ON THE EXPLORATION OF NEXT-GENERATION INTERCONNECT DESIGN FOR CHIP MULTI-PROCESSORS

By

ZHONGQI LI

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2012
To my father and mother
ACKNOWLEDGMENTS

First and foremost, I offer sincere gratitude to my advisor, Dr. Tao Li, who has guided me throughout my PhD pursuit with his great knowledge and patience. It has been an exceptional experience to work with Dr. Li in the past years. His mentoring is inspiring and her dedication to work is contagious. The dissertation would have been next to impossible without his vision and research support.

I acknowledge my committee members at University of Florida: Dr. Renato Figueiredo, Dr. Ann Gordon-Ross, and Dr. Peng Jiang. I am truly thankful for the time and efforts that they spent on reviewing and commenting my research proposal and dissertation defense.

I am fortunate to work with a cheerful group of fellow students at Dr. Li’s lab. I appreciate all the warm help and encouragement that I received from the lab members during my personal and professional time. My research would have been less colorful without the witty remarks in the lab now and then.

Finally, I am deeply indebted to my parents. They have provided me with immense understanding and moral support all these years. I have enjoyed every moment we spent together with care and love.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACKNOWLEDGMENTS</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>LIST OF TABLES</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>LIST OF FIGURES</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>ABSTRACT</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>THE INTRODUCTION TO ON-CHIP PHOTONIC COMMUNICATION</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Introduction to Interconnection Network</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>A Sample NoC Architecture</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Network Topology</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Router Architecture</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Background of Photonic Communication</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>The structure of ring resonators</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>Application of ring resonators</td>
<td>23</td>
</tr>
<tr>
<td>2</td>
<td>THE THERMALLY RESILIENT PHOTONIC NETWORK-ON-CHIP ARCHITECTURE</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>A Characterization of Thermal Impact on Photonic NoCs</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>Motivation</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>Structure of Thermal Resilient Photonic NoC System</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>Impact of Temperature on Ring Resonators</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Photonic Network Architecture</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>Thermally Resilient Photonic NoC Architecture</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>Circuit-level Technique</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>Architecture-level Technique</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>Operating System-level Technique</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>Experimental Setup</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>Evaluation Results</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>NoC Latency</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>BER and MER</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>Power Consumption</td>
<td>55</td>
</tr>
<tr>
<td>3</td>
<td>THE ARCHITECTURE OF HIERACHICAL PHOTONIC NOC</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>Motivation</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>The Proposed Hierarchical Photonic NoC Architecture</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>An Overview of Hierarchical Photonic NoC Architecture</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>Dynamic Resource Allocation in Photonic Network</td>
<td>63</td>
</tr>
</tbody>
</table>
## 4 EXPLORING PHOTONIC INTERFACE FOR OFF-CHIP PHASE CHANGE MEMORY SYSTEMS

- **Motivation** .......................................................... 89
- **Background** .................................................................. 91
  - Phase-Change Random Access Memory ........................................ 91
  - The Memory Devices Organization and LPDDR2 Protocol .................. 93
  - OptiPCM System Organization .............................................. 95
- **Sub-channel Division Technology** ........................................ 97
  - Fixed Channel Division .................................................... 97
  - Dynamic Channel Division ................................................. 100
  - The Structure of PIs .......................................................... 100
  - The Design of Memory Controller .......................................... 103
- **Experimental Setup** ....................................................... 107
  - Simulation Methodology ...................................................... 107
  - Power Model of the Communication Bus .................................... 109
- **Performance Evaluation** .................................................. 113
  - Power Consumption Breakdown ........................................... 113
  - Latency Evaluation under Different Memory Configurations .......... 115
  - System Throughput under Different Number of Ranks .................. 116
  - Channel Width Impact on OptiPCM ....................................... 117

## 5 RELATED WORKS .................................................................. 120

## 6 CONCLUSION ...................................................................... 125

## LIST OF REFERENCES .......................................................... 128

## BIOGRAPHICAL SKETCH ...................................................... 138
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>50</td>
</tr>
<tr>
<td>2-2</td>
<td>50</td>
</tr>
<tr>
<td>2-3</td>
<td>52</td>
</tr>
<tr>
<td>2-4</td>
<td>52</td>
</tr>
<tr>
<td>3-1</td>
<td>79</td>
</tr>
<tr>
<td>4-1</td>
<td>108</td>
</tr>
<tr>
<td>4-2</td>
<td>108</td>
</tr>
<tr>
<td>4-3</td>
<td>109</td>
</tr>
<tr>
<td>4-4</td>
<td>109</td>
</tr>
<tr>
<td>2-1</td>
<td>50</td>
</tr>
<tr>
<td>2-2</td>
<td>50</td>
</tr>
<tr>
<td>2-3</td>
<td>52</td>
</tr>
<tr>
<td>2-4</td>
<td>52</td>
</tr>
<tr>
<td>3-1</td>
<td>79</td>
</tr>
<tr>
<td>4-1</td>
<td>108</td>
</tr>
<tr>
<td>4-2</td>
<td>108</td>
</tr>
<tr>
<td>4-3</td>
<td>109</td>
</tr>
<tr>
<td>4-4</td>
<td>109</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>The structure of a 2D Mesh network</td>
<td>18</td>
</tr>
<tr>
<td>1-2</td>
<td>The architecture of a typical router in Mesh or Torus architecture</td>
<td>20</td>
</tr>
<tr>
<td>1-3</td>
<td>The structure of a ring resonator</td>
<td>22</td>
</tr>
<tr>
<td>1-4</td>
<td>A typical optical communication system</td>
<td>25</td>
</tr>
<tr>
<td>2-1</td>
<td>Representative schematics of ring-resonator building blocks</td>
<td>31</td>
</tr>
<tr>
<td>2-2</td>
<td>Simplified layout of a ring modulator</td>
<td>32</td>
</tr>
<tr>
<td>2-3</td>
<td>Transmission spectra affected by DC bias voltages and temperature</td>
<td>33</td>
</tr>
<tr>
<td>2-4</td>
<td>Impact of temperature shift</td>
<td>36</td>
</tr>
<tr>
<td>2-5</td>
<td>Placement of temperature-detecting resonators</td>
<td>38</td>
</tr>
<tr>
<td>2-6</td>
<td>Photonic network layout</td>
<td>39</td>
</tr>
<tr>
<td>2-7</td>
<td>5 x 6 folded torus network augmented with access points</td>
<td>41</td>
</tr>
<tr>
<td>2-8</td>
<td>Schematic diagram of the bias circuit used for compensating small range</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>temperature variations</td>
<td></td>
</tr>
<tr>
<td>2-9</td>
<td>Paths selected by the proposed routing algorithms under various thermal</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>scenarios</td>
<td></td>
</tr>
<tr>
<td>2-10</td>
<td>Operating system-level workload relocation</td>
<td>48</td>
</tr>
<tr>
<td>2-11</td>
<td>Thermal maps of the generated scenarios</td>
<td>51</td>
</tr>
<tr>
<td>2-12</td>
<td>NoC Latency</td>
<td>54</td>
</tr>
<tr>
<td>2-13</td>
<td>Average BER of the network</td>
<td>58</td>
</tr>
<tr>
<td>2-14</td>
<td>Average MER of the network</td>
<td>59</td>
</tr>
<tr>
<td>2-15</td>
<td>Comparison of network power consumption</td>
<td>59</td>
</tr>
<tr>
<td>3-1</td>
<td>An overview of ESPN architecture</td>
<td>62</td>
</tr>
<tr>
<td>3-2</td>
<td>The VCSEL sources</td>
<td>63</td>
</tr>
<tr>
<td>3-3</td>
<td>The network components</td>
<td>67</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>3DI</td>
<td>Three-Dimensional Integration</td>
<td></td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Rate</td>
<td></td>
</tr>
<tr>
<td>BW</td>
<td>Bufferwrite</td>
<td></td>
</tr>
<tr>
<td>CMOS</td>
<td>Complementary metal–oxide–semiconductor</td>
<td></td>
</tr>
<tr>
<td>CMP</td>
<td>Chip multiprocessor</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
<td></td>
</tr>
<tr>
<td>DDR</td>
<td>Double Data Rate</td>
<td></td>
</tr>
<tr>
<td>DLL</td>
<td>Delay-Lock Loop</td>
<td></td>
</tr>
<tr>
<td>DIMM</td>
<td>Dual in-line memory module</td>
<td></td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processor</td>
<td></td>
</tr>
<tr>
<td>DBR</td>
<td>Distributed Bragg Reflector</td>
<td></td>
</tr>
<tr>
<td>DCD</td>
<td>Dynamical channel division</td>
<td></td>
</tr>
<tr>
<td>DRAM</td>
<td>Dynamic random-access memory</td>
<td></td>
</tr>
<tr>
<td>DWDM</td>
<td>Dense Wavelength-Division Multiplexing</td>
<td></td>
</tr>
<tr>
<td>DQS</td>
<td>Data strobe signals</td>
<td></td>
</tr>
<tr>
<td>ECC</td>
<td>Error Correcting Code</td>
<td></td>
</tr>
<tr>
<td>ER</td>
<td>Extinction Ratio</td>
<td></td>
</tr>
<tr>
<td>ESPN</td>
<td>Energy-Star Photonic NoC</td>
<td></td>
</tr>
<tr>
<td>FCD</td>
<td>Fixed channel division</td>
<td></td>
</tr>
<tr>
<td>FIFO</td>
<td>First In, First Out</td>
<td></td>
</tr>
<tr>
<td>FR-FCFS</td>
<td>First-Ready First-Come-First-Serve</td>
<td></td>
</tr>
<tr>
<td>FSR</td>
<td>Free Spectral Range</td>
<td></td>
</tr>
<tr>
<td>IP</td>
<td>Intellectual property</td>
<td></td>
</tr>
<tr>
<td>IPC</td>
<td>Instructions Per Cycle</td>
<td></td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>JEDEC</td>
<td>Joint Electron Devices Engineering Council</td>
<td></td>
</tr>
<tr>
<td>LPDDR</td>
<td>Low-Power Double Data Rate</td>
<td></td>
</tr>
<tr>
<td>LPDDR-NVM</td>
<td>Low-Power Double Data Rate Non-Volatile Memory</td>
<td></td>
</tr>
<tr>
<td>LVCMOS</td>
<td>Low Voltage Complementary Metal Oxide Semiconductor</td>
<td></td>
</tr>
<tr>
<td>MER</td>
<td>Message Error Rate</td>
<td></td>
</tr>
<tr>
<td>MMI</td>
<td>Multimode Interference</td>
<td></td>
</tr>
<tr>
<td>MMP</td>
<td>Markov Modulated Process</td>
<td></td>
</tr>
<tr>
<td>NRZ</td>
<td>Non-return-to-zero</td>
<td></td>
</tr>
<tr>
<td>NoC</td>
<td>Network-on-Chip</td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>Operating system</td>
<td></td>
</tr>
<tr>
<td>PCB</td>
<td>Printed Circuit Board</td>
<td></td>
</tr>
<tr>
<td>PCM</td>
<td>Phase Change Memory</td>
<td></td>
</tr>
<tr>
<td>PHOP</td>
<td>Path Hop</td>
<td></td>
</tr>
<tr>
<td>PSE</td>
<td>Photonic Switching Elements</td>
<td></td>
</tr>
<tr>
<td>RC</td>
<td>Routing computation</td>
<td></td>
</tr>
<tr>
<td>RHOP</td>
<td>Request Hops</td>
<td></td>
</tr>
<tr>
<td>RMS</td>
<td>Root Mean Square</td>
<td></td>
</tr>
<tr>
<td>RTD</td>
<td>Resistance Temperature Detector</td>
<td></td>
</tr>
<tr>
<td>SA</td>
<td>Switch allocation</td>
<td></td>
</tr>
<tr>
<td>ST</td>
<td>Switch traversal</td>
<td></td>
</tr>
<tr>
<td>SoC</td>
<td>System-on-Chip</td>
<td></td>
</tr>
<tr>
<td>SPF</td>
<td>Shortest Distance First</td>
<td></td>
</tr>
<tr>
<td>SSTL</td>
<td>Stub Series Terminated Logic</td>
<td></td>
</tr>
<tr>
<td>TDM</td>
<td>Time Division Multiplexing</td>
<td></td>
</tr>
<tr>
<td>TF</td>
<td>Temperature First</td>
<td></td>
</tr>
<tr>
<td>Acronym</td>
<td>Full Form</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>TSV</td>
<td>Through-silicon via</td>
<td></td>
</tr>
<tr>
<td>ToC</td>
<td>Thermo-optic coefficient</td>
<td></td>
</tr>
<tr>
<td>VA</td>
<td>Virtual-channel allocation</td>
<td></td>
</tr>
<tr>
<td>VC</td>
<td>Virtual Channel</td>
<td></td>
</tr>
<tr>
<td>VCSEL</td>
<td>Vertical-Cavity Surface-Emitting Laser</td>
<td></td>
</tr>
<tr>
<td>WDM</td>
<td>Wavelength Division Multiplexing</td>
<td></td>
</tr>
</tbody>
</table>
ON THE EXPLORATION OF NEXT-GENERATION INTERCONNECT DESIGN FOR CHIP MULTI-PROCESSORS

By

Zhongqi Li

December 2012

Chair: Tao Li
Major: Electrical and Computer Engineering

With the emergence of multi- and many-core processors, the required bandwidth to support effective on-chip communication is expected to grow rapidly. According to ITRS, conventional electrical interconnect will become a power and performance bottleneck for future on-chip communication. As a result, photonic Network-on-Chips (NoCs) are drawing increased attention as an alternative to achieve low power and high-bandwidth interconnects in the multi-/many-core era.

Nevertheless, the design of energy-efficient photonic NoCs faces many new challenges. Our work exploits several aspects of next generation photonic NoC design. For example, photonic NoCs are sensitive to ambient temperature variations because their basic constituents, ring resonators, are themselves sensitive to those variations. We propose a thermally resilient photonic NoC architecture design that supports reliable and low bit error rate (BER) on-chip communications in the presence of large temperature variations.

Also, we advocate the hierarchical photonic NoC architecture to optimize energy utilization via a three-pronged approach: (1) by enabling dynamic resource provisioning,
it adapts photonic network resources based on runtime traffic characteristics; (2) by leveraging power-efficient routers, it minimizes power used for compensating optical signal loss; and (3) by utilizing all-optical adaptive routing, it improves energy efficiency by intelligently exploiting existing network resources without introducing high latency and power hungry auxiliary routing mechanisms.

We also exploit the utilization of photonic channels in the Phase Change Memories (PCMs) to build a high-performance and energy proportional system, the memory devices need to be reorganized so that (1) smaller rank preserves unnecessary power waste in contemporary computer systems with small-sized cache lines; and (2) concurrent operations of phase change memory devices hide the long write access latency.
CHAPTER 1
THE INTRODUCTION TO ON-CHIP PHOTONIC COMMUNICATION

Introduction to Interconnection Network

An interconnection network is a programmable system that transports data between terminals [1]. An interconnection network is usually programmable in the sense that it makes different connections at different points in time. The network may deliver a message from a terminal to another in one cycle and then use the same resources to deliver a message for other terminals in the next cycle. The network is a system because it is composed of many components: buffers, channels, switches, and controls that work together to deliver data.

The interconnection network may work at many scales. The on-chip networks deliver data between processor cores, caches, and arithmetic units within a single processor. The system-level networks may tie processors to off-chip memories or input ports to output ports. Finally, the local-area and wide-area networks connect disparate systems together within an enterprise or across the globe.

The interconnection network between processor and memory largely determines the memory latency and memory bandwidth, which are two key performance factors, in a computer system. The interconnection network between processors and caches is related to Instructions per Cycle (IPC) directly. The performance of an interconnection network in a communication switch largely determines the capacity (data rate and number of ports) of the switch [1]. Since the demand for interconnection, especially on-chip communication, has grown more rapidly than the capability of the underlying wires, interconnection is now attracting more attentions as a critical bottleneck in most systems.
A Sample NoC Architecture

In order to meet the growing computation-intensive applications and the needs of low-power, high-performance systems, the number of computing resources in single-chip is enormously increasing and current VLSI technology can provide support to such an extensive integration of transistors in a single chip. Especially, when adding many computing resources such as Central Processing Units (CPUs), Digital Signal Processors (DSPs), specific Intellectual Properties (IPs), etc. to build a System-on-Chip (SoC), the interconnection between each other becomes an important bottleneck. In most existing SoC applications, a shared bus interconnection which implements an arbitration logic is used to serialize several bus access requests. This type of bus solution is usually adopted to communicate with each integrated processing unit due to its low-cost and simple control characteristics. However, such a shared bus interconnection has some natural limitation from the perspective of scalability since only one master at a time can utilize the bus. This requires serialized communication of all bus accesses controlled by the arbitrator. Therefore, more advanced interconnection schemes should be taken in environments where the number of bus requesters is large and their required bandwidth for interconnection is beyond current bus solutions.

The NoC architecture is proposed to address such scalable bandwidth requirement issues. The NoC generally uses the on-chip packet-switched micro-network of interconnects. Its basic idea is derived from the traditional large-scale distributed computing networks. The scalable and modular nature of NoCs and their support for efficient on-chip communication lead to the NoC-based system implementations. Even though the current large-scale network technologies are well developed and their
supporting features are excellent, their complicated configurations and implementation complexity make it hard to be adopted as an on-chip interconnection methodology.

**Network Topology**

In order to meet typical SoCs or multi-core processing environment, the basic module of network interconnection like switching elements, the routing algorithm and its packet definition should be light-weighted to result in the implemental solutions on single chips. The NoC approach has clear advantages over the traditional busses and most notably system throughput. And hierarchies of crossbars or multilayered busses have characteristics somewhere in between traditional busses and NoC, however they still fall far short of the NoC with respect to performance and complexity. We will use an example to explain the components in a typical NoC system.
Figure 1-1 presents a sample NoC structured as a 4-by-4 mesh which provides global chip-level communication. Instead of busses and dedicated point-to-point links, a more general 2D-Mesh network is adapted, employing a grid of routing nodes spread out across the chip, connected by communication links. In this dissertation, we will
adapt a simplified perspective in which the NoC contains the following fundamental components.

1. Network adapters implement the interface by which cores (IP blocks) connect to the NoC. Their function is to decouple computation (the cores) from communication (the network).

2. Switching nodes route the data according to chosen protocols. They implement the routing strategy.

3. Links connect the nodes, providing the raw bandwidth. They may consist of one or more logical or physical channels.

Figure 1-1 covers only the topological aspects of the NoC. The NoC employs packet or circuit switching or something entirely different and be implemented using asynchronous, synchronous, or other logic.

**Router Architecture**

The switching node usually contains a router. The architecture of the router is depicted in Figure 1-2. The data transmitted between the processors are usually encapsulated into packets. A typical packet encloses a cache line, an invalidation packet, or part of DMA block data. A packet usually contains the data section and the header.

In each router, the incoming packets are first received and stored in an input buffer. Then the control logic circuits in the router makes a routing decision and channel arbitration. Finally, the granted packet will traverse through a crossbar to the next router, and this process repeats until the packet arrives at its destination.

Each head flit of a packet must proceed through the steps of buffer write (BW), routing computation (RC), virtual-channel allocation (VA), switch allocation (SA), and switch traversal (ST). A head flit, on arriving at an input port, is first decoded and buffered according to its input virtual channel (VC) in the BW pipeline stage.
The VC is an important aspect of NOC. In the case that a VC splits a single physical channel into two channels, it is virtually providing two paths for the packets to be routed. There can be two to eight virtual channels in each physical channel. The use of VCs can reduce the network latency at the expense of area, power consumption, and production cost of the NOC implementation [2].

![Diagram of a typical router in Mesh or Torus architecture]

Figure 1-2. The architecture of a typical router in Mesh or Torus architecture

In the next stage, the routing logic performs RC to determine the output port for the packet. The header then arbitrates for a VC corresponding to its output port in the VA stage. Upon successful allocation of a VC, the header flit proceeds to the SA stage where it arbitrates for the switch input and output ports. On winning the output port, the
flit then proceeds to the ST stage, where it traverses the crossbar. Finally, the flit is passed to the next node through external links in the link traversal (LT) stage. Body and tail flits follow a similar pipeline except that they simply inherit the VC allocated by the head flit. Thus, the time between the header flit of a packet to be received by the router and the downstream node starts to receive the packet without considering the contention could be computed as:

$$T = T_{BW} + T_{RC} + T_{VA} + T_{SA} + T_{ST} + T_{LT}$$

**Background of Photonic Communication**

**The structure of ring resonators**

In recent years, the integrated ring resonators have emerged in the last few years in integrated optics and have been applied into many applications. The integrated ring resonators require no facets or gratings for optical feedback and are thus particularly suited for monolithic integration with other components [3]. In this way, the response from coupled ring resonators can be custom designed by the use of different coupling configurations. Thus, the responses from the ring resonator filters can be designed to have both a flat top and steep rolloff.

A typical layout of the channel dropping filter is shown in Figure 1-3 [4]. This can be regarded as the standard configuration for an integrated ring resonator channel dropping filter. In this example, two straight waveguides also known as the bus or the port waveguides are coupled either by directional couplers through the evanescent field or by multimode interference (MMI) couplers to the ring resonator. A simpler configuration is obtained, when the second bus or port waveguide is removed. Then the filter is typically referred to as “notch” filter because of the unique filter characteristic.
Figure 1-3 shows a Prototype of ring resonator channel drop filter (Ring resonate at frequency $\lambda_2$ and $\lambda_A$. Figure is redrawn from [3]). The ring resonator filters can be described by certain characteristics which are also generally used to describe optical filters. One important characteristic is the distance between resonance peaks, which is called the free spectral range (FSR) [3]. A simple approximation can be obtained for the FSR by using the propagation constant $\beta = \kappa \cdot n_{eff} = \frac{2\pi n_{eff}}{\lambda}$, where $\beta$ is the propagation constant. The vacuum wavenumber $\kappa$ is related to the wavelength $\lambda$ through: $\kappa = \frac{2\pi}{\lambda}$. Using the vacuum wavenumber, the effective refractive index $n_{eff}$ can be introduced easily into the ring coupling relations. By neglecting the wavelength dependency of the effective refractive index.
This equation leads to the FSR $\Delta \lambda$, which is the difference between the vacuum wavelengths corresponding to two peak resonant conditions.

$$\frac{\partial \beta}{\partial \lambda} = -\frac{\beta}{\lambda} + \kappa \frac{\partial n_{\text{eff}}}{\partial \lambda} \approx -\frac{\beta}{\lambda}$$

$$\text{FSR} = \Delta \lambda = -\frac{2\pi}{L} \left( \frac{\partial \beta}{\partial \lambda} \right)^{-1} \approx \frac{\lambda^2}{n_{\text{eff}} L}$$

This equation is also for the resonant condition next to a resonance found for the propagation constant. In the above equations, $\lambda$ is the wavelength, and $L$ is the circumference of the ring which is given by $L = 2\pi r$, where $r$ is the radius of the ring measured from its center to the center of the waveguides. Thus the phase $\theta = \frac{\omega L}{c} = \frac{\kappa c_{\text{air}} L}{c} = \kappa \cdot n_{\text{eff}} \cdot 2\pi r = 4\pi^2 n_{\text{eff}} \frac{r}{\lambda}$.

**Application of ring resonators**

The communication fabric emerges as the critical performance factor when tens or hundreds of cores are integrated into a single chip. Therefore, a high performance network is essential for efficient inter-core communication. By sharing channels and paths, packets can be routed to their destinations with optimum bandwidth, latency, and power. However, electrical NoCs do not scale well because of large latencies associated with conventional RC wires and stringent power requirements [5, 6].

Recently, photonic NoCs have been attracting plenty of attention [7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. Compared to electrical NoCs, photonic NoCs offer higher bandwidth density, lower latency [17], and power consumption that is independent of path length. These characteristics seem to be an answer to the shortcomings of electrical NoCs.

Moreover, Wavelength and Time Division Multiplexing (WDM and TDM) allow several channels to share an optical waveguide for transmitting information, thus
increasing bandwidth density. An optical waveguide is a structure constructed from two materials having different refractive indices. This allows the waveguide to confine and guide light waves via total internal reflection. Unlike electrical wires, energy is only expended at the end points, which reduces power consumption significantly. Since optical signals travel at a speed close to that of light, latencies are also improved. Recent advances in integrating photonic devices with microelectronics using current Complementary metal–oxide–semiconductor (CMOS) technology have made possible the realization of high-speed, low-power modulators, switches, and detectors that are essential to the design of photonic NoCs [18, 19]. The basic building block for these devices is a ring resonator. Ring resonators are waveguides shaped as rings. Resonance occurs when a ring selectively couples one wavelength from a close-by waveguide and ignores the rest. The significance of this ability is that ring resonators can act as filters, switches, modulators, and detectors. Unfortunately, this ability can be compromised due to the effect of temperature variations on refractive index [20, 21], causing the resonance frequency to shift.

Integrated silicon-photonic technology could be used as an ideal candidate for the large scale connection of the multi-core processors due to its low-latency and high scalability. Silicon nanophotonics have made complete photonic on-stack communication systems a promising alternative to electrical communication systems. Nanophotonics significantly improve the interconnect bandwidth density by approximately two orders of magnitude and yields to over 10× power reduction [22]. Figure 1-4 shows the basic optical communication components including the laser source(s), the optical waveguides, the modulators, and the photodetectors. The laser
source(s) multiplex a number of different wavelengths of laser lights into a single waveguide, using the dense wavelength-division multiplexing (DWDM). One feasible laser source, vertical-cavity surface-emitting laser (VCSEL), is a type of semiconductor laser diode with laser beam emission perpendicular from the top surface. The modulators then modulate laser lights to carry the optical bits using the Mach-Zehnder interferometer [3]. The SiGe photodetectors couple and absorb the laser lights at their resonant wavelengths and then convert into current flows to be amplified for the final electrical bits.

Figure 1-4. A typical optical communication system

Apart from the basic photonic components, the turn resonators which are properly tuned couple the traversing optical signals and drop them to the intersecting waveguides. A turn resonator works in a set of resonant frequencies which is derived from its material and structural properties. When the resonant frequency of the turn resonator is different from the traversing wavelength(s) of the optical light(s), the light(s) pass through the waveguide intersection uninterrupted (the red light in Figure 1-1); otherwise they are coupled into the resonator and dropped to the intersecting waveguide (the green light). The material and structural properties of the passive
resonators are predetermined when manufacturing and kept constant during run-time, as (e) in Figure 1-4. The frequency of the active resonator could be tuned during run-time to support different waveguide connections. The frequency tuning is achieved by adjusting the effective index of the resonator and is generally achieved in one of the following ways. The heat tuning applies or cancels the heat on resonator to change the effective index, which usually requires several microseconds. The electrical tuning applies a voltage on the p-n contact and injects electrical current into the resonator to tune the effective index of the ring waveguide as (c), which requires ~100 ps. Another way is to apply optical pump pulses to inject free carriers through two-photon absorption inside the ring resonator and hence tune the effective index of the ring resonator as (d) [23]. The optical pulse tuning has the lowest latency among the three (~40 ps) [24], and is suitable to control the distant resonators which otherwise suffers extra delay from electrical control wire. In our design we apply the electrical tuning at the memory controller while the optical tuning at the memory device.
A Characterization of Thermal Impact on Photonic NoCs

While proposed architectures \[13, 14, 15, 16\] employ a photonic network layer placed on the top of a silicon chip, we use the design proposed in [16] as a representative 3D chip to characterize the impact of thermal variations on the reliability of photonic NoCs. In this section we first describe the photonic network architecture; we then discuss BER as an indicator of thermal effects, quantify the BER due to temperature variations, and address temperature sensing issues.

Our simulated architecture is based on 3D integration, where a photonic NoC is implemented as a layer of optical devices on the top of the silicon chip. Such an arrangement reduces fabrication complexity, chip dimensions, and total cost. A 2D folded torus hybrid NoC topology is used in our study since it is compatible with the tiled chip multiprocessor (CMP) chip, allows the use of low-radix switches, and allows light waves to intersect without significant cross talk. The hybrid NoC architecture [16] combines a photonic circuit-switched network with an electrical packet-switched control network to reduce power consumption while achieving high bandwidth and low latency.

In this study we assume a 2D 30-core processor tiled in a $5 \times 6$ arrangement. To transmit a message, a path setup packet is first sent on the electrical control network. As the packet is routed through the network, it reserves the corresponding photonic switches along its path. Once the optical path is established, the message is transmitted through the photonic network.
**Motivation**

In the ring resonator based photonic NoC, resonance occurs when a ring selectively couples one wavelength from a close-by waveguide and ignores the rest. The significance of this ability is that ring resonators can act as filters, switches, modulators, and detectors. However, this ability can be compromised due to the effect of temperature variations on refractive index [20, 21], causing the resonance frequency to shift. Because a variation in temperature causes a change in the refractive index, it can potentially disrupt the proper operation of photonic devices. For instance, ring resonators can be brought in or out of resonance due to a small variation in temperature. A resonance shift of 0.11 nm/K has been reported in ring resonators [21]. In addition, prior work [25] has reported high BER when a thermal shift as small as several degrees K caused a significant shift from the base resonant wavelength. Thus, small temperature variations can introduce large BER, or even cause faulty operation in photonic NoCs. Conventionally, metal strip heaters embedded around ring resonators [26] or overlaid on top of the silicon oxide cladding [27] are used to control the temperature of the resonators. However, these heaters require substantial electrical tuning power, exacerbate on-chip thermal effects, and are not suitable for use in large-scale photonic NoCs due their bulkiness and extensive wiring requirement. Other methods resort to overlaying a polymer coating with negative temperature coefficient [28, 29]. Unfortunately, polymer is not compatible with CMOS processes yet. The ITRS Roadmap [30] projects that three-dimensional chip stacking for three-dimensional integration (3DI) is a viable solution for latency and power dissipation limitations. Hybrid photonic/electrical NoCs [13, 14, 15, 16] have been proposed to be built on a separate layer on top of the core layer with through-silicon-vias (TSVs) connecting the two layers.
Although latency and power dissipation are improved, thermal effects are compounded due to heat generated by other layers. Since heat is not easily removed from multilayered integration, techniques to counter thermal effects on photonic NoCs at the architectural and operating system levels become imperative.

To mitigate temperature effects on photonic NoCs, we propose Aurora, a thermally resilient photonic NoC architecture design that can tolerate a wide range of temperature variations. Our proposed cross-layer solution targets the device, architecture, and operating system layers where each can significantly improve the reliability of the photonic NoC. More attractively, combining our proposed techniques provides significant reliability improvements and, as a side benefit, better power efficiency. Our first proposed technique deals with temperature variations within a small range. To achieve this at the device level, we adopt the method proposed in [25], which varies the bias current through a ring resonator to compensate for small local temperature variations. At the architectural level and for thermal variations across a large range, we propose to reroute the messages through cooler regions of the chip to their destinations. At the Operating system (OS) level, we use thermal/congestion-aware co-scheduling to reorganize the thermal profile of the chip to further lower BER. To the best of our knowledge, we present the first effort on improving thermal reliability of photonic NoCs at the architecture and operating system levels.

**Structure of Thermal Resilient Photonic NoC System**

Ring resonators are not only applied in the optical networks, the resonators have recently been demonstrated to be used as sensors and biosensors as well. Extensive research strived to create optical devices that can modulate, guide, and detect light signals efficiently while leveraging current CMOS processes. Of those devices, ring
resonators are finding wide acceptance in the photonic and architecture communities for serving as a basic building block for various photonic circuits ranging from modulators to switches and multiplexers. Their compact size, low power consumption, low insertion loss, and high extinction ratio (ER) per unit length, make them ideal for use in on-chip optical networks [25]. In this section, we give an overview of the structure and operation of ring resonators, the role of the refractive index, and the effects of temperature variations on their operation.

In our simulated architecture, we assume that 64 wavelengths are used for modulation, resulting in 64 modulators and 64 photodetectors for a total of 128 ring resonators per core. A total of 4680 ring resonators are used to build the photonic network. In order to increase bandwidth density, path multiplicity can be used, where additional parallel waveguides are added to the network. These new paths will need additional modulators, multiplexers, demultiplexers, photodetectors, and switches. This will dramatically increase the number of ring resonators used in the photonic NoC. A typical example of photonic NoC is shown in Figure 2-1.
Impact of Temperature on Ring Resonators

A ring resonator is built by placing a ring next to a straight waveguide, as shown in Figure 2-2. The ring’s circumference is designed to be a multiple integer of the wavelength traveling through the straight waveguide. The index of refraction of the materials that form the ring waveguide plays an important role in determining the resonance frequency. Resonance occurs when coupled light circulates inside the ring and is reinforced by interference while light traveling in the waveguide is suppressed. Changing the refractive index changes the resonance frequency. To control the refractive index, the method of free-carrier injection [31] is used due to its high speed. In this method, two highly doped regions that form a PIN junction surrounding the ring are built to form a modulator [32], as shown in Figure 2-2. By applying a voltage Vm to the P
and N regions, free-charge carriers are injected into the ring, causing its effective refractive index to change. By injecting more free carriers into the ring, the refractive index decreases. On the contrary, extracting free-charge carriers increases the refractive index. Thus, PIN carrier injection and extraction effectively modulates the refractive index of the ring resonator. A ring resonator can be in one of two states. In the ON state, there are no free carriers in the ring since the PIN junction is reverse-biased. By design, the resonance wavelength of the ring is same as the wavelength of the light, hence resonance is ON, and light is coupled into the ring. This coupling causes the optical signal to circulate inside the ring, and prevents the signal from passing through the waveguide. In the OFF state, the PIN junction is forward-biased, and thus free carriers are injected into the ring. The injection of free carriers changes the refractive index and in turn shifts the resonance wavelength. Since the resonant wavelength is now different from the wavelength of the light signal, resonance is OFF and the light continues its path unobstructed through the straight waveguide.

Figure 2-2. Simplified layout of a ring modulator
As described above, resonance occurs only at some specific frequencies where light is coupled into the ring. The wavelength at which resonance occurs [20] is governed by:

\[ \lambda_0 = n_{\text{eff}} L / m, \]  

where \( n_{\text{eff}} \) is the effective refractive index of the optical mode, \( L \) is equal to \( 2\pi r \), where \( r \) is the radius of the ring, \( m \) is an integer number, and \( \lambda_0 \) is the resonant wavelength. A shift of the effective refractive index results in a shift of the resonant wavelength [21]:

\[ \Delta \lambda = \lambda_0 (\Delta n_{\text{eff}} / n_{\text{eff}}), \]  

where \( \Delta \lambda \) is a change in the resonant wavelength, \( \lambda_0 \) is the resonant wavelength, \( \Delta n_{\text{eff}} \) is a change in the effective refractive index, and \( n_{\text{eff}} \) is the effective refractive index.

Figure 2-3. Transmission spectra affected by DC bias voltages and temperature (a) Transmission spectra of a modulator under two different DC bias voltages, (b) Transmission spectra shifts due to changes in temperature.
Figure 2-3 (a) shows the transmission spectra of a modulator at a nominal operating temperature $\Delta T = 0K$. The figure shows that for a small positive increase in bias current, the spectrum shifts to the left due to the decrease of the refractive index of the silicon caused by the injection of free carriers in the ring. Consequently, resonance occurs at a shorter wavelength than the original one. This shift means that a light wave at the original wavelength of $\lambda_0$ will be allowed to pass since it has a high transmission value. Before the shift, its transmission value at $\lambda_0$ was small, and the wave was suppressed. Let $n_{new} = n_{eff} - \Delta n$, where $\Delta n$ is a change in the refractive index, and $\lambda_{new}$ is the new resonant wavelength. Substituting $n_{new}$ in Eq. (2-2) gives $\lambda_{new} = (n_{eff} - \Delta n)L/m = n_{eff}L/m - \Delta nL/m = \lambda_0 - \Delta \lambda$, where $\Delta \lambda$ is a change in the resonant wavelength.

In addition to the carrier injection method described above, the refractive index can also be altered by temperature variations. Due to silicon’s relatively large thermo-optic effect [33], ring resonators are sensitive to temperature variations. The thermo-optic coefficient (TOC) is given by $\Delta n/\Delta T = 1.86 \times 10^{-4}/K$. As in the carrier injection case, temperature variations also affect the refractive index and result in shifting the resonance wavelength. A resonance shift of 0.11 nm/K from the original resonance wavelength has been reported in [21]. Such resonance shifts are undesirable and increase the BER in systems that use resonant electro-optic modulators and switches. Figure 2-3(b) shows transmission spectra shifts due to 2 K and 4 K temperature shifts. The original spectrum is at a nominal operating temperature $\Delta T = 0K$ and constant bias current. It is interesting to point out that temperature variation and free-carrier injection have opposite effects on the resonance frequency. For example, an increase in
temperature causes an increase in the refractive index, and a corresponding shift of the spectrum towards the right. Thus, it is possible that electro-optic and thermo-optic effects can compensate each other.

Undesirable thermal shifts will cause large BER and even faulty operation of a photonic NoC. With a rise in temperature, rings will not resonate at the intended frequency. Modulators, switches, multiplexers, and demultiplexers will produce erroneous outputs if thermal shifts are not addressed. Figure 2-3 illustrates several scenarios showing the intended and the actual outputs when a ring fails to resonate at the intended frequency due to a rise in temperature.

In 3D packaging, the photonic network is usually implemented on top of the core layer. It experiences larger non-uniform temperature variations, depending on the temperature of the cores below. Since the photonic layer consists of thousands of ring resonators, the operation of the photonic network will be drastically compromised by the variations in temperature. As described in previous section, these variations affect the refractive index of the ring resonators, causing the transmission spectra of the resonators to shift unpredictably. For example, a few degrees rise in temperature can cause a photonic switching element to malfunction by diverting light when it should not.

To obtain eye diagrams and BER of ring resonators, we simulated optical links with OptiSystem, an optical communication system simulation software [34]. The simulated optical channel consists of a VCSEL source, signal generator that generates 10 Gbps pseudorandom Non-return-to-zero (NRZ) code, a modulator to modulate the NRZ code to optical signals, intermediate resonator, and demodulator. The resonance frequency was varied to simulate the effect of temperature variation.
As seen in Figure 2-4 (a), BER increases with variation in temperature and reaches $10^{-12}$ at a temperature variation of ~ 3.5 degrees K. This value is sufficient for reliable on-chip communication [35]. Figure 2-4 (b) shows the eye diagrams for different temperature variations. Eye diagrams are used to qualitatively examine signal integrity and signal to noise ratio in a communication system. As the temperature varies, the quality of the eye diagrams deteriorates indicating reduced signal integrity.

To obtain runtime chip temperature, we ran multi-core oriented workloads on a cycle-accurate, multiprocessor simulator and the generated power traces are then fed into HotSpot [36]. We modified Garnet [37] to simulate the photonic NoC. We used average BER as an indicator to provide a measure of how temperature variations affect the operation of our simulated photonic network. We obtained BER along the optical path by evaluating the temperature of the involved photonic devices. We observed that if temperature variations were left unaddressed, the average BER across the network would be unacceptably high (greater than $10^{-1}$) and all messages would be corrupted.
during transmission, implying the need for a thermally resilient photonic NoC architecture.

**Temperature-detecting Resonators**

The temperature information of resonators is necessary for maintaining their initial operating conditions. Integrated temperature sensors like thermistors and Resistance Temperature Detectors (RTDs) are usually used to measure the temperature within a chip. However, these conventional integrated sensors require large areas, making them unsuitable for large-scale photonic networks that contain thousands [16] or even millions [15] of ring resonators.

In Aurora, we employ resonators to measure temperature [38] because of their small area overhead and compatibility with CMOS technology. In these resonators, the amplitude of the output is related to temperature variation. Resonators used for temperature detection are coupled to waveguides through splitters to minimize signal loading. In the implementation, the output signal of a resonator is amplified and converted by a Root Mean Square (RMS) detector into DC current whose level indicates the amount of frequency shift (Figure 2-5. (c)). A temperature-detecting resonator along with its detection and control circuitry are deployed in each switch and modulator set. Due to the small size of modulator sets and switches (around 640 and 70 µm in diameter), we assume that the temperature measured is the temperature of the whole set or switch. The placement of these resonators within the modulator sets and the switches is shown in Figures 2-5 (a) and (b).
Photonic Network Architecture

To characterize the impact of thermal variations on the reliability of photonic NoCs, we assume an optical network similar to the one proposed in [16]. In this section we first describe the photonic network architecture, we then discuss BER as an indicator of thermal effects and quantify the BER due to temperature variations.

Our simulated architecture is based on 3D integration, where a photonic NoC is implemented as a layer of optical devices on top of the silicon chip. Such an arrangement reduces fabrication complexity, chip dimensions, and total cost. A 2D folded torus hybrid NoC topology is used in our study since it is compatible with the tiled CMP chip, allows the use of low-radix switches, and allows light waves to intersect without significant cross talk. The hybrid NoC architecture combines a photonic circuit-
switched network with an electronic packet-switched control network to reduce power consumption while achieving high bandwidth and low latency.

Figure 2-6. Photonic network layout (a) 5 × 6 folded torus network, (b) Access point (c) Modulators and detectors, (d) 4 × 4 switch

In this chapter we assume a 2D 30-core processor tiled in a 5 × 6 arrangement. The detailed processor, memory and NoC configuration can be found in Section 5. Figure 2-6 (a) shows the layout of the 2D grid of optical waveguides with switches at the intersection points. An electronic sub-network of similar layout (not shown) is overlaid on the photonic network. This network is used for control and short messages. Each core connects to the photonic network through an access point. Access points enable the injection and ejection of messages without interference with through traffic, and avoid blocking between injected and ejected traffic. Figure 2-6 (b) provides a magnified in view of an access point excluding a torus switch. As can be seen, an access point consists of a gateway and 3 switches. A gateway, shown as Figure 2-6 (c), acts as a
A photonic network interface which connects each core to the folded torus network. A gateway converts electronic signals to optical and optical signals to electronic. It contains optical modulators and detectors whose structure is based on ring resonators. A gateway is connected to a switch through its West port, while the other 2 switches are for injection and ejection of messages. As shown in Figure 2-6 (d), injection, ejection, and torus switches are $4 \times 4$ switches controlled by an electronic router. Each $4 \times 4$ switch is made of four Photonic Switching Elements (PSE). A PSE is a $2 \times 2$-switching element capable of switching the direction of a light signal. It is based on a ring resonator structure where two rings are placed at the intersection of two waveguides.

Figure 2-7 illustrates the topology of the simulated $5 \times 6$ folded torus network augmented with the access points. Access points comprise injection and ejection switches that lie on additional waveguides to facilitate injection and ejection. A gateway and switch unit is connected to two injection switches and one ejection switch. Torus switches are used to route messages between the cores. To transmit a message, a path setup packet is first sent on the electronic control network once the destination address is known. As the packet is routed through the network, it reserves photonic switches along the path to be followed by the photonic message. A next-hop decision is made at every router along the path, depending on the routing algorithm. The process of reserving the photonic path is completed when the packet reaches its destination. To indicate that a path is now open, a short light pulse is transmitted through the waveguide back to the source. The source realizes that the optical path is established and sends out the message through the photonic network. At the end of the message, a path-teardown packet is sent to release all resources and free the path. An acknowledgement packet may be sent on the electronic control network if guaranteed-delivery is requested.
In our simulated architecture, a single core contains four 4 × 4 switches and one gateway. We assume that 16 wavelengths are used for modulation, resulting in 16 modulators and 16 receivers for a total of 32 ring resonators. A total of 1920 ring resonators are used to build the photonic network. In order to increase bandwidth density, path multiplicity can be used, where additional parallel waveguides are added to the network. These new paths will need additional modulators, multiplexers, demultiplexers, photodetectors, and switches. This will dramatically increase the number of ring resonators used in the photonic NoC.

**Thermally Resilient Photonic NoC Architecture**

We propose a holistic approach to mitigate the effect of temperature variations on the operation of photonic NoCs. Our techniques target circuit, architecture, and OS levels respectively. For small temperature variations, we adopt a circuit-level technique
that adjusts the bias current flowing through ring resonators to locally compensate for thermal effects. At the architecture level and for larger temperature variations, we reroute messages away from higher temperature regions through cooler regions to their destinations. At the OS level, we employ a thermal/congestion-aware co-scheduling technique to further reduce BER. More attractively, our solutions at the circuit, architecture, and OS levels can be further integrated with each other to reduce BER.

Circuit-level Technique

We use the circuit-level technique proposed in [25] to combat temperature variations within a small range (e.g. 15 K). The heat generated by the flow of an appropriate DC bias current through a ring resonator is used to maintain the original operating conditions. The amount of Joule heat generated in the device is proportional to the value of the bias current. As the temperature varies, the bias current is varied to compensate for changes in local temperature in order to maintain the resonant frequency at its original value. Figure 2-8 shows the schematic diagram used to control the bias current through a PIN resonator-based modulator. Only sectors of the ring and the N region are shown for clarity. A bias tee network combines a modulating signal with the DC bias to modulate the refractive index of the resonator via free-carrier injection and extraction. The inductor and the capacitor provide isolation between the DC bias and the RF bit generator inputs. In [25], the modulation was maintained for a temperature rise of 15 K by changing the base operating condition from 1.36 mA at 0.2 V to 345 µA at 2.2 V bias. In nominal operation, reducing the bias current does not have an effect on the modulation process since the high-speed RF signal injects the required amount of carriers to perform switching. The use of this technique is limited to small variations in temperature since the amount of wavelength shift using the free-carrier
injection and extraction method is limited to about 2 nm. In contrast, the amount of wavelength shift due to temperature variations can be up to 20 nm [39].

Figure 2-8. Schematic diagram of the bias circuit used for compensating small range temperature variations

**Architecture-level Technique**

The circuit-level solution could mitigate the impact of small variations in temperature. However, due to the variance of running workloads, some regions of the chip area may experience temperature variations beyond the compensation range of the circuit-level solution. We propose re-routing messages away from resonators within these regions, and through cool regions to their destinations. We propose two techniques based on the shortest-distance algorithm: shortest-path first (SPF) and temperature first (TF). SPF selects the path with the lowest MER among all shortest paths available. On a tie, the algorithm selects the path with the lowest utilization. TF selects the path with the lowest temperature (i.e. the lowest MER path between source
and destination) when the circuit-level technique is unable to compensate. On a tie, the algorithm considers route length and route utilization in order to mitigate link delay and avoid congestion.

Figure 2-9 illustrates the routes generated by the proposed algorithms under various thermal scenarios. The regions where the DC bias current was able to compensate for the resonance frequency shifts are indicated in white. The regions that are beyond the compensation range of the DC bias current are indicated in orange. Source and destination nodes are indicated in blue. The paths selected by SPF algorithm are indicated by “A”, and the paths selected by TF algorithm are indicated by “B”. As shown, our proposed routing algorithms search for a shortest-distance path to the destination by avoiding hot regions and hence incur low MER. Messages that fail to find a cool path towards the destination incur a higher MER than messages that succeed. Messages that fail to be delivered are retransmitted after a timeout period.

The routing path is calculated by the source node. In order to compute a routing path, the source node gathers temperature information of the resonators, which is distributed to all nodes through the electrical network. Before sending a packet, the source node first calculates the MER at each resonator along the routing path according to $\text{mer}_i = 1 - (1 - \text{ber}_i)^n$, where $\text{ber}_i$ is the BER of one resonator and $n$ is the number of bits in one message. Then, the MER is obtained by multiplying all MERs for each resonator on that path, i.e. $\text{mer} = 1 - \prod_r (1 - \text{mer}_r)$, where $r$ is the number of resonators in that path. After that, the source node performs either SPF or TF algorithm utilizing $\text{mer}$ as the weight of a path. Then the source node selects the path with the minimum weight among the shortest-distance paths (SPF) or the path with the minimal
weight among all paths (TF). Aurora employs an electrical/optical hybrid network structure, and path establishment is performed via the electrical network, so it is reasonable to assume that no error occurs when establishing the path. Deadlock in the electrical network can be avoided by using virtual channel flow control. On the other hand, the photonic network is inherently deadlock-free due to circuit-switching and predetermined routing path. Livelock is also avoided due to the predetermined routing path.

Note that as the number of cores increases, the number of paths available for transmission also increases. Therefore, it is expected that the proposed routing algorithms scale well in large-scale multi-/many-core systems. However, if the source and/or destination cores are located in hot regions themselves, a high MER is inevitable regardless of the selected path. In these situations, thermal management solutions such as dynamic clock disabling and dynamic frequency scaling can be invoked to halt or power off hot cores for a period of time [40] to guarantee reliable communication.
Operating System-level Technique

To further mitigate the effect of temperature variations on the photonic network and reduce the MER, we propose a thermal/congestion-aware co-scheduling scheme at the operating system level. The operating system distributes workloads across the multi-core substrate in order to reorganize the temperature profile of the chip. The OS prioritizes the outer cores of the chip rather than the inner cores when mapping the workloads to the cores. Usually, a set of related workloads occupies adjacent cores and the communications demand within that set is high. We treat related workloads as one set when performing thermal/congestion-aware co-scheduling. Figure 2-10 (a)
shows a scenario in which this co-scheduling technique relocates 4 workload sets (T1’-T4’) to new locations (T1-T4). Workload sets can be rotated when necessary as in the case of T3. If the outer cores are already occupied by other workloads, rescheduling will only be performed when a workload set can be mapped as a block in order to maintain efficient communication among the set. Figure 2-10 (b) shows the pseudo-code of the co-scheduling algorithm.

This thermal/congestion-aware co-scheduling algorithm provides two benefits: First, relocating workloads to the edges of the chip helps reduce both peak and average chip temperatures since the edges of a chip are more efficient in transferring the heat to the ambience than the center of the chip. Second, chip utilization and performance are increased. Due to fragmentation, a new workload may be prevented from being allocated to contiguous cores, resulting in increased communication latency.

Maintaining the shape of the workload sets and preferentially mapping workloads to the outer cores alleviates the impact of fragmentation. Third, the utilization of links located on the edge of the chip is increased. Using traditional adaptive routing algorithms, messages tend to be routed through the center of the chip, resulting in significant congestion in that area [1]. With co-scheduling, workloads at outer cores may take advantage of side links within a chip. However, the average packet travelling distance will be increased after applying this co-scheduling technique. Fortunately, as we will show in next section, this drawback could be largely compensated by photonic networks due to the inherent high-speed and low-power nature of light. This makes our thermal/congestion-aware co-scheduling highly suitable for photonic networks.
Experimental Setup

In our study, we used Simics and GEMS simulation frameworks. Simics [41] provides a full-system, functional simulation framework whereas GEMS [42] provides a cycle-accurate timing simulator which models timing of multiprocessor memory systems. We used GARNET [37], which is a detailed cycle-accurate on-chip network model incorporated inside the GEMS framework, and extended it to support the proposed Aurora architecture. All simulations are performed on the 5×6 network. Table 2-1 summarizes the parameters of the simulated chip. We evaluated our techniques using a set of representative synthetic traffic patterns (i.e. uniform random, transpose, bit-complement and tornado [1]). Garnet generates traffic during a period of 1 million cycles (including 1K warm-up cycles). We assume that the E/O and O/E conversions are carried out at 640 Gbps (64 wavelengths, 10 Gbps each). Since the time needed to
establish an optical path is quite costly, especially under heavily loaded situations, the size of messages in photonic networks should be larger than those in traditional electrical networks to increase network performance. Nevertheless, extraordinarily large messages may block the network due to the lack of virtual channels and buffers in the photonic network. Thus, in our simulations, we set the maximum message size to 13312 bits, which is a trade-off between link efficiency and blocking probability. Consequently, maximum message transmission time on the photonic network is 208 cycles.

We simulated a 30-core processor with a shared 2 M Byte cache to generate the temperature profiles. We assume 3GHz frequency and a 45 nm technology with a supply voltage of 1.2 V. Each core is 4 mm × 4 mm for a total chip area of 20 mm × 24 mm. The baseline processor and memory architecture are summarized in Table 2-2. To evaluate our proposed techniques, we modeled all of the above components. To evaluate the efficiency of our proposed schemes under a wide range of temperature profiles, we constructed various thermal scenarios using the method. Table 2-3 summarizes the characteristics of thermal scenarios used to evaluate our techniques. Figure 2-11 presents the thermal map of each generated scenario.
Table 2-1. Chip parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of cores</td>
<td>30 arranged as 5×6 in a folded torus</td>
</tr>
<tr>
<td>Convection resistance</td>
<td>0.07 K/W</td>
</tr>
<tr>
<td>Convection capacitance</td>
<td>240.4 J/K</td>
</tr>
<tr>
<td>Area of demodulator/photodetector set</td>
<td>660 µm × 40 µm</td>
</tr>
<tr>
<td>Area of switch</td>
<td>70 µm × 70 µm</td>
</tr>
<tr>
<td>Number of resonators in demodulator/photodetector set</td>
<td>3840</td>
</tr>
<tr>
<td>Number of resonators in switches</td>
<td>720</td>
</tr>
<tr>
<td>Number of resonators in temperature-detecting units</td>
<td>120</td>
</tr>
<tr>
<td>Total number of ring resonators</td>
<td>4680</td>
</tr>
</tbody>
</table>

Table 2-2. Baseline machine parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>4-wide fetch/issue/commit</td>
</tr>
<tr>
<td>IQ, ROB, LSQ</td>
<td>64 Issue Queue, 96 ROB entries, 48 LSQ entries</td>
</tr>
<tr>
<td>TLB</td>
<td>128 entries(ITLB), 256 entries(DTLB), 4-way, 200 cycle</td>
</tr>
<tr>
<td>Branch Pred.</td>
<td>2 K entries Gshare, 10-bit global history, 32 entries RAS</td>
</tr>
<tr>
<td>I/D L1 Cache</td>
<td>64 KB, 4-way, 64 Byte/line, 2 ports, 3 cycle</td>
</tr>
<tr>
<td>Integer ALU</td>
<td>4 I-ALU, 2 I-MUL/DIV, 2 Load/Store</td>
</tr>
<tr>
<td>FP ALU</td>
<td>2 FP-ALU, 2 FP-MUL/DIV/SQRT</td>
</tr>
<tr>
<td>L2 Cache</td>
<td>Private 512K, 4-way, 128 Byte/line, 12 cycle</td>
</tr>
</tbody>
</table>

**Evaluation Results**

In this study, we evaluate the reliability and performance characteristics of the proposed Aurora architecture using different architecture- and OS-level thermal management schemes. Table 2-4 summarizes the evaluated techniques. We assume that the circuit-level technique is always activated to achieve thermal stability on small range temperature variations.
Figure 2-11. Thermal maps of the generated scenarios
Table 2-3. Thermal scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Synopsis</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1 Center block</td>
<td>A block of hot cores in the center force traffic to use the edges as paths, Figure 2-9(a)</td>
</tr>
<tr>
<td>S2 Corner block</td>
<td>More than half of the hot cores are located at the corner, Figure 2-9(b)</td>
</tr>
<tr>
<td>S3 Winding path</td>
<td>Hot regions force traffic to follow a winding path to destination, Figure 2-9(c)</td>
</tr>
<tr>
<td>S4 Narrow strait</td>
<td>Hot regions on both sides, dividing the processor into two sections, Figure 2-9(d)</td>
</tr>
<tr>
<td>S5 Random 1</td>
<td>Randomly generated hot regions, Figure 2-9(e)</td>
</tr>
<tr>
<td>S6 Random 2</td>
<td>Randomly generated hot regions, Figure 2-9(f)</td>
</tr>
</tbody>
</table>

Table 2-4. The evaluated techniques

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Routing Algorithm</th>
<th>OS-level Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD (Baseline)</td>
<td>Shortest-distance</td>
<td>No</td>
</tr>
<tr>
<td>SD+OS</td>
<td>Shortest-distance</td>
<td>Yes</td>
</tr>
<tr>
<td>SPF</td>
<td>Shortest-Path First</td>
<td>No</td>
</tr>
<tr>
<td>TF</td>
<td>Temperature First</td>
<td>No</td>
</tr>
<tr>
<td>SPF+OS</td>
<td>Shortest-Path First</td>
<td>Yes</td>
</tr>
<tr>
<td>TF+OS</td>
<td>Temperature First</td>
<td>Yes</td>
</tr>
</tbody>
</table>

NoC Latency

Figure 2-12 shows the average latency of the simulated photonic NoCs under four traffic patterns (uniform random, transpose, bit-complement and tornado) and various thermal management techniques. As described in previous section, the architecture-level techniques decrease the average BER but can introduce additional congestion since messages tend to traverse through cool regions. In general, we observed that the average network latency increases by 5-50% compared to the baseline cases. In addition, we found that the worst performance occurs when a hot region occupies a significant fraction of the chip area and leaves only narrow straits for message routing, as shown in scenarios 1 and 4 in Figure 2-11. The network latency in these cases increases by 1 ~ 4 times. The network latency of SPF falls in between those of the TF
and the baseline, since the SPF takes both the path length and BER into consideration. Figure 2-10 further shows that in most cases, network latency can be reduced by combing the OS-level technique with architecture-level technique. Compared with the SPF and TF cases, the average latency reductions of SPF+OS and TF+OS are 6% and 27% respectively. This is because our proposed OS-level technique diminishes the high temperature regions within the chip and hence provides additional routing alternatives.

Note that if we retransmit the messages which are ruined by errors, it may incur additional latency overhead. In this case, the latency of the baseline case would increase significantly more than our proposed techniques. This is because our proposed architecture- and OS-level techniques dramatically reduce the MER (as will be shown in the next subsection), thus reducing the message retransmission probability.

**BER and MER**

Figure 2-13 shows the average BER for our simulated photonic NoC using various thermal management techniques. The first three bars in each group represent the BER after applying architecture-level techniques (i.e. SD, SPF, and TF). The next three bars show the BER after applying both architecture- and OS-level techniques, (i.e. SD+OS, SPF+OS, and TF+OS). As indicated, BER is reduced by 10% and 49% after applying the architecture-level technique (SPF and TF) alone. On average, combining the architecture and OS-level techniques can further reduce BER by 93% and 92% for SPF+OS and TF+OS respectively.
We observed that in Figure 2-13, the average BER of the SD (baseline) case in scenario 1 is about $1 \times 10^{-3}$, while it is $2 \times 10^{-4}$ for scenario 2. This indicates that the average BER depends on the thermal map of the chip. The high BER in scenario 1 is attributed to the routes traversing the high-temperature region in the center of the chip. After applying the architecture-level technique to scenario 1, BER is significantly
reduced compared to scenario 2 since more messages are rerouted through the cooler paths. Furthermore, applying the OS-level technique provides more cool paths through the center than scenario 2 by relocating high temperature regions to the outer cores.

Among the SD, TF and SPF cases, TF achieves the best BER performance followed by SPF. This is because TF depends upon the heat distribution in the network, and thus tends to route messages through the regions with least MER; whereas SPF uses temperature information as well as the number of hops from source to destination. There is a tradeoff between delay and error rate improvement shown by SPF and TF algorithms. For cases with high congestion, TF shows more improvement in BER 60%-80% at the expense of increasing network delay. The above observations are also valid for TF+OS and SPF+OS cases.

We also recorded the average MER which indicates the ratio of messages that fail in delivery to total messages as shown in Figure 2-14. SPF and TF show 6% and 30% improvement compared to the baseline case, whereas SPF+OS and TF+OS can achieve 76% and 84% improvement on average in our simulation scenarios.

**Power Consumption**

Total power consumption in Aurora is mainly attributed to:

1. Heat generated by the DC bias current (direct localized heating) for each ring resonator
2. Energy consumed by the network for the transmission of messages

The static energy of the network is also converted to a per-bit scale and integrated into part 2 as in [43].

Compared to conventional metal strip heaters, maintaining the operating temperature by varying the DC bias current consumes about 50% less energy [25] due
to direct localized heating. The metal strip heaters are implemented in a metal layer atop the photonic layer. Due to the top cladding oxide between the metal layer and the waveguides, the metal strips cannot directly heat the resonators and thus are power-inefficient. In contrast, the DC bias current provides localized heating in the PIN junction surrounding the resonator and thus is more efficient. In our simulation, we employ one metal strip heater for each ring resonator. We assume that the size of the heater is 2 µm × 2 µm × 5 µm and its surface heat release rate is 1 mW/µm³. The thickness of the top cladding oxide is assumed to be 1 µm.

We also modeled the power consumption for both the electrical and photonic networks. For the electrical network, the dynamic power consumed due to data transmission is obtained through ORION [44]. The total power consumed on our 2D 5×6 mesh electrical network is calculated as in [43]. For the photonic network, the resonators consume energy when free carriers are injecting into the rings. The in-plane Poly-Si energy consumed is 100 fJ/bit [45]. Assuming advanced driver circuits with poly-Si carrier lifetimes of 0.1-1 ns and modulation speed of 10 Gbps, the power consumed by each modulator is approximately 200 fJ/bit [45]. The energy consumption is also related to link MER since retransmission of messages which fail in delivery will cost additional energy.

For six thermal scenarios, we compare the power consumption of a network using conventional metal strip heaters to a network using the DC bias control method, as shown in Figure 2-15. The DC bias current driven heater is about twice as power efficient as the conventional metal strip heaters. Since applying our architecture and OS-level techniques reduce MER, message retransmission ratio decreases which
further reduces the power consumption of Aurora. On average, the DC bias method consumes 33% less total power than the metal strip heater.

Moreover, by leveraging the architecture level and the OS level co-scheduling techniques, Aurora could further save another 4% power (TF+OS scheme) because of decreasing message retransmissions.
Figure 2-13. Average BER of the network
Figure 2-14. Average MER of the network

Figure 2-15. Comparison of network power consumption. The “Heater” scheme is a similar hybrid network using conventional metal strip heaters to compensate for temperature variations in resonators.
Motivation

Unlike electrical NoCs, static power dominates the overall photonic NoC power budget (e.g. 75% reported in [14]). Worse, the energy conversion efficiency of the laser sources is low (e.g. 50% reported in [46]), which further aggravates the total power loss. While the static power of photonic NoCs is fixed owing to the predetermined network design and the constant laser source injection, the network traffic manifests substantial runtime variation [47, 48]. When the traffic is below the provisioned network bandwidth, the NoCs will manifest a significant static power overhead. Furthermore, a large portion of the laser power is lost when traversing through the ring resonators along the traffic path. The optical switches in photonic NoCs contain modulators, photo-detectors, and turn resonators; all of which are made from ring resonators. These ring resonators act as band-pass filters, causing pass-band attenuation and power loss on the traversing optical signals. For instance, the intermediate ring resonators within optical switches [13, 43] can cause 40% optical power loss in an 8×8 mesh network. In addition, these ring resonators have to be thermally tuned to function, which incurs significant heating power. The above static power overheads make deploying on-chip optical components (e.g. ring resonators and waveguides) power-expensive, and demand a good utilization of the provisioned network resources. Moreover, due to the lack of optical logic gates and storage, existing photonic NoC routing approaches are either static or relying on additional components (such as duplicated optical networks [49] and electrical buffers [13]) to achieve adaptivity. These methodologies, however, fail to exploit existing
photonic network resource effectively and increase the overall NoC latency and power due to the inclusion of auxiliary components.

In summary, the emergence of photonic NoCs calls for a new set of techniques to optimize their energy efficiency. To this end, we propose ESPN, an energy-star photonic NoC architecture. Specifically, we make the following contributions:

1. We propose a dynamic photonic NoC design that allows network resources to adapt with run-time traffic characteristics. In our design, the network resources are partitioned and supplied with separate laser sources to enable dynamic network resource management strategies via traffic-aware bandwidth provisioning.

2. We propose a power-efficient router design (e.g. RapidEngy), which alleviates the impact of power loss on the traversing signals due to the intermediate modulator/photo-detector arrays.

3. We propose all-optical adaptive routing to accelerate data communication. Our adaptive routing leverages low-latency optical links to establish data paths and thus avoids introducing high latency and power hungry auxiliary routing components.

The Proposed Hierarchical Photonic NoC Architecture

An Overview of Hierarchical Photonic NoC Architecture

Figure 3-1 provides an overview of the proposed Hierarchical Photonic NoC architecture design. Hierarchical Photonic NoC is an architecture targeting future high-throughput systems, so our exploration and evaluation targets 22nm technology [50]. Hierarchical Photonic NoC consists of one multi-processor chip and two laser source chips connected by off-chip optical fibers and electrical wires on the Printed Circuit Board (PCB). The multi-processor chip consists of three vertically stacked dies using 3D packaging technology [17]. The processor & caches die contains processor cores, private L1/L2 caches and electrical routers. The control die, which operates as the interface between the processor & caches die and the optical die, integrates driving circuits, sense amplifiers, and control circuits for the optical components (e.g. the
ON/OFF switch of turn resonators and modulators/photo-detectors). The optical die, which integrates the waveguides and ring resonators, is connected to the control die using Through-Silicon Vias (TSVs) [51]. These optical components are built using CMOS-compatible monolithic integration to reduce cost [52].

Figure 3-1. An overview of ESPN architecture

ESPN employs a 2D mesh dynamic optical network which consists of several sub-networks to support traffic-aware dynamic network resource allocation through real-time tuning of the laser sources; each of the sub-network is supplied with individual external laser source. The external laser lights are provided by two VCSEL source chips [53] and coupled into the multi-processor chip via off-chip fibers. The laser lights are separately conducted to the waveguides in horizontal and vertical directions through on-chip splitters. A basic switch element of ESPN consists of an electrical router and an optical switch. The electrical router surrounds a processor core and locates on the processor &
cache die; the optical switch is located on the optical die. In addition, ESPN uses the RapidEngy router and all optical adaptive routing to further improve the energy efficiency of photonic NoCs.

**Dynamic Resource Allocation in Photonic Network**

ESPN achieves dynamic resource allocation by partitioning the interconnection network into multiple sub-networks. Each sub-network provides a fraction of the aggregated bandwidth and is driven by separate lasers. The bit widths (i.e. wavelengths) of the data channels are divided among the sub-networks. The sub-networks can be dynamically activated/deactivated based on the run-time bandwidth estimation. To further minimize the power consumption of an inactive sub-network, the driving circuitries, and heaters (we assume each heater is dedicated to one turn resonator or shared by one modulator/photo-detector array [15]), along with the photonic components are turned off.

![Diagram of VCSEL sources and organization](image)

Figure 3-2. The VCSEL sources (a). The organization of a VCSEL source (b). The organization of a VCSEL array

Unlike hierarchical electrical NoCs [54], ESPN leverages the characteristics of photonic components such as the controllable laser source to achieve low-overhead...
sub-network switching. ESPN employs two types of optical channels to connect each tile in the 2D mesh network in order to support all-adaptive routing. A data channel is used for data transmission; and a routing channel, which travels along the data channel, carries routing and control information. In this study, we investigate two sub-network partition techniques. In both techniques, the wavelengths of the data channels are evenly distributed among the sub-networks. The routing channels are either shared by all the sub-networks (called data channel splitting) or duplicated across the sub-networks (called full splitting).

Low-latency, high-density laser sources are required to facilitate the switching of sub-networks. In this study, we employ VCSEL sources [55], each of which is a perpendicular emission type of semiconductor laser diode, to power photonic links for fast sub-network activation/deactivation operations. VCSEL has the advantages of low-cost mass production and can achieve high integration density due to its vertical nature.

The organization of a VCSEL source is illustrated in Figure 3-2(a). The laser source consists of two Distributed Bragg Reflector (DBR) mirrors [56, 57] with an active region that contains one or more quantum wells for the laser light generation in between. When a voltage is applied between P-DBR and N-DBR, the generated current flow drives the p-n junction to emit laser light from the bottom of the chip. The VCSEL switching is achieved by applying and removing the forward operating voltage and the DC bias current between the two reflector mirrors [57].

The VCSEL source can operate at a high speed. Its switching delay is mainly determined by the “turn-on” latency [58], which is the delay in the emission of light from the laser after applying the driving current [17]. This delay is typically the time for the
driving current to fill the electron up to the laser emission threshold level, which varies
from 10 to 100 ps, depending on the type of driving circuitry [58]. The dissipated power
is mainly determined by the average current, which is negligible during this short period.
Multiple VCSEL sources can be organized as wavelength-division arrays, which
multiplex multiple optical signals within a single optical waveguide by leveraging a
microlens array and an external focusing lens as shown in Figure 3-2(b). Our design
integrates multiple VCSEL arrays within two laser source chips. These VCSEL arrays
are supplied to different sub-networks and controlled individually. Assuming that (1)
each laser source chip contains four VCSEL arrays used for implementing four 2D
mesh sub-networks, (2) each VCSEL array consists of 64 wavelength division
multiplexing (WDM) laser sources, (3) the VCSEL center-to-center space is 250 um [53]
and the laser sources are organized as a 32×8 matrix, the dimensions of a VCSEL array
chip is 0.8cm×0.2cm×0.03cm.

Both the optical switch and the network interface need to be modified to support
the dynamically partitioned photonic network. In ESPN, each electrical router is
connected to a processor with its private L1/L2 caches, as shown in Figure 3-3(a). The
routers and processors are located on the processor & cache die, atop the
corresponding optical switches. The electrical router and the optical switch
communicate with each other through TSVs and optical/electrical signal converters on
the control die. Each electrical router contains four input and four output interfaces.
Each output interface is shared by two output directions; while an input interface is
dedicated to an input direction to support all-optical adaptive routing (detailed in next
section).
**Network Interface:** As shown in Figure 3-3(b), the output interface uses a message First In, First Out (FIFO) to distribute incoming messages (from local processor) among the active sub-networks. Message transmitters are deployed to support optical adaptive routing, each of which serves two sub-networks in different output directions. A message transmitter is available when not transmitting data and its connected sub-networks are active. The sub-network selector assigns messages to the available transmitters in a round-robin fashion. When deactivating sub-network(s), ESPN employs two mechanisms to avoid destroying in-flight messages. First, the laser sources of the deactivated sub-network(s) remain on for $T_{\text{MAX}}$ cycles in order to complete the transmission of traversing messages in network, where $T_{\text{MAX}}$ is the round-trip transmission cycle between two nodes with the maximal distance (We evaluated $T_{\text{MAX}} = 3$ in 8×8 network in this study). Second, a message in message transmitter will not be eliminated immediately after being sent to the network owing to potential re-transmission. The message is destroyed only after reaching its destination. Similar to the output interface, the input interface employs a central FIFO to buffer traffic from active sub-networks.
Central Controller: The state of sub-networks is determined by the network pressure, which indicates the ratio of communication demand and available bandwidth.

In our design, the network pressure of each output interface is estimated as $2^{\left\lfloor \frac{n}{n_0/8} \right\rfloor}$, where $n$ is its FIFO data count and $n_0$ is the FIFO capacity. This allows our design to better adapt with bursty and un-predictable NoC traffic such as the hotspots. The output interfaces periodically send their network pressure to the central controller. ESPN employs an electrical H-tree network composed of differential transmission lines (T-line) [51] located on the control die to collect network pressure and control the laser source. This H-tree network is similar to the common H-tree clock distribution network but with reverse signal flow and light load. The central controller aggregates the network pressure, and then identifies the number of active sub-networks (via the Network Status Lookup Table), expressed as $\left\{ \begin{array}{ll} m, & p > p_{\text{max}} \\ mp_{\text{max}}, & p \leq p_{\text{max}} \end{array} \right.$, where $m$ is the total number of sub-
networks, \( p \) is the current network pressure and \( P_{\text{max}} \) is the threshold to activate all sub-networks, as shown in Figure 3-3(c). The laser controller generates the laser source control information based on the required number of active sub-networks.

When the network pressure fluctuates, the subnetwork status should be adjusted correspondingly. The latency of sub-network activation/deactivation consists of the following components: delay in the H-tree network, delay in the central controller, delay from the central controller to laser, and the laser operation delay. Assuming a 3cm × 3cm multi-processor chip, the H-tree differential T-line delay in 22nm is estimated as 8.04 ps/mm [52]. So the H-tree transmission delay is estimated to be 240 ps. The delay in the central controller is assumed as 3 cycles (600 ps under 5GHz clock). The delay to the laser source chips is determined by the distance between the laser source chip and the processor chip and is assumed as 200 ps. The laser operation delay is 10 ps to 100 ps and we use 50 ps in our simulations.

**RapidEngy Optical Switch**

Apart from dynamic network resource allocation, ESPN employs low-loss optical switches (i.e. RapidEngy) to further optimize energy efficiency. Note that each optical switch in 2D photonic mesh network requires five pairs of ports to connect the four adjacent switches and the local node. Although a 5×5 switch can be implemented using crossbar-based design in the electrical domain, its photonic implementation is quite challenging. To implement the optical switch, [13] adopts a 4×4 optical crossbar to direct messages to different directions, as shown in Figure 3-4(a). The crossbar connects different input ports to different output ports by electrically tuning the turn resonators to ON or OFF state (shown in Figure 3-4(c)). Additionally, the optical switch adopts four modulator and four photo-detector arrays as the interface between the local node and
the beneath crossbar. Each modulator array (EO, WO, SO, NO) modulates messages from the local node to one output direction and each photo-detector array (EI, WI, SI, NI) demodulates messages from one input direction to the local node.

![Diagram of optical switches](image)

Figure 3-4. The design of optical switches (a) A 4×4 crossbar switch with dedicated optical modulator/demodulator array [59] (b) RapidEngy switch (c) The required turn resonators states for switch in Figure 3-4(a) (ON: the turn resonator is on-resonance and signal turns. OFF: the turn resonator is off-resonance and signal does not turn) (d) The required turn resonators states for RapidEngy

Unfortunately, the modulator and photo-detector arrays incur severe pass-band attenuation to the traversing messages. For example, the messages from West to East are affected by the WI photo-detector array and EO modulator array as shown in Figure 3-4(a). In RapidEngy, we propose to rearrange the modulator and photo-detector arrays
to avoid affecting traversing messages, as illustrated in Figure 3-4(b). Now the same messages no longer pass any modulator array and photo-detector array. Similar to the 4×4 crossbar switch, the messages from/to the local node are modulated/demodulated at corresponding modulator/photo-detector array. For example, Figure 3-4(b) shows a message from South converted by the SI photo-detector array and a message to West modulated by the WO modulator array.

Due to the relocation of the modulators and photo-detectors, RapidEngy introduces additional resonators and waveguide crossings, resulting in signal loss. For instance, the traversing message from West to East experiences two ON resonators in RapidEngy (resonators 9 and 12, as shown in Figures 4(b) and (d)) compared to one ON resonator in the 4×4 switch (resonator 6, as shown in Figures 3-4(a) and (c)). Also the number of traversing waveguide crossings is increased. Nevertheless, the additional turn resonators and waveguide crossings exert much less impact on the signal compared with the modulators/photo-detectors. Our simulation results show that in a data channel splitting network consisting of 4 sub-networks, the power loss in RapidEngy is 1.61 dB less than that in the switch shown in Figure 3-4(a).

**All Optical Adaptive Routing**

Although our dynamic resource allocation reduces network power considerably, it could incur performance degradation due to the reduced network bandwidth. Adaptive routing achieves load balance and could compensate for the reduced network bandwidth; nevertheless existing optical-based routing algorithms [6] are mostly static due to the inherent buffer-less nature of photonic NoCs. The use of auxiliary electrical network routing [16] introduces additional hardware and performance overhead. To overcome these limitations, we propose an all-optical adaptive routing scheme by
leveraging the low-latency optical network to route messages. To our knowledge, this is the first work that explores optical adaptive routing in mesh network without relying on the high-power electrical components.

Our proposed all-optical adaptive routing first establishes an optical circuit switch path between the source and destination node and then transmits messages via that path. To achieve good tradeoff between routing complexity and network performance, we adopt minimal adaptive shortest-distance routing algorithm [1], which searches for the optimal routing path among all the shortest distance paths. The path establishment consists of the following scenarios:

A. The source node sends request signals along the shortest path(s) to check their availability. While proceeding, a request signal reserves photonic links along the path for the upcoming message.

B. In case the request signal encounters a blocked link and fails to reach the destination, the signal carrying the information of the blocked position is transmitted back along its reverse path and releases the reserved links.

C. If the request signal reaches its destination, the signal that indicates successful link acquisition is transmitted back to the source node. The message transmission then starts along the reserved links.

D. If all the request signals are blocked and fail to reach the destination, the source node retries the next path.

Each scenario is described below in detail.

**A. The Traversal of Request Signal:** The request signal travels along the request channel, which is part of the routing channel. The routing channel also contains the response channel driven by reverse laser light. The request channel consists of an even number of waveguides (two waveguides are shown in Figure 3-5). The wavelengths of the request signal are organized as two groups: Path Hops (PHOP) and Request Hops (RHOP). The PHOP stripes across the two waveguides and is divided into several
sections (PHOP₁ to PHOPₙ), which sequentially records the routing information in n hops. Each PHOPₓ consists of four optical bits (wavelengths), which represent four possible turn directions. At each switch, if the downstream switch is available, the active PHOP bit drives the turn resonator of the corresponding direction to route the request signal and upcoming message.

Figure 3-5 illustrates a case in which the request signal traverses through three hops (go straight, turn left, and finally received by the local node). Each switch snoops on the PHOP₁ wavelengths to detect the turn direction of request signal. PHOP₁ is eliminated after the signal is routed through the current switch. So PHOP₂ needs to be moved to PHOP₁ to be detected by the next hop. As a result, all PHOPᵢ needs to be moved to PHOPᵢ₋₁. To achieve this, we apply either physical shift or frequency translation mechanism proposed in [13] to one waveguide, which respectively moves the optical bits to the same or different wavelengths at another waveguide. For example, Figure 3-5 shows that in hop 1, PHOP₃,₅ are frequency translated to PHOP₂, ₄ (different wavelengths) while PHOP₂,₄,₆ are physically shifted to PHOP₁, 3, ₅ (the same wavelengths). Our proposed design implements the all-optical adaptive routing by leveraging the RHOP signals and response channel rather than static X-Y routing in [1].

B. Path Availability Identification: After sending the request signal, the source node needs to be notified on path availability. If the requested path is currently blocked, the source node will be notified about the block position and then plan an alternative path, which is achieved using RHOP and REPLY. The RHOP is part of the request signal and is duplicated across the request waveguides. An active RHOPᵢ indicates that the current distance to the destination node is i-1 hops. RHOP is decreased at each hop
by frequency translating the activated bit to the next one. For example, in Figure 3-5, in each hop the RHOP$_1$ is eliminated and RHOP$_{2-6}$ is frequency translated to RHOP$_{1-5}$ in the other waveguide. When the request signal encounters a blocked link or reaches its destination, the REPLY is modulated by physical shifting the RHOP signal from the request waveguides to the response channel and then transmitted back to the source node. The source node examines the REPLY to decide whether the path has been successfully established.
Figure 3-5. The request signal in routing examination and forwarding
Figure 3-6 illustrates an example in the context of a 4×2 2D mesh topology. In this example, node 1 needs to send a message to node 7, while the link between nodes 3 and 7 is currently blocked. The source node simultaneously generates up to two request signals (on a minimal adaptive routing basis along both coordinates) to accelerate link establishment. In this example, node 1 generates two request signals to its South (request A) and East (request B) output ports. The request A is frequency translated from RHOP$_3$ to RHOP$_2$ at node 2, indicating the message proceeds. Due to the blocked link, node 3 modulates the REPLY by physical shifting RHOP to response channel and then eliminates request A. The request B successfully reaches its destination, node 7. Thus node 1 receives two REPLYs from the two response channels and identifies the REPLY from the East port as a successful link establishment. During the transmission of request signal, in case that several request signals contend for one output port, the highest priority will be given to the one that is closest to its destination by activating the corresponding RHOP signals. Such distance-class ordering mechanism ensures a deadlock-free network [1].

**C. Data Transmission:** The source node starts transmitting data along the reserved links once the REPLY indicates that a path has been established. When the data transmission completes, an optical pulse traverses back along the path to tear down the links.

**D. Alternative Path Selection:** In case that all the request signals are blocked and fail to reach the destination, the source node will plan an alternative path. The source node first examines the returned REPLY and locates the blocked position. It then retries the next path using the unblocked links plus a detour for the blocked links.
That path is still one of the shortest-distance paths between the source and the destination. For example, in Figure 3-6, if both requests fail to reach the destination, a possible alternative to request A is links 1-2, 2-6 and 6-7.
Figure 3-6. An example of blocked link in adaptive routing (activated RHOP bit highlighted)
Experimental Methodology

Machine Configuration and Workloads

Our evaluation is performed using a simulator developed from Simics/GEMS [41] framework. We used GARNET [37], a detailed cycle-accurate on-chip network model incorporated within the GEMS framework, and extended it to support our proposed optical NoC architecture. All simulations are performed on an 8×8 mesh network as listed in Table 3-1. We explore different sub-network partition schemes while keeping the data channel bandwidth, i.e. the product of the number of wavelengths per waveguide and the number of bundled waveguides per data channel, constant. The baseline case, “E-deterministic”, includes an auxiliary electrical network configured as in [16]. Among all studied design alternatives, the “O-Deterministic” uses the optical channels and X-Y static routing to establish path, similar to [13]. The “O-Adaptive” adopts all-optical adaptive routing. The “ESPN (F-m-n)” and “ESPN (P-m-n)” are fully splitting and data channel splitting ESPN respectively. We also compare the performances of the conventional four-port switch shown in Figure 3-4(a) (with the prefix “Sub-“) to RapidEngy (without the prefix “Sub-“). Our modeled system consists of 64 processing cores with private L1 and L2 caches fabricated using 22nm processing technology. We assume the interconnect network clock is 5GHz with a supply voltage of 0.5V.
Table 3-1. The evaluated NoC design

<table>
<thead>
<tr>
<th>NoC Design</th>
<th>Link Establishment</th>
<th>Routing Scheme</th>
<th>Network Division</th>
<th>Wavelengths</th>
<th>Data Bus Width (Bytes)</th>
<th>Number of Sub-networks</th>
<th>Optical Switch Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-Deterministic</td>
<td>Electrical</td>
<td>X-Y Static</td>
<td>-</td>
<td>64</td>
<td>8</td>
<td>1</td>
<td>Four-Port</td>
</tr>
<tr>
<td>O-Deterministic</td>
<td>Optical</td>
<td>X-Y Static</td>
<td>-</td>
<td>64</td>
<td>8</td>
<td>1</td>
<td>Four-port</td>
</tr>
<tr>
<td>Sub-O-Adaptive</td>
<td>Optical</td>
<td>Adaptive</td>
<td>-</td>
<td>64</td>
<td>8</td>
<td>1</td>
<td>Four-Port</td>
</tr>
<tr>
<td>O-Adaptive</td>
<td>Optical</td>
<td>Adaptive</td>
<td>-</td>
<td>64</td>
<td>8</td>
<td>1</td>
<td>RapidEngy</td>
</tr>
<tr>
<td>Sub-ESPN (F-m-n)</td>
<td>Optical</td>
<td>Adaptive</td>
<td>Fully Splitting</td>
<td>64 / m</td>
<td>8 / n</td>
<td>m × n</td>
<td>Four-Port</td>
</tr>
<tr>
<td>ESPN (F-m-n)</td>
<td>Optical</td>
<td>Adaptive</td>
<td>Fully Splitting</td>
<td>64 / m</td>
<td>8 / n</td>
<td>m × n</td>
<td>RapidEngy</td>
</tr>
<tr>
<td>Sub-ESPN (P-m-n)</td>
<td>Optical</td>
<td>Adaptive</td>
<td>Data Splitting</td>
<td>64 / m</td>
<td>8 / n</td>
<td>m × n</td>
<td>Four-Port</td>
</tr>
<tr>
<td>ESPN (P-m-n)</td>
<td>Optical</td>
<td>Adaptive</td>
<td>Data Splitting</td>
<td>64 / m</td>
<td>8 / n</td>
<td>m × n</td>
<td>RapidEngy</td>
</tr>
</tbody>
</table>

We used 128-state MMP synthetic traffic (i.e. Bit-compliment, Tornado, Bit-reverse, and Random) in our simulations. The MMP synthetic traffic generates a time-varying NoC utilization by modulating the rate of a Bernoulli injection process on the states of a Markov chain [6]. The injected messages in MMP synthetic traffic are 64 bytes (the cache line size) and 8 bytes (the invalidation message size) respectively. In addition to synthetic traffic, we used real-world workloads PARSEC [60] and SPLASH-2 [61]. We run the PARSEC and SPLASH-2 benchmarks on top of Simics/GEMS and extract the network traffic traces. In order to preserve the traffic characteristic of different benchmarks while stressing the network more than the originally extracted traces, we adopted the evaluation methodology used in [47]. We normalized the PARSEC and SPLASH-2 traffic traces of all the benchmarks to accommodate the bandwidth of the optical network by setting the normalized average traffic rate to 0.4 times the network bandwidth. This scaling maintains the unbalanced nature of the traffic load, and stresses the network more than the real traffic load.
**Power Estimation Methodology**

We used the statistics reported in [62, 63] for the optical network power estimation. The electrical energy coupling efficiency of the laser source ranges from 30% [63] to 50% [62, 64]. We used the median value of 40% in this study. Another important factor of the total power consumption is the required optical detection power for photo-detectors, which is related to the expected Bit Error Rate (BER). We adopted BER of $10^{-15}$ in our study and [65] shows that each photo-detector requires at least 5 μW power under 5Gb/s modulation rate. By default, all turn resonators are set to OFF state and tuning energy is required when switching to ON state [3]. This energy is assumed to be 100 fJ/bit [45]. Besides, the power consumed by each modulator is approximately 200 fJ/bit using advanced driver circuits with poly-Si carrier lifetimes of 0.1-1 ns and modulation speed of 5 Gb/s [45].

In a typical photonic NoC with auxiliary electrical network [16], the total power consumption is the sum of power dissipated by both optical and electrical networks. The power consumption of the electrical network is modeled based on [22, 44], which assumes the energy required to transmit one bit under 22nm technology is 0.83 pJ plus 0.34 pJ/mm link power.

**Evaluation**

In this section, we explore the design spaces of ESPN and evaluate the performance and power benefits of the proposed techniques.

**The Optimal Network Power-Latency Product (PLP)**

The threshold to activate all sub-networks ($p_{max}$) determines the tradeoff between network power and latency. We measure the normalized network Power-Latency
Product (PLP) metric to determine the optimal $p_{\text{max}}$. As $p_{\text{max}}$ increases, fewer sub-networks are activated, which results in increased network latency and PLP. In contrast, reducing $p_{\text{max}}$ increases network power and PLP.

Figure 3-7. The power-latency product (PLP) of different networks (a) ESPN (P) (b) ESPN (F) (the average of four synthetic traffic patterns)

Figure 3-7 shows the average network PLP of ESPN (P) and ESPN (F) on synthetic traffic patterns with different injection rates. We observe that the optimal $p_{\text{max}}$ varies with network configurations. For example, the optimal $p_{\text{max}}$ on ESPN (P-1-2) and ESPN (P-2-1) is 512 while the optimal $p_{\text{max}}$ on ESPN (P-2-2) is 640. We also observe that increasing traffic injection rate increases PLP by up to 20 times (e.g. as injection rate increases from 0.35 to 0.40) due to severe network congestion. The optimal $p_{\text{max}}$ is insensitive to the injection rate and remains stable in most cases. In this study, we choose 512 for ESPN (F-1-2), ESPN (F-2-1), ESPN (P-1-2), ESPN (P-2-1), and 640 for ESPN (F-2-2) and ESPN (P-2-2). We apply the same $p_{\text{max}}$ for Sub-ESPN( ) and ESPN( ) since the network performance is not sensitive to switch architecture.
Network Performance

Our proposed all-optical adaptive routing reduces both the path request latency and the number of attempts to establish a path. In this section, we evaluate the network performance from these two aspects as well as the overall network latency.

The Path Request Latency

In all-optical adaptive routing the path request latency is crucial since the source node may attempt to establish a link multiple times before succeeding. This latency is characterized by the source resonator modulation latency, interim resonator drive latency, destination resonator modulation latency, and optical link latency. For each hop, the request signals fall into one of the three cases: (a) frequency translated and then forwarded to the next hop (passed), (b) transmitted back to the source node through the response channel (blocked), or (c) received by the local node (received). In case (a), the request signal needs to be received and driven to the corresponding resonator, the same as in case (b), except that the to-be-driven resonator is at the response channel rather than the request channel. In case (c), the request signal drives resonators to eliminate itself. In all cases, the latency is determined by three factors: (1) the time to receive PHOP and RHOP signals from the request channel, (2) the latency of performing a single-level CMOS logic to establish a path, and (3) the time for driving the physically shifted and frequency translated resonators. We use the latency parameters from [13, 35, 50, 59] to estimate the request attempt round-trip delay. The optical network, including resonators and peripheral circuitry, operates at 5 GHz.
The Path Establishment Attempts

Compared with deterministic routing, the all-optical adaptive routing reduces the path establishment attempts by choosing alternative paths, and therefore reduces the average network latency. We compare the number of path establishment attempts with different routing algorithms using synthetic traffic patterns. Figure 3-7(a) shows the cumulative distribution of the results under low traffic (injection rate = 0.3). We observe that with all-optical adaptive routing, 87% paths are established within 3 attempts, compared to 81% using deterministic routing.

![Cumulative Distribution](image)

Figure 3-8. The number of path establishment attempts under (a) light traffic (b) heavy traffic (the average of four synthetic traffic patterns)

In a congested network, the all-optical adaptive routing exhibits better efficiency. As shown in Figure 3-8 (b), under heavy traffic and with O-adaptive, 75% paths can be established within 3 establishment attempts, compared to 64% using deterministic routing. By dividing the network into multiple sub-networks and supplying with dedicated routing channels, ESPN (F-1-2) and ESPN (F-2-2) further increase this ratio to 77% and 81%.
Network Latency

Figure 3-9 shows the network latency under 128-state MMP synthetic traffic. Since the switch architecture does not affect network performance, Sub-ESPN exhibits the same performance as ESPN. E-deterministic and O-deterministic exhibit the worst performance owing to the deterministic routing. O-adaptive and ESPN (P) benefit from adaptive routing and thus improve performance by 20%-25%. However, due to the reduced bandwidth and subnetwork switch delay, ESPN (P) incurs 1%-3% performance degradation compared to O-adaptive. ESPN (F) gains a 5%-10% performance improvement over ESPN (P) by deploying dedicated routing channels in each sub-network.

Figure 3-9. Network latency under 128-state MMP synthetic traffic (a) Bit-compliment (b) Tornado (c) Bit-reverse (d) Random
Power and Energy Efficiency

Synthetic Traffic Patterns

Figure 3-10 shows power breakdown of the investigated NoC design. We observe that among all the sub-network partitions, ESPN (P-2-2) shows the best power efficiency, which yields 50% savings on average compared to E-deterministic. When the number of wavelengths within each waveguide drops from 64 to 32, an additional 10% power saving is observed due to the alleviated optical coupling loss in modulators and photo-detectors. On ESPN (F), the power for the routing channels increases due to the deployment of dedicated resources in each sub-network. ESPN (F-2-2) reduces power by 46% compared to that of E-deterministic. We observe that ESPN consumes 26% less power than Sub-ESPN owing to alleviating the impact of modulators and photo-detectors along the traversing path. The power savings vary since the average number of hops that a message traverses in the 8×8 network varies with traffic patterns. Figure 3-10 also shows the energy consumed per message. As can be seen, ESPN (P-2-2) and ESPN (F-2-2) save 57% and 58% energy respectively compared to the baseline case. The above two network configurations have different performance and power characteristics but exhibit similar energy per message profile.
Figure 3-10. Power breakdown on synthetic traffic
PARSEC and SPLASH-2 Benchmarks

Figures 3-11 and 3-12 show the normalized NoC power and energy efficiency on SPLASH-2 and PARSEC workloads. ESPN (P-2-2) and ESPN (F-2-2) reduce NoC power by 51% and 48% compared to the baseline case (E-deterministic). In general, the proposed sub-network partition techniques achieve more power savings on benchmarks that manifest higher traffic fluctuation (e.g. fmm and canneal). This is because ESPN benefits from traffic fluctuation and is able to deactivate the unused sub-networks, thus decreasing the overall NoC power.

Besides the power-efficient architecture, our adaptive routing circuit switch reduces the application execution time and therefore further improves the energy efficiency. The best case occurs on canneal, where ESPN (P-2-2) saves 68% of the total energy. On average, ESPN (F-2-2) and ESPN (P-2-2) reduce the execution time by 22% and 16% compared to the baseline case, resulting in 60% and 59% of the total energy saving, respectively. We also observe that, similar to synthetic traffic, the energy consumption of ESPN (P-2-2) and ESPN (F-2-2) are very similar (less than 2%).
Figure 3-11. The normalized power consumption on SPLASH-2 and PARSEC Benchmarks

Figure 3-12. The normalized energy consumption on SPLASH-2 and PARSEC Benchmarks
CHAPTER 4
EXPLORING PHOTONIC INTERFACE FOR OFF-CHIP PHASE CHANGE MEMORY SYSTEMS

Motivation

Current computer systems pose challenges on memory energy conservation, especially on the energy reduction of main memory. For memory intensive applications, the main memory is one of the major power consumers [66, 67]. Recently, several non-volatile memory technologies (e.g. Phase Change Memories, or PCMs) have emerged as alternative to Dynamic random-access memory (DRAM) solutions by avoiding the power wall with low leakage cells.

At current technology nodes, the intensive current injection in PCM cells motivates energy-proportional design. Prior study [68] breaks the conventional memory ranks into multiple small ranks, namely “mini-rank” to preserve energy consumption. Its power reduction majorly comes from the reduced number of chips and narrower row buffers (i.e. sense amplifiers) [66] involved in each activation and precharge operation. Also, the memory background power is reduced owing to the better utilization of low power mode. Nevertheless, the mini-rank design is not naturally supported in electrical domain due to the potential InterSymbol Interference (ISI) problem caused by the load of multiple ranks per channel [69]. Zheng et al. [68] employs a dedicated interface chip between memory chips and the communication bus at the cost of power waste and additional data transmission cycles. Besides, the rank performance is constrained by the narrow connections between this chip and memory devices. Other solutions to ISI problem, e.g. the fully-buffered dual in-line memory module (DIMM) [70] and the “fly-by” topology [71], also introduce intolerant transmission latency [72] and thus not applicable.
In order to overcome the limitation of electrical mini-rank design, we propose OptiPCM, which is an extension to the legacy memory architecture that takes advantage of the recent advances in CMOS-compatible nano-scale silicon photonic integrated circuitry [73, 74]. In OptiPCM, the photonic channels connecting PCM arrays are built on the monolithically integrated silicon-photonic waveguides. Far beyond conventional electrical bus, the silicon photonic bus is able to load a large number of memory devices even under very high frequency provided enough injection power. Thus, the ISI effect which limits the electrical mini-rank performance is successfully removed by the photonic communication. Besides, OptiPCM is able to provide increased memory-level-parallelism to hide the long PCM access latency by taking advantage of the large number of devices per channel. Interestingly, unlike the predetermined electrical links, the optical paths can be easily reconfigured, which enables the traffic-aware bandwidth allocation.

State-of-the-art Double Data Rate (DDR) protocol does not provide support for non-volatile memory interface. Protocol such as Low-Power Double Data Rate Non-Volatile Memory (LPDDR2-NVM) [69, 75] provides support for PCM memory interfacing, but lowers overall performance as compared to DDRx. To recuperate such performance loss and to retain low power consumption, we use high-bandwidth, low-latency and low-power photonic communication links to connect PCM chips and follow LPDDR2-NVM based protocol.

In summary, the following are the contributions of this section:

1. We propose to apply photonic links to connect large number of independent PCM chips. Apart from the widely recognized high-bandwidth and low-power characteristic, our design takes full advantage of the high load capacity of optical
channels, which provides support for the mini-rank design and hides the long PCM access latency.

2. We apply the fixed channel division technology to amortize the rank-to-rank turnaround time, which is critical for OptiPCM design with wide bus and numerous ranks.

3. We introduce the dynamic channel division to compensate for the potential low channel utilization in fixed channel division technology. Dividing the channel based on the traffic utilizes the channel bandwidth more efficiently while amortizing the rank-to-rank turnaround penalty in heavy traffic.

Background

Phase-Change Random Access Memory

Currently, PCM is being considered as one of the most promising technologies for next generation non-volatile memory. The emerging PCM technology has many advantages such as random access, non-volatility, superior scalability, fast read cycle and manufacturing compatibility with existing CMOS process. PCMs differ from the DRAMs in the organization of the cell. Each PCM cell employs reversible phase change materials to store information. These materials are usually made using a chalcogenide alloy of germanium, antimony and tellurium (GeSbTe) called GST. Figure 4-1 shows the basic structure of a PCM cell, which consists of a standard NMOS transistor and a phase change device. PCMs leverage the differences between the two states in the electrical resistivity of GST to store information. In the amorphous state, the high resistance is used to represent a binary “0”, while in the crystalline state, low resistance denotes a “1”. The GST phase can be changed by heating a region of phase change material to a high temperature threshold using electrical-pulse generated Joule heat [76].
Figure 4-1. A single PCM cell
Figure 4-2. The organization of a rank of memory device

The Memory Devices Organization and LPDDR2 Protocol

Figure 4-2 shows the contemporary PCM chip organization. The organization of PCM is highly similar to that of DRAM. In this example, eight PCM chips are ganged together and work in lockstep to respond to the commands issued by the memory controller. The links used to communicate between the ranks and memory controller is called channels and the ganged chips form a rank.

We employ the channel following LPDDR2-NVM protocol in our study [69]. The original LPDDR2 is proposed to become the technology of choice for embedded and mobile applications thanks to its low-power characteristics, e.g. Malladi et al. [77] leverages LPDDR2 based DRAM in the data centers. LPDDR2 saves a significant proportion of power (over 50%) under similar device density and performance conditions. This saving majorly comes from the reduced working voltage (1.2V), the removal of certain circuits like the Delay-Lock Loops (DLLs), the advanced power management, and the shrunken pin count. LPDDR2-NVM officially provides support for non-volatile memory such as flash memory and PCM. In LPDDR2, the signals between
memory device and memory controller fall into four categories: the command signals, the address signals, the data signals, and the miscellaneous signals. The command and address signals are unidirectional and extend from memory controller to the memory. They provide the commands and bank/row/column address to the memory chips. The data bus is a bidirectional bus whose bandwidth is the aggregation of each memory device. For example, Figure 4-2 shows an example of eight 8-bit chips forming a 64 bit data bus. Data bus also contains the data strobe signals (DQS) used for the data alignment. Apart from the four types of signals, the LPDDR2-NVM protocol also contains miscellaneous signals such as the clock signal and temperature sensor signals.

Each PCM chip is organized into multiple banks. A bank is an independently controllable unit and is composed of several PCM arrays. The PCM array consists of the cells which are organized into 2D arrays and accessible through a row address and a column address. In contemporary design, each read/write access to one rank will activate one row specified by the row address across all PCM arrays in one bank and loads data from the cells in that row to row buffers. The column address specifies one bit from the row buffer of each PCM array, and multiple PCM arrays provides multiple bits. Contemporary memories used in CMP system usually provide the data from several columns in a burst to meet the size of a single cache line.
OptiPCM System Organization

Figure 4-3. The example of 16 mini-rank prototype design of the OptiPCM system

OptiPCM employs a set of non-volatile memory chips and DIMM (Dual In-line Memory Modules) following LPDDR2-NVM protocol as shown in Figure 4-3. OptiPCM replaces the conventional electrical channels in conventional memories with optical channels. A VCSEL chip is employed as the external laser source and provides laser lights for the optical channels. The Dense Wavelength Division Multiplexing (DWDM) silicon photonic technology provides highly aggregated pin-bandwidth density, which improves the bandwidth density by two orders of magnitude than that of electrical buses. The communication channels in OptiPCM consist of two bundles of unidirectional waveguides (in purple and blue in Figure 4-3). The outbound channel carries 64-byte wide data, addresses and commands from the memory controller to the memory while the return data travels through inbound channel. The modulator arrays are duplicated across the outbound channels to different memory ranks ("M"s in Figure 4-3). The memory controller modulates the laser lights in the outbound channel by properly controlling the modulators arrays. OptiPCM adopts connected outbound/inbound channel where the laser source only injects laser lights into the
outbound channel. The connected channel design is based on the half-duplex LPDDR2
data bus where the data can only be transmitted in one direction at any instant, and is
distinct from previous optical memory designs with separate outbound/inbound
channels [78]. The connected outbound/inbound channel shares one laser source,
hence nearly halves the static photonic power in separate outbound/inbound channel.
To support the photonic channel, the conventional electrical DIMM is redesigned as a
CMOS-compatible integrated photonic interface (PI) in place of the conventional
electrical pins. PI converts the optical signals from outbound channel to electrical
signals for write commands. For read commands, PI recaptures laser lights from
outbound channel to inbound channel and then modulates the laser lights to carry the
return data. The memory controller generates and distributes optical clock through
outbound channel to each memory rank in order to avoid the need for centralized signal
retiming unit and global timing synchronization among memory ranks. The clock
wavelength parallels the data wavelengths, with the clock signal traveling with the data
signals.

In contemporary design, multiple PCM chips in a rank work in a tandem and share
the DIMM, timing synchronization circuit and power supply circuit. OptiPCM breaks the
PCM chips from one rank into multiple mini-ranks, each of which is a single PCM chip
with its own power supply and timing synchronization circuits. Unlike the original mini-
rank design, which breaks each rank up to eight mini-ranks, OptiPCM breaks each rank
into 16 or even more smaller memory ranks. In OptiPCM, eight PCM chips are placed
on one memory board as contemporary memory. The increased ranks require the laser
source to inject enough laser power for the compensation of waveguide propagation
loss. The injection power is constrained at the first modulator, which must be below the threshold (around 10-20 mW [79]) that induces nonlinear effects. According to our evaluation, though 10 mW modulation power is sufficient for 64 ranks, more than 32 ranks starts to the performance owing to frequent inter-rank switch and yields overwhelming manufacturing cost.

**Sub-channel Division Technology**

OptiPCM surpasses the original electrical LPDDR2-NVM systems in terms of memory level parallelism and power consumption. Nevertheless, the prototype design could be further optimized to reduce the rank-to-rank switch penalty which is non-trivial when the number of ranks is large. In this section, we leverage both fixed and dynamic channel division to overcome the rank-to-rank turnaround penalty.

**Fixed Channel Division**

The Rank-to-Rank turnaround penalty exists in the high-frequency global-synchronous memory system. Unlike the commands to same open memory bank that can be issued and pipelined back to back, the consecutive commands to different memory ranks relies on the system-level synchronization mechanisms [71]. Thus, the shared data bus must be idle for some period of time between data bursts from different ranks. In the electrical design, the synchronization circuits are used to align the strobe signal DQs to DQs on data bus. Modern LPDDR2 protocol removes the DLL circuits that are usually implemented in the DDRx for synchronization; however, LPDDR2 also suffers from the latency of synchronization circuit. This latency dominates the rank-to-rank synchronization penalty, which depends on the system-level synchronization mechanisms and usually costs 1 to 3 cycles [69]. Due to the unpredictable photodetector/modulator delay and the requirement for optical clock synchronization,
the synchronization circuit is also required in the optical domain and the ranks suffer from rank-to-rank turnaround penalty. It is possible to increase the number of DQs aligned to one DQS. In electrical domain, eight DQs are aligned to one DQS so that the physical designers can easily route the wires. In optical domain, the wavelengths travels within same waveguide demonstrate highly similar physical characteristics, so OptiPCM aligns all data signals of one mini-rank to one DQS.

The rank-to-rank turnaround penalty in conventional LPDDR2-NVM with the help of command reordering is shown in Figure 4-4(a). In this example, the data burst (tBL) and synchronization (tRTRS) operation interleave on data bus. In contemporary CMP system, the typical tBL length is 4 cycles (64 bytes cache line), and the tRTRS rises with increasing frequency of bus clock (e.g. approximately 3 cycles in DDR3 [71]). In OptiPCM, the tBL is reduced to 1 cycle in the 64-byte wide data bus and the number of ranks is large, which could incur more significant bandwidth waste.
Figure 4-4. The timing penalty caused by rank-to-rank switch (a). Memory access timing without channel division for consecutive write to three ranks (PA_n = Pre-activate rank n, ACT_n = Activate rank n, WR_n = Write rank n, RAH_n = Raw Address High bits, RAL_n = Raw Address Low bits. [tRP = 5, tRCD = 4, tWL = 1, tBL = 4, tRTRS = 3]) (b). Memory access timing with channel division for consecutive write to four ranks [tRP = 5, tRCD = 4, tWL = 1, tBL = 16, tRTRS = 3])

In the fixed channel division, the data channel is equally divided into several sub-channels. Each sub-channel is dedicated to one rank. The memory controller allocates one sub-channel rather than the whole data channel to one memory access. Although the width of each sub-channel decreases, the sub-channel division amortizes the rank-to-rank synchronization penalty as shown in Figure 4-4(b). In this example, the narrow sub-channel extends the original tBL from 4 to 16, but inter-rank switch penalizes only one sub-channel and thus quarters the overall rank-to-rank switch overhead.
Dynamic Channel Division

Although the sub-channel division benefits the system performance, it may fail to utilize the provisioning channel bandwidth sufficiently especially under non-memory intensive applications. To address this issue, we propose the dynamic channel division, which dynamically adjusts the sub-channel width (i.e. number of wavelengths) based on the incoming traffic. When there is only one waiting command in the command queues, the memory controller allocates the whole channel to it. When two or more commands in the command queues compete for the channel, the memory controller seeks to equally divide the channel width among these commands using the current wavelengths availability information. However, if the equal division is unobtainable, one or more request(s) will be delayed to next assignment. In the non-memory intensive applications the dynamic channel division features low latency data channel; while the rank-to-rank switch penalty is amortized in the memory-intensive applications. The dynamic channel division requires extra design module for memory device and the memory controller as we will discuss in next section.

The Structure of PIs

In OptiPCM, the PIs are deployed in place of the conventional electrical DIMMs as shown in Figure 4-5(a). PI directs photonic signals to appropriate ranks and convert between electrical/optical signals. One RW signal in each sub-channel travels along with the data signals to indicate the current direction of one sub-channel. As shown in Figure 4-5(a), all RW bits from outbound waveguides are first directed by R1 to the vertical waveguide, and then separated by R2 resonators as the pulse control signals of R3 [15] by injecting or canceling the free carriers. The R3 resonators then direct the signals to either the ranks (or optical crossbar in dynamic channel division) or the
inbound waveguides based on their status. In the prototype design, the signals are
directed to the ranks and the built-in photodetectors convert the optical signals to
electrical signals.

Figure 4-5. The structures of important photonic components (a). The photonic interface
(PI) (b). The optical crossbar in dynamic-channel division
To support the fixed channel division, the PI simply directs the signals in one sub-channel to the corresponding rank using passive turn resonators (which is not shown in Figure 4-5). For the dynamic channel division, the PI uses a group of optical bits named the connectivity bits and encodes the connection information into them. An optical crossbar under the control of the connectivity bits (shown in the red dashed-dotted box in Figure 4-5(a)) switches the data between the sub-channels and the ranks.

An \( n \times w \) optical crossbar is required to switch the data between \( n \) ranks and \( w \) wavelengths wide data bus. The structure of the crossbar is depicted in Figure 4-5(b). The optical crossbar (shown in the red dashed-dotted box in Figure 4-5(b)) contains \( n \times w \) basic unit. One basic unit contains one passive resonator used to extract one connectivity bit and one optical tuning resonator that directs the data from one sub-channel to one rank. For example, an 8 sub-channel with 64 bytes wide data bus requires \( 512 \times 8 = 4096 \) such components. The PI requires no crossbar-like components for the read operation. As Figure 4-5(a) shows, the modulator arrays from different ranks modulate data to the same waveguide at different wavelengths to avoid interference.

Increased ranks and sub-channel division requires extra photonic components and incurs extra limitation and overhead. The extra resonators and increased length of waveguides attenuate the traversing laser lights. However, the maximal power that the modulator could support must be below the threshold at which nonlinear effects are induced, which is typically \( 10 \text{–} 20 \text{ mW} \). The optical power experiences up to 1 filter drop loss, 4095 filter through loss, 1 modulator insertion loss, and 1 photodetector loss when the system employs dynamic wavelength assignment with 8 ranks and 64 bytes
wide data bus. This part of loss is around 6.1 dB. Current technology node presents the waveguides which have a propagation loss of 0.3 dB/cm [80]. Assuming 10 mW modulation power, the length of waveguide could be up to 89 cm according to the calculation, which is sufficient to route waveguides to memory boards.

The OptiPCM uses the internal row buffers within PCM chips to convert different widths of sub-channel and internal data. The data from the sub-channel fill the row buffer in one or more cycles and is written into the memory cells in one batch. The data read out from memory cells is also temporarily stored in the buffers and then consecutively sent to the sub-channels.

The overhead of the dynamic channel division mainly comes from the optical crossbar and the extra internal width conversion circuit. The calculation results from optical latency model [5, 73, 81, 82] indicate that the optical crossbar incurs less than 100 ps latency, which is negligible under 533MHz clock (the highest frequency supported by Joint Electron Devices Engineering Council (JEDEC) LPDDR2). We synthesize this circuit in Synopsys Design Compiler [83] and find that the critical path latency is also within one cycle. We also include its power consumption information under different traffic rates.

The Design of Memory Controller
Figure 4-6. The structure of a memory controller
Figure 4-7. Finite state machine in the Enhanced Wavelength Assigner

Note that the memory controller needs to be modified to support the dynamic channel division. OptiPCM uses the First-Ready First-Come-First-Serve (FR-FCFS) memory controller [84]. In FR-FCFS memory controller, the transaction arbiter accepts requests (i.e. transactions) from multiple processors or I/O devices and arbitrates for them. Once a transaction wins arbitration and enters into the memory controller, it is
decomposed to a sequence of memory commands and mapped to a command queue. The command queues are arranged in such a way that there is one queue per rank. Then, commands are scheduled to the memory devices through the optical signaling interface depending on the command scheduling policy. OptiPCM uses the same transaction scheduling and address mapping mechanism as conventional memory controller, while enhancing the command scheduler to support dynamic channel division, as shown in Figure 4-6.

The enhanced FR-FCFS command scheduler used in the prototype design and fixed channel division is highly similar to conventional command scheduler. The workflow of enhanced command scheduler is depicted in Figure 4-7. The memory commands in the command queue fall into five categories, i.e. activate, preactivate (in place of the precharge command of volatile memories), read, write, and miscellaneous commands (e.g. power down). The command scheduler round-robinly checks the command queues and then allocates the active, preactive and control command in the conventional way. Once the scheduler finds an issuable write/read command from one command queue, it checks the commands at the top of each rank command queues. The scheduler then stores all the read and write commands into a pool and seeks to equally divide the available channel width among them provided that two or more commands are in the pool.

The channels are divided in the unit of one byte, and the total channel width is divisible by this allocated sub-channel width. The memory controller sends unmodulated laser lights to memory devices for read command and modulates the optical signals carrying the data to the memory devices for write command. The enhanced command
scheduler controls turn resonators by varying the electrical signals. In the fixed channel division, the turn resonators could be implemented as passive resonator and needs no additional control. In contrast, the memory controller controls the direction of data channels in prototype design and controls the turn resonators to dynamically adjust the sub-channel width and connectivity in the dynamic channel division.

**Experimental Setup**

**Simulation Methodology**

We evaluate the power consumption and performance improvement of OptiPCM using Simics [41], a multi-processor system simulator, and DRAMSim2 [85], a cycle-accurate memory system simulator. In our simulation, we mix several benchmarks from single-threaded SPEC2006 [86] benchmark suits to generate various memory stress. We also choose four multi-threaded benchmarks from PARSEC benchmark suit [60]. All the benchmark configurations are listed in Table 4-1. All the tests are executed on a quad-core system (2 threads / core) with 1GB memory as mentioned in Table 4-2. We test the following simulation scenarios as shown in Table 4-3. In our study, each memory board accommodates 8 memory chips and we assume that the Error Correcting Code (ECC) bits are stored along with their associated data bits in the same page [68].
### Table 4-1. Simulation benchmarks

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Workloads</th>
<th>Traffic(Mbps)</th>
<th>wr</th>
<th>rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPEC-1</td>
<td>bzip * 2, gcc * 2, sjeng * 2, lbm * 2</td>
<td>399.9</td>
<td>1027.7</td>
<td></td>
</tr>
<tr>
<td>SPEC-2</td>
<td>Milc, bzip * 2, gcc * 2, sjeng * 2, lbm</td>
<td>772.1</td>
<td>1182.3</td>
<td></td>
</tr>
<tr>
<td>SPEC-3</td>
<td>Milc, GemsFDTD, bzip * 2, gcc * 2, sjeng, lbm</td>
<td>822.0</td>
<td>1181.0</td>
<td></td>
</tr>
<tr>
<td>SPEC-4</td>
<td>Milc, GemsFDTD, mcf, bzip * 2, gcc, sjeng, lbm</td>
<td>804.9</td>
<td>1967.3</td>
<td></td>
</tr>
<tr>
<td>SPEC-5</td>
<td>Milc, GemsFDTD, mcf, cacbusADM, bzip, gcc, sjeng, lbm</td>
<td>963.2</td>
<td>1842.0</td>
<td></td>
</tr>
<tr>
<td>SPEC-6</td>
<td>Milc, GemsFDTD, mcf, cacbusADM * 2, bzip, gcc, sjeng</td>
<td>798.8</td>
<td>2039.8</td>
<td></td>
</tr>
<tr>
<td>SPEC-7</td>
<td>Milc, GemsFDTD, mcf * 2, cacbusADM * 2, bzip, gcc</td>
<td>608.6</td>
<td>2590.8</td>
<td></td>
</tr>
<tr>
<td>SPEC-8</td>
<td>Milc, GemsFDTD * 2, mcf * 2, cacbusADM * 2, bzip</td>
<td>595.9</td>
<td>2556.5</td>
<td></td>
</tr>
<tr>
<td>SPEC-9</td>
<td>Milc * 2, GemsFDTD * 2, mcf * 2, cacbusADM * 2</td>
<td>654.2</td>
<td>2831.7</td>
<td></td>
</tr>
<tr>
<td>blacksch</td>
<td>Financial Analysis, 65,536 options</td>
<td>2.1166</td>
<td>5.1035</td>
<td></td>
</tr>
<tr>
<td>swaption</td>
<td>Financial Analysis, 16 swaptions, 20,000 simulations</td>
<td>7.8720</td>
<td>2.0904</td>
<td></td>
</tr>
<tr>
<td>freqmine</td>
<td>Data Mining, 990,000 transactions</td>
<td>24.978</td>
<td>16.141</td>
<td></td>
</tr>
<tr>
<td>x264</td>
<td>Media Processing 128 frames, 640×360 pixels</td>
<td>114.08</td>
<td>138.40</td>
<td></td>
</tr>
</tbody>
</table>

### Table 4-2. Machine configuration

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>4 cores, Pentium-4, 1.0 GHz, In-Order, 4 IntALU, 2 FPALU</td>
</tr>
<tr>
<td>Width</td>
<td>4-wide fetch/issue/commit</td>
</tr>
<tr>
<td>TLB</td>
<td>128 entries(ITLB), 256 entries(DTLB), 4-way, 200 cycle</td>
</tr>
<tr>
<td>Branch Pred.</td>
<td>2 K entries Gshare, 10-bit global history, 32 entries RAS</td>
</tr>
<tr>
<td>L1 Cache</td>
<td>64 KB, 8-way, 64 Byte/line, 2 ports, 3 cycle</td>
</tr>
<tr>
<td>Integer ALU</td>
<td>4 I-ALU, 2 I-MUL/DIV, 2 Load/Store</td>
</tr>
<tr>
<td>FP ALU</td>
<td>2 FP-ALU, 2 FP-MUL/DIV/SQRT</td>
</tr>
<tr>
<td>L2 Cache</td>
<td>512 KB, 8-way, 64 Byte/line, 12 cycle</td>
</tr>
<tr>
<td>Data Channel</td>
<td>Electrical: 64 bit/channel, Optical: 64 byte/channel ;533MHz double data rate</td>
</tr>
<tr>
<td>Memory</td>
<td>1 Giga Bytes</td>
</tr>
<tr>
<td>Capacity</td>
<td></td>
</tr>
</tbody>
</table>
Table 4-3. Simulation scenarios (The suffix “-n” implicates the number of PCM chips deployed in system)

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASE</td>
<td>The conventional electrical LPDDRx-compatible channel with eight chips in one rank (baseline case)</td>
</tr>
<tr>
<td>MINI-n</td>
<td>The mini-rank configuration electrical LPDDRx channel with small ranks as in [68]</td>
</tr>
<tr>
<td>PRT-n</td>
<td>Prototype OptiPCM design</td>
</tr>
<tr>
<td>FCD-n</td>
<td>Prototype OptiPCM design with fixed channel division</td>
</tr>
<tr>
<td>DCD-n</td>
<td>Prototype OptiPCM with dynamic channel division</td>
</tr>
</tbody>
</table>

**Power Model of the Communication Bus**

![Diagram of communication bus](image)

Figure 4-8. The power modeling of the LPDDR2-NVM (the equivalent driver impedance $R_{ON}$ is equally devided into two parts: $R_{ONPU}$ and $R_{ONPD}$. The value is typically chosen to be $120\,\Omega$ when measuring at $0.5 \times V_{DDQ}$)

Table 4-4. Optical loss in various components

<table>
<thead>
<tr>
<th>Optical components</th>
<th>Attenuation</th>
<th>Optical components</th>
<th>Attenuation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optical coupler</td>
<td>1 dB</td>
<td>Optical splitter</td>
<td>0.2 dB</td>
</tr>
<tr>
<td>Interlayer coupling loss</td>
<td>1 dB</td>
<td>Filter through</td>
<td>$1^4$ to $1^2$ dB</td>
</tr>
<tr>
<td>Filter drop</td>
<td>1.5 dB</td>
<td>Photo detector</td>
<td>0.1 dB</td>
</tr>
<tr>
<td>Waveguide loss</td>
<td>0.3dB/cm</td>
<td>Bending loss</td>
<td>0.5 dB</td>
</tr>
<tr>
<td>Non-linear loss</td>
<td>1 dB</td>
<td>Modulator insertion loss</td>
<td>0 ~ 1 dB</td>
</tr>
<tr>
<td>Waveguide crossing</td>
<td>0.05dB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

109
**Electrical Links:** The electrical link between memory and memory controller is modeled as [74]. The transmission line between the PCM and the processor could be characterized using a simple RC model. In LPDDR2-NVM standard, DQ termination may not be used to conserve on power dissipation or board space. LPDDR2-NVM standard also uses the Low Voltage Complementary Metal Oxide Semiconductor (LVCMOS) logic level rather than Stub Series Terminated Logic (SSTL) in DDRx. Figure 4-8 shows the LPDDR2-NVM communication bus model. The power consumed on the DQ bus could be calculated as: \( P(\text{per DQ}) = C_{LOAD} \times V_{REF}^2 \times (2 \times f_{CK}) \), where the DQ data rate frequency is twice the system clock frequency [86]. For the differential transmission line, the voltage supplied to the DQ bus \( V_{REF} \) is 0.5 times \( V_{DDQ} \).

**Optical Links:** The power consumed on the optical channels is an aggregation of both the static power and the dynamic power. The static power consists of the required detection power for each resonator, the tuning power of the resonators when they are tuned to be ON state, the traversing optical power loss, and the power consumed by the heater of the resonator. The dynamic power is consumed by the modulators and photodetectors when modulating and detecting optical data. The important factor that affects the total static power consumption is the required optical detection power for a single photodetector. Prior study [65] shows that the power consumption of the photodetectors is related to the BER. We adopted expected BER of \( 10^{-15} \) [5, 23] to ensure reliable end-to-end communications, which required 5 μW sensing power per photodetector [65]. The power consumption for different photonic components is summarized in Table 4-4 [7, 11]. By default, all the ring resonators are set to OFF state. The energy is required when they are tuned to ON state [3] and this in-plane Poly-Si
energy per resonator is assumed to be 0.5 mW [18]. Assuming advanced driver circuits with poly-Si carrier lifetimes of 0.1-1 ns, the power consumed by each modulator is approximately 200 fJ/bit [45]. The energy coupling efficiency of the laser source ranges from 30% [63] to 50% [7]. We use the median value of 40% in our power model.

**PCM Devices:** The main difference between the DRAM and PCM is the organization of the cells. DRAM employs the 1T1C cell while PCM employs the 1T1R cell (shown in Figure 4-1). In the non-volatile memory, the preactive commands that load the row address buffer replace the precharge command in the volatile memory.

![Figure 4-9. The power consumption under different memory states [88] per memory chip (P: dynamic power consumption of the peripheral circuits; C: power consumption of the cells; L: leakage power consumption; units: mW; n: number of modified bits per row)](image)

The PCM power has two major consumers: the peripheral circuits and the cells. We adopt the power consumption profile from CACTI 6.5 [89] for the peripheral circuits.
and the power data of the PCM model extended from [78] in our study. The PCMs work in different states when operating, as shown in Figure 4-9. Its timing parameter is shown in Table 4-4. In reading and writing states, the memory returns the latched data from the row buffers in response of the column access command. We obtain the sense amplifier power and the row buffer power data from CACTI. The preactive operation is analogous to the precharge in the DRAM accesses, which resets the row buffer to the idle state once the minimum preactive latency (tRP) is satisfied. In the preactive operation, the data stored in the row buffer is written back to the memory cells using partial write where only the modified bits will be written [66]. In the active state, a row of data from the PCM cells are sensed, amplified and then latched into the sense amplifiers. All but power-down states consume leakage power. The idle PCM device in power-down state could save the leakage power however incurs longer exit latency going back to the idle or active state to serve incoming requests. By leveraging the mini-rank design, it is possible to fine-tune the state of each rank without impeding the data operation in other ranks.

In power-down state, the memory device is still supplied with power however most of the peripheral circuits like input/output buffers are deactivated [90]. The LPDDR2-NVM protocol supports the power-down state with stopped clock. The only overhead of resuming the clock is a NOP command before the next access command could be applied. The deep power-down state eliminates power to both the peripheral circuitry and memory array and will be supported in future LPDDRx protocol. In this study we choose to use the power-down state with stopped clock. Applying this state effectively reduces the power consumption of the PCM device, while incurring limited entering and

112
exiting overhead. We use CACTI to estimate the leakage power consumed by the peripheral circuits. PCMs belong to the class of non-volatile memory, hence there is almost no leakage power consumed by the cells [78]. The data from Micron P8P PCM datasheet [88] shows that less than 100 uA current per memory chip will be consumed in the low power state. So we assume that the power consumption in power-down state is negligible, which is consistent with [91].

Performance Evaluation

Power Consumption Breakdown

The PCM device power consumption could be categorized into three groups using the power analysis above: background, operation, and read/write power. The background power is consumed all the time when the memory chip powers on except for the power-down state. The device consumes operation power when performing activation or preactive operations. The read/write power is consumed when the device reads or writes data. We record the power in Figure 4-10 and find that most of the overall power consumption is decreased when more ranks are deployed in OptiPCM. For example, the PRT-64, FCD-64 and DCD-64 reduce the overall power by 34.8%, 27.8% and 21.8% respectively compared with the baseline case. Due to the limitation of pages, we only show the results from half of the benchmarks; the remainders exhibit similar behavior.
Figure 4-10. The breakdown of power consumption in OptiPCM

The overall power reduction comes from the different groups. The memory cell R/W power is caused by the write/read operations. This part of power is majorly proportional to the number of memory accesses and increases with the reduced execution time. For example, the PRT-64, FDC-64 and DDC-64 increase the power by 10.1%, 26.3%, and 40.3%. The memory operation power occupies a large proportion of the overall power consumption owing to the low-power LPDDRx interface and low-standby power. The operation power reduces significantly by 84.8%, 83.5% and 82.1% in PRT-64, FDC-64 and DDC-64 modes due to the reduced width of activated sense amplifiers in smaller ranks. The memory background power is also preserved owing to the better utilization of low-power state. The smaller rank will be staying at idle state more frequently than original design; the negligible cell leakage power helps saving the power consumption in that state as well. Thus, the PRT-64, FDC-64 and DDC-64 consume 42.7%, 40.2%, and 36.7% less background power on average. Moreover, the photonic channel saves 44.1% power on average compared with electrical channels though its power increases with number of ranks.
Latency Evaluation under Different Memory Configurations

Figure 4-11. The latency behavior in different test scenarios

Figure 4-11 shows the latency improvement under different network configurations. Overall, the SPEC 2006 benchmark suit exhibits much higher latency than that of PARSEC owing to the significantly larger memory accesses. Among the network configurations, we observe that the conventional mini-rank design incurs an average of 16.9% performance degradation, especially in the low-traffic PARSEC benchmarks (on average of 27.3%) due to the insufficient channel utilization. After employing the photonic channels, the memory latency is reduced (i.e. PRT-32) by 13.9% compared with the BASE. This is caused by the increased ranks that hide the long PCM access latency, and the high bus width that photonic waveguides provide. The fixed channel division (FCD-32) improves the latency by 30.6% on average of SPEC and 3.6% on average of PARSEC benchmarks. The rank-to-rank turnaround penalty is significantly amortized in the high-traffic benchmarks (SPEC) owing to fixed channel division; however, the slight performance improvement is mainly due to the increased bus width.
The dynamic channel division (DCD) compensates for the performance degradation under low-traffic since it shows similar behavior with the PRT in low traffic while amortizes the rank-to-rank turnaround penalty when the traffic is heavy. The dynamic channel division (DCD-32) gains an average performance improvement by 32.0% and 30.9% in SPEC and PARSEC compared with baseline case. The performance improvement comes from three aspects: 1) the increased ranks provide better parallelism 2) the increased width of photonic communication channel 3). The dynamical allocation of channel width reduces the rank switch penalty while preserving the low latency when the traffic is low.

**System Throughput under Different Number of Ranks**

We evaluate the system throughput under different number of ranks. Increasing ranks could provide better memory-level parallelism to hide the long latency; however, the width of each sub-channel may be reduced. We observe from Figure 4-12 that in most cases the system throughput improves with the increased ranks. The memory-intensive benchmarks exhibit more significant throughput improvement than the non-memory intensive benchmarks. For example, the DDC-32 and FCD-32 exhibits 28.1% and 39.7% improvement compared with BASE on average of all the SPEC benchmarks, compared with 3.0% and 15.5% improvement on average of PARSEC benchmarks. This performance variation is due to the traffic difference; the increased number of ranks can hide the latency difference better when the traffic amount is high.

Figure 4-12 shows that more ranks do not always help the system performance much. For example, the FCD-8, FCD-16, FCD-32, and FCD-64 promote the system performance by 19.8%, 26.2%, 28.1% and 28.7% respectively; the DCD-8, DCD-16, DCD-32, and DCD-64 improve the system performance by 30.0%, 37.3%, 39.8%, and
40.4% respectively. In both cases, -32 configuration is sufficient for parallelism, so -64 configuration demonstrates quite slight performance improvement while introduces in additional power consumption and manufacturing cost. On PARSEC benchmarks, the increased number of ranks even hurt the performance in FCD due to the reduced sun-channel width.

![Figure 4-12. Normalized memory throughput under different rank number (All results are normalized to BASE)](image)

FCD exhibits better performance compared with DCD especially on PARSEC benchmarks. For example, in -32 and -64 configurations, the DCD demonstrates 9.1% and 9.2% improvement on FCD in SPEC and 12.1% and 13.4% improvement on PARSEC. The DCD performs better on PARSEC since in could utilize the channel width better under light traffic compared with FCD.

**Channel Width Impact on OptiPCM**
We also evaluated the system performance under different data bus widths, from the contemporary 8 bytes to a whole cache line (64 bytes). The BASE following contemporary electrical channel supports up to 8 bytes wide data bus. We simulate the BASE configuration, though unimplementable, with up to 64 bytes wide bus (shown in dash in Figure 4-13). Figure 4-13 (a) shows that wide bus helps improving the performance of all the configurations. In PRT, FCD and DCD, the 64-byte bus improves the performance of 8-byte bus by 18.1% in SPEC and 24.0% in PARSEC. The 64-byte FCD improves 8-byte FCD performance by 31.1% in SPEC and 30.2% on PARSEC, which is the highest among the three network configurations. The FCD suffers from high latency especially in the low traffic benchmarks due to the very narrow sub-channel. In the memory-intensive applications with FCD configuration, the memory accesses are able to fully utilize the sub-channels and the system achieves better performance.

Figure 4-13(b) also shows that on the low traffic benchmarks (PARSEC), the increased ranks exhibit limited impact on the performance. For example, the FCD-32 exhibits a 4.0% improvement on FCD-16 on average of the PARSEC benchmarks, but
DCD-32 improves DCD-16 by only 0.9%, compared with 6.3% and 9.1% on SPEC. The latency of the channel affects the performance considerably in low traffic benchmarks. For example, the 64 bytes width FCD shows 27.7% performance degradation compared with 8 bytes width FCD.
CHAPTER 5
RELATED WORKS

Recent advances in silicon-based photonic devices have inspired the possibility of realizing photonic NoCs that satisfy the communications requirement of future multi-core processors.

There are many researches focusing on the critical components of photonic network. A common factor among the above topologies is that they rely on high-speed modulators, switches, multiplexers, de-multiplexers, or detectors which utilize ring resonators as building blocks. The thermal reliability of ring resonators is left unaddressed. Temperature-aware routing was proposed for electrical NoCs using ThermalHerd [92], a distributed run time thermal management scheme. ThermalHerd tries to mitigate the hotspots generated at a router by reducing its workload using different routing schemes. High-speed modulators have been demonstrated using two different structures: Mach-Zehnder interferometers [93] and resonant structures [32]. Among those, Mach-Zehnder interferometers have better thermal stability [20]. However, they are bulky and consume greater power. Ring resonators are smaller in size but highly sensitive to temperature variations [20]. The effect of temperature variations on the operation of ring resonators has been mitigated by the use of metal heaters placed on top of waveguides. However, this method consumes significant power and is cumbersome to implement for large number of ring resonators. Improvements on metal heaters have been proposed by Manipatruni et al. [25] by using localized heating due to PIN junction DC bias current. Nawrocka et al. [21] have also observed the undesirable shifting of resonance frequency due to temperature variations. To mitigate the thermal impact on ring resonators, Biberaman et al. [94] used integrated
metal heaters to tune individual rings and align their resonant modes to the signals’
wavelengths. However, for a chip with thousands of ring resonators, it is difficult to
monitor and control each ring separately. Another approach taken by [95, 96] is to
induce pressure in the resonator by using oxidation methods to compensate for the
effect of temperature variations on the resonance frequency shift. Guha et al. [97]
proposed passive temperature compensation by coupling a ring resonator to a Mach-
Zehnder interferometer. A laser source can be tuned to follow the resonant frequency
shift of the rings to compensate for temperature effects. Tunable lasers are commonly
used in communications and testing; Wang et al. [98] used a tunable laser with steps of
0.01 nm over the wavelength range of 1520-1580 nm to characterize micro-ring
resonators. Although all the above techniques have been demonstrated for a single or
small number of resonators, they are difficult to implement in large-scale photonic NoCs
because of implementation difficulty or the unwieldy number of terminals to control the
heaters. Until a proper solution at the device level is found, architecture and OS-level
solutions must provide reliable operation under wide thermal variations. To our
knowledge, there has been no prior work on designing thermal resistant photonic NoCs
at architecture and system levels.

Many researchers adopted the photonic techniques in the interconnection among
the processing cores targeting at the reduction of the overall chip power consumption.
However, prior solutions usually impose new challenges such as the auxiliary electrical
network or oversimplified routing method. Shacham et al. [16] have proposed a 2D
folded torus topology utilizes a circuit-switched network. The electrical packet-switched
network is charged with the setup and breakdown of an optical path before transmitting
a data packet; the electrical network, however, incurs additional power and latency cost. Our ESPN architecture employs a single optical network and thus achieves a low-power high-speed network. Vantrease et al. [15] proposed a fully-connected high-bandwidth low-latency optical crossbar architecture. Every node possesses a dedicated channel through which other nodes can transmit data. The arbitration of the channels is performed through requests to the token channel. Although it can achieve a significant performance improvement, a system with N nodes requires at least N + 1 serpentine waveguides, which causes layout difficulty and dramatic static power. Pan et al. [14] proposed a clustered hybrid electrical/optical architecture to address the scalability problem in [15]. The crossbar is partitioned into multiple crossbars in order to localize arbitration. In [47], a flexible crossbar is proposed to minimize the static power consumption by fully sharing a reduced number of channels across the network through doubling the modulators and photo-detectors. It copes with spatial traffic imbalance among cores but does not address the temporal traffic imbalance. Cianchetti et al. [13] proposed a switch-based on-chip photonic network that uses source-based routing and reconfigurable optical switches. It introduces electrical buffers into the network design to address the contention problem. However, the power efficiency and network performance in heavy traffic degrade significantly due to the frequent accesses of electrical buffers. Tan et al. [99] propose a micro-ring-resonator-based generic wavelength-routed optical router. They construct the optical routers by minimizing the number of passive micro ring resonators used. Ye et al. [100] takes advantage of both electrical and optical routers and interconnects in a hierarchical manner. It employs adaptive power control mechanism, low-latency control protocols, and hybrid optical-
electrical routers. However, the power consumption and performance is still restricted by the electrical network. In this dissertation, our ESPN architecture avoids the use of electrical buffers by leveraging optical routing to establish routing paths.

Some other researchers start working on the photonic connected memories; but most of the previous works are targeting at the DRAM based memory architecture. They usually focus on leveraging the photonic links to support the memory device in order to preserve power and achieve higher performance. For example, Beamer et al. [78] propose to use photonically interconnected memory by using a monolithically integrated silicon-photonic technology to replace electrical I/O and the electrical links across the memory chips. Hadke et al. [101] leverage the low latency optical links to replace the multi-hop store-and-forward network in the fully buffered DIMM [70] and successfully support up to 32 DIMMs. Hendry et al. [102] propose the photonic network-on-chip with integrated memory I/O interfaces to build a uniform photonic network-on-chip communication system using the low-latency and low-power photonic links. Udupi et al. [103] propose a 3D-stacked memory architecture by adding an interface die to handle the conversion between optics and electronics. The majority of prior photonically connected memory architectures focus on leveraging the photonic channels to reduce the power consumption on communication channels. The power saving on communication channels, however, is very limited in LPDDR2-NVM environment where the channel consumes less than 20% of the memory system power [67]. Malladi et al. [77] leverages LPDDR2 based DRAM in the data centers. They architect server memory systems using mobile DRAM devices, trading peak bandwidth for lower energy consumption per bit and more efficient idle modes.
our work and prior works is: in our design the major part of the power saving and performance improvement are gained from more effective management of PCM, not just from the channels. Our design substantially improves the PCM performance and achieves significant power saving from the management of PCM power consumption thanks to the large number of ranks enabled by photonic channels.
CHAPTER 6
CONCLUSION

Photonic NoCs provide many benefits over their electrical counterpart such as low latency, high bandwidth density, and repeater-less long range transmission. Recent advances in silicon photonics and integration with microelectronics present an opportunity to exploit the benefits of photonic NoCs in future multi-core processors. To implement photonic NoCs, high-speed modulators, switches, multiplexers, demultiplexers, and detectors must be integrated into the chip. Recent work has demonstrated that ring resonators can be used as versatile devices to build photonic NoCs. However, architecting photonic NoCs presents new challenges. In particular, thermal effects are a major concern. For example, temperature variations cause a shift in the resonance frequency prompting the resonator to respond to a different frequency. By responding to a different frequency, unintended operations occur or data is corrupted, thereby introducing high BER or even causing faulty operation in a photonic NoC. Thermal effects deteriorate the reliability and performance of photonic NoCs. Therefore, the successful implementation of photonic NoCs hinges on the ability to overcome thermal challenges. Current device-level methods to counter the effect of temperature variations are difficult to implement in CMOS processes or not suitable for large scale on-chip networks.

We address the effect of temperature variations on photonic NoCs and propose cross-layer solutions targeted at the circuit, architecture, and OS levels to mitigate their effect and improve reliability. At the circuit layer we maintain the original temperature by varying the bias current through ring resonators. At the architecture layer, we re-route messages away from hot regions and through cooler regions to their destinations. At the
OS layer, we employ thermal/congestion-aware co-scheduling to relocate workloads to the outer cores where heat dissipation is more efficient. This encourages messages to use the cooler center of the processor. The solutions can be integrated with each other to further reduce the BER and improve reliability. We have shown that the average BER and MER are reduced by 94% (96%) and 76% (84%) for SPF+OS (and TF+OS) techniques respectively. We have also shown that the power consumption can be reduced by up to 36% by applying the three techniques.

Also, substantial static power consumption limits the scalability of photonic NoCs. We presented an energy-efficient photonic NoC architecture to enable traffic-aware dynamic allocation of network resources. We showed that dynamical coupling of communication resources and network traffic could save up to 50% of the total optical NoC power. We also employ an optical-based adaptive routing circuit switch, which reduces the application execution time by 22%. By integrating these technologies, our structure achieves 51% and 60% power and energy savings compared to the baseline case.

We also explored a high-performance and energy-efficient memory communication infrastructure for PCMs. The widely used LPDDRx-NVM protocol provides an implementation of energy efficient memory interface for non-volatile memories, however, degrades the system performance. We propose to leverage photonic channels in place of the conventional electrical channels. The photonic channels save energy from both the memory organization and communication channel. Meanwhile, our OptiPCM structure supports large number of memory devices on photonic channels and is able to process a large number of concurrent accesses and
thus improves the system performance. The channel division provides better channel utilizations and further improves the system efficiency. Our simulation results show that PRT, DCD and FCD design saves up to 34.8%, 27.8%, and 21.8% overall power consumption while increases the system performance by 22.3%, 28.7%, and 40.4% respectively.
LIST OF REFERENCES


[34] OptiSystem simulator, ver. 7.0, Optiwave, Ottawa, Canada.


134


BIOGRAPHICAL SKETCH

Zhongqi Li was born in Chengdu, China. He received his B.S. and M.S. degree in electrical engineering from University of Electronic Science and Technology of China, Chengdu, China, in 2006 and 2009, respectively. He received his PhD degree in electrical and computer engineering from the University of Florida, Gainesville, FL in 2012. He is a recipient of UF Alumni Graduate Fellowship. He has interned as a processor modeling engineer at Marvell Semiconductor in 2012.

His current research interests include the CPU/GPU architecture, and the interconnect network of the heterogeneous processor.