the merits of the adaptive mechanism, we compare EcoLaser with adaptive laser control (*Adaptive*) with two static control mechanisms: *Static-1*, with 1 cycle stay-on time, and *Static-10*, with 10-cycle stay-on time. Static-1 is the quickest to turn the laser off; Static-10 saves the most laser energy per packet among all static schemes when average across injection rates. Finally, to contrast EcoLaser against a power-equivalent design with no laser control, we evaluate a design similar to the baseline without laser

TABLE 5. Nanophotonic Parameters and Laser Power

		Radix-16	Radix-64
	per Unit	**Total**	**Total**
DWDM		64	16
WG Loss	0.3 *dB/cm*	3 *dB*	3 *dB*
Nonlinearity	1 *dB*	1 *dB*	1 *dB*
Modulator Ins.	0.5 *dB*	0.5 *dB*	0.5 *dB*
Ring Through	0.01 *dB*	10.24 *dB*	10.24 *dB*
Filter Drop	1.5 *dB*	1.2 *dB*	1.2 *dB*
Photodetector	0.1 *dB*	0.1 *dB*	0.1 *dB*
Total Loss		**16.04 *dB***	**16.04 *dB***
Detector		**-20 *dBm***	**-20 *dBm***
Laser Power Per Wavelength		**0.401 *mW***	**0.401 *mW***
Total LaserPower		**20.1 *W***	**78.1*W***

control, but with interconnect width scaled down to approximate the average energy savings of EcoLaser across applications (Power_Eq). For each study, we compare the performance (user instructions per sec), energy per instruction (EPI), and energy-delay-product (EDP) of Adaptive, perfect laser control (Perfect), baseline no-control (No-Ctrl), Static-1, Static-10, and Power_Eq.

To evaluate the performance and energy efficiency benefits of ProLaser, we evaluate the load-latency and energy-per-flit of ProLaser, Simple and EcoLaser schemes, and compare them against a baseline without laser control (*No-Ctrl*), and a perfect control scheme with full knowledge of future messages (*Perfect*). We compare the performance (user instructions per sec), energy per instruction (EPI) of CMesh, the baseline scheme without laser control (No-Ctrl), Power_Eq, EcoLaser [16], Simple (Section 3.4.1), ProLaser, and perfect laser control (Perfect).

3.5.3 Laser Power Consumption Calculation

Table 1 shows the optical loss parameters for the modulators, demodulators, drop filters, and detectors introduced in [2] and assumed in this work. The modulation and demodulation energy is 150 *fJ/bit* at 10 *GHz* [2]. The laser power per wavelength and total laser power are calculated in Table 1 using the ana-

lytical models introduced in [33]. Because the number of turned-off rings on a single optical path is high for a radix-64 crossbar, we limit the network to 16 DWDM. The total laser power in Table 1 includes the laser power for both data and reservation channels, plus the laser efficiency of 10%, so it is the wall plug power for the laser. The data bus is 300-bits wide, so it can push a data message in one processor cycle (both edges of a 5 *GHz* clock).

3.5.4 Sensitivity to Optical Parameters

Unfortunately, there is little consensus on the optical loss parameters used or projected in literature. In same cases, parameters exhibit a variance over 10x across publications. However, we observe that the design of an optical interconnect highly depends on the losses of the optical components used. For example, if the off-ring through loss on the radix-16 crossbar was 10x higher (i.e., 0.1*dB*) the interconnect wouldn't employ 64-way DWDM, as this would increase the laser power to unsustainable levels. Rather, the interconnect would be optimized with a lower 6-way DWDM and it would employ more waveguides, resulting in a total optical loss (and hence laser power) similar to the interconnect modeled in our work. In the extreme case where the off-ring loss were to increase by 10x, and on top of that the modulator insertion, drop loss, detection and non-linearity losses were to double, a 4-way DWDM would accommodate the increased losses and keep the total laser power at the same level.

In either case, the fraction of laser energy that ProLaser saves depends on the network utilization, not on the optical loss parameters. Moreover, the higher the total optical loss, the more power in absolute terms ProLaser would save, which would have a higher impact on the performance of the processor if this power is given back to the cores. Thus, in this work, we remain conservative in our estimates of optical losses.

3.5.5 Resonant Ring Heater Modeling

To calculate the total ring heating power we extend the method by Nitta *et al.* [51] by additionally accounting for the heating of the photonic die by the operation of the cores. We model the thermal characteristics of a 3D-stacked architecture where the photonic die sits underneath the logic die using the 3D-chip extension of HotSpot [67]. When a workload executes, we calculate the ring heating power required to maintain the entire photonic die at the micro-ring trimming temperature during the entire execution. In addition, we account for the individual ring trimming power required to overcome process variations, as described in [33].

3.6 Experimental Results for EcoLaser

3.6.1 Network Performance

EcoLaser turns off the lasers and saves energy at the cost of higher message latency. At low injection rates, EcoLaser on SWMR has a 4-cycle latency overhead, which is lower than the 5-cycle laser turn-on delay, as some messages catch the laser on (Figure 24). The overhead decreases for higher injection rates as more messages catch the laser on. Similarly, EcoLaser on MWSR exhibits 8 cycles latency overhead, instead of the full 11-cycle laser turn-on delay, as the token design allows senders to transmit immediately when they find the laser on. Static schemes with high laser stay-on time "*K*" (keeping the laser on at least *K* cycles), increase the likelihood of finding the laser on, so they have lower latency overhead and provide higher throughput. However, they don't save much energy at low injection rates, as they may needlessly leave the lasers on. Static schemes with lower *K* turn off the lasers quickly, saving significant laser energy at low injection rates. However, they don't provide enough throughput under heavy utilization, increasing the overall energy consumption. Adaptive outperforms all Static schemes because it adjusts *K* at runtime, thus

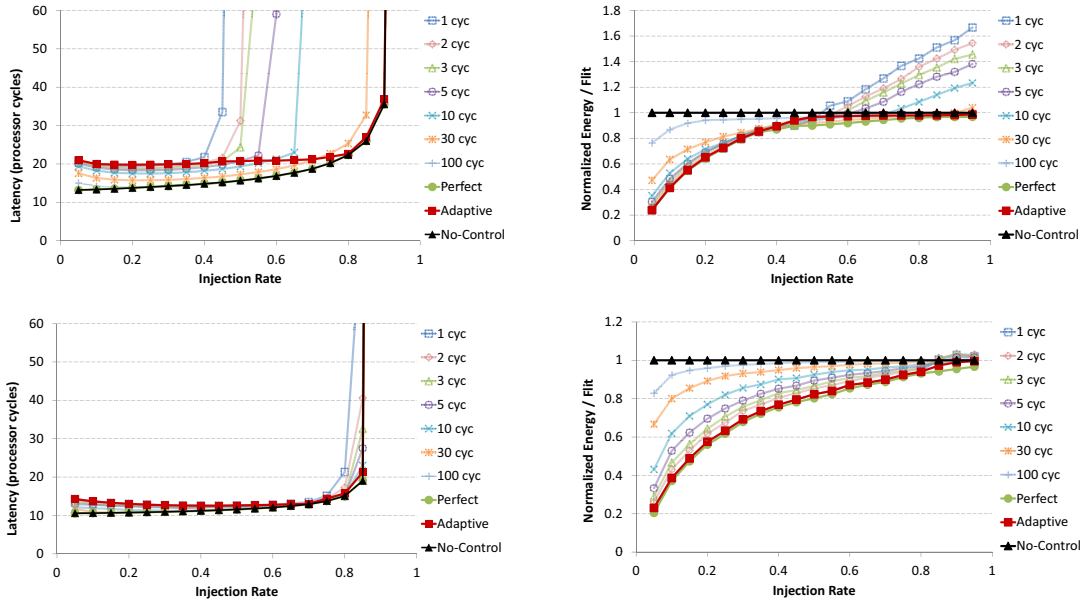


FIGURE 24: Load-Latency and Energy per Flit for radix-16 MWSR(top row) and SWMR (bottom row) Crossbars.

it achieves high energy savings at low injection rates, and high throughput at high injection rates. Adaptive's performance improvement over Static schemes is higher for MWSR, because it sends turn-on requests through the token stream (which takes longer), while SWMR can turn on or keep the laser on much quicker. Overall, Adaptive's energy consumption is within 2-3% of the Perfect scheme.

As the system scales, the contention on MWSR token arbitration increases, therefore the static schemes become more inefficient. Thus, Adaptive saves more laser energy than the static schemes as the system

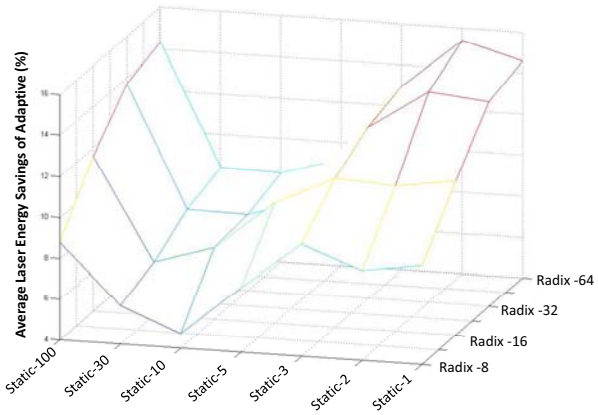


FIGURE 25: MWSR Scalability Analysis

scales (Figure 25). As Figure 25 indicates, the energy savings of Adaptive grow for higher radix crossbars, indicating its scalability. On average, Adaptive on MWSR_XBAR_64 saves 17% laser energy compared to No-Ctrl, which is only 2% higher energy consumption that Perfect's. We observe that Static-10 is the most energy-efficient of the static schemes for all crossbar sizes.

3.6.2 Performance cost of Laser Control

EcoLaser is expected to degrade performance compared to No-Ctrl, as sometimes transmission is delayed while the laser turns on. In reality, however, EcoLaser recoups the losses and even increases performance by minimizing thermal emergencies and core throttling that DVFS employs to keep a chip within safe operating temperatures. Controlling the laser lowers the power consumption by a significant margin compared to No-Ctrl, which allows for a cooler chip, reduces core throttling, and increases performance. Thus, even though EcoLaser trades off network latency for energy savings, a realistic power-limited system may exhibit higher performance with EcoLaser because the cores will be throttled less often.

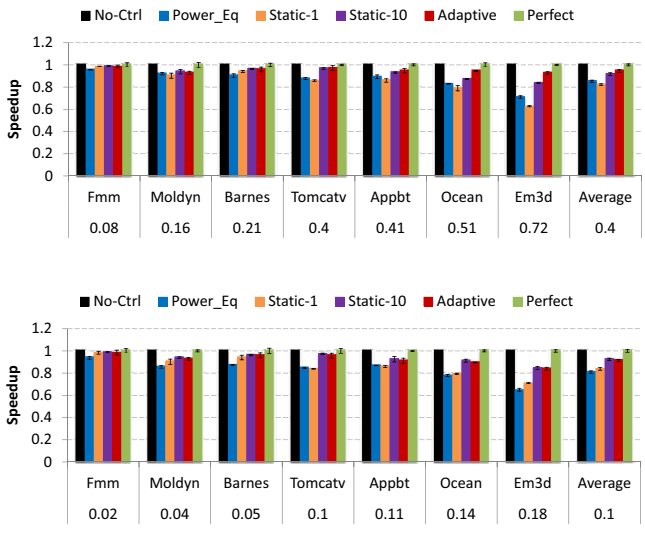


FIGURE 26: Speedup for radix-16 (top) and radix-64 (bottom) MWSR on a hypothetical multicore without thermal constraints.

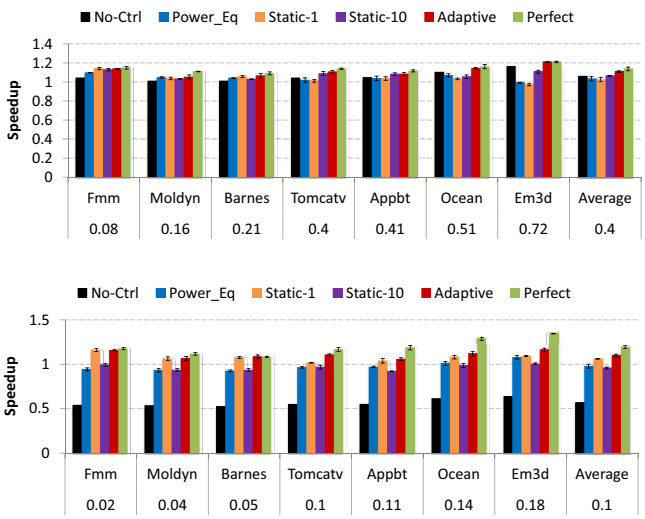


FIGURE 27: Speedup over CMesh for radix-16 (top) and radix-64 (bottom) MWSR crossbars under realistic thermal constraints.

We analyze the two effects (increasing the network latency, and reducing core throttling) separately. We analyze the performance cost of EcoLaser by evaluating it on a multicore that is not subject to thermal constraints, thus cores are not throttled and run at maximum frequency (5 GHz). Our workload suite includes both memory-intensive workloads that generate high traffic and are sensitive to interconnect latency (em3d, ocean, appbt, tomcatv), as well as compute-intensive workloads that are less sensitive to message latency (fmm, moldyn, barnes). Figure 26 summarizes our findings. The injection rate of each application appears below its name. Overall, laser control saves more energy on real-world workloads than on synthetic random traffic patterns, because real-world workloads typically have bursty (and sparse) memory access patterns.

In radix-16 MWSR, Static-1 saves the most laser energy (49% on average) at the expense of slowing down the memory intensive workloads. Static-10 achieves high throughput, but it wastes laser energy at compute-intensive workloads (saves 32% on average). Adaptive combines the benefits of both: it saves 45% of the laser energy on average for radix-16 and 68% for radix-64 MWSR crossbars, at the cost of 4.8% and

7.5% slowdown respectively, while on SWMR it saves 53% and 72% of the laser energy for radix-16 and radix-64 respectively, with only 4% slowdown.

Power_Eq is a scaled-down version of No-Ctrl (150-bit flits for radix-16, and 100-bit flits for radix-64) to approximate Adaptive's laser energy consumption. While it achieves similar energy savings, Power_Eq suffers from high serialization delays and underperforms EcoLaser. Thus, saving laser energy by reducing the width of the interconnect is not a good alternative to laser control.

3.6.3 Impact of EcoLaser on a Realistic Multicore

Under realistic thermal (power) constraints, DVFS in No-Ctrl throttles the cores to keep the chip within a safe temperature. EcoLaser, however, reduces the laser power and results in a cooler chip, less core throttling, and higher performance. The static schemes typically work well at only one end of the spectrum. Static-1 speeds up workloads with low injection rates, as it saves the most power and reduces throttling, but slows down memory-intensive workloads due to frequent laser turn-on delays (Figure 27-top). Static-10 speeds up workloads with high injection rates, as it increases the likelihood that a sender finds the laser on

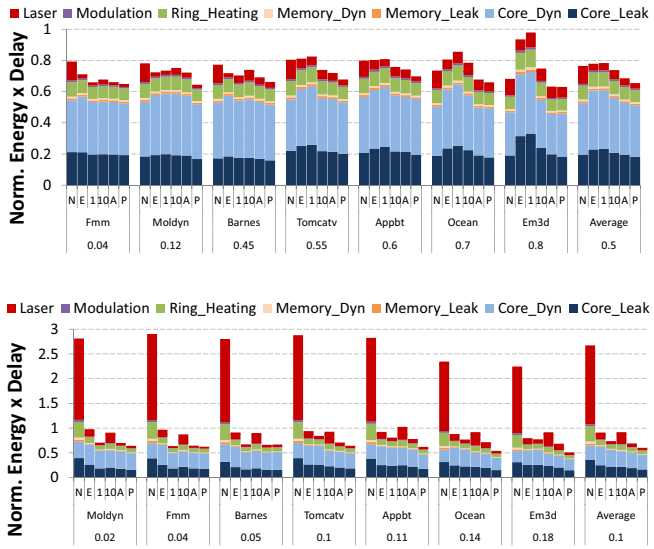


FIGURE 28: Energy x Delay Product in radix-16 (top) and radix-64 (bottom) MWSR crossbar. The evaluated designs are from left to right: No-Ctrl (N), Power_Eq (E), Static-1 (1), Static-10 (10), Adaptive (A), and Perfect (P).

and transmits without delay, but wastes power when the injection rate is low and leads to more core throttling. Power_Eq achieves low laser power, but at the expense of serialization delays due to its limited width. Overall, the performance and energy-delay product (EDP, Figure 28) of the static schemes is much worse than that of Perfect's. Thus, static laser control or reduced width often lead to slow and energy-inefficient systems.

Adaptive EcoLaser tracks the workload's needs, and provides both low power and high throughput. The impact of EcoLaser is more pronounced on 64-radix crossbars, because their energy savings are a significant fraction of the total chip power, and hence allow the cores to run faster. For example, Perfect runs fmm at 3.25 GHz, Adaptive at 3.2 GHz, and No-Ctrl at only 1.5 GHz. For the same reason, No-Ctrl is 1.7x slower than CMesh even though it has higher bandwidth and lower latency. Compared to No-Ctrl, adaptive EcoLaser on radix-64 MWSR and SWMR crossbars is 2x faster and has 74-77% lower EDP on average (10% faster and 20% lower EDP for radix-16). In all cases, Adaptive's performance and EDP are within 2-6% of Perfect's.

3.7 Experimental Results for ProLaser

3.7.1 Network Performance

Laser control saves energy by turning off the lasers whenever the data bus idle. Energy savings come with the potential cost of increased message latency, because messages may have to wait for the laser to turn back on. ProLaser scheme we proposed turns the laser on proactively by anticipating upcoming messages, so that majority of the messages don't have to wait. Hiding the laser turn-on delay, ProLaser minimizes the latency overhead of the laser control on the network performance. On top of that, by keeping the data-only portion of the data bus inactive while sending small (dataless) messages, ProLaser achieves higher energy savings than the previously proposed EcoLaser scheme. We investigate the laser energy savings and the

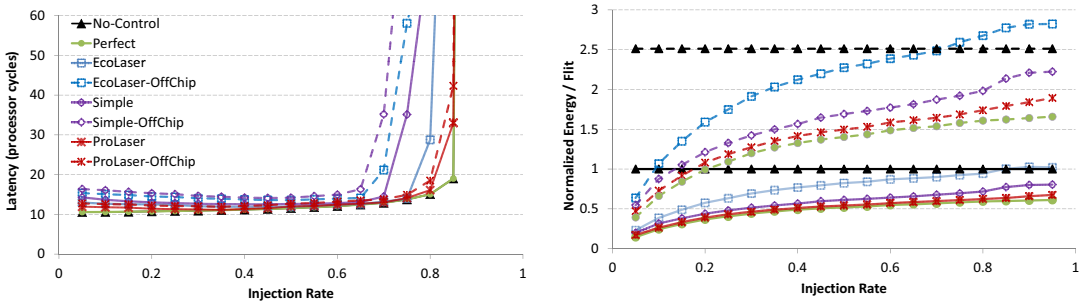


FIGURE 29: Load-Latency (left) and Energy-per-Flit (right) for a radix-16 SWMR crossbar.

network performance trade-off of ProLaser scheme on a radix-16 SWMR crossbar using random traffic pattern and compare it against No-Ctrl, EcoLaser [16], Simple (Section 3.4.1) and Perfect schemes. We extend this evaluation for both on-chip and off-chip laser sources.

The Ge-based on-chip lasers assumed in this work exhibit 5-cycle (1 ns) laser turn-on delay, however Eco-Laser exhibits 4-cycle delay overhead at low injection rates, because some of the messages find the laser active and transmit immediately (Figure 29). This overhead slightly decreases when the injection rate increases, because more messages find the laser active. Simple scheme exhibits slightly higher laser turn-on delay overhead, because data messages can't catch the laser active as the data-only portion of the data-bus is turned off more frequently. On the other hand, ProLaser shows only 1-cycle delay overhead at low injection rates, because it foresees majority of the messages and activate the laser ahead of time. EcoLaser and Simple saturate faster as the injection rate grows, providing 5-10% lower throughput respectively, because of high laser turn-on delay overhead. ProLaser doesn't suffer from throughput decrease, because it hides the laser turn-on delay by turning the lasers on proactively.

Controlling off-chip laser source requires control signals to be sent back to the off-chip laser source and the light generated by the laser source to travel to the sender, which incur additional latency overhead. We estimate that, signaling the laser source and source to sender light travel takes 2 cycles each way (2 cm waveguide plus 4 cm fiber travel). While the performance of EcoLaser and Simple suffers from this additional

latency overhead with additional decrease in their throughput, ProLaser manages to hide most of it and shows slight decrease in its performance.

Laser control achieves high energy savings by turning the lasers off when the crossbar utilization is low. EcoLaser's energy savings disappear as the injection rate grows, whereas Simple and ProLaser manages to save more by keeping the data-only portion of the data bus inactive longer (Figure 29). ProLaser consumes lower energy per flit than Simple because it provides higher throughput with lower message latency. On average over injection rates, ProLaser consumes 33% lower laser energy per flit compared to EcoLaser, and achieves savings 3% lower than Perfect.

When using an off-chip laser source, the coupling losses required to carry the light on to the chip (~4dB) increases the laser power consumption by 2.5x. Coupling losses and the additional control latency overhead makes EcoLaser, Simple and ProLaser consume higher laser energy per flit with an off-chip laser source (Figure 29). On average over injection rates, ProLaser consumes 35% lower laser energy per flit compared to EcoLaser, and achieves savings 6% lower than Perfect.

3.7.2 Performance cost of Laser Control

ProLaser trades off laser energy savings to increased message latency, therefore it is expected to lower the performance of the system compared to No-Ctrl. However, in a realistic system with on-chip lasers, laser energy saved by ProLaser may decrease the thermal emergencies and decrease the need for core throttling, thus increase the performance. Previous work showed that, EcoLaser [16] lowers the on-chip laser power consumption, which allows better cooling, reduces core throttling, and increases performance of a realistic multicore implementing a thermal management system like DVFS. ProLaser lowers the power consumption by a significant margin, and has less laser turn-on latency overhead compared to EcoLaser, therefore a

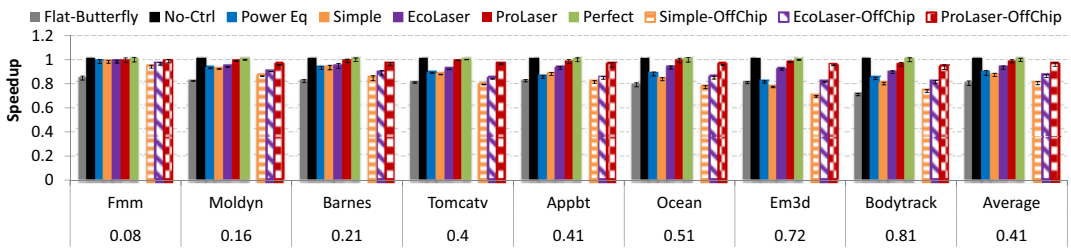


FIGURE 30: Speedup for radix-16 SWMR on a hypothetical multicore without thermal constraints.

realistic power-limited system may exhibit higher performance with ProLaser because the cores will not be throttled as much.

We aim to analyze the effect of network latency increase and reduced need for core throttling separately. First, we analyze the performance cost of ProLaser by evaluating it on a multicore that is not subject to thermal constraints, thus cores run at maximum frequency (5 GHz) without being throttled. Our workload suite includes both memory-intensive workloads that generate high traffic and are sensitive to interconnect latency (bodytrack, em3d, ocean, appbt, tomcatv), as well as compute-intensive workloads that have low injection rates and are less sensitive to message latency (fmm, moldyn, barnes). Figure 30 summarizes our findings. The injection rate of each application appears below its name.We also present the performance of multicore with a traditional electrical network (Flat-Butterfly) for reference.

Simple laser control has the highest laser turn-on latency overhead, therefore it under performs other schemes when running memory intensive workloads (Figure 30). On average, Simple saves 60% of the on-chip laser energy while causing 13% slowdown compared to No-Ctrl. EcoLaser outperforms Simple, causing 7% slowdown while saving only 35% of laser energy on average. With the support of partial activation of the data bus and proactive laser turn on, ProLaser saves 63% of the laser energy (88% maximum) while causing only 1.5% slowdown on average when compared to No-Ctrl. Furthermore, laser energy savings of ProLaser is in the vicinity of 2% - 3% of Perfect running real-world workloads. Overall, ProLaser exhibits higher energy savings on real-world workloads than on synthetic random traffic patterns, because real-

world workloads typically have bursty (and sparse) memory access patterns. Laser control schemes controlling off-chip laser sources achieve similar laser energy savings, however they cause higher slowdown in performance because of higher turn-on latency overhead. Additional latency overhead hurts the Simple's performance the most, while ProLaser hides the additional latency. Controlling off-chip laser with ProLaser causes only 4% slowdown, while saving 61% of the laser energy on average.

Power_Eq approximates ProLaser's laser energy consumption by scaling down its width (100-bit flits instead of 300-bit flits), but otherwise is similar to No-Ctrl. While achieving similar energy savings, Power_Eq suffers from high serialization delays and under performs ProLaser. Thus, saving laser energy by reducing the width of the interconnect is not a good alternative to laser control.

3.7.3 Impact of EcoLaser on a Realistic Multicore

Wall-plug power consumption of the on-chip laser sources can be a significant portion of the multiprocessor's power budget, because of the low laser efficiency levels (15% efficiency [38]). Under realistic thermal (power) constraints, DVFS in No-Ctrl throttles the cores to keep the chip within a safe temperature. ProLaser, however, reduces the laser power and results in a cooler chip with less core throttling, and higher performance.

Off-chip laser sources consume ~2.5x higher laser power compared to on-chip counterparts because of the coupling losses. However, majority of the off-chip laser power (Wall-plug power) is dissipated away from the multicore chip, therefore we don't expect to observe ProLaser's performance increase by cooling effect with off-chip laser sources. On the other hand, the power consumption of the off-chip lasers can be as high

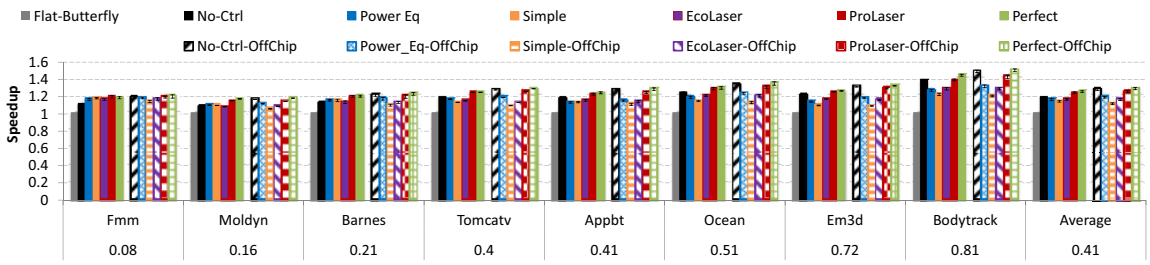


FIGURE 31: Speedup over Flat-Butterfly for radix-16 SWMR crossbar under realistic thermal constraints.

as the power budget of the multiprocessor (because of high coupling losses and low efficiency levels), so the impact of ProLaser's energy savings on the total energy of the system will be significant.

It is important to note that, the impact of ProLaser's energy savings on system performance and energy efficiency depends on the total laser power consumption of the photonic network. In order to make a fair evaluation of the impact of ProLaser, we investigate two case studies: radix-16 crossbar and Firefly topology. The radix-16 crossbar approximates a worst case scenario for ProLaser. It has low power consumption (similar to the power consumption of Flat-Butterfly) and its high concentration factor (4) creates heavier traffic. The low power consumption and heavy traffic limit ProLaser's opportunity. Previously proposed, high performance optical interconnect Firefly [57] corresponds to a better case for ProLaser. It has high laser power consumption (4x laser power of radix-16) and a low concentration factor (1), which results in light traffic, thus giving ample opportunity to ProLaser to conserve laser power.

3.7.4 Case Study: Radix-16 SWMR

The wall plug power consumption for on-chip lasers on radix-16 crossbar is 14.1W. All laser control schemes saves significant portion of this power and cause minimal slowdown when running compute-intensive workloads, therefore they outperform No-Ctrl (Figure 31). However, Simple and EcoLaser under perform No-Ctrl when running memory-intensive workloads, because of their high laser turn-on latency overhead and low laser energy savings. Power_Eq achieves similar energy savings to ProLaser but suffers

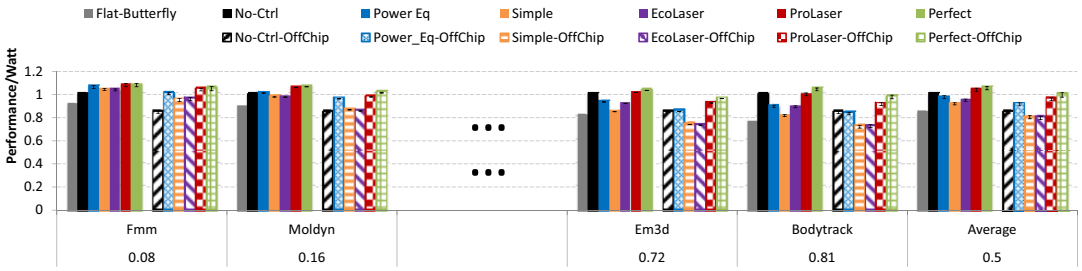


FIGURE 32: Performance per Watt over No-Ctrl for radix-16 SWMR crossbar under realistic thermal constraints.

from serialization delays. On the other hand, ProLaser outperforms No-Ctrl for all workloads (1.05x on average), because it achieves high performance with high laser energy savings. ProLaser outperforms Eco-Laser by 1.07x and has 6% lower EPI on average (Figure 31).

For radix-16 crossbar, off-chip lasers consume 37.1 W and only 2.25 W of this power is dissipated on the multiprocessor. Majority of the off-chip laser power (Wall-plug power) is dissipated away from the multi-core chip which allows microprocessor to run faster, however off-chip lasers also increase the total power consumption of the system significantly. As a result, even thought the radix-16 crossbar with on-chip lasers is 7% slower, it provides 16% more performance per watt consumed when compared to the one with off-chip lasers. All laser control schemes with off-chip lasers are slower than No-Ctrl, because the cooling effect of laser power savings is insignificant. However, laser energy saved by ProLaser is a significant portion of the total laser energy consumption, which reduces the total energy consumption of the multiprocessor. ProLaser-OffChip is only 2% slower than No-Ctrl-OffChip (Figure 31)), but has 13% lower EPI on

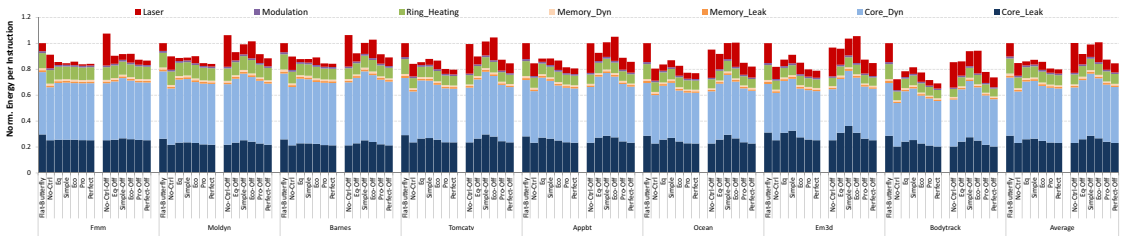


FIGURE 33: Energy Per Instruction for radix-16 SWMR crossbar. The evaluated designs are from left to right: Flat-B., No-Ctrl, Power_Eq (Eq), Simple, EcoLaser (Eco), ProLaser (Pro), and Perfect and their Off-chip implementations (-Off).

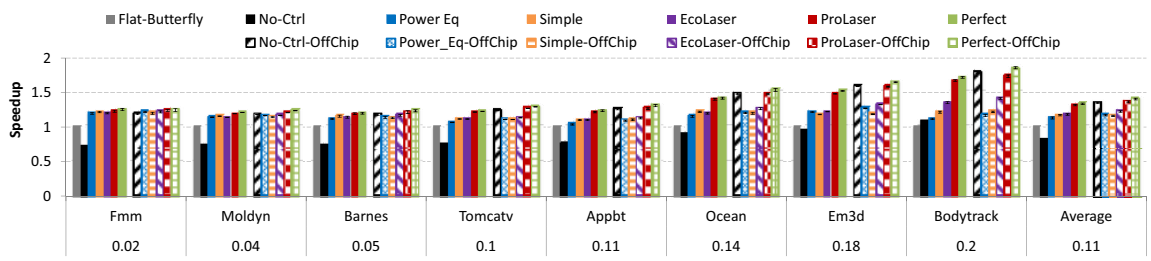


FIGURE 34: Speedup over Flat-Butterfly for Firefly topology under realistic thermal constraints.

average (Figure 33). On the other hand, EcoLaser is both slower and consumes higher energy than No-Ctrl for off-chip lasers. This shows us that, ProLaser's partial control on the data bus and proactive laser turn-on is essential for controlling off-chip lasers. ProLaser outperforms EcoLaser by 1.08x, and has 14% lower EPI than EcoLaser. In all cases, ProLaser's performance and EPI are within 2-3% of Perfect's (Figure 33).

3.7.5 Case Study: Firefly

Firefly topology consists of 4 radix-16 SWMR crossbars, therefore the wall plug power consumption for on-chip lasers is 56.5W. All laser control schemes outperform No-Ctrl on all workloads, because laser energy savings is a considerable portion of the multiprocessor's power budget (Figure 34). ProLaser outperforms No-Ctrl for all workloads (1.6x on average) and has 40% lower EPI on average. ProLaser outperforms EcoLaser by 1.1x and has 9% lower EPI on average (Figure 36). We observe that, the previously proposed high-performance Firefly [57] topology wouldn't be able to deliver promised performance on a realistic multiprocessor (No-Ctrl). However, ProLaser enables Firefly to deliver its performance to the fullest by making it more energy proportional.

Off-chip lasers consume 152 W for Firefly, and only 9 W of this power is dissipated on the multicore processor, which means the majority of the power is dissipated away from the processor and the cooling effect of laser power savings is still insignificant. The power consumption of on-chip lasers slows down the processor cores due to throttling, but on the other hand off-chip lasers more than double the total power consumption of the system. As a result, a Firefly with on-chip lasers is 65% slower, however it provides 22%

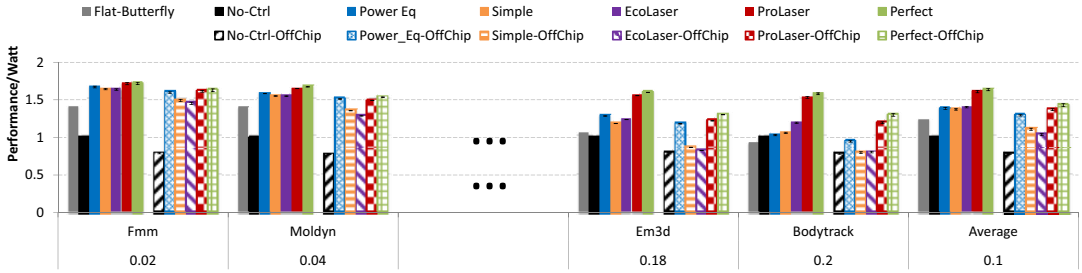


FIGURE 35: Performance per Watt over No-Ctrl for Firefly topology under realistic thermal constraints.

more performance per watt consumed when compared to the Firefly powered by off-chip lasers. On average ProLaser-OffChip is slightly slower than No-Ctrl-OffChip, but has 44% lower EPI on average (52% maximum, Figure 36). ProLaser outperforms EcoLaser by 1.11x, and has 21% lower EPI on average than EcoLaser. In all cases, ProLaser's performance and EPI are within 3-4% of Perfect's, which shows that ProLaser is harvesting the majority of the possible laser energy savings.

It is important to note that, when Firefly [57] topology is implemented using an off-chip laser source, on average, 55% of the total system energy would be consumed by this laser source (No-Ctrl-OffChip, Figure 36). However, ProLaser can reduce laser energy consumption by 5.1x on average by controlling the off-chip lasers, leaving only 17% of the total system energy to the lasers to consume making the photonic interconnect more energy proportional.

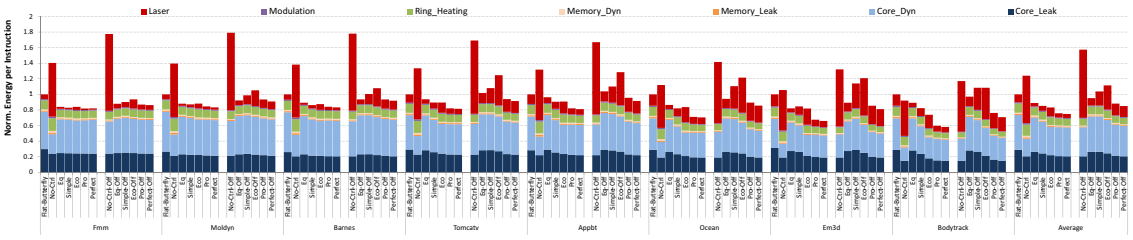


FIGURE 36: Energy Per Instruction for Firefly topology.The evaluated designs are from left to right: Flat-B., No-Ctrl, Power_Eq (Eq), Simple, EcoLaser (Eco), ProLaser (Pro), and Perfect and their Off-chip implementations (-Off).

3.7.6 Laser Turn-on Latency Tolerance

ProLaser foresees the majority of the messages and activates the lasers proactively, thus it can tolerate higher laser turn-on delays with minimal performance penalty. In contrast, EcoLaser lacks a proactive laser turn-on mechanism and is susceptible to high laser turn-on delays. Figure 37 reports the average message latency and the laser energy savings for No-Ctrl, EcoLaser, and ProLaser as a function of different laser turn-on delays, under uniform traffic with an injection rate similar to the average injection rate of our benchmark suite (0.11 packets/router/cycle). The core speed is set to 2.5 GHz to reflect the average speed of the power-limited multicores evaluated in Section 3.7.3.

Figure 37 shows that the network performance and the laser energy savings (LES) decrease with increasing laser turn-on delay, which emphasizes the need for fast lasers. Laser control schemes become inefficient with high laser turn-on delays, because they slow down the system and end up consuming more laser energy to send messages. EcoLaser can tolerate up to only 3 *ns* laser turn-on delay, where is saves 18% of the laser energy and increases message latency by 50%, and becomes impractical beyond that. A ProLaser scheme without Bloom filters that relies on early L2-tag lookup has 3-4 cycles between the L2 tag lookup (laser turn-on) and the L2 hit, so it can tolerate higher laser turn-on delays than EcoLaser (up to 5 *ns*). Pro-Laser with Bloom filters hides the increased laser turn-on delay even more, because the laser controller has

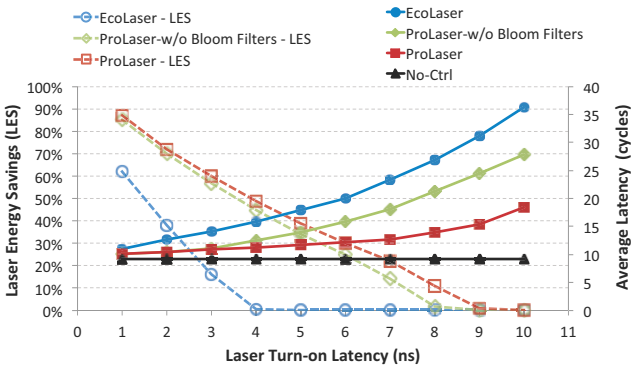


FIGURE 37: Laser Turn-on Latency Tolerance.

14 cycles to turn on the lasers between the Bloom filter lookup and L2 cache hit. ProLaser can tolerate up to 7 *ns* laser turn-on delay, where it still saves 23% of the laser energy compared to No-Ctrl. In conclusion, early laser turn-on prediction with Bloom filters allows ProLaser to withstand 2.3x higher turn-on delays than the state of the art (EcoLaser). This allows ProLaser with Bloom filters to remain an effective laser control scheme even under relatively high laser turn-on delays, when competing schemes fail.

Chapter 4

Introducing Laser Control in a Flattened Butterfly Network

4.1 Motivation

Silicon-photonics have emerged as the prime candidate technology for energy-efficient chip-to-chip interconnects because they provide low-latency, high-bandwidth and energy-efficient communication over long distances. Scaled-out systems, such as multi-chip systems and the datacenters exploit scalable photonic network topologies, such as “Flattened butterfly” topology. Previously proposed laser power-gating schemes improve the energy efficiency of the on-chip interconnects (Chapter 3), however they fail to extend to the flattened butterfly topology. Flattened butterfly is a scalable topology which provides path-diversity between source and destination pairs, so it can provide high throughput across thousands of nodes, while keeping the hardware cost at bay. Energy efficiency and proportionality is desirable for not only the on-chip photonic interconnects, but also for the multi-chip systems and the datacenters with photonic networks, because they also waste laser energy when idle.

Flattened butterfly has been proposed as an on-chip electrical interconnect [34], which uses long electrical links running across the chip in both dimensions to connect the row and column neighboring routers efficiently. For a photonic flattened butterfly, a similar implementation using waveguides would result in high number of waveguide intersections which require high-power lasers and reduce the energy efficiency. A serpentine shaped waveguide layout can be used to avoid the waveguide intersections, however, due to it’s shape, it wouldn’t connect the row and column neighboring routers directly, which introduces additional latency and laser energy consumption. We introduce the Divergent Flattened Butterfly

layout (D-FBFLY) which is a waveguide layout for photonic implementation of the flattened butterfly topology. D-FBFLY aims to connect the row and column neighboring routers in an efficient way while avoiding the waveguide crossings, which leads to a high-performance and energy-efficient implementation of the photonic flattened butterfly topology. Both the on-chip and the multi-chip implementations (similar to [39]) can exploit D-FBFLY. Compared to the serpentine shaped layout, D-FBFLY can save up to 50% of the laser energy (multi-chip scale), and achieve 1.08x-1.16x overall speedup.

Laser power-gating is a promising technique to reduce the high laser power consumption of the photonic interconnects, however, it reduces the performance when messages have to wait for the laser turn-on. On a flattened butterfly, power-gating photonic links naively may result in significant performance degradation, because message may end up waiting for the laser turn-on multiple times. We propose SLAC, a laser control scheme for flattened butterfly network which turns off majority of the network to save laser energy, while maintaining a fully connected network which removes the laser turn-on latency from the critical path and causes minimal (next to nothing) performance decrease. SLAC turns off majority of the network when the utilization is low to save energy, and activates additional stages when the utilization is high to provide better performance. From an on-chip interconnect to a multi-chip system to a datacenter network, any network with flattened butterfly topology can take advantage of SLAC. Our results show that, for on-chip and multi-chip applications, SLAC can save up to 67% laser energy while reducing the performance by only 2% while running real-world workloads. On a flattened butterfly datacenter network, SLAC saves 79% laser energy on average while running the traces collected from a university server [6].

(a) (b)

FIGURE 38: Electrical link (a) and Serpentine waveguide (b) layout for flattened butterfly topology

4.2 Divergent Flattened Butterfly Layout

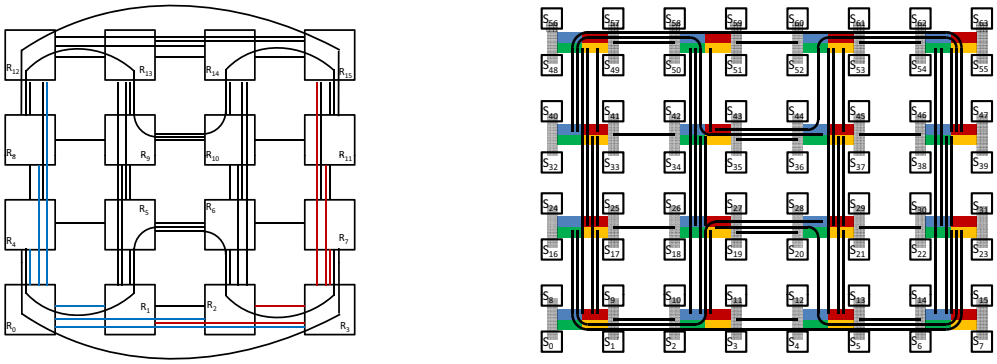
Traditional on-chip flattened butterfly network[34] uses long electrical links to connect the row and the column neighboring routers (Figure 38a). These links provide efficient message transfer between the routers because they connect the source and the destination pairs in the shortest way. A photonic implementation of FBFLY with a similar layout to electrical on-chip FBFLY[34] is not practical, because it requires waveguide crossings. When waveguides cross, every wavelength in a waveguide imposes crosstalk over every other wavelength in the crossing waveguide, which reduces the signal quality. In order to maintain the quality of the communication, high laser power is needed which reduces the energy efficiency and makes the photonic FBFLY impractical

The serpentine waveguide layout shown in Figure 38b avoids waveguide crossings, but causes unnecessary message latency between hops and additional laser power consumption due to long waveguides. Joshi et al. introduced serpentine waveguide layout for a Clos network [33], and he showed that messages may end up travelling across the chip twice due to intermediate hops, which increases the average message latency. In the case of photonic FBFLY, on this serpentine waveguide layout (S-FBFLY), a message from the Router 1 to the Router 2 (Figure 38b) has to travel the whole chip twice (over the Router 12 and the Router 14),

whereas it takes a short direct link with the electrical implementation (Figure 38a). Secondly, S-FBFLY requires longer links compared to the electrical layout, for example the link connecting the Router 0 to the Router 4 is approximately 2.5x longer. The amount of optical loss in a waveguide is proportional to the waveguide length, so these longer links require more powerful lasers which reduce the energy-efficiency.

In this paper we propose the Divergent Flattened Butterfly (D-FBFLY) layout (Figure 39) which aims to connect the routers using shortest links possible while avoiding crossings. The D-FBFLY achieves this by allowing waveguides run across the chip in one dimension (i.e. y-dimension) and routing the waveguides in the other dimension (i.e. x-dimension) around and in between them to connect the routers in the shortest possible way with out intersecting. The Figure 39 presents the proposed D-FBFLY layout, which shows the straight waveguides connecting routers in the x-dimension and the ones diverging and wrapping around in the y-dimension. An example for the waveguide wrapping around would be the one connecting the Router 8 to the Router 10.

It is important to note that, in D-FBFLY all of the routers are connected to their immediate neighbors via a short straight waveguide. Therefore a message that needs to travel across the chip twice with S-FBFLY (from the Router 1 to the Router 2) can take a small and direct hop with D-FBFLY. Another important point is, the longest link of D-FBFLY is 2.5x shorter than the longest link of the S-FBFLY (from the Router 0 to



(a) **(b)**

FIGURE 39: On-chip (a) and Multi-chip (b) Divergent Flattened Butterfly Layout

TABLE 6. Architectural Parameters.

CMP Size	64 cores, 480mm^2
Processing Cores	ULTRASPARC III ISA, up to 5Ghz, OoO, 4-wide dispatch/retirement, 96-entry ROB
L1 Cache	Split I/D, 64KB 2-way, 2-cycle load-to-use, 2 ports, 64-byte blocks, 32 MSHRs, 16-entry victim cache
L2 Cache	Shared, 512 KB per core, 16 way, 64-byte blocks, 14 cycle-hit, 32 MSHRs, 16-entry victim cache
Memory Controllers	One per 4 cores, 1 channel per Memory Controller Round-robin page interleaving
Main Memory	Optically connected memory [2], 10ns access
Networks	radix-16 SWMR and Firefly

the Router 4). As a result, D-FBFLY achieves lower average message latency while consuming less laser energy.

4.2.1 .Experimental Methodology

4.2.1.1 Interconnect Performance and Energy Analysis

To evaluate the performance and the energy consumption of D-FBFLY layout for flattened butterfly on-chip network in isolation from the interference of the other system components or application characteristics, we employ a cycle-accurate network simulator based on Booksim 2.0 [15], which models a 4-ary 3-flat flattened butterfly network servicing a random uniform traffic (with concentration of 4). The simulator models a three-cycle router, with the 1-cycle E/O and O/E conversions. We assume a 480 mm^2 chip, which means the S-FBFLY employs a 10 *cm* waveguide with a round trip time of 5 cycles. The link latency (1-3 cycles) is calculated based on the traversed waveguide length. The buffers are 20-flits deep, with a flit size of 300 bits. The maximum core frequency is 5 *GHz*, and the optical interconnect runs at 10 *GHz*. Latency is measured as the time required for the network to process a sample of injected packets. We evaluate the load-latency characteristics of D-FBFLY and compare it against an S-FBFLY, a radix-64 SWMR photonic crossbar[57] and an electrical flattened butterfly network. In order to make a fair comparison we equate the

average power consumption of radix-64 SWMR crossbar and electrical flattened butterfly to the power consumption of D-FBFLY by adjusting their data-path width (flit size).

For the multi-chip (wafer) scale flattened butterfly implementation, we model an 8-ary 3-flat flattened butterfly network where we assume S-FBFLY uses a 75 cm waveguide with a round trip time of 38 cycles. The flit size is 50 bits. The link latency (2-15 cycles) is calculated based on the length of the traversed waveguide length.

4.2.1.2 Multicore System Performance and Energy Analysis

To evaluate the impact of D-FBFLY layout on a realistic multicore system, we model a 64-core processor on a full-system cycle-accurate simulator based on Flexus 4.0 [27,75] integrated with Booksim 2.0 [15] and DRAMSim 2.0 [62]. Table 2 details the architectural modeling parameters for the on-chip analysis. We assume a shared and physically distributed L2 cache and directories. The memory controllers are uniformly distributed on the chip, and they use the same physical interconnect with the VCs to avoid deadlock. All messages below L1 cache traverses the interconnect. The power consumption of the electrical interconnect is calculated using DSENT [69]. We target a 16 *nm* technology, and have updated our tool chain accordingly based on ITRS projections [23]. The simulated system executes a selection of benchmarks from SPLASH-2, PARSEC and other scientific workloads. For the multi-chip implementation analysis, we conduct a similar size simulation assuming each thread is located in a different site.

4.2.1.3 Laser Power Consumption Calculation

We compare the laser power savings of D-FBFLY over S-FBFLY for both the on-chip and the multi-chip implementations. Table 7 shows the optical loss parameters for the modulators, demodulators, drop filters, and detectors introduced in [2] which are assumed for on-chip implementation, as well as the optical loss

parameters introduced in [39] which are assumed for multi-chip integration. The modulation and demodulation energy is 150 *fJ/bit* at 10 *GHz* [2] for both implementations. The laser power per wavelength and the total laser power are calculated in Table 1 using the analytical models introduced in [33].

For the on-chip implementation, the data-path width for the on-chip D-FBFLY is 300-bits (so it can push a data message in one processor cycle using both edges of a 5 *GHz* clock). The on-chip D-FBFLY can be powered by both the on-chip and the off-chip laser sources, so we calculated the laser power consumption for both (note that an off-chip laser source has higher efficiency but it introduces additional coupler loss).

For the multi-chip implementation the data-path width for the D-FBFLY is 50-bits. The multi-chip D-FBFLY requires powerful lasers, so we assumed only off-chip laser sources. In [39], authors assume an aggressive waveguide loss parameter of 0.05 dB/cm, so we calculated the laser power consumption for both the traditional and the aggressive waveguide loss assumptions (aggressive assumption is noted with a * in Table 7).

TABLE 7. Nanophotonic Parameters and Laser Power.

On-Chip		**S-FBFLY**	**D-FBFLY**	**Multi-Chip**		**S-FBFLY**	**D-FBFLY**
	per Unit	**Total**	**Total**		**per Unit**	**Total**	**Total**
DWDM		64	64	DWDM		16	16
Splitter	0.2 *dB*	0.6 *dB*	0.6 *dB*	WG Loss	0.3 *dB/cm*	7.5 *dB*	4.5 *dB*
WG Loss	0.3 *dB/cm*	1.5 *dB*	0.75 *dB*	WG Loss*	*0.05 dB/cm*	*1.25 dB*	*0.75 dB*
Nonlinearity	1 *dB*	1 *dB*	1 *dB*	Bridge WG Loss	*1 dB*	*1 dB*	*1 dB*
Modulator Ins.	0.5 *dB*	0.5 *dB*	0.5 *dB*	Modulator Ins.	4 *dB*	4 *dB*	4 *dB*
Ring Through	0.01 *dB*	0.63 *dB*	10.24 *dB*	Ring Through	0.05 *dB*	0.8 *dB*	0.8 *dB*
Filter Drop	1.2 *dB*	1.2 *dB*	1.2 *dB*	Filter Drop	1 *dB*	1 *dB*	1.2 *dB*
Receiver Margin	4 *dB*	4 *dB*	4 *dB*	Receiver Margin	4 *dB*	4 *dB*	4 *dB*
Coupler	2 *dB*	2 dB	2 *dB*	Coupler	2 *dB*	6 *dB*	6 *dB*
Total Loss		**9.43 *dB***	**8.68 *dB***	**Total Loss**		**24.3 *dB***	**21.3 *dB***
Detector		**-20 *dBm***	**-20 *dBm***	**Detector**		**-20 *dBm***	**-20 *dBm***
Laser Power Per Wavelength		**0.087 *mW***	**0.073 *mW***	**Laser Power Per Wavelength**		**2.6915 *mW***	**1.34896 *mW***
On-Chip LaserPower	*10% Eff.*	**25.25 *W***	**21.25 *W***	**Total LaserPower**	*30% Eff.*	**397.91 *W***	**199.43*W***
Off-Chip LaserPower	*30% Eff.*	**13.21 *W***	**11.11 *W***	**Total LaserPower***	*30% Eff.*	**94.3 *W***	**84.1*W***

4.2.2 .Experimental Results

4.2.2.1 On-Chip Implementation Power and Performance

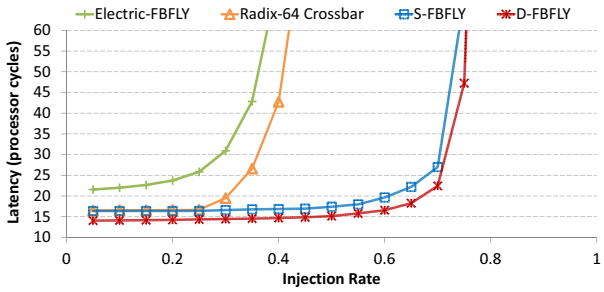
We compared the D-FBFLY to an S-FBFLY on-chip interconnect with the similar bisection width. D-FBFLY achieves better performance with lower laser power consumption, because it uses shorter links to connect the routers. By reducing the length of the photonic links, D-FBFLY consumes 16% lower laser power compared to the S-FBFLY (Table 1). Figure 40a presents the load-latency characteristics of the D-FBFLY compared against the S-FBFLY, radix64-crossbar and electrical flattened butterfly (Electric-FBFLY). Under random uniform traffic, D-FBFLY achieves 14 cycle zero-load message latency on average which is 2.4-cycles lower than the S-FBFLY. D-FBFLY provides 1.88x and 2.14x higher throughput compared to an equal power radix-64 crossbar and the Electric-FBFLY respectively (Figure 40).

Figure 40b presents the performance analysis of a multi-core processor with a D-FBFLY compared against a S-FBFLY, radix64-crossbar and an electrical flattened butterfly (Electric-FBFLY). D-FBFLY provides low latency and high throughput communication, so a multi-core with D-FBFLY is 1.08x faster than the S-FBFLY on average (1.17x maximum) when running real world workloads. The performance impact of D-FBFLY is more predominant for memory-intensive workloads of which performance highly depend on the network performance.D-FBFLY outperforms the radix-64 crossbar and the Electric-FBFLY by 1.21x and 1.33x respectively.

4.2.2.2 Multi-Chip Implementation Power and Performance

We compared D-FBFLY to the S-FBFLY layout on a multi-chip implementation (on a wafer size silicon interposer) with a similar bisection width. D-FBFLY has 9-cycles lower zero-load average message latency

(a)



(b)

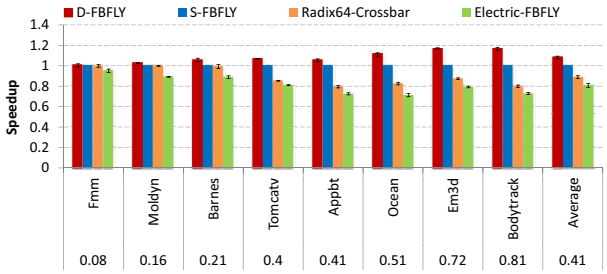
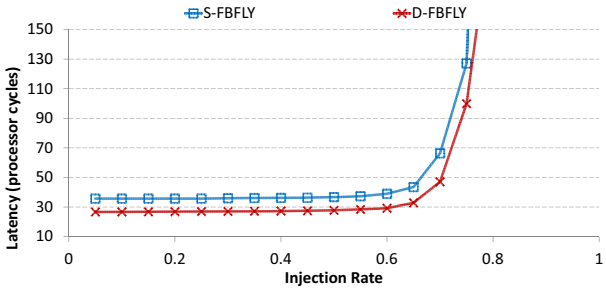


FIGURE 40: Load-Latency (a) and Speedup (b) for on-chip D-FBFLY and S-FBFLY layouts

(26.6 cycles on average) under random uniform traffic. D-FBFLY is1.16x faster than S-FBFLY on average (1.22x maximum) while running real-world workloads (Figure 41).

(a)



(b)

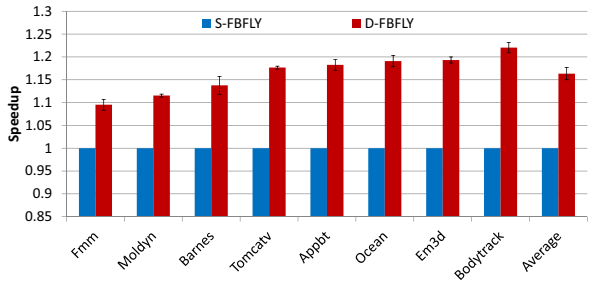


FIGURE 41: Load-Latency (a) and Speedup (b) for wafer size D-FBFLY and S-FBFLY layouts

The laser energy savings of D-FBFLY is more significant for the multi-chip implementation because the photonic links are much longer. D-FBFLY reduces the laser power consumption by 50% compared to the S-FBFLY because it doesn't need long links. The aggressive link loss assumption reduces the impact of the waveguide loss on the network energy consumption, however even with the 0.05 dB/cm loss D-FBFLY achieves 11% lower laser energy consumption compared to the S-FBLY.

4.3 Stage Laser Control Scheme

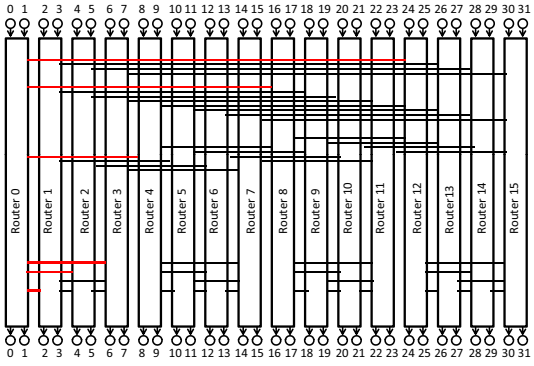
The laser control schemes aim to save laser energy by turning the lasers off whenever the photonic links are not utilized. Energy savings come at the cost of increased message latency, because messages have to wait for the laser to turn on before transmission, when they find the laser off.

The naive approach to the laser power-gating in FBFLY would be turning off the photonic links whenever they are idle (**Naive**). In this case, all of the paths between router pairs can be turned off simultaneously, forcing messages to wait for the lasers to turn on before transmission. Furthermore, a packet which is routed over multiple hops can experience the laser turn-on delay multiple times, as all of the photonic links along the path could be turned off. This cumulative laser turn-on delay effect can have significant impact on the performance and should be avoided.

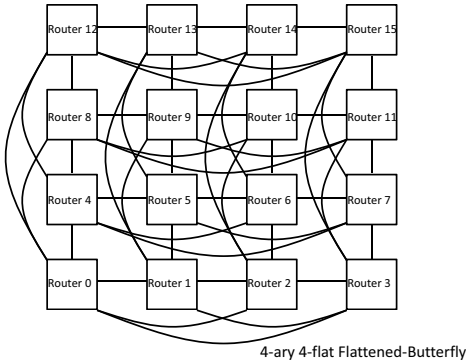
Flattened butterfly provides high path diversity which increases the chances of packets avoiding the laser turn-on latency. We present a 4-ary 3-flat FBFLY configuration in Figure 42a, where the "Router 0" can send a message to the "Router 15" using either the "Router 3" or the "Router 12". So, if the photonic link between the "Router 3" and the "Router 15" is turned off, the messages can still be directed through the

"Router 12" without waiting for that link to be turned on. Another important point to note is, by steering the traffic through the "Router 12" the opportunity to turn the laser off for the photonic link between the "Router 3" and the "Router 15" is maximized.

Removing the laser turn-on latency from the critical path of the messages reduces the performance penalty of the laser power-gating. A k-ary n-flat FBFLY network consist of n k-ary (n-1)-flat FBFLY networks connected together (Figure 42b). We can provide full connectivity even if we turn off the n-1 of these k-ary (n-1)-flat FBFLY networks (Stages), given that all of the other stages have active connections to the active stage. Turning off the Stages save laser energy, while turning on additional stages will increase the path diversity and provide better performance. We propose Stage Laser Control Scheme (**SLAC**) which turns off the stages when the utilization is low to save energy and activates additional stages when the utilization



(a)



(b)

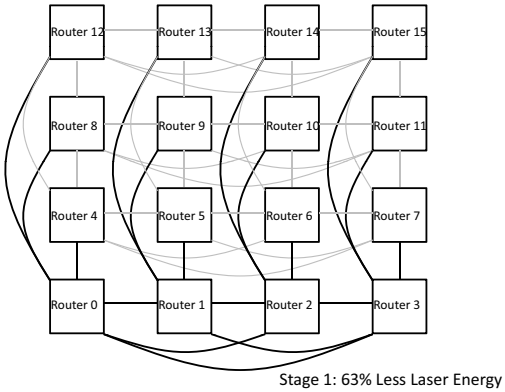
FIGURE 42: Flattened Butterfly Configurations

is high to provide better performance. SLAC observes the message traffic through inspecting the buffer utilization levels to decide on the number of stages to activate on the fly. If one of the input buffers of a routers in an active stage goes over a set threshold, that router broadcasts a message to all other routers to activate a new stage. Similarly if one of the input buffers that activated another stage is below a certain threshold, a stage turn-off message is broadcasted to all other routers, and the most recently turned on stage will be turned off (after emptying its buffers). SLAC always adoptively routes the traffic through the active stages (a stage that received a turn-off signal is not considered active, so it doesn't receive new messages) which balances the traffic, avoids message's waiting for laser turn-on time (completely hiding the turn-on latency) and avoids unnecessary stage activation which maximizes the laser energy savings.

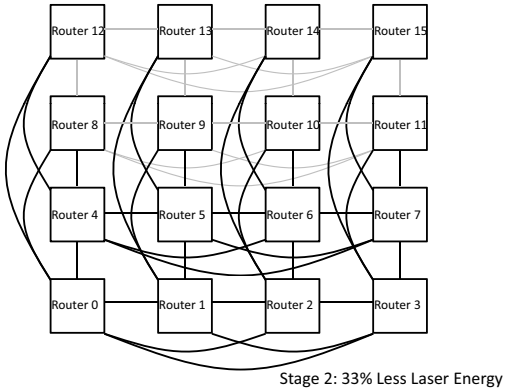
In Figure 43, we show how SLAC works on a 4-ary 3-flat FBFLY. This network consist of 4 stages (one row of routers), which SLAC activates adaptively. The set of black links in Figure 43a shows SLAC with 1 stage activated (Stage 1), and Figure 43b shows two stages activated (Stage 2). On Stage 1, the FBFLY network consumes 63% less laser energy compared to a conventional FBFLY network with photonic links always on (No-Ctrl). However, Stage 1 can cause high contention, because there is only one path between each router pair, which may reduce the performance. Stage 2 saves 33% of the total laser energy while providing multiple paths between the source and the destination pairs, which can be exploited to provide higher throughput (via dynamic routing algorithm).

SLAC monitors the traffic to switch between stages to achieve high energy efficiency when the traffic is low, and high throughput under heavy traffic. SLAC always keeps the Stage 1 on, and activates additional stages in ascending order when needed. SLAC monitors the input buffer utilization levels to estimate the network utilization, which is proposed as a lightweight and accurate method [88]. When a router in an active stage has an input buffer over a certain threshold, a stage activation message is broadcasted. After the routers in the de-active stages receive the turn-on signal, next available stage is activated. The newly

activated stage broadcasts a message after its activation, so all other routers update their routing table, so they can send through the newly activated stage. The newly activated stage can be turned off when the router which turned it on becomes underutilized. A stage turn-off message is broadcasted when the activating buffer, which was overutilized, goes below a certain threshold. This stage turn-off message deactivates the last activated stage. Right before the deactivation, the stage broadcasts a message to update all other router's routing table, so it stops receiving messages. Stage turns-off once all of the messages in the output buffers of the routers in that stage is empty. For example, when in Stage 1, if an input buffer in Router 0 goes over 75% utilization, it broadcasts a turn on message, so Router 4,5,6,7 turn on their links. Once the Stage 2 is active, it broadcasts a message so all others update the list of the stages they can send messages to. If the same input buffer in Router 0 goes below 25% utilization, it broadcasts a turn-off message, so the Stage 2 can be turned off. All other routers update their routing table after the turn-off signal broadcast. In



(a)



(b)

FIGURE 43: Laser-gating stages for the Flattened Butterfly Network

our evaluation we model all of the additional message broadcasts and latencies in both the turn-on and the turn-off sequences.

SLAC employs an adaptive routing algorithm which increases the opportunities to save laser energy when the traffic is low by using active stages only, and activates additional links only when it detects heavier traffic. The routing algorithm randomly selects the active stage to use, so it balances the traffic. SLAC's routing algorithm is deadlock free because it uses a dimension ordered routing scheme. When a packet is generated, our routing algorithm first checks if the destination stage is active, if so, it routes package to the active stage first and then routes the destination router within the active stage. If the destination router is not in an active stage, routing algorithm selects an active stage to use randomly and makes three hops using the active stage. In short, our routing algorithm routes packages in north-south direction first, and then routes them west-east later.With the dimension ordered routing, the turns from East to South and West to South is prohibited, therefore the routing algorithm avoids forming cycles and stays deadlock free.

4.3.1 .Experimental Methodology

4.3.1.1 Interconnect Performance and Energy Analysis

To evaluate the performance and the energy consumption of SLAC for FBFLY on-chip network in isolation from the interference of the other system components or the application characteristics, we employ a cycle-accurate network simulator based on Booksim 2.0 [15], which models a 4-ary 3-flat FBFLY network servicing random uniform traffic (with concentration of 4). The simulator models a three-cycle router, with 1-cycle E/O and O/E conversions. We assume a 480 mm^2 chip, where the link latency (1-3 cycles) is calculated based on the traversed waveguide length. The buffers are 20-flits deep, with a flit size of 300 bits. The maximum core frequency is 5 *GHz*, and the optical interconnect runs at 10 *GHz*. Latency is measured as the time required for the network to process a sample of injected packets.We evaluate the load-latency

TABLE 8. Architectural Parameters.

CMP Size	64 cores, 480mm^2
Processing Cores	ULTRASPARC III ISA, up to 5Ghz, OoO, 4-wide dispatch/retirement, 96-entry ROB
L1 Cache	Split I/D, 64KB 2-way, 2-cycle load-to-use, 2 ports, 64-byte blocks, 32 MSHRs, 16-entry victim cache
L2 Cache	Shared, 512 KB per core, 16 way, 64-byte blocks, 14 cycle-hit, 32 MSHRs, 16-entry victim cache
Memory Controllers	One per 4 cores, 1 channel per Memory Controller Round-robin page interleaving
Main Memory	Optically connected memory [2], 10ns access
Networks	radix-16 SWMR and Firefly

characteristics of SLAC and compare it against a flattened butterfly that always keeps the lasers on (No-Ctrl), a Naive control scheme and an electrical flattened butterfly network (Electric-FBFLY). In order to make a fair comparison we equate the average power consumption of Electric-FBFLY to the power consumption of No-Ctrl by adjusting their data-path width (flit size).

For the multi-chip (wafer) scale flattened butterfly implementation, we model an 8-ary 3-flat flattened butterfly network where the link latency (2-15 cycles) is calculated based on the length of the traversed waveguide length. The flit size is 50 bits.

The FBFLY network for the Datacenter is an 8-ary 3-flat FBFLY network with the concentration of 8, so it supports up to 512 nodes. The router delay is 200ns, and the link latency (100-200 ns) is calculated based on the traversed optical fiber length. The flit size is 300 bits.

4.3.1.2 Multicore System Performance and Energy Analysis

To evaluate the impact of SLAC on a realistic multicore system, we model a 64-core processor on a full-system cycle-accurate simulator based on Flexus 4.0 [27,75] integrated with Booksim 2.0 [15] and DRAMSim 2.0 [62]. Table 2 details the architectural modeling parameters for the on-chip analysis. We

assume a shared and physically distributed L2 cache and directories. The memory controllers are uniformly distributed on the chip, and they use the same physical interconnect with the VCs to avoid possible deadlock. All messages below L1 cache traverses the interconnect. The power consumption of the electrical interconnect is calculated using DSENT [69]. We target a 16 *nm* technology, and have updated our tool chain accordingly based on ITRS projections [23]. The simulated system executes a selection of benchmarks from SPLASH-2, PARSEC and other scientific workloads. For the multi-chip implementation analysis, we conduct a similar size simulation assuming each thread is located in a different site. To evaluate the performance of the SLAC on the datacenter scale FBFLY, we used snippets of the traces collected from routers in a datacenter(EDU1 and EDU2) [6]. EDU1 and EDU2 consist of packages passing through a single router in a datacenter, so we scaled the workload to reflect and all-to-all traffic on the FBFLY network. We injected a different copy of the package trace at each FBFLY router starting from a random location within the trace, and we measured the average message delivery latency to estimate the network performance.

TABLE 9. Nanophotonic Parameters and Laser Power.

On-Chip		**FBFLY**	**Multi-Chip**		**FBFLY**	**PtoP**
	per Unit	**Total**		**per Unit**	**Total**	**Total**
DWDM		64	DWDM		16	16
Splitter	0.2 *dB*	0.6 *dB*	WG Loss	0.3 *dB/cm*	4.5 *dB*	10.5 *dB*
WG Loss	0.3 *dB/cm*	0.75 *dB*	WG Loss*	*0.05 dB/cm*	*0.75 dB*	*1.75 dB*
Nonlinearity	1 *dB*	1 *dB*	Bridge WG Loss	*1 dB*	*1 dB*	*1 dB*
Modulator Ins.	0.5 *dB*	0.5 *dB*	Modulator Ins.	4 *dB*	4 *dB*	4 *dB*
Ring Through	0.01 *dB*	10.24 *dB*	Ring Through	0.05 *dB*	0.8 *dB*	0.8 *dB*
Filter Drop	1.2 *dB*	1.2 *dB*	Filter Drop	1 *dB*	1 *dB*	1 *dB*
Receiver Margin	4 *dB*	4 *dB*	Receiver Margin	4 *dB*	4 *dB*	4 *dB*
Coupler	2 *dB*	2 *dB*	Coupler	2 *dB*	6 *dB*	6 *dB*
Total Loss		**8.68 *dB***	**Total Loss**		**21.3 *dB***	**27.3 *dB***
Detector		**-20 *dBm***	**Detector**		**-20 *dBm***	**-20 *dBm***
Laser Power Per Wavelength		**0.073 *mW***	**Laser Power Per Wavelength**		**1.34896 *mW***	**4.7863 *mW***
On-Chip LaserPower	*10% Eff.*	**21.25 *W***	**Total LaserPower**	*30% Eff.*	**199.43*W***	**124.73*W***
Off-Chip LaserPower	*30% Eff.*	**11.11 *W***	**Total LaserPower***	*30% Eff.*	**84.1*W***	**43.96*W***

4.3.1.3 Laser Power Consumption Calculation

We calculate the laser power savings of SLAC and compare it against the equal power networks for both on-chip and multi-chip implementations. Table 1 shows the optical loss parameters for the modulators, demodulators, drop filters, and detectors introduced in [2] which are assumed for the on-chip implementation, and optical loss parameters introduced in [39] which are assumed for the multi-chip integration. The modulation and demodulation energy is 150 *fJ/bit* at 10 *GHz* [2] for both. The laser power per wavelength and the total laser power are calculated in Table 1 using the analytical models introduced in [33]. Note that, we assumed the on-chip laser efficiency of 10% and the off-chip laser efficiency of 30%. In [39] authors assume an aggressive waveguide loss parameter of 0.05 dB/cm, so we calculated the laser power consumption for both the traditional (0.3 dB/cm) and the aggressive waveguide loss assumptions (aggressive assumption is noted with a *). The laser turn-on latency for the on-chip laser is 1.5ns, and for the off-chip laser source (comb laser) is 1 us.

4.3.2 .Experimental Results

4.3.2.1 Impact of SLAC on Performance and Energy of Flattened Butterfly Network

SLAC increases the message latency due to non-minimal routing, but provides high throughput. Figure 44a presents the load-latency characteristics of SLAC compared against a FBFLY with no laser power-gating (No-Ctrl), with Naive Control and an electrical flattened butterfly (Electric-FBFLY). Under random uniform traffic, SLAC achieves 16.9 cycle zero-load message latency on average which is 2.8-cycles higher than No-Ctrl. On the other hand, the throughput provided by SLAC under higher injection rates is almost equal to No-Ctrl's, and 1.15x and 2.14x higher than the Naive's and the Electric-FBFLY's respectively. Naive control incurs additional 10.8 cycle zero-load message latency over No-Ctrl, because of the cumulative laser turn-on delay (messages have to wait for the laser turn-on almost at every hop most of the time).

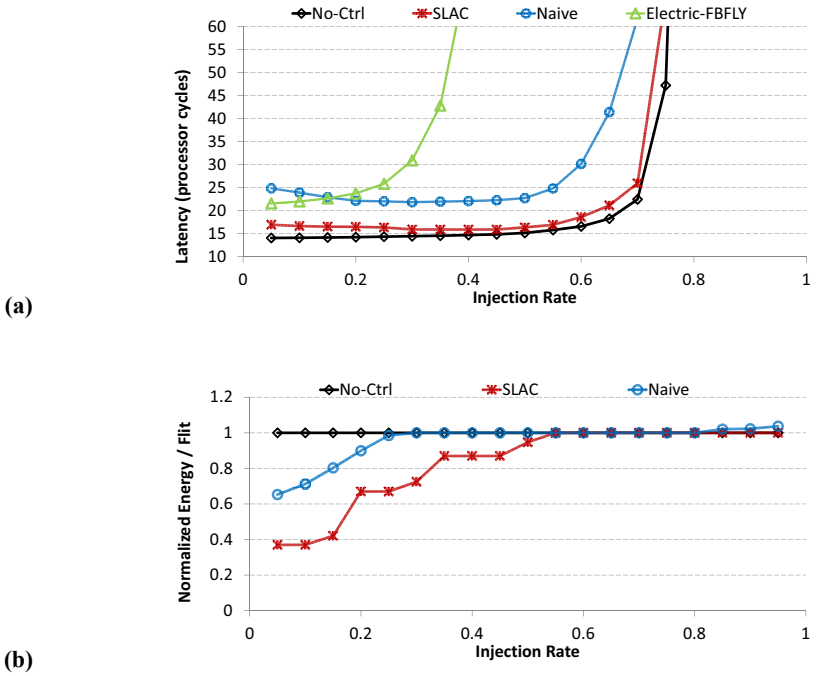


FIGURE 44: Load-Latency (a) and Laser Energy per Flit (b) for Flattened Butterfly topology with No-Ctrl, SLAC, Naive Control

Figure 44b presents the Laser Energy per Flit (EPF) characteristics of SLAC compared against the No-Ctrl and the Naive Control. SLAC trades off a small latency increase to high laser energy savings up to 63%. The steps observed in the EPF graph corresponds to the new stages activations. The Naive control achieves less energy savings, because it doesn't reuse activated links, wasting additional laser turn-on time and laser energy.

4.3.2.2 Impact of SLAC on Performance and Energy of a Multi-core Processor

SLAC achieves high laser energy savings but increases the average message latency slightly, because it uses non-minimal routing which prefers to use active links. In this section, we investigate the performance impact of SLAC on a multi-core processor with a 4-ary 3-flat FBFLY. Figure 45a shows the speedup of SLAC compared against the No-Ctrl, the Naive control and the Electric-FBFLY (power equivalent to No-Ctrl). The performance of SLAC is only 2% away from the No-Ctrl. SLAC outperforms the Naive control

and the Electric-FBFLY by 1.1x and 1.31x respectively, because it provides higher throughput under heavier traffic by turning on additional stages. Figure 45b presents the laser energy consumption per flit, where SLAC saves 43% laser energy on average (59% maximum). Naive control manages to save some laser energy while running the workloads with lighter traffic, however it slows down the execution significantly when the traffic demand is high, and ends up consuming higher laser energy. This shows us the importance of providing high performance (by maintaining full connectivity and additional stage activation) in achieving laser energy savings.

The energy savings of SLAC depends on the traffic rate, however its energy savings stay significant across all of the workloads. In Figure 46 we present the fraction of time spent in each stage for SLAC when running the appbt, fmm and bodytrack workloads. For the workloads with low message traffic (fmm), SLAC stays in Stage 1, maximizing energy savings. For the ones with higher traffic demand (bodytrack), SLAC tends to turn on higher stages to provide better performance. For the appbt workload, the fraction of time

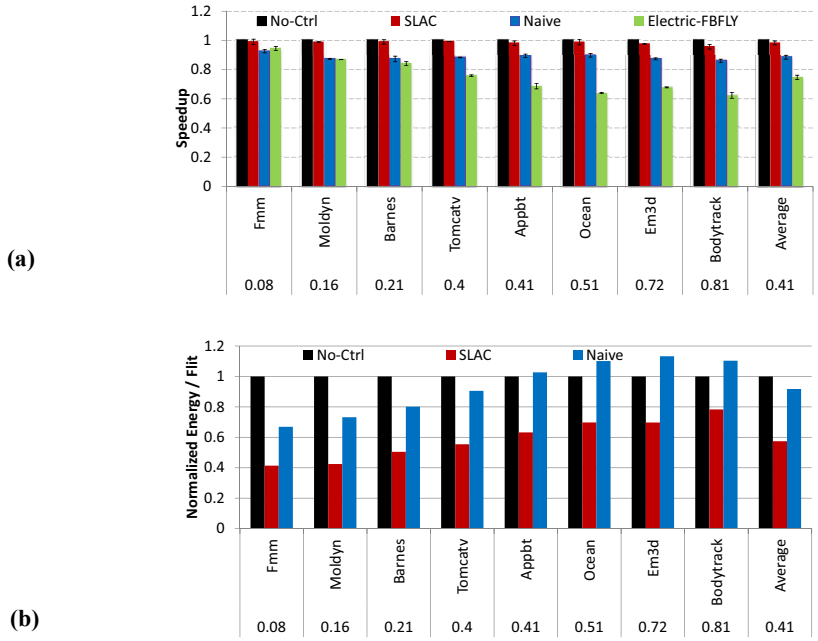


FIGURE 45: Speedup (a) and Laser Energy per Flit (b) for a multicore with No-Ctrl, SLAC, Naive Control and Electric-FBFLY

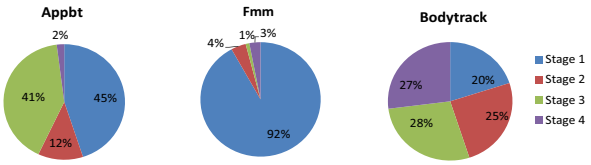


FIGURE 46: Fraction of time spent in each Stage

spent in the Stage 3 is higher than the fraction of time spent in Stage 2, which shows the bursty message traffic behavior of this workload.

4.3.2.3 Impact of SLAC on the Performance and the Energy Consumption of a Multi-chip System

Flattened butterfly networks are highly scalable and can connect up to thousands of nodes together, for that reason they are preferred for the multi-chip integration systems (a wafer scale implementation similar to Macrochip[39]). SLAC can be implemented on the wafer scale photonic FBFLY networks too, and its energy savings impact would be more significant due to higher laser power consumption of the wafer scale network. In this section, we present the performance and laser energy saving characteristics of SLAC implemented on a wafer scale 8-ary 2-flat FBFLY network. We compare our results against the FBFLY with no control (No-Ctrl), and a point to point (PtoP) network which is previously proposed in [39]. To make a fair comparison, we compare SLAC against equal power PtoP network, however there is little consensus on the waveguide loss parameter assumption which has a direct impact on the laser power consumption, so we consider both the aggressive waveguides [39] (with 0.05 dB/cm loss) and the traditional waveguides [8] (with 0.3 dB/cm loss). With the aggressive waveguides, a power equivalent PtoP network can support 25-bit links. With the traditional waveguides PtoP network can only support 4-bit wide links.

Figure 47a shows the speedup of SLAC for the wafer scale network. On average, SLAC is only 3% slower than the No-Ctrl, and 1.44x and 1.86x faster than the Naive control and the PtoP respectively. Even with

the aggressive waveguides, SLAC is 1.14x faster than the PtoP network proposed in [19]. Figure 47b shows the laser energy per flit comparison. SLAC saves 57% of the laser energy on average (66% maximum), whereas PtoP save between 4-17% on average, and Naive causes an increase in the energy consumption by 10%.

4.3.2.4 Impact of SLAC on the Performance and the Energy Consumption of a Datacenter Network

Flattened butterfly have been proposed as datacenter networks, because they provide low latency, high throughput communication, and they can scale out while keeping cost at bay. SLAC can be exploited to improve energy efficiency of a photonic datacenter network with flattened butterfly topology. A datacenter scale network is expected to employ optical fibers powered by external lasers. “Comb” lasers is a popular choice for an external laser, and they can be turned on and off within 1 us [28]. Different than the on-chip counterparts, following the laser turn-on a clock and data recovery is necessary in the datacenter networks. A clock data recovery (CDR) system can synchronize in 200 *ns* [28], which should be included in the laser

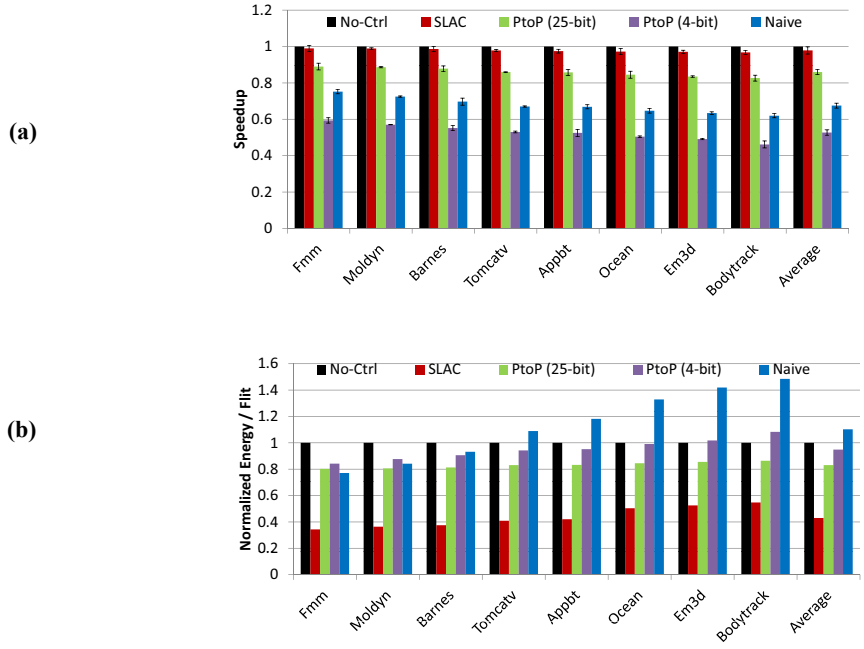
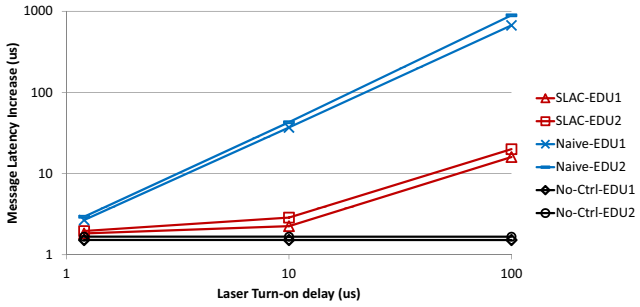


FIGURE 47: Speedup (a) and Laser Energy per Flit (b) for a Multi-chip with No-Ctrl, SLAC and Naive

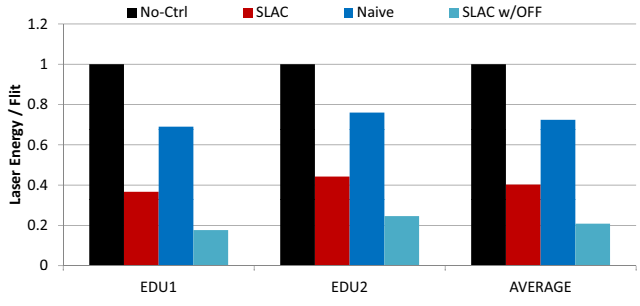
turn-on delay. SLAC removes the laser turn-on delay from the critical path, so it tolerates even higher laser turn-on delays.

Figure 48a presents the message latency increase SLAC and the Naive control causes as a function of the laser turn-on delay under EDU1 and EDU2 traces. A 1.2 us laser turn-on delay results in 0.29 - 0.35 us increase for SLAC and 1.14 - 1.27 us increase for the Naive control. As the turn-on delay increases, the message latency increases slowly for SLAC and much faster for Naive, because messages have to wait for the laser turn-on more frequently in Naive. The 10 us laser turn-on delay results in 0.75 - 1.2 us increase for SLAC and 35.4 - 41.3 us increase for the Naive control. With 100us laser turn-on delay the Naive control causes more than 1ms additional delay to the messages, while SLAC keeps it under 20 us.

The datacenter message traces EDU1 and EDU2 exhibit sparse and bursty message injection trends, therefore SLAC can turn-off most of its stages during the low traffic and achieve high laser energy savings. The



(a)



(b)

FIGURE 48: Message Latency(a) and Laser Energy per Flit (b) for a Datacenter Networks with No-Ctrl, SLAC, Naive Control

Figure 48 shows the laser energy per flit for SLAC and the Naive control when running database workloads. SLAC saves 60% of the laser energy while Naive only saves 28%.

Figure 49a shows the fraction of time spent in each stage during the execution, and due to sparse arrival of the messages, most of the time is spent in Stage 1. SLAC aims to remove the laser turn-on latency from the critical path, so it keeps the Stage 1 always turned on, which means laser energy could still be wasted when there are no messages in the network. In order to harvest this wasted laser energy while hiding the laser turn-on delay, we proposed an optimization for SLAC (SLAC w/OFF) that turns off all of the Stages, and predicts an upcoming message ahead of time with the help of the OS. The OS can take advantage of the packet preparation latency of the TCP/IP network to turn on the lasers ahead of time to hide the laser turn-on latency completely. Previous research [43] showed that it takes 950 ns for a process to send a message to the socket interface, and 260 ns later IP layer is called, and IP layer takes 450 ns to prepare the package, and the network driver constructs the output package in 430 ns. This means the SLAC w/OFF laser control will have 2.1 us to turn the lasers on, which can completely hide the 1.2 us laser turn on latency, so SLAC w/OFF can turn off the whole network without incurring any additional message delay. Our results show that SLAC w/OFF turns off all of the stages completely during the 54 to 62% of the whole execution (Figure 49b), and saves 79% of the laser energy compared the No-Ctrl (Figure 48).

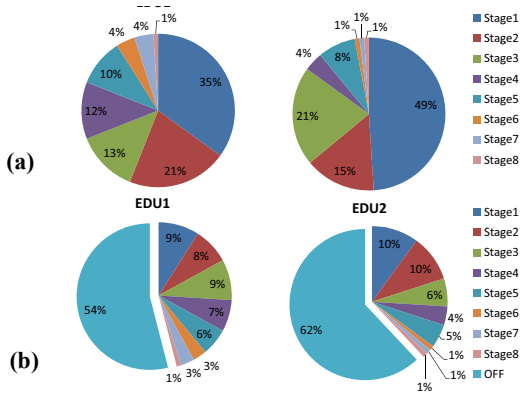


FIGURE 49: Fraction of time spent in each Stage

Chapter 5

System Level Thermal Tuning Considerations

5.1 Motivation

Silicon photonics provide high-bandwidth, low-latency, and energy-efficient communication in many-core processors. However, the high optical loss of photonic components, together with the low efficiency of WDM-compatible lasers, and thermal susceptibility of the microring resonators increase the laser power and ring-heater power consumption significantly. While previously proposed power-gating techniques [17,19,18] reduce the laser power consumption, the high ring-heating power consumption remains a problem that needs to be addressed.

Silicon-photonic devices can be manufactured alongside CMOS logic even on the same die [11], designers typically assume a simplified process where the photonic components are housed within a photonic die, which is 3D-stacked to a logic die that contains cores, caches, and other electronic components. Due to this arrangement, the thermal variations of the logic die directly couple to the photonic devices. These thermal variations may occur rapidly depending on the workload [25], are both spatial and temporal in nature, and can exceed $30^{o}C$ difference. As current silicon-photonic designs are predominantly based on microring resonators which are highly temperature-sensitive devices, these thermal fluctuations in turn throw the microring resonators off-resonance and prevent the optical interconnect from functioning. To keep the microrings resonating at their appropriate wavelengths designers employ trimming, a technique that dynamically shifts the microring's resonant wavelength towards the red through heating, or shifts it

towards the blue through current injection. Trimming by current injection causes instability and thermal runaways [51], thus microrings are typically maintained at a constant temperature using the heaters only. Because only the heaters are used, the microrings are tuned to temperatures above the maximum temperature that the microprocessor reaches.

Unfortunately this means that the heaters need to work continuously to keep the microrings at such high temperature, and at the same time the majority of the heating power is wasted as it dissipates through the package to the heat sink. As a result, it is common for microring heaters to consume upwards of 40W [51], mostly of which is wasted. To make matters worse, this thermal energy heats up the logic layer to temperatures very close to its operational limit, which forces the system to throttle the cores, thereby reducing performance. The runaway heat also increases the frequency and magnitude of thermal emergencies, and accelerates the aging of the logic die.

The solution we propose is rather simple: thermally decouple the 3D-stacked logic die from the photonics die by introducing an insulating layer between them to maintain higher thermal stability and easier trimming. More specifically, our contributions are:

- We propose *Parka*, a nanophotonic NoC that encases the photonic die in a thermal insulator that keeps its temperature stable with low energy expenditure, while minimizing the spatial and temporal thermal coupling between logic and silicon-photonic components.
- We quantify the ring heating power consumption for a large-scale multicore under a variety of insulation methods and cooling solutions.
- We evaluate the performance impact of thermal decoupling on a multicore running a range of scientific workloads, under realistic physical constraints.

Our results indicate that Parka reduces the ring heating power by 3.8x on average across our workload suite. Moreover, the energy savings allow for providing a higher power budget to the cores, which enables them to run faster. Parka on a radix-16 crossbar allows the multicore to achieve 11-23% speedup (34% max) over a baseline scheme with no insulation, depending on the cooling solution used. Lastly we show that, Parka combined with microfluidic cooling solution can reduce the temperature fluctuations significantly, so that it can remove the constant need for using ring-heaters to regulate the photonic die temperature.

5.2 Photonic Die Insulation with Parka

The basic building block of silicon-photonic interconnects is the microring resonators, which are designed to resonate at a specific wavelength to realize add/drop filters and modulators. The microring resonators are very susceptible to temperature changes, because the refractive index of Si changes with temperature, in turn changing the resonance wavelength. Trimming keeps the microrings resonating at their appropriate wavelengths by dynamically shifting the microring's resonant wavelength towards the red through heating, or towards the blue through current injection. Microrings are typically kept at a constant temperature using the heaters only, as current injection causes instability and thermal runaways [51]. The strong thermal coupling of the logic and photonic dies means that trimming by heating requires that the photonic die is heated to a temperature above the maximum temperature of the logic die.

This ring-heating power is mainly wasted, as it dissipates through the processor stack into the logic layer and eventually through the heat-sink, which is designed to remove heat from the processor stack. This heats the logic layer close to the limits of safe operating temperatures. A thermal emergency occurs when the logic die temperature exceeds the safe operation limits, where the cores are throttled or turned off to

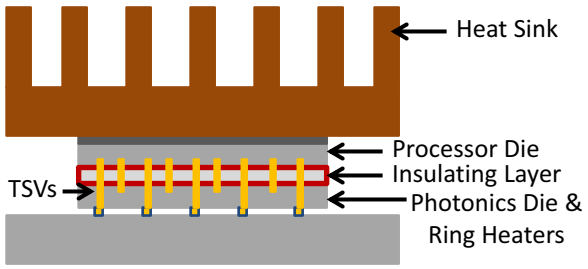


FIGURE 50: Proposed Parka Architecture

lower the temperature. Therefore, high ring heating power consumption makes the multicore processor more susceptible to thermal emergencies, and may decrease its performance significantly.

Parka reduces the wasted energy and the heating of the logic layer by thermally decoupling the 3D-stacked logic die from the photonics die through an insulating layer between them, as shown in Figure 50. The insulation layer increases the thermal resistivity of the heat path from the photonics layer to the heat-sink, and (a) allows for easier trimming by trapping the heat within the photonics layer, (b) reduces the temperature variation in the photonics layer, and (c) minimizes the heating of the logic die induced by the microring heaters. The processor die is placed close to the heat-sink to allow better cooling, while an oxidized macro porous Si layer [50] realizes the thermal insulation, because the porous Si has 100x lower thermal conductivity compared to Si [50]. The porous Si layer is 150 *um* thick, as we find that a thinner layer does not provide adequate thermal insulation. The power delivery and communication between the dies is maintained through high aspect ratio TSVs [25, 66, 78].

Adding the insulation layer is expected to increase the manufacturing cost only marginally. The porous Si insulation layer can be readily integrated into the CMOS process by passing a plain silicon die through a simple electrochemical process that oxidizes it [50]. This silicon die is not subject to the regular yield-induced costs of dies that implement complex logic and require multiple mask exposures and several metal layers, and thus it is significantly cheaper. The addition of the porous Si layer also does not affect the number of TSVs and the number of pins in the package, which together with the logic and photonic dies consti-

tute the dominant cost factors [20,79]. The thickness of the insulation layer impacts the TSVs' height, but the cost is highly insensitive to it [20,79]. The additional layer will incur 3D-bonding costs, but these will increase the total cost by less than 1.5% [20,79].

Insulation can be achieved also by a 5 *um*-thick air or vacuum cavity etched between layers, a technique for which prototypes have been successfully manufactured and characterized [78]. Air has a thermal resistivity of 40 *m-K/W*, which is 40 times higher than porous Si, so it would be an even better insulator. However, this technique is more challenging to employ than oxidized porous Si. Thus, we maintain our conservative assumptions using porous Si insulators and do not consider alternative insulation techniques further. It is important to note that Parka does not depend on the exact insulator technology used. As processes mature and better materials and techniques become available, they can be employed by Parka to achieve even higher power savings than the ones we show in this paper.

5.3 Experimental Methodology

5.3.1 Ring Heater Power Consumption Analysis

We model a photonic die with microrings tuned to $90\ ^{o}C$ ($363.15\ ^{o}K$), which is the maximum temperature that the logic die can reach. To calculate the total ring heating power we extend the method by Nitta *et al.* [51] by estimating the ring-heater power consumption while accounting for the heating of the photonic die by the operation of the cores. While one can assume that the heaters are employed to shift the resonant wavelengths of the microrings only momentarily according to the local temperature, keeping a stable temperature for the die as a whole is a more realistic approach [51].

We model a multicore where 50 *um*-thick logic and photonic dies are 3D-stacked, and separated by a 150 *um* porous Si insulation layer, as shown in Figure 50. The thermal resistivity is 0.01 *m-K/W* for Si, and

TABLE 10. ARCHITECTURAL PARAMETERS.

CMP Size	64 cores, 480mm^2
Processing Cores	ULTRASPARC III ISA, up to 5Ghz, OoO, 4-wide dispatch/retirement, 96-entry ROB
L1 Cache	Split I/D, 64KB 2-way, 2-cycle load-to-use, 2 ports, 64-byte blocks, 32 MSHRs, 16-entry victim cache
L2 Cache	Shared, 512 KB per core, 16 way, 64-byte blocks, 14 cycle-hit, 32 MSHRs, 16-entry victim cache
Memory Controllers	One per 4 cores, 1 channel per Memory Controller Round-robin page interleaving
Main Memory	Optically connected memory [2], 10ns access
Networks	SWMR crossbar, radix-16

1 *m-K/W* for the porous Si insulator [50]. We evaluate the ring-heater power consumption of Parka using the 3D extension of HotSpot [67], a thermal modeling tool based on an equivalent circuit of thermal resistances and capacitances. We evaluate Parka's impact on the heat transfer rate between dies via a transient thermal analysis at 300 *us* time steps. The ambient temperature is fixed at $45\ ^oC$ (318.15 ^oK).

Our model accounts for the thermal impact of TSVs, as they are highly conductive, and also for the individual ring trimming power required to overcome process variations, as described in [33]. We model a design that employs a total of 76,800 microrings, which are driven by one TSV each. We model high-aspect ratio TSVs with 10 *um* diameter [66]. All the TSVs together cover a 6 mm^2 area, which corresponds to 1.25% of the chip area and contributes only 0.5% to the total cost [20,79]. It is important to note that this is not an overhead that Parka imposes to the system; rather, it is the overhead of 3D-stacking the photonic and the logic dies, and it is incurred by both Parka and the baseline system.

5.3.2 Multi-core System Performance and Energy Analysis

To evaluate the impact of Parka on a realistic multicore system, we model a multicore processor on a full-system cycle-accurate simulator based on Flexus 4.0 [27, 75] integrated with Booksim 2.0 [15] and DRAMSim 2.0 [62]. Figure 4 describes our simulation tool chain. We target a 16 *nm* technology, and have

updated our tool chain accordingly based on ITRS projections [23]. We collect runtime statistics from full-system simulations, and use them to calculate the power consumption of the system using McPAT [46], and the power consumption of the optical networks using the analytical power model by Joshi *et al.* [33]. The analytical model we use for the power calculation of the photonic components results in similar overall power estimates as DSENT [69], but it also provides an easy breakdown of the power consumed by each one of the nanophotonic components in our network. We estimate the temperature of the chip using the 3D extension of HotSpot 5.0 [67]. The estimated temperature is then used to refine the leakage power estimate. We adjust the voltage and frequency of the logic die based on the stable-state power and temperature estimates (Figure 4), and we repeat the process until the system reaches a stable state and additional iterations result in no further changes on temperature and overall power consumption.

Using the methodology above, we simulate a 64-core multicore system. By scaling existing core designs down to 16 *nm* we estimate that 64 cores would require a 480 mm^2 die. Table 2 details the architectural modeling parameters. We model realistic multicore systems that employ dynamic thermal management by throttling the voltage and the frequency of the chip to keep it within safe operational temperatures

TABLE 11. WORKLOAD DETAILS.

Suite	Workload	Description
NAS	**appbt**	Independent equations system solver *32x32x32 grid, 1e-12 tolerance, 8e-4 time step, 1.2 SSOR iteration relaxation factor*
SPEC-CPU	**tomcatv**	Vectorized mesh generation; parallel version of 101.tomcatv from SPEC-FP *4,096 array size, 10 iterations*
SPLASH-2	**barnes**	Barnes-Hut hierarchical N-body simulation *64K particles., 2.0 subdiv. tol., 10.0 fleaves, 2.0 fcells, 0.025 time step, 0.05 softening*
	fmm	Particle simulation via adaptive fast multipole *131K particles, two clusters, plummer distr., 1e-6 precision, 30 steps, 0.025 step duration*
	ocean	Eddy & boundary oceanic currents simulator *1026 x 1026 grid, 20,000 meters, 9,600 sec, 1e-7 tolerance*
PARSEC	**bodytrack**	Annealed particle filter to track human body *4 cameras, 4 frames, 4,000 particles, 5 annealing layers (simlarge)*
Other Scientific	**moldyn**	Molecular dynamics simulation *19,652 molecules, max interactions 3,200,000*
	em3d	Electromagnetic force simulation *768K nodes, degree 2, span 5, 15% remote*

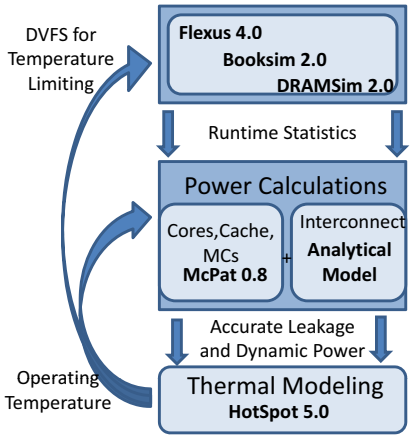


FIGURE 51: Simulation flow chart.

(below 90 ^{o}C, i.e., 363.15 ^{o}K). The simulated multicore executes a selection of SPLASH-2 and PARSEC benchmarks, and other scientific workloads. The workload parameters are detailed in Table 11.

5.3.3 Interconnect and Nanophotonic Parameters

We employ a cycle-accurate network simulator based on Booksim 2.0 [15], which models a radix-16 SWMR crossbar. The simulator models a single-cycle router, with 1-cycle E/O and O/E conversions. We assume a 480 mm^2 chip, which employs a 10 *cm* waveguide with a round trip time of 5 cycles. The link latency (1-5 cycles) is calculated based on the traversed waveguide length. The buffers are 20-flits deep, with a flit size of 300 bits. The maximum core frequency is 5 *GHz*, and the optical interconnect runs at

TABLE 12. NANOPHOTONIC PARAMETERS.

	per Unit	Radix-16 Total
DWDM		16
WG Loss	0.3 *dB/cm[8]*	3 *dB*
Nonlinearity	1 *dB*	1 *dB*
Modulator Ins.	0.5 *dB*	0.5 *dB*
Ring Through	0.01 *dB*	2.56 *dB*
Filter Drop	1.2 *dB*	1.2 *dB*
Photodetector	0.1 *dB*	0.1 *dB*
Total Loss		8.36 *dB*
Detector		-20 *dBm*
Mod./Demod. Energy (10 GHz)		150 *fJ/bit*

10 *GHz*. We derive the nanophotonic parameters from [2] and detail them in Table 12. The data bus is 300-bits wide (300 wavelengths with 16-way DWDM) powered by an off-chip laser source.

Unfortunately, there is little consensus on the optical loss parameters used or projected in literature, as parameters exhibit a variance over 10x across publications. However, the design of an optical interconnect highly depends on the losses of the optical components used. If the off-ring through loss on the radix-16 crossbar was 10x higher (i.e., 0.1dB), the interconnect wouldn't employ 64-way DWDM, as this would increase the laser power to unsustainable levels. Rather, it would be optimized with a lower DWDM (using more waveguides), keeping the total optical loss (and hence laser power) the same. In our work we limit the network to 16 DWDM because the number of turned-off rings on a single optical path of a crossbar is high, so limiting the DWDM helps keep the total optical loss at reasonable levels. 16-way DWDM has already been demonstrated and it is a widely-accepted parameter.

5.3.4 Modeling Cooling Solutions

The ring-heating power requirement depends highly on the cooling solution. Aggressive cooling solutions are capable of faster heat removal from the processor stack, which is likely to force the ring heaters to work even harder to keep the photonic layer at the tuned temperature. Therefore, the thermal decoupling that Parka advocates will be more important when better cooling solutions are employed. To evaluate the impact of Parka across cooling solutions we model both forced-air cooling (convective thermal resistance R_{conv} = 0.25 *K/W*) and a liquid cooling solution (R_{conv} = 0.15 *K/W* [65], 10 *ml/min* per cavity flow rate).

For the liquid cooling solution we assume that microchannels facilitate forced convective interlayer cooling with single-phase fluids, in particular water. While other single-phase fluids with higher thermal capacitance exist, they are toxic and thus impractical to deploy. We model high-aspect ratio TSVs with 10 *um* diameter [66], located and etched within 100 *um*-wide microchannel walls as in [63]. We assume

uniformly distributed microchannels, and equivalent fluid flow rate through each channel in the same layer. Although variation of the fluid flow due to nonuniform heat flux can exist, variations stay below 2% for single-phase flows and have negligible impact on the cooling system's performance [63]. The fluid pump and valve consume 1.3 *W* per 10 *ml/min* flow, and the power is linear to the volumetric fluid flow [63].

5.4 Experimental Results

5.4.1 Impact on the Ring-Heating Power Consumption

Parka thermally decouples the photonics die from the processor die using a porous Si insulating layer which reduces the thermal fluctuations caused by the processor layer, and traps the heat in the photonics die allowing for easier trimming. In this section we evaluate the ring-heating power consumption of Parka on a 64-core processor, and compare it against an architecture with no insulation.

First we evaluate the thermal shielding effect of the insulating layer by observing the temperature variation in the photonics die resulting from temperature fluctuations in the processor die. We increase the power consumption in the processor layer (from its idle level) to its maximum allowed level, and observe the temperature change in the photonics layer (Figure 17). The processor die stays at 66 ^{o}C (339.15 ^{o}K) when in the idle state, and its temperature reaches 90 ^{o}C (363.15 ^{o}K) rapidly when it is turned on (~18 *ms*). The temperature of the photonics die closely tracks the temperature change of the processor layer when there is no insulation. However, for Parka, it takes twice (Figure 17) as long for photonics layer to reach 90^{o}C, because of the thermal shielding effect of the insulating layer. Note that the insulating layer not only shields the fluctuations towards the higher temperature levels, but it also shields from the dips in the tem-

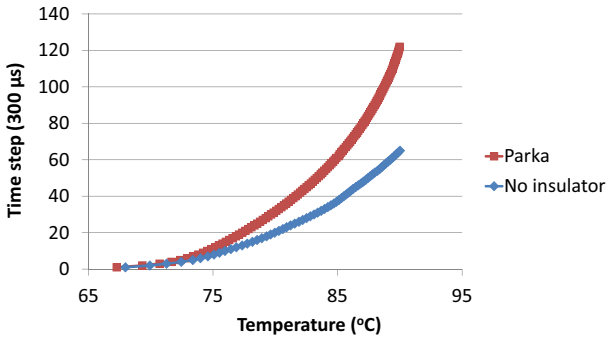


FIGURE 52: Transient analysis of temperature fluctuations in the photonics die.

perature. Parka allows for easier trimming because it shields the photonics layer from the short temperature fluctuations in the processor layer.

Thermally decoupling the photonics layer from the rest of the processor stack allows for trimming with less ring heater power consumption, because it doesn't allow the heat (generated by the ring-heaters) dissipate through the heat sink easily. The insulating layer increases the thermal resistance on the heat path to the heat sink, so it traps the heat within the photonics die. Therefore, Parka's ring-heaters can bring the whole photonics die to a stable temperature level which is higher than the maximum execution temperature at the processor layer with less power. Figure 26 shows a scenario where we present the both shielding and heat trapping effect of Parka. Figure 26.a shows a snapshot (at time t_0) of the thermal map of the processor die when running a real workload (appbt). We assume that, at time t_0 all of the processors stop, and only dissipate the leakage power until time t_1. We estimate that the processor die leakage power is ~30 *W* when idle. Figure 26.b shows the temperature maps of the photonics layer at time t_1. We observe that photonics layer stayed at a higher temperature for Parka compared to no insulation (or retained the heat better because of the insulating layer). Note that we assume that the ring-heaters are also off until time t_1. At time t_1, the ring-heaters are turned on to bring the photonics layer to a stable 90^oC, and Figure 26.c shows the power

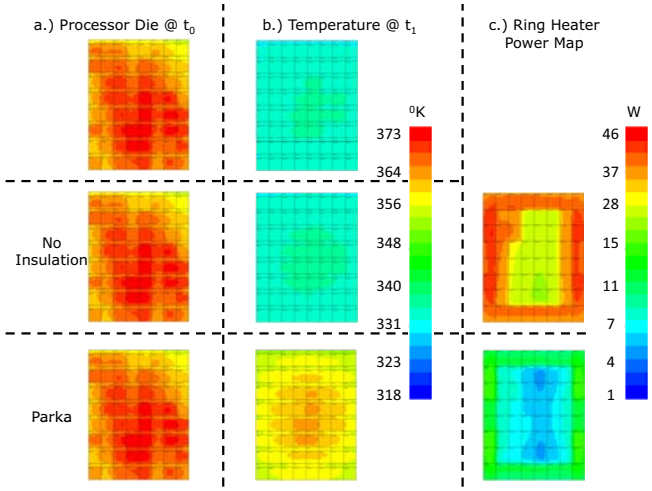


FIGURE 53: Case study: Impact of thermal insulation on the photonics layer temperature and the ring-heating power consumption

distribution of these ring-heaters. We observe that the photonics layer stays at a higher temperature for Parka compared to no insulation, as it retains the heat due to the insulating layer.

In the example in this figure we assume that the ring heaters are also off until time t_1. At time t_1, the ring heaters are turned on to bring the photonics layer to a stable 90 ^oC (363.15 ^oK), and Figure 26.c shows the power distribution of these ring heaters. We observe that Parka requires less ring-heating power. There are two reasons for this: first, the photonics layer is at a higher temperature at time t_1, so there is a smaller temperature difference (to 90 ^oC) to cover. Second, it is easier to close this temperature difference with Parka because the heat generated by the ring heaters stays within the photonics die.

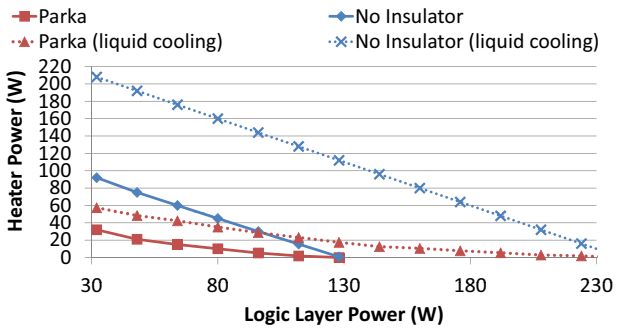


FIGURE 54: Ring -Heating Power vs. Processor Die Power

The amount of ring-heating power required to keep the photonics layer at a stable $90^{o}C$ highly depends on the power consumption of the processor die. When processor die is idle, ring heaters have to work harder to warm up the photonics die. In Figure 54, we show the ring-heating power requirement for different processor die power consumption levels.We observe that for every processor die utilization level Parka consumes less ring-heating power than no insulation case. The maximum amount of ring-heating power required for Parka is 3x lower then the maximum ring-heating power required with No-insulation (Processor die leakage power is ~30W when idle).

We observe that with liquid cooling the operational temperature at the processor layer stays under 90 ^{o}C when the processor die consumes up to 250 *W* (Figure 55.a), while forced-air cooling can sustain at best only up to 130 *W* and passive cooling less than 100 *W*. More importantly, we observe that the magnitude of the thermal fluctuations on the processor layer is higher under an aggressive cooling solution, because higher utilization levels are permitted within the power budget, and the idle temperature is lower due to

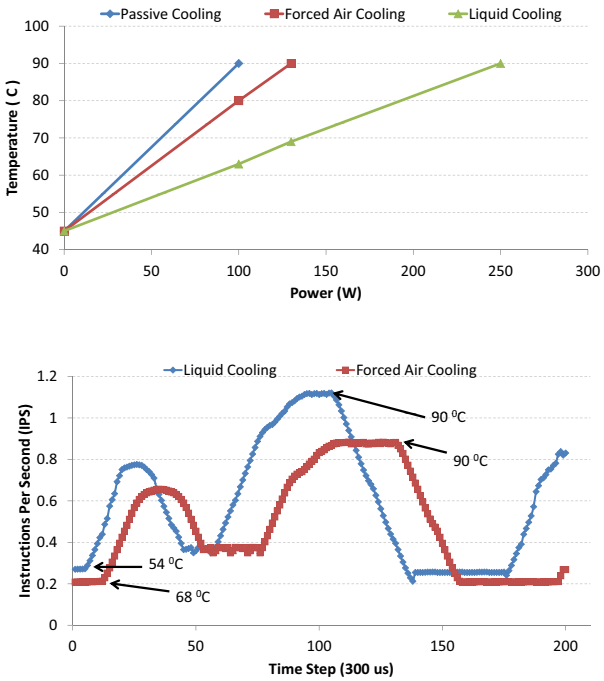


FIGURE 55: a.)Processor die temperature vs. processor power consumption, b.) Temperature trace (running appbt) of a multicore.

better cooling. Figure 55.b shows the instructions per second attained during the execution of a given code fragment of an application (appbt) when liquid or forced-air cooling are employed. We observe that liquid cooling allows higher performance, but also that the temperature under liquid cooling fluctuates between 54–90 ^{o}C, while for the same exact execution segment run under forced-air cooling the temperature fluctuates between 68–90 ^{o}C. Thus, the temperature fluctuation range on the simulated multicore is 14 ^{o}C wider with liquid cooling compared to forced-air cooling when running the same code fragment (Figure 55.b). The maximum temporal temperature fluctuation at a given point of the processor is 23 ^{o}C with forced-air cooling and 40 ^{o}C with liquid cooling solution.

As processor temperatures fluctuate during execution, the ring-heaters have to step in to keep the photonics layer at a stable temperature. We analyze this effect by running a collection of diverse workloads on our simulated multicore system and calculating the average ring-heating power consumed by each application (Figure 56). We observe that the temperature fluctuations are higher when running memory-intensive workloads (e.g., bodytrack, em3d, ocean, appbt), hence the ring-heating power consumption is also higher. On average ring heaters consume 16.9 *W* (22.4 *W* maximum) when there is no insulation. Parka allows for easier trimming by shielding from short fluctuations and trapping the heat, so it consumes on average 3.8x less ring heating power (4.4 *W* on average).

The liquid cooling solution keeps the processor cooler and allows for cores to run faster, however this results in higher temperature fluctuations at the photonics layer. On top of that, with better heat dissipation

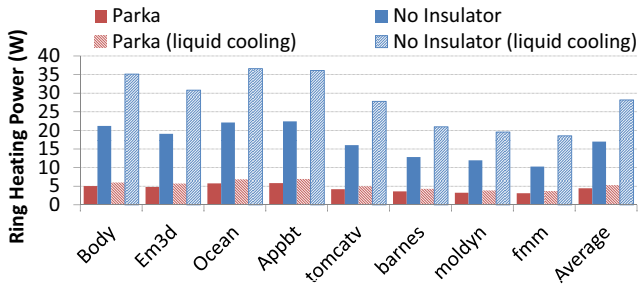


FIGURE 56: Average ring-heating power consumption while running real world applications

from the photonics layer, ring-heaters have to consume more power to keep the photonics layer at a stable temperature. Figure 56 shows that the ring heaters have to consume 28.2 *W* on average when there is no insulation and a liquid cooling solution is employed. However, employing an insulating layer in this case reduces the ring-heating power consumption by 5.4x on average (5.2 *W*),. Thus, Parka is essential when using aggressive cooling solutions.

5.4.2 Impact on Processor Temperature

Ring heaters warm up and keep the photonics die at a slightly higher temperature than the maximum operating temperature of the processor [51]. However, while heating the photonics die, the ring heaters also heat the processor die when there is no insulation. Heating the processor die forces it to operate close to its maximum operating temperature, even when it is idle. In this case, even a small increase in the utilization can cause a temperature spike which pushes the processor out of the safe operating limits causing it to throttle, and reducing performance. Thus, in the absence of an insulating layer the processor becomes highly vulnerable to thermal emergencies.

On the other hand, ring heaters consume 3.8x less power on average with Parka, and thus the processor layer remains cooler, because the overall power consumption in the processor stack is lower (leakage power is exponentially related to temperature). For example, Figure 57 shows that when compute compo-

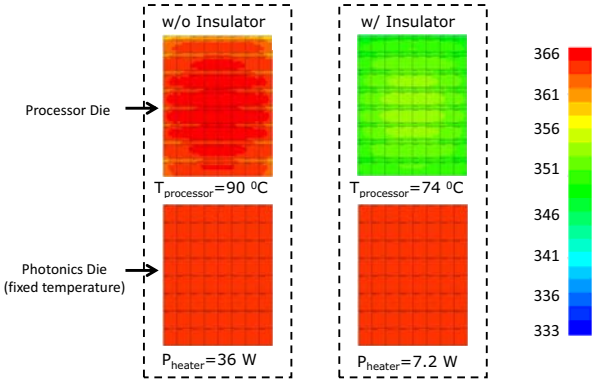


FIGURE 57: Parka's impact on the processor die temperature

nents consume 90 *W* at the logic layer, the ring heaters consume 36 *W* when there is no insulation, but only 7.2 *W* with Parka. As a result, the logic layer stays at 74 ^{o}C (347.15 ^{o}K) with Parka, while it reaches ~90 ^{o}C without insulation.

The ring heaters keep the processor die very close to the limit of safe operating temperature, so any increase in the processor utilization can push the processor into thermal emergencies. We present such an example in the execution window shown in Figure 58.a. The activity increase around time steps 12 and 62 push the processor temperature over 90 ^{o}C when there is no insulation, whereas with Parka the processor stays cooler and avoids the thermal emergencies. When running real applications, the processor runs into thermal emergencies up to 19% of the execution time (2% on average) when there is no insulation (Figure 58.b) The cores need to be throttled or completely turned off during a thermal emergency to allow for the processor to cool down and avoid permanent damage, so we expect that these thermal emergencies

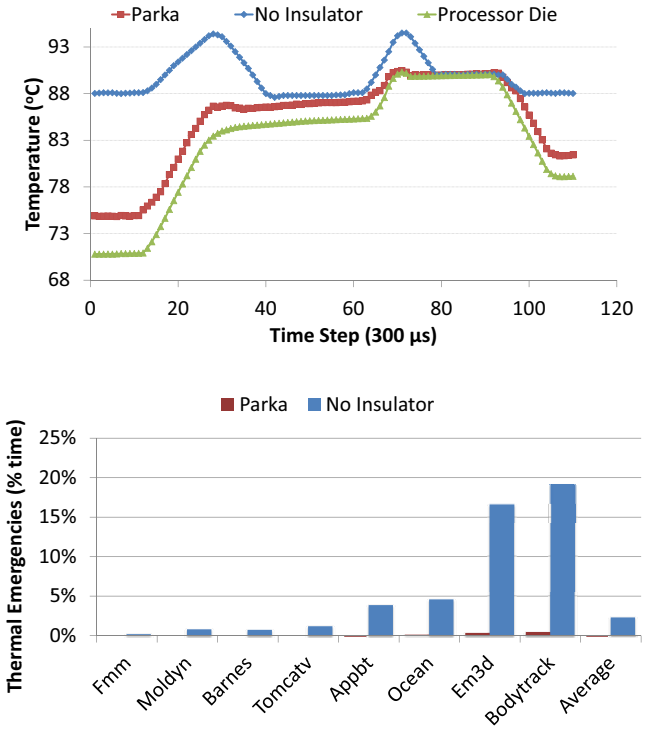


FIGURE 58: Temperature trace (appbt) presenting thermal emergencies in a multicore, and the percentage of execution time spent under thermal emergencies.

will significantly reduce the processor's performance. In contrast, Parka's processor die largely avoids thermal emergencies, and only experiences them for less then 1% of the execution time (Figure 58.b).

5.4.3 Impact on A Realistic Multicore

Under realistic thermal (power) constraints, DTM (Dynamic Thermal Management system) in the processor throttles the cores to keep the chip within a safe temperature.The insulating layer, however, reduces the ring-heating power and results in a cooler chip, causes less core throttling, and provides higher performance. Overall, Parka reduces the ring-heating power consumption by 3.8x, which allows for it's cores to run faster. As a result the processor with the insulating layer runs 11% faster on average (18% maximum) than the processor without the insulating layer.

Ring-heating power consumption is more significant when a more aggressive cooling solution is employed, so the power savings of Parka is greater. With a liquid cooling solution, Parka outperforms the processor without the insulation by 23% on average (34% maximum).

5.5 Photonic Die Insulation with Microfluidic Cooling

Interlayer liquid cooling using microchannels etched on the back of the substrates of individual layers is a viable and scalable cooling solution for 3D designs. Water based microfluidic liquid cooling solu-

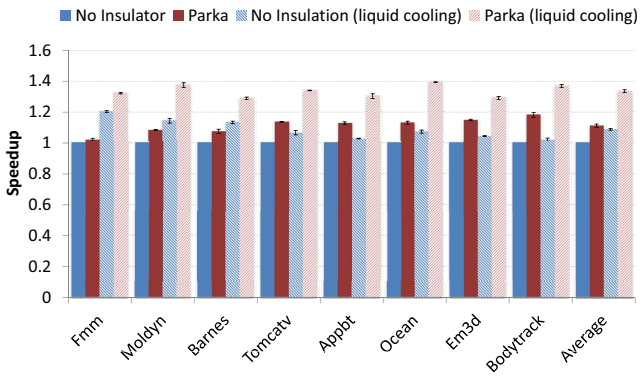


FIGURE 59: Realistic Multicore Performance with Parka.

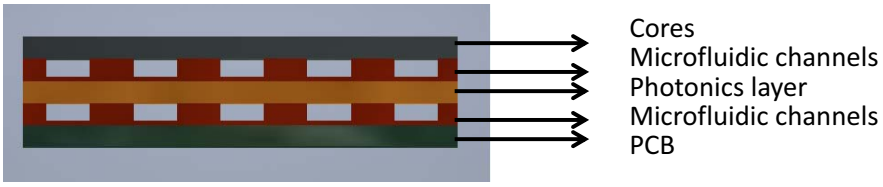


FIGURE 60: Liquid cooling solutions.

tions have demonstrated superior cooling capabilities, so they allow for cores to run faster (dissipate more power) while reducing the local thermal emergencies efficiently. This stabilizing effect of the microfluidic cooling solutions make them a good candidate to reduce the ring heating power consumption. On the other hand, microfluidics create a heat gradient across the chip (the liquid heats up as it flows throughout the chip) and the heat generated by the ring heaters can dissipate much easily, because of the better cooling capacity of the microfluidic cooling.

In Figure 60, we present a processor with a 3D-stacked photonic layer where microfluidic channels are etched under both the processor layer and the photonics layer. This design provides superior cooling, so the processor can dissipate up to 400W while staying within the safe operating temperature (90 ^{0}C). However, even with this design, the maximum temperature fluctuation at a given point while running real-world applications can be as high as 36.5 ^{0}C. In order to improve the temperature stability we propose to place an insulation layer between the photonics layer and the processor layer (Figure 61) similar to PARKA.

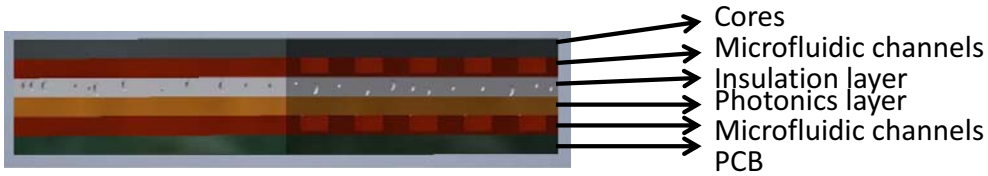


FIGURE 61: Liquid cooling solution with PARKA.

5.5.1 Evaluation Methodology

To evaluate the thermal stability of the photonics layer with the microfluidic cooling solution, we use the "3D-Ice" [001], which is a thermal modeling tool based on equivalent circuit of thermal resistances and capacitances. We model 3D-stacked silicon dies (processor die and photonics die) with 50 *um* thickness, separated by 50 *um high and* 50 *um wide* microfluidic channels, and a 100 *um* porous silicon insulation layer (Figure 61). The liquid choice for the coolant is water and its flow rate is limited to 30 ml/min per cavity, because higher flow rates could cause erosion and pressure damage in the channels. The thermal resistivity for Si is 0.01 *m-K/W*, and for the porous Si insulation layer is 1 *m-K/W* [50]. The TSVs running through the insulation layer are highly conductive, so we also include the thermal impact of TSVs in our model. The ambient temperature is fixed at *45* ^{0}C. In our analysis, we observe the maximum temperature fluctuation at a given point on the photonics layer while the power consumption density of the core layer fluctuates between 0.5 W/cm^2 (leakage only) to 4 W/cm^2 for both 100 mm^2 and 400 mm^2 chips.

5.5.2 Experimental Results

In Figure 62, we present the maximum temperature fluctuation on the 100 mm^2 chip when the core power consumption goes from 50W (leakage only at idle) to 400W power consumption, with no insulation (shown in Figure 60), with insulation (shown in Figure 61). When there is no insulation, the temperature can vary up to 19 ^{0}C with 20 *ml/min* per cavity (10 cavities) flow rate and 17.6 ^{0}C with 30 *ml/min* per cavity coolant flow rate. When we place a porous silicon insulating layer between, the temperature fluctuation goes down the 2.8 ^{0}C which can be compensated by the microrings. The operating frequency of the microrings change 0.9 $nm/^{0}C$ [51], which means only 0.25 *nm* shift in the modulation (demodulation) wavelength when PARKA is coupled with the microfluidic cooling solution. The microrings presented in [2] has a full half width measurement (FHWM) of 0.65 *nm*, so they can compensate the 0.25 *nm* shift. In

conclusion, PARKA reduces the maximum temperature.fluctuation on a 100 mm^2 chip by 6.3x, so the photonic network can operate without needing the constant heating of the ring-heaters.

Microfluidic cooling solutions are proposed for high performance computing applications due to their superior cooling capacity, therefore we investigate the maximum temperature fluctuation on a 400 mm^2 chip with similar core power densities. Without the insulation, the temperature fluctuation can be as high as 36.5 ^0C (30 *ml/min* per cavity), and it goes down to 6.2 ^0C with the addition of the insulating layer. Note that even the 6.2 ^0C causes 0.56 *nm* shift in the operating wavelength, which would interrupt the communication. We observe that, when the 400 mm^2 chip stays below the 2.4 W/cm^2 power consumption density, the maximum fluctuation stays below 2.8 ^0C which can be compensated by the microrings. As a result, the 400 mm^2 chip needs to be throttled 60% of the time in order to avoid the constant need for using ring-heaters to regulate the photonic die temperature, which would limit its performance significantly.

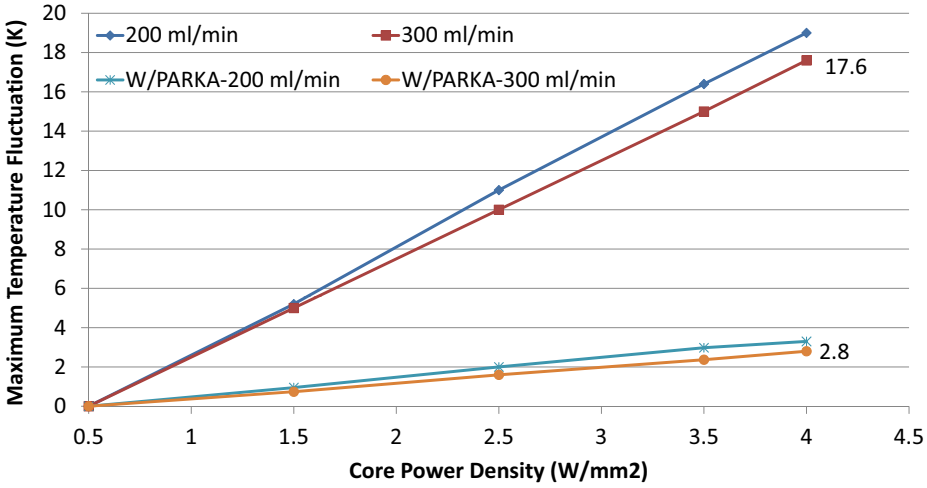


FIGURE 62: Maximum temperature fluctuation at the photonics layes with the microfluidic cooling solution, with and without an insulation layer.

5.6 Limitations and Challenges

Parka reduces the ring-heating power consumption by thermally decoupling the photonics die from the processor die using a porous Si insulating layer. Adding the insulation layer is expected to increase the manufacturing cost only marginally. The porous Si insulation layer can be readily integrated into the CMOS process by passing a plain silicon die through a simple electrochemical process that oxidizes it [50]. This silicon die is not subject to the regular yield-induced costs of dies that implement complex logic and require multiple mask exposures and several metal layers, and thus it is significantly cheaper. The addition of the porous Si layer also does not affect the number of TSVs and the number of pins in the package, which together with the logic and photonic dies constitute the dominant cost factors [20,79]. The additional layer will incur 3D-bonding costs, but these will increase the total cost by less than 1.5% [20,79].

We observe that the thickness of the insulation layer impacts the ring-heating power savings. We assumed 150 *um* thick porous Si layer, as we find that a thinner layer does not provide adequate thermal insulation. The power delivery and communication between the dies is maintained through high aspect ratio TSVs [25, 66, 78]. The manufacturing of this TSVs through the porous Si might be challenging. How-

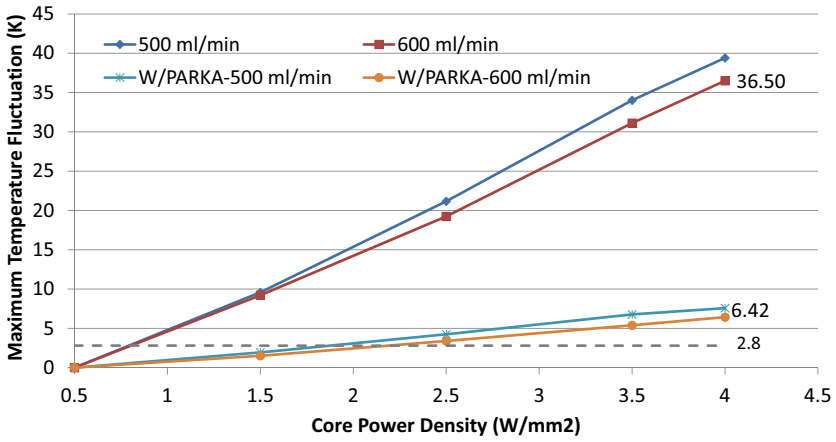


FIGURE 63: Maximum temperature fluctuation at the photonics layes with the microfluidic cooling solution, with and without an insulation layer.

ever Parka does not depend on the exact insulator technology used, so we can use 5 *um*-thick air or vacuum cavity etched between layers if the manufacturing of TSVs through the porous Si deemed too challenging.

For the liquid cooling solution we assume that microchannels facilitate forced convective interlayer cooling with single-phase fluids, in particular water. We model high-aspect ratio TSVs with 10 *um* diameter [66], located and etched within 100 *um*-wide microchannel walls as in [63]. We assume uniformly distributed microchannels, and equivalent fluid flow rate through each channel in the same layer. We assume flow rates in the range of 10-30 *ml/min* to avoid erosion or damage to the microfluidic channels. The fluid pump and valve consume 1.3 *W* per 10 *ml/min* flow, and the power is linear to the volumetric fluid flow [63]. However, we expect the microfluidic cooling solution to recoup this power consumption through lower leakage power and faster processing speeds, because it keeps the cores cooler allowing them to run faster.

Chapter 6

Discussion and Future Work

In this work, we showed that the photonic interconnects is a key element to the high-performance and energy-efficient processor design, and their energy efficiency can be improved via laser power-gating and thermal insulation techniques we proposed. We modeled our proposed schemes in detail to show the performance and the energy impact most accurately. However, there are some improvements that can take our schemes one step further, in terms of power and performance benefits. In this section, we will discuss these improvements and their expected impact.

Firstly, the laser efficiency highly depends on the operating temperature of the laser. According to [29], when the temperature increases from 25 ^0C *to 35* ^0C*,* the effective reflectivity decreases, which leads to an increase in the cavity loss, which reduces the wall-plug efficiency of the laser. In example, [37] shows that, the laser efficiency decreases from 15% to 10%, when temperature increases from *20* ^0C to 80 ^0C*.* As we previously showed, the temperatures of a high performance chip fluctuates rapidly, therefore, when we are calculating the on-chip laser power consumption, we should take laser temperature into account. If the laser efficiency decreases with the high temperature, the impact of ProLaser increases, because, it saves the wasted power. Also, the importance of microfluidic cooling increases, because, it keeps the processor cool and reduces the temperature fluctuations. In the future, we can combine ProLaser with Parka to improve the laser-power and the ring-heating power savings further.

Secondly, previously different power-gating techniques on electrical interconnects has been proposed, which aim to reduce the leakage power consumption of electrical routers [12]. Different than these previous schemes that work on the electrical interconnects, ProLaser aims to power-gate the optical links which

are the biggest source of the static power consumption in the photonic interconnects. We observed that Pro-Laser's electrical routers (of the radix-16 and the Firefly topologies) only consumes 2-6W, so we didn't expect a big energy reduction by implementing the electrical power-gating. However, another photonic topology which has higher radix routers (flattened-butterfly) may benefit from the electrical power-gating too. Furthermore, the wake-up latency for electrical routers have been shown to be on the order of 5-8 cycles [12], which means that, this wake-up latency can be overlapped with the laser turn-on latency, so we expect to see minimal performance impact. As a result, integrating the previously proposed power-gating schemes on the electrical routers [12] can improve the energy savings of laser-power gating schemes we proposed.

Lastly, we showed that, Parka with microfluidic cooling solution can remove the constant need for using ring-heaters to regulate the photonic die temperature, when applied on relatively smaller chips (100 mm^2), but failed to do so with the larger high-performance chips (400 mm^2). This means, a Galaxy like disintegrated processor (which connects smaller chiplets together using a photonic interconnect) can take full advantage of the microfluidic cooling solution, whereas, a single-chip processor can't. In the future, we should explore the performance and energy efficiency impact of Galaxy with microfluidic cooling solutions.

Related Work

Several on-chip interconnect networks exploiting optical signaling have been proposed. Previously, Beamer *et al*. [5] explained how multi-socket systems can provide higher hardware parallelism while using smaller dies with high production yield. Batten *et al*. [2] proposed to connect a many-core processor to the DRAM memory using monolithic silicon. Koka *et al*. [39] discuss the design and implementation of a silicon-photonic network for a large multi-die "macrochip" system. In contrast to these architectures, Galaxy leverages optical fibers to create a high-bandwidth, scalable, low-latency photonic interconnect that can support both processor disintegration and multi-chip integration, and at the same time enable cheap cooling solutions.

Different on-chip interconnect networks have been proposed that exploit CMOS-compatible photonics for future multicore processors. The Corona [74] architecture and many others [73,56,55], implement a monolithic MWSR crossbar topology to support on-chip communication. The hierarchical Firefly architecture [57] advocates the use of partitioned nanophotonic SWMR crossbars to connect clusters of electrically-connected mesh networks. Firefly improves power efficiency and provides uniform global bandwidth between all clusters. These network topologies, can exploit Laser power-gating techniques such as EcoLaser and ProLaser to achieve higher laser energy efficiency while maintaining their performance.

Previous work has explored segregating the interconnect used for core communication from the interconnect used for communication with the cache [32,48] to lower the network cost or to optimize for data placement and partitioning. However, such designs have not been proposed or evaluated in the context of

photonic interconnects. ProLaser segregates the data portion of the photonic interconnect from the control portion and manages them separately, to maximize power savings without hurting performance.

The high laser and ring-heating power consumption reduce the energy efficiency of the nanophotonic interconnects. Thonnart *et al.* [72] proposes powering down the unused units of an electrical interconnect to reduce static power consumption. Zhou *et al.* [81] propose a mechanism that controls active splitters to tune channel bandwidth on a binary tree network and increase channel utilization, which leads to higher energy efficiency. Kurian *et al.* [42] propose an optical R-SWMR crossbar and electrical hybrid interconnection network, and improve performance by utilizing the coherence protocol. Chen *et al.* [10] proposes a technique to distribute laser power across multiple busses based on the changes in the bandwidth demand to improve energy-efficiency in a multi-bus NoC. Kurian *et al.* [42] mention that a Ge-based laser can be controlled to improve the laser energy efficiency, but they do not present nor evaluate a detailed laser-control scheme. Nitta *et al*. [52] show the energy inefficiency of photonic interconnects under low utilization, and propose to improve efficiency by recapturing the energy of photons which are not used for communication. In contrast to previous work, we propose ProLaser, a laser control mechanism applicable to both on-chip and off-chip lasers that improves the laser energy efficiency for R-SWMR crossbars, while providing high bandwidth and performance. Furthermore, all these works are orthogonal to Parka and can be used in addition to Parka to achieve even higher energy efficiency. Parka covers the photonic die with an insulation layer that keeps its temperature stable with low energy expenditure, while minimizing the spatial and temporal thermal coupling between logic and silicon-photonic components.

Nitta et al.[51] showed that the microring resonators are highly susceptible to thermal fluctuations, and he proposed to use additional redundant microrings to compensate for wavelengths shifts due to temperature changes (by creating a thermal control window TCW). Parka is orthogonal to this scheme as it stabilizes the temperature of the microrings with low energy consumption. Both of these techniques can be

applied together and they can benefit from each other, because Parka would allow for smaller TCW, so fewer number of redundant microrings would be used, and Nitta's TCW can relax the strict temperature stability requirement of Parka, so ring-heaters would need to work less.

Microfluidic cooling solutions have been proposed for high performance applications [68], and when combined with Parka they can remove the need for the constant ring-heating for the temperature stabilization.There are several techniques that can be used to resolve the thermal challenges of the silicon microring resonator devices. Methods to reduce the thermal dependence of microrings to tolerable levels include athermalization using negative thermo-optic materials or the embedment of the microring in a thermally-balanced interferometric structure. However, it is challenging to integrate the necessary polymer and TiO_2 materials into a CMOS-compatible fabrication process, and the interferometric structure still suffers from susceptibility to fabrication tolerances, increases the footprint of the microring, and it is challenging to adapt the technique to larger microring switch fabrics [53]. Thus, control-based techniques that aim to detect and react to the resonance shift due to thermal fluctuations are preferable, and several prototypes have been shown to withstand thermal variations across a wide temperature range up to 32–60 ^{o}K [83]. It is beyond the scope of this work to provide a detailed review and comparison of such techniques. However, the interested reader could refer to some of the excellent surveys on this topic that are available in the literature, e.g., Padmaraju and Bergman [53].

Chapter 7

Conclusion

In this work, we showed that photonic interconnect can be used to design high-performance energy-efficient multi-chip processors that can break free of the power, bandwidth and yield limitations which single-chip designs are subject to. We introduced Galaxy, a multi-chip architecture which builds a many-core "virtual chip" by connecting multiple smaller chiplets through optical fibers. Galaxy is designed to push back the power constraints, in addition to overcoming the area and bandwidth limitations, while matching the high performance of tightly-coupled chips. We demonstrate that Galaxy achieves 1.8-3.4x average speedup over competing single-chip designs, and achieves 2.6x lower energy-delay product (6.8x maximum).

Secondly, we aim to address energy-efficiency problems of nanophotonic interconnects by introducing laser control and efficient ring heating solutions. We proposed EcoLaser, a collection of static and adaptive laser control mechanisms that react to the demands of the aggregate workload by opportunistically turning the laser off during periods of low activity to save energy, and leaving it on during periods of high activity in order to meet the high bandwidth demand. Our results indicate that EcoLaser saves up to 77% laser energy for radix-16 and radix-64 SWMR and MWSR crossbars and achieve up to 2x speedup and has 74-77% lower EDP on average compared to a conventional design with no laser control. I improved upon EcoLaser by introducing ProLaser, which is a laser control scheme that outperforms Eco-

Laser (saves up to 88% of laser energy) scheme by keeping the majority of the data-bus inactive while sending small (dataless) messages, and anticipating upcoming messages to turn the lasers on ahead of time. Our results show that laser control is a powerful technique that improves the energy-efficiency of the photonic interconnect, so I extended it to a more scalable Flattened Butterfly [34] network, which will save up to 67% of laser energy (79% for datacenter networks) while causing 2% performance decrease by removing the laser turn-on latency from the critical path.

The nanophotonic devices are highly susceptible to temperature-induced changes, and this forces designers to employ power hungry ring heaters. In a multicore processor there is a potential for significant variation in temperature, so micro-ring resonators must be stabilized at a higher temperature using ring heaters which may consume significant amount of energy. We propose “Parka”, a nanophotonic NoC that encases the photonic die in a thermal insulator that keeps its temperature stable with low energy expenditure, while minimizing the spatial and temporal thermal coupling between logic and silicon-photonic components.Our results indicate that Parka reduces the ring heating power by 3.8x on average across our workload suite. Moreover, the energy savings allow for providing a higher power budget to the cores, which enables them to run faster. Parka on a radix-16 crossbar allows the multicore to achieve 11-23% speedup (34% max) over a baseline scheme with no insulation, depending on the cooling solution used.

In conclusion, providing high-bandwidth, low-latency and energy-efficient communication, photonic interconnects are prime candidate in high-performance and energy-efficient single-chip and multi-chip processor design. The laser power-gating and thermal insulation techniques we presented improve the energy efficiency of them photonic interconnects which allows them to deliver the promised performance and energy-efficiency benefits.

Bibliography

[1] [1] L. A. Barroso and U. Holzle. The case for energy-proportional computing. *IEEE Computer*, 40(12):33- 37, 2007.

[2] [2] C. Batten, A. Joshi, J. Orcutt, A. Khilo, B. Moss, C. W. Holzwarth, M. A. Popovic, H. Li, H. I. Smith, J. L. Hoyt, F. X. Kartner, R. J. Ram, V. Stojanovic, and K. Asanovic. Building many-core processor-to-dram networks with monolithic cmos silicon photonics. *IEEE Micro*, 29(4):8- 21, 2009.

[3] [3] C. Batten, A. Joshi, V. Stojanovic, and K. Asanovic. Designing chip-level nanophotonic interconnection networks. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2(2):137- 153, 2012.

[4] [4] S. Beamer. *Designing Multisocket Systems with Silicon Photonics*. PhD thesis, University of California at Berkeley, Berkeley, CA, USA, 2009.

[5] [5] S. Beamer, K. Asanovic, C. Batten, A. Joshi, and V. Stojanovic. Designing multi-socket systems using silicon photonics. In *Proc. of the Int l Conference on Supercomputing (ICS)*, pages 521- 522, Yorktown Heights, NY, 2009. ACM.

[6] [6] T. Benson, A. Akella, and D. A. Maltz. Network traffic characteristics of data centers in the wild. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC 10, pages 267- 280, New York, NY, USA, 2010. ACM.

[7] [7] R. E. Camacho-Aguilera, Y. Cai, N. Patel, J. T. Bessette, M. Romagnoli, L. C. Kimerling, and

J. Michel. An electrically pumped germanium laser. *Optics Express*, 20(10):11316- 11320, May 2012.

[8] [8] J. Cardenas, C. Poitras, J. Robinson, K. Preston, L. Chen, and M. Lipson. Low loss etchless silicon photonic waveguides. *Optics Express*, 17(6):4752- 4757, 2009.

[9] [9] M. Chang, J. Cong, A. Kaplan, M. Naik, G. Reinman, E. Socher, and S.-W. Tam. CMP network-on-chip overlaid with multi-band rf-interconnect. In *Proceedings of the 14th IEEE Symposium on High-Performance Computer Architecture*, pages 191- 202, February 2008.

[10] [10] C. Chen and A. Joshi. Runtime management of laser power in silicon-photonic multibus noc architecture. *Selected Topics in Quantum Electronics, IEEE Journal of*, 19(2):3700713- 3700713, March 2013.

[11] [11] G. Chen, H. Chen, M. Haurylau, N. Nelson, P. M. Fauchet, E. Friedman, and D. Albonesi. Predictions of cmos compatible on-chip optical interconnect. In *7th International Workshop on System-Level Interconnect Prediction (SLIP)*, pages 13- 20, San Francisco, CA, 2005.

[12] [12] L. Chen, L. Zhao, R. Wang, and T. Pinkston. Mp3: Minimizing performance penalty for power-gating of clos network-on-chip. In *High Performance Computer Architecture (HPCA), 2014 IEEE 20th International Symposium on*, pages 296- 307, Feb 2014.

[13] [13] M. Cianchetti, N. Sherwood-Droz, and C. Batten. Implementing System-in-Package with Nanophotonic Interconnect. *Workshop on the Interaction between Nanophotonic Devices and Systems*, 2010.

[14] [14] M. J. Cianchetti, J. C. Kerekes, and D. H. Albonesi. Phastlane: a rapid transit optical routing net-

work. In *Proceedings of the 36th Annual International Symposium on Computer Architecture*, ISCA 09, pages 441- 450, 2009.

[15] [15] W. J. Dally and T. B. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishing Inc., 2004.

[16] [16] Y. Demir and N. Hardavellas. Ecolaser: An adaptive laser control for energy efficient on-chip photonic interconnects. Technical Report NU-EECS-14-02, Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA, April 2014.

[17] [17] Y. Demir and N. Hardavellas. Ecolaser: An adaptive laser control for energy efficient on-chip photonic interconnects. In *Proceedings of the International Symposium on Low-Power Electronics and Design*, ISLPED 14, August 2014.

[18] [18] Y. Demir and N. Hardavellas. Lac: Integrating laser control in a photonic interconnect. In *IEEE Photonics Conference (IPC)*, pages 28- 29, Oct 2014.

[19] [19] Y. Demir and N. Hardavellas. Towards energy-efficient photonic interconnects. In *Proceedings of Optical Interconnects XV, SPIE Photonics West*, February 2015.

[20] [20] X. Dong, J. Zhao, and Y. Xie. Fabrication cost analysis and cost-aware design space exploration for 3-d ics. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(12), December 2010.

[21] [21] G.-H. Duan, A. Shen, A. Akrout, F. V. Dijk, F. Lelarge, F. Pommereau, O. LeGouezigou, J.-G. Provost, H. Gariah, F. Blache, F. Mallecot, K. Merghem, A. Martinez, and A. Ramdane. High performance inp-based quantum dash semiconductor mode-locked lasers for optical communications.

Bell Labs Technical Journal, 14(3):63- 84, 2009.

[22] [22] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger. Dark silicon and the end of multicore scaling. In *Proceedings of the 38th Annual International Symposium on Computer Architecture*, ISCA 11, pages 365- 376, 2011.

[23] [23] European Semiconductor Industry Association (ESIA), Japan Electronics and Information Technology Industries Association (JEITA), Korean Semiconductor Industry Association (KSIA), Taiwan Semiconductor Industry Association (TSIA), and United States Semiconductor Industry Association (SIA). The international technology roadmap for semiconductors (itrs). http://www.itrs.net/ , 2012 Edition.

[24] [24] A. W. Fang, H. Park, O. Cohen, R. Jones, M. J. Paniccia, and J. E. Bowers. Electrically pumped hybrid AlGaInAs-silicon evanescent laser. *Optics Express*, 14(20):9203- 9210, Oct 2006.

[25] [25] A. C. Fischer, S. J. Bleiker, T. Haraldsson, N. Roxhed, G. Stemme, and F. Niklaus. Very high aspect ratio through-silicon vias (tsvs) fabricated using automated magnetic assembly of nickel wires. *Journal of Micromechanics and Microengineering*, 22(10):105001, 2012.

[26] [26] N. Hardavellas, M. Ferdman, B. Falsafi, and A. Ailamaki. Toward dark silicon in servers. *IEEE Micro*, 31(4):6- 15, July-August 2011.

[27] [27] N. Hardavellas, S. Somogyi, T. F. Wenisch, R. E. Wunderlich, S. Chen, J. Kim, B. Falsafi, J. C. Hoe, and A. G. Nowatzyk. SimFlex: a fast, accurate, flexible full-system simulation framework for performance evaluation of server architecture. *SIGMETRICS Performance Evaluation Review, Special Issue on Tools for Computer Architecture Research*, 31(4):31- 35, April 2004.

[28] [28] M. Heck and J. Bowers. Energy efficient and energy proportional optical interconnects for multi-core processors: Driving the need for on-chip sources. *Selected Topics in Quantum Electronics, IEEE Journal of*, 20(4):1- 12, July 2014.

[29] [29] H. Hisham, G. Mahdiraji, A. Abas, M. Mahdi, and F. Adikan. Characterization of transient response in fiber grating fabry-perot lasers. *IEEE Photonics Journal*, 4(6):2353- 2371, Dec 2012.

[30] [30] H. Hisham, G. Mahdiraji, A. Abas, M. Mahdi, and F. Adikan. Characterization of turn-on time delay in a fiber grating fabry-perot lasers. *IEEE Photonics Journal*, 4(5):1662- 1678, Oct 2012.

[31] [31] M. Horowitz. Scaling, power and the future of cmos. In *Proceedings of the 20th International Conference on VLSI Design*, page 23, jan. 2007.

[32] [32] J. Huh, C. Kim, H. Shafi, L. Zhang, D. Burger, and S. W. Keckler. A nuca substrate for flexible cmp cache sharing. In *Proceedings of the 19th Annual International Conference on Supercomputing*, ICS 05, pages 31- 40, New York, NY, USA, 2005. ACM.

[33] [33] 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 *Proceedings of the IEEE International Symposium on Networks-on-Chip (NOCS)*, pages 124- 133, 2009.

[34] [34] J. Kim, W. J. Dally, and D. Abts. Flattened butterfly: A cost-efficient topology for high-radix networks. In *Proceedings of the 34th Annual International Symposium on Computer Architecture*, ISCA 07, pages 126- 137, June 2007.

[35] [35] L. C. Kimerling. Scaling functionality with silicon photonics: Achievement and potential. http://www.orc.soton.ac.uk/fileadmin/seminar_pdf/UKSP_Showcase_-_Lionel_Kimerling.pdf, Novem-

ber 2013.

[36] [36] N. Kirman, M. Kirman, R. K. Dokania, J. F. Martinez, A. B. Apsel, M. A. Watkins, and D. H. Albonesi. Leveraging optical technology in future bus-based chip multiprocessors. In *Proceedings of the 39th IEEE/ACM Annual International Symposium on Microarchitecture*, MICRO 39, pages 492-503, 2006.

[37] [37] B. Koch, E. Norberg, B. Kim, J. Hutchinson, J.-H. Shin, G. Fish, and A. Fang. Integrated silicon photonic laser sources for telecom and datacom. In *Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC), 2013*, pages 1- 3, March 2013.

[38] [38] B. R. Koch, E. J. Norberg, B. Kim, J. Hutchinson, J.-H. Shin, G. Fish, and A. Fang. Integrated silicon photonic laser sources for telecom and datacom. In *Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2013*, page PDP5C.8. Optical Society of America, 2013.

[39] [39] P. Koka, M. McCracken, H. Schwetman, X. Zheng, R. Ho, and A. Krishnamoorthy. Silicon-photonic network architectures for scalable, power-efficient multi-chip systems. In *Proceedings of the 37th Annual International Symposium on Computer Architecture*, ISCA 10, pages 117- 128, Saint-Malo, France, 2010. ACM.

[40] [40] E. Kotelnikov, A. Katsnelson, K. Patel, and I. Kudryashov. High-power single-mode ingaasp/inp laser diodes for pulsed operation. *Proceedings of SPIE*, 8277:827715- 827715- 6, 2012.

[41] [41] A. Krishnamoorthy, R. Ho, X. Zheng, H. Schwetman, J. Lexau, P. Koka, G. Li, I. Shubin, and J. Cunningham. Computer systems based on silicon photonic interconnects. *Proceedings of the*

IEEE, 97(7):1337 - 1361, july 2009.

[42] [42] G. Kurian, C. Sun, C.-H. Chen, J. Miller, J. Michel, L. Wei, D. Antoniadis, L.-S. Peh, L. Kimerling, V. Stojanovic, and A. Agarwal. Cross-layer energy and performance evaluation of a nanophotonic manycore processor system using real application workloads. In *26th IEEE International Parallel Distributed Processing Symposium (IPDPS)*, pages 1117- 1130, 2012.

[43] [43] S. Larsen, P. Sarangam, and R. Huggahalli. Architectural breakdown of end-to-end latency in a tcp/ip network. In *Computer Architecture and High Performance Computing, 2007. SBAC-PAD 2007. 19th International Symposium on*, pages 195- 202, Oct 2007.

[44] [44] B. Lee, F. Doany, S. Assefa, W. Green, M. Yang, C. Schow, C. Jahnes, S. Zhang, J. Singer, V. Kopp, J. Kash, and Y. Vlasov. 20um-pitch eight-channel monolithic fiber array coupling 160 Gb/s/channel to silicon nanophotonic chip. In *Conference on Optical Fiber Communications and National Fiber Optic Engineers Conference (OFC/NFOEC)*, pages 1 - 3, March 2010.

[45] [45] G. Li, J. Yao, H. Thacker, A. Mekis, X. Zheng, I. Shubin, Y. Luo, J. hyoung Lee, K. Raj, J. E. Cunningham, and A. V. Krishnamoorthy. Ultralow-loss, high-density soi optical waveguide routing for macrochip interconnects. *Optics Express*, 20(11):12035- 12039, May 2012.

[46] [46] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi. Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of the 42nd IEEE/ACM Annual International Symposium on Microarchitecture*, MICRO-42, pages 469- 480, 2009.

[47] [47] J. Liu, X. Sun, R. Camacho-Aguilera, L. C. Kimerling, and J. Michel. Ge-on-si laser operating at room temperature. *Opt. Lett.*, 35(5):679- 681, Mar 2010.

[48] [48] P. Lotfi-Kamran, B. Grot, and B. Falsafi. Noc-out: Microarchitecting a scale-out processor. In *Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-45, pages 177- 187, Washington, DC, USA, 2012. IEEE Computer Society.

[49] [49] R. Merritt. ARM CTO: Power surge could create dark silicon. http://www.eetimes.com/electronics-news/4085396/ARM-CTO-power-surge-could-create-dark-silicon-, Oct. 2009.

[50] [50] B. Mondal, P. Basu, B. Reddy, H. Saha, P. Bhattacharya, and C. Roychoudhury. Oxidized macro porous silicon layer as an effective material for thermal insulation in thermal effect microsystems. In *International Conference on Emerging Trends in Electronic and Photonic Devices Systems*, pages 202- 206, Dec 2009.

[51] [51] C. Nitta, M. Farrens, and V. Akella. Addressing system-level trimming issues in on-chip nanophotonic networks. In *17th IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 122- 131, 2011.

[52] [52] C. Nitta, M. Farrens, and V. Akella. Dcof: An arbitration free directly connected optical fabric. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2(2):169- 182, June 2012.

[53] [53] K. Padmaraju and K. Bergman. Resolving the thermal challenges for silicon microring resonator devices. *Nanophotonics*, 3(4-5):269–- 281, September 2013.

[54] [54] Y. Pan, Y. Demir, N. Hardavellas, J. Kim, and G. Memik. Exploring benefits and designs of optically connected disintegrated processor architecture. *Workshop on the Interaction between Nanophotonic Devices and Systems (in conj. with MICRO-43)*, December 2010.

[55] [55] Y. Pan, J. Kim, and G. Memik. Flexishare: Channel sharing for an energy-efficient nanophoton-

ic crossbar. In *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 1- 12, Bangalore, India, Jan. 2010.

[56] [56] Y. Pan, J. Kim, and G. Memik. Featherweight: low-cost optical arbitration with qos support. In *Proceedings of the 44th IEEE/ACM Annual International Symposium on Microarchitecture*, MICRO-44, pages 105- 116, 2011.

[57] [57] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A. Choudhary. Firefly: Illuminating future network-on-chip with nanophotonics. In *Proceedings of the 36th Annual International Symposium on Computer Architecture*, ISCA 09, Austin, TX, 2009.

[58] [58] M. Paniccia and J. Bowers. First electrically pumped hybrid pumped hybrid silicon laser silicon laser. http://www.intel.com/content/dam/www/public/us/en/documents/technology-briefs/intel-labs-hybrid-silicon-laser-announcement.pdf, September 2006.

[59] [59] K. Petermann. *Laser Diode Modulation and Noise*, volume 3 of *Advances in Optoelectronics (ADOP)*. Springer, 1988.

[60] [60] J. Poulton, R. Palmer, A. Fuller, T. Greer, J. Eyles, W. Dally, and M. Horowitz. A 14-mW 6.25-Gb/s transceiver in 90-nm CMOS. *IEEE Journal of Solid-State Circuits*, 42(12):2745- 2757, 2007.

[61] [61] B. M. Rogers, A. Krishna, G. B. Bell, K. Vu, X. Jiang, and Y. Solihin. Scaling the bandwidth wall: challenges in and avenues for cmp scaling. In *Proceedings of the 36th Annual International Symposium on Computer Architecture*, ISCA 09, pages 371- 382, 2009.

[62] [62] P. Rosenfeld, E. Cooper-Balis, and B. Jacob. Dramsim2: A cycle accurate memory system simulator. *Computer Architecture Letters*, 10(1):16- 19, 2011.

[63] [63] M. M. Sabry, A. K. Coskun, D. Atienza, T. S. Rosing, and T. Brunschwiler. Energy-efficient multiobjective thermal control for liquid-cooled 3-d stacked architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(12):1883- 1896, December 2011.

[64] [64] E. Safi, A. Moshovos, and A. Veneris. L-cbf: A low-power, fast counting bloom filter architecture. *IEEE Transactions on Very Large Scale Integration Systems*, 16(6):628- 638, June 2008.

[65] [65] K. Sankaranarayanan, B. H. Meyer, W. Huang, R. Ribando, H. Haj-Hariri, M. R. Stan, and K. Skadron. Architectural implications of spatial thermal filtering. *Integration VLSI Journal*, 46(1):44- 56, Jan. 2013.

[66] [66] T. Sarvey, Y. Zhang, Y. Zhang, H. Oh, and M. Bakir. Thermal and electrical effects of staggered micropin-fin dimensions for cooling of 3d microsystems. In *IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, pages 205- 212, May 2014.

[67] [67] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *Proceedings of the Annual International Symposium on Computer Architecture (ISCA)*, ISCA 03, pages 2- 13, 2003.

[68] [68] A. Sridhar, A. Vincenzi, D. Atienza, and T. Brunschwiler. 3d-ice: A compact thermal model for early-stage design of liquid-cooled ics. *Computers, IEEE Transactions on*, 63(10):2576- 2589, Oct 2014.

[69] [69] C. Sun, C.-H. O. Chen, G. Kurian, L. Wei, J. Miller, A. Agarwal, L.-S. Peh, and V. Stojanovic. Dsent - a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling. In *6th IEEE/ACM International Symposium on Networks-on-Chip*, pages 201- 210, 2012.

[70] [70] Y. Tamir and G. Frazier. Dynamically-allocated multi-queue buffers for VLSI communication switches. *IEEE Transactions on Computers*, pages 725- 737, 1992.

[71] [71] S. Tanaka, S.-H. Jeong, S. Sekiguchi, T. Kurahashi, Y. Tanaka, and K. Morito. Highly-efficient, low-noise si hybrid laser using flip-chip bonded soa. In *IEEE Optical Interconnects Conference*, pages 12- 13, 2012.

[72] [72] Y. Thonnart, E. Beigne, A. Valentian, and P. Vivet. Automatic power regulation based on an asynchronous activity detection and its application to anoc node leakage reduction. In *14th IEEE International Symposium on Asynchronous Circuits and Systems*, pages 48- 57, 2008.

[73] [73] D. Vantrease, N. L. Binkert, R. Schreiber, and M. H. Lipasti. Light speed arbitration and flow control for nanophotonic interconnects. In *Proceedings of the 42nd IEEE/ACM Annual International Symposium on Microarchitecture*, pages 304- 315, New York, NY, 2009.

[74] [74] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. P. Jouppi, M. Fiorentino, A. Davis, N. Binkert, R. G. Beausoleil, and J. H. Ahn. Corona: System implications of emerging nanophotonic technology. In *Proceedings of the 35th Annual International Symposium on Computer Architecture*, ISCA 08, pages 153- 164, 2008.

[75] [75] T. F. Wenisch, R. E. Wunderlich, M. Ferdman, A. Ailamaki, B. Falsafi, and J. C. Hoe. SimFlex: statistical sampling of computer system simulation. *IEEE Micro*, 26(4):18- 31, Jul-Aug 2006.

[76] [76] P. Wolf, P. Moser, G. Larisch, W. Hofmann, H. Li, J. Lott, C.-Y. Lu, S. Chuang, and D. Bimberg. Energy-efficient and temperature-stable high-speed VCSELs for optical interconnects. In *15th International Conference on Transparent Optical Networks (ICTON)*, pages 1- 5, June 2013.

[77] [77] M. Yang. A comparison of using icepak and flotherm in electronic cooling. In *Proceedings of the 7th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITHERM)*, volume 1, pages 240 - 246, may 2000.

[78] [78] Y. Zhang, H. Oh, and M. Bakir. Within-tier cooling and thermal isolation technologies for heterogeneous 3d ics. In *2013 IEEE International 3D Systems Integration Conference (3DIC)*, pages 1-6, Oct 2013.

[79] [79] J. Zhao, X. Dong, and Y. Xie. Cost-aware three-dimensional (3d) many-core multiprocessor design. In *47th ACM/IEEE Design Automation Conference*, DAC-2010, June 2010.

[80] [80] X. Zheng, J. E. Cunningham, I. Shubin, J. Simons, M. Asghari, D. Feng, H. Lei, D. Zheng, H. Liang, C. chih Kung, J. Luff, T. Sze, D. Cohen, and A. V. Krishnamoorthy. Optical proximity communication using reflective mirrors. *Optics Express*, 16(19):15052 15058, September 2008.

[81] [81] L. Zhou and A. Kodi. Probe: Prediction-based optical bandwidth scaling for energy-efficient nocs. In *Seventh IEEE/ACM International Symposium on Networks on Chip (NoCS)*, pages 1- 8, 2013.

[82] [82] A. Zilkie, B. Bijlani, P. Seddighian, D. C. Lee, W. Qian, J. Fong, R. Shafiiha, D. Feng, B. Luff, X. Zheng, J. Cunningham, A. V. Krishnamoorthy, and M. Asghari. High-efficiency hybrid III-V/Si external cavity DBR laser for 3um SOI waveguides. In *9th IEEE International Conference on Group IV Photonics (GFP)*, pages 317- 319, 2012.

[83] [83] W. Zortman, A. Lentine, D. Trotter, and M. Watts. Integrated cmos compatible low power 10gbps silicon photonic heater-modulator. In *National Fiber Optic Engineers Conference and Optical Fiber Communication Conference and Exposition (OFC/NFOEC)*, pages 1- 3, March 2012.