2021

Energy-efficient architectures for chip-scale networks and memory systems using silicon-photonics technology

BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Dissertation

ENERGY-EFFICIENT ARCHITECTURES FOR CHIP-SCALE NETWORKS AND MEMORY SYSTEMS USING SILICON-PHOTONICS TECHNOLOGY

by

ADITYA NARAYAN

B.E., Birla Institute of Technology and Science, Pilani, 2012
M.S., University of Pennsylvania, 2014

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2021

Approved by

First Reader

Ayse K. Coskun, Ph.D.
Professor of Electrical and Computer Engineering

Second Reader

Ajay J. Joshi, Ph.D.
Associate Professor of Electrical and Computer Engineering

Third Reader

Pascal Vivet, Ph.D.
Scientific Director of Digital Systems and Integrated Circuits Division, CEA-LIST

Fourth Reader

Milos A. Popovic, Ph.D.
Associate Professor of Electrical and Computer Engineering

Nobody ever figures out what life is all about, and it doesn't matter. Explore the world. Nearly everything is really interesting if you go into it deeply enough.

Richard Feynman

Acknowledgments

This thesis is the result of a rich blend of mentorship of my research advisors, guidance from my committee members, interesting discussions with collaborators, and the constant support and encouragement from friends and family.

I express my deepest gratitude to my PhD advisor, Prof. Ayse K. Coskun for her guidance and support throughout my PhD studies. Her advising style has helped me evolve into a critical thinker and develop a research aptitude that has greatly influenced me in maturing as a researcher. She has provided me with positive feedback to improve my writing and presentation skills. Our research discussions have been very engaging, and I appreciate the time and effort she put in my work. Her constant encouragement has enabled me to develop my ideas into research publications and deliver well-received talks at conferences.

I want to thank my PhD co-advisor, Prof. Ajay Joshi for his valuable feedback and suggestions on my research. I enjoyed the lively discussions and brainstorming sessions during my work on optically-controlled phase change memory. His inputs on addressing a problem from different angles has greatly shaped my way of thinking about a problem.

I want to extend my gratitude to Dr. Pascal Vivet, who was my manager at CEA-Leti during my internship. He and his team at CEA-Leti made me extremely comfortable during my stay in France and ensured a great educational experience. His feedback on thermal modeling and characterization of 2.5D and 3D chips improved my understanding of thermal implications in manycore chips. I would also like to acknowledge the rest of my committee members, Prof. Milos Popovic and Prof. Martin Herbordt, who provided insightful comments and feedback that helped me improve my dissertation significantly.

I want to extend special thanks to Dr. Tiansheng Zhang for being a great mentor

during my initial PhD years, and a good friend with whom I exchanged many interesting discussions. I also acknowledge Yvain Thonnart, who was my mentor during my internship at CEA-Leti. He has been instrumental in my understanding of silicon-photonic link technology. He patiently explained the operation of optical devices, and provided essential feedback on the device-level implications of my architectural and system proposals.

I owe many thanks to my collaborators and co-authors for our joint productive work during my PhD: Prof. Satish Narayanasamy and Dr. Shaizeen Aga from University of Michigan, Ann Arbor; Dr. Cesar Fuguet Tortolero from CEA-Leti; Prof. Andrew Kahng and Dr. Vaishnav Srinivas from University of California, San Diego; Dr. Yenai Ma and Furkan Eris from Boston University.

I thank my good friend, fellow PhD candidate and my flatmate, Soham Sinha, with whom I have shared both the fun moments and the struggles during my PhD years. I thoroughly enjoyed all our technical discussions on computer architecture and operating systems, and on various other topics including movies, politics, and economics. I am very thankful to all my lab mates and colleagues from PeacLab, ICSG group, CAADLab, and SeclaBU who made my PhD cheerful and memorable.

I want to extend sincere thanks to my parents, my brother, my parents in-law, and my brother-in-law, whose constant support has guided me through my PhD journey. Finally, I thank my wife, Sravya Kotaru, who has been my true source of inspiration to pursue a PhD. Her passion for science fueled a desire in me to explore the beautiful world of computing systems ever so deeply. She has stood by me through the difficult ordeals, always cheered me, and brought out the best in me. It would also be remiss of me not to mention her editing skills that have made me a better writer.

The research that forms the basis of this dissertation has been partially funded by NSF and the CARNOT Institute in France.

The contents of Chapter 3 contains reprints from the papers Aditya Narayan, Yvain Thonnart, Pascal Vivet, Cesar Fuguet Tortolero, and Ayse K. Coskun, "WAVES: Wavelength selection for Power-efficient 2.5D-integrated Photonic NoCs", in Design, Automation & Test in Europe Conference & Exhibition (DATE), 2019; Aditya Narayan, Yvain Thonnart, Pascal Vivet, and Ayse K. Coskun, "PROWAVES: Proactive Runtime Wavelength Selection for Energy-efficient Photonic NoCs", in Transactions on Computer-Aided Design (TCAD), 2020; Aditya Narayan, Yvain Thonnart, Pascal Vivet, Ajay Joshi, and Ayse K. Coskun, "System-level Evaluation of Chip-Scale Silicon Photonic Networks for Emerging Data-Intensive Applications", in Design, Automation & Test in Europe Conference & Exhibition (DATE), 2020; and Aditya Narayan, Ajay Joshi, and Ayse K. Coskun, "Bandwidth Allocation in Silicon-Photonic Networks Using Application Instrumentation", in High Performance Extreme Computing Conference (HPEC), 2020.

The contents of Chapter 5 are part reprints of the paper Aditya Narayan, Tiansheng Zhang, Shaizeen Aga, Satish Narayanasamy, and Ayse Coskun, "MOCA: Memory Object Classification and Allocation in Heterogeneous Memory Systems", in International Parallel and Distributed Processing Symposium (IPDPS), 2018.

ENERGY-EFFICIENT ARCHITECTURES FOR CHIP-SCALE NETWORKS AND MEMORY SYSTEMS USING SILICON-PHOTONICS TECHNOLOGY

ADITYA NARAYAN

Boston University, College of Engineering, 2021

Major Professors: Ayse K. Coskun, Ph.D.
Professor of Electrical and Computer Engineering

Ajay Joshi, Ph.D.
Associate Professor of Electrical and Computer Engineering

ABSTRACT

Today's supercomputers and cloud systems run many data-centric applications such as machine learning, graph algorithms, and cognitive processing, which have large data footprints and complex data access patterns. With computational capacity of large-scale systems projected to rise up to $50 GFLOPS/W$, the target energy-per-bit budget for data movement is expected to reach as low as $0.1 pJ/bit$, assuming $200 bits/FLOP$ for data transfers. This tight energy budget impacts the design of both chip-scale networks and main memory systems. Conventional electrical links used in chip-scale networks ($0.5-3 pJ/bit$) and DRAM systems used in main memory ($> 30 pJ/bit$) fail to provide sustained performance at low energy budgets. This thesis builds on the promising research on silicon-photonic technology to design system architectures and system management policies for chip-scale networks and main memory systems. The adoption of silicon-photonic links as chip-scale networks, however,

is hampered by the high sensitivity of optical devices towards thermal and process variations. These device sensitivities result in high power overheads at high-speed communications. Moreover, applications differ in their resource utilization, resulting in application-specific thermal profiles and bandwidth needs. Similarly, optically-controlled memory systems designed using conventional electrical-based architectures require additional circuitry for electrical-to-optical and optical-to-electrical conversions within memory. These conversions increase the energy and latency per memory access. Due to these issues, chip-scale networks and memory systems designed using silicon-photonics technology leave much of their benefits underutilized.

This thesis argues for the need to rearchitect memory systems and redesign network management policies such that they are aware of the application variability and the underlying device characteristics of silicon-photonic technology. We claim that such a cross-layer design enables a high-throughput and energy-efficient *unified silicon-photonic link and main memory system*. This thesis undertakes the cross-layer design with silicon-photonic technology in two fronts. First, we study the varying network bandwidth requirements across different applications and also within a given application. To address this variability, we develop bandwidth allocation policies that account for application needs and device sensitivities to ensure power-efficient operation of silicon-photonic links. Second, we design a novel architecture of an optically-controlled main memory system that is directly interfaced with silicon-photonic links using a novel read and write access protocol. Such a system ensures low-energy and high-throughput access from the processor to a high-density memory. To further address the diversity in application memory characteristics, we explore heterogeneous memory systems with multiple memory modules that provide varied power-performance benefits. We design a memory management policy for such systems that allocates pages at the granularity of memory objects within an application.

Contents

List of Tables

List of Figures

List of Abbreviations

3D	.....	3 Dimensional
2.5D	.....	2.5 Dimensional
AI	.....	Artificial Intelligence
ALU	.....	Arithmetic Logic Unit
ARIMA	.....	Autoregressive Integrated Moving Average
BEOL	.....	Back End of Line
DRAM	.....	Dynamic Random Access Memory
EDP	.....	Energy-Delay Product
FWHM	.....	Full-Width at Half Maximum
HBM	.....	High-Bandwidth Memory
HMC	.....	Hybrid Memory Cube
HPC	.....	High Performance Computing
LLC	.....	Last Level Cache
LUT	.....	Lookup Table
MLP	.....	Memory-level Parallelism
MPKI	.....	Misses per kilo Instructions
MRR	.....	Microring Resonator
MRRG	.....	Microring Resonator Group
NLP	.....	Natural Language Processing
NVM	.....	Non-Volatile Memory
OS	.....	Operating System
PCM	.....	Phase Change Memory
PTE	.....	Page Table Entry
PV	.....	Process Variations
RAM	.....	Random Access Memory
ROI	.....	Region of Interest
SCC	.....	Single-chip Cloud Computer
TIA	.....	Transimpedance Amplifier
TLB	.....	Translation Lookaside Buffer
TSV	.....	Through-silicon Vias
TV	.....	Thermal Variations

Chapter 1

Introduction

We currently live in an age of unprecedented amounts of data. Data is ubiquitous. Generation over generation, human ingenuity to store data has advanced exponentially, beginning with the ancient cave paintings to the Voyager Golden Record to a flash drive that is the size of a thumb. Our ability to develop means to store and process data has been a major driving force behind societal advancement.

With the digital revolution in the 21^{st} century, the ability to efficiently generate, process, and store this data is becoming critical across many sectors. Computational genomics is a field that is fast progressing towards extreme data-centric computing. Since the inception of the Human Genome Project in 2003 to map the entire human genomic sequence, the genomic data has been doubling every 11 months and is expected to surpass the total data requirements of Youtube and Twitter by 2025 (Cirillo and Valencia, 2019). As another example, a year of particle collisions at the Large Hadron Collider generates about one million petabytes (Hesla, 2012). Scientists now record only part of the raw data, but imagine the level of our understanding of the universe if they had the means to store the entire raw data. As a final example, to study the interaction between humans and computers, natural language processing models have seen major breakthroughs in linguistics, artificial intelligence, and cryptography techniques. In 2019, Nvidia released a model, Megatron, with 8.5 billion parameters (Nvidia, 2019), while Microsoft developed Turing-NLG with 17 billion parameters (Microsoft, 2019). The GPT-3 model developed by OpenAI in 2021 uses

174 billion parameters and requires $350GB$ of memory (Floridi and Chiriatti, 2020). It seems inevitable that we will reach levels of petabytes of data per day within the next decade.

Rapid proliferation in application datasets and their computational complexity has been pushing the demand for denser integration of compute cores and memory modules on a single chip. Manycore chips are already a big part of modern supercomputers and data centers. Mellanox's TILE-Gx72 is a 72-core system-on-chip that is used in intelligent networking, multimedia, and cloud applications (Mellanox, 2015). Intel's Xeon Phi series integrates up to 72 cores and multiple memory modules in a single chip (Sodani, 2015; Bradford et al., 2017), while AMD's EPYC processor family integrates 64 cores in a single chip (Lepak et al., 2017). The emergence of GPUs for machine learning and AI applications have yielded chip designs with thousands of lightweight cores. Nvidia's Turing GPUs have more than 4000 CUDA cores (Nvidia, 2018) and AMD's Navi/RDNA GPUs have more than 2500 cores (AMD, 2019).

The data-centric nature of emerging applications is pushing the design focus of manycore systems from how fast tasks can be executed to how fast data can be moved and how efficiently data can be accessed from memory systems. This has prompted the design of cost-effective and energy-efficient integration of compute cores, memory modules, and chip-scale networks in computer systems that serve supercomputers, data centers, and cloud systems. Critical design challenges in such dense manycore chips arise from (1) how fast the communication network can service requests among the different compute units and memory units, (2) how fast data can be read from and written to the memory unit, and (3) what degree of parallelism is offered by the network system and the memory unit in servicing these requests.

The prominent communication network in manycore chips relies on electrical link technology, which provides a maximum bandwidth of $112Gb/s$ at $10-50pJ/bit$ (Wade

et al., 2020; Pasricha and Nikdast, 2020). With roadmaps for on-chip bandwidth suggesting upwards of $1Tbps$ (Kim and Kim, 2014), it is impractical that electrical links can meet such demands due to their technological limitations, cost challenges, and energy constraints. On the main memory front, DRAM is the conventional technology used in most commercial servers and data centers. DDR4 DRAM provides a capacity of tens of GB with a bit density of $0.14Gb/mm^2$ at $40pJ/bit$ for read/write accesses. However, power consumption in DRAM, especially the leakage power, grows substantially with technology scaling, with current DRAM consuming 40% of total system power (Mutlu, 2018; Paul et al., 2015). Moreover, DRAM internal bandwidth is not scaling at the same rate as application requirements. Thus, the challenges in data movement and data access are forcing a paradigm shift in the network and memory design to attain energy-efficient execution of data-centric applications.

This thesis explores emerging chip-scale network and memory system solutions based on silicon-photonic link and optical integration technology. Device research has demonstrated silicon-photonic links as high-bandwidth and low-latency fabrics for chip-scale communication, and phase change materials with optical control as a scalable and non-volatile memory technology. A key missing link has been adapting such devices in manycore chips and developing the necessary architecture and system-level solutions that are tailored to the optical properties of these devices. This thesis claims that designing system architectures and management policies that are aware of the application variabilities and device characteristics is essential towards achieving an energy-efficient unified "silicon-photonic link and optically-controlled memory" system. To this end, this thesis develops runtime power-management policies for silicon-photonic links, architectural designs to integrate optically-controlled phase change memory with silicon-photonic links, and memory management policies for improving the energy efficiency of heterogeneous memory systems.

1.1 Designing Energy-efficient Silicon-Photonic Links

The rapid data growth and resulting compute and memory capacity in manycore systems make data movement a significant burden in chip-scale networks. Using conventional electrical links, an L1/L2 cache access on the same chip takes only about $0.1 - 0.2pJ/bit$ for data transfers, whereas the chip-scale data access to an L3 cache or main memory can often take up to $10 - 50pJ/bit$ (Pasricha and Nikdast, 2020). The latter energy numbers are $100\times$ higher than the energy efficiency budgets of supercomputers, cloud systems and data centers (Bergman, 2018). Enabling faster data movement at improved energy-per-bit over the chip-scale networks is, therefore, a key goal to address.

With advances in CMOS integration of silicon-photonics technology, chip-scale networks using silicon-photonic links are being developed. In 2020, Ayar Labs commercialized TeraPHY, a system that uses silicon-photonic links for chip-scale communication, providing up to $2Tbps$ bandwidth, and that is currently integrated into Intel's Stratix10 FPGA (Wade et al., 2020). Mellanox, now part of Nvidia, developed an optical transceiver of data rates up to $500Gps$ for GPU-accelerated computing (Rumley et al., 2017). Unlike electrical links, silicon-photonic links are able to deliver bandwidths on the order of $> 1Tbps$ at reduced latency and negligible data-dependent power. Despite these promising benefits, silicon-photonic links suffer from increased power overhead at higher data rates. This overhead results from laser sources emitting optical signals and the power dissipated in electrical circuitry for serialization, modulation, and filtering of optical signals. Furthermore, optical devices such as MRRs are highly sensitive to thermal variations, requiring additional heating power for thermal tuning. This power overhead increases the network energy-per-bit for chip-scale communication. Figure 1·1 demonstrates the increasing energy-per-bit of silicon-photonic links with increasing data rates (Bahadori et al., 2017b).

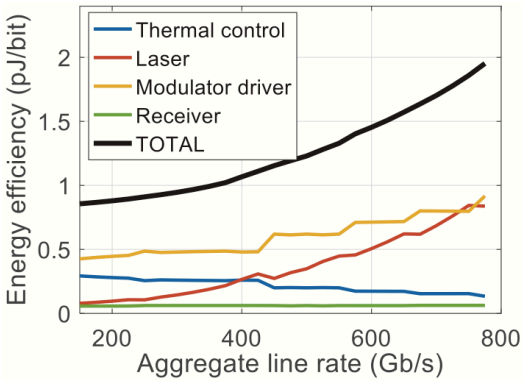


Figure 1·1: Energy-per-bit consumption in silicon-photonic links with increasing data rate (Bahadori et al., 2017b).

Power management in silicon-photonic links is a challenging task due to this direct trade-off between bandwidth and energy. Using a cross-layer approach enables a deeper understanding of the device-level sensitivities of optical devices, architectural and design parameters, impact of system-level bandwidth requirements, power and thermal profile, and implementation of the software stack. We, therefore, introduce three primary techniques for energy-efficient operation of silicon-photonic links as chip-scale networks:

1. **Bandwidth allocation for silicon-photonic links:** We propose **WAVE**length **S**election (*WAVES*) policies for power-efficient execution of silicon-photonic links. *WAVES* uses the bandwidth requirement of an application to activate the minimum number of optical channels for that application. Our first *WAVES* policy, **S**tatic **O**racle-**WAVES** (*SO-WAVES*) uses the average bandwidth requirement for an application to select the number of optical channels (Narayan et al., 2019). However, *SO-WAVES* does not account for the runtime dynamic trends in application's bandwidth requirement. Our **PRO**active **WAVES** policy (*PROWAVES*) predicts the network activity for future application phases using a time-series forecasting model to select the number of optical channels (Narayan et al., 2020b). Using graph and HPC workloads from standard

benchmark suites, we demonstrate substantial power savings with *SO-WAVES* and *PROWAVES* compared to a system that uses all of its optical channels.

2. **MRR thermal remapping during runtime application execution:** Due to the high sensitivity of MRRs towards PV and TV, we develop a method that accounts for the fabrication PV and chip-scale TV at each application phase. This thesis models the low-level thermal control loop at the system-level for the first time to capture the effects of TV-induced shifts and the resultant heating power. Modeling the thermal control loop enables *SO-WAVES* and *PROWAVES* to perform MRR remapping due to TV-induced shifts at application runtime and activate the optimal set of optical channels with lowest heating power (Narayan et al., 2020b).

3. **Application instrumentation assisted bandwidth allocation:** The communication traffic in chip-scale networks highly depends on the software implementation of the application. Our proposed system-level policies, *SO-WAVES* and *PROWAVES*, do not account for this dependence. We, therefore, design a framework to perform application instrumentation at the software-level that can assist our runtime *WAVES* policies to further improve the energy efficiency of silicon-photonic links (Narayan et al., 2020a).

1.2 Designing Scalable and High-throughput Main Memory

In addition to data movement, the key factors affecting the energy efficiency of data-centric applications are data storage and data access in the main memory. DRAM has been the prominent main memory used in the majority of computing systems. Unfortunately, DRAM technology faces critical scaling challenges at sub $20nm$ nodes. At lower technology nodes, leakage current in DRAM is higher, resulting in high idle power in DRAM cells (Mutlu, 2013; Kang et al., 2014; Lefurgy et al., 2003).

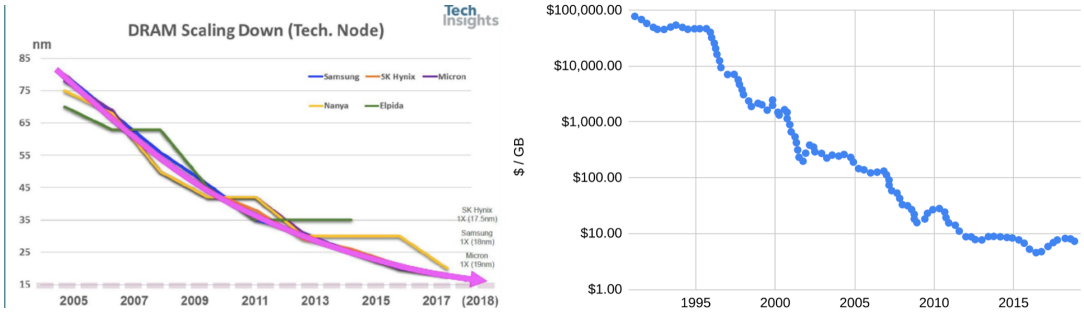


Figure 1·2: (a) DRAM technology scaling from 2005 to 2018 adapted from (Bergal, 2019). (b) Price per GB of DRAM from 1991 to 2019, according to the Objective Analysis graph (Hertz, 2021).

Moreover, DRAM cells uses capacitors to store charge, which leak charge over time and require periodic refresh to rewrite the data. These challenges put a significant burden on memory vendors working to scale down the technology nodes of memory chips. Figure 1·2 shows that DRAM scaling and the price per DRAM capacity have slowed down since 2010. To compound these issues, the limited memory bandwidth of DRAM systems fails to meet the increased bandwidth demands arising from parallel accesses of most data-centric applications.

We, therefore, need a main memory system that is amenable to technology scaling, has high bit density, meets the high capacity and bandwidth demands of data-centric applications, and uses low energy for memory accesses. NVM systems provide a scalable and non-volatile memory alternative with increased bit density and zero leakage power. A promising class of NVMs are PCMs with electrical control (EPCMs) owing to their higher reliability, increased bit density, and better write endurance (Bedeschi et al., 2008; Burr et al., 2010; Wuttig et al., 2017; Nirschl et al., 2007). Though EPCMs are highly scalable with increased bit density than DRAM, incur significant performance and energy overhead. Recent advances in device research have demonstrated phase change materials with optical control. These optically-controlled PCM cells, OPCM cells, demonstrate significantly higher bit density per cell compared

to other NVM cells, in addition to data non-volatility and high scaling. Moreover, the OPCM cells provide the opportunity for direct access with silicon-photonic links, thereby providing orders of magnitude higher memory bandwidth. Unfortunately, the current memory architectures for DRAM and EPCM systems are designed for electrical addressing and encounter major design challenges (such as increased latency, high energy, thermal issues) when adapted for OPCM cells. A main memory system using OPCM cells, therefore, requires a complete redesign of the microarchitecture, read/write access protocol, and address mapping.

In addition to the high bandwidth requirements in data-centric applications, these applications also exhibit significant diversity in their memory characteristics. For example, a highly parallel video rendering application exhibits high memory parallelism, while an iterative graph application exhibits very poor memory parallelism. Since memory modules are primarily designed to optimize either latency, bandwidth or power, a homogeneous memory system (such as DDRx, HBM, RLDRAM, LPDDRx, etc.) falls substantially short of addressing the diverse memory characteristics of applications. It is, therefore, beneficial to design a heterogeneous memory system with multiple memory modules, where different modules are optimized for different metrics. The performance of such a heterogeneous memory system is contingent on the memory management policy that can allocate pages based on the heterogeneity in memory characteristics within a given application.

The major contributions of this thesis towards designing an energy-efficient and high-throughput main memory system focus on two fronts.

1. **Architecting OPCM memory system with silicon-photonic links:** This thesis presents the first architectural design of a main memory system using OPCM cells that is directly accessed using silicon-photonic links. Our proposed **CO**mbined **S**ystem of Optical Phase Change **M**emory and **O**ptical Link**S**

(*COSMOS*) uses a novel read and write access protocol for accessing the memory cells in a multibanked OPCM array. *COSMOS* also uses an E-O-E control unit to map the standard DRAM protocol commands, data and addresses into optical signals to access the OPCM array. Owing to the increased bit density of OPCM cells and the high-bandwidth-density of silicon-photonic links, *COSMOS* demonstrates significant performance and energy benefits over EPCM systems. Moreover, *COSMOS* is the first NVM system with comparable performance and energy as DRAM systems, with increased bit density, higher scalability, non-volatility and zero leakage power.

2. **Memory management for heterogeneous memory systems:** We demonstrate that memory objects allocated in heap space exhibit substantial diversity in their memory characteristics. To address this diversity, this thesis presents **M**emory **O**bject **C**lassification and **A**llocation (**MOCA**) (Narayan et al., 2018). *MOCA* uses the memory intensity and MLP of memory objects to classify their memory characteristics, and uses this information at runtime to allocate them in the appropriate memory module in a heterogeneous memory system.

1.3 Dissertation Organization

The remainder of the thesis begins with a background on silicon-photonic link technology and optically-controlled phase change memory cells, and a review of related work in Chapter 2. Chapter 3 presents our system-level power-management policies for reducing the photonic power in silicon-photonic links. Chapter 4 describes the architecture of *COSMOS*, a non-volatile OPCM main memory system, where the memory cells are directly interfaced using silicon-photonic links. Chapter 5 presents our memory management policy, *MOCA*, for heterogeneous memory systems. Chapter 6 concludes this thesis and discusses the open problems and future research directions.

Chapter 2

Background and Context

This chapter starts with an introduction on 2.5D-integrated manycore systems as a promising alternative to 2D and 3D-stacked systems. We discuss the challenges for data-centric applications running on manycore systems arising from the limitations of chip-scale networks and main memory. The chapter then introduces silicon-photonic links as high-bandwidth-density and low-latency networks and reviews the existing works on designing energy-efficient silicon-photonic links. We then present the operation of an optically-controlled PCM cell and its promise in designing a non-volatile and high-throughput main memory that can be directly interfaced with silicon-photonic links. Later, the chapter discusses heterogeneous memory systems with different power-performance characteristics and the existing works on memory manegement in such systems. The chapter concludes with an overview of the distinguishing aspects of this thesis compared to the existing works.

2.1 2.5D Manycore Computing Systems

The growing need for data-centric processing is driving the design of manycore chips with hundreds of logic cores. The design of such a densely integrated manycore chip in conventional 2D fabrication results in large die sizes and reduced manufacturing yields, contributing to high fabrication costs (Gelsinger, 2001). Since the late 2000s, 3D integration has been explored as an alternative to 2D manycore chips. 3D-integrated chips enable vertical stacking of multiples dies using dense TSVs, which

provide high-bandwidth-density between multiple dies. However, the increased transistor density with vertical stacking leads to high power density and high chip temperatures. Consequently, sophisticated cooling techniques and complex packaging solutions for 3D-integrated chips contribute to increased costs (Kandlikar, 2014).

2.5D integration has gained popularity as an alternative technology to 2D and 3D integration. In 2.5D-integrated manycore chips, multiple smaller chiplets are integrated on a large interposer. 2.5D-integrated chips are more cost-effective than 2D chips, as breaking down a large monolithic chip into multiple smaller chiplets improves the manufacturing yield (Stow et al., 2016). 2.5D-integrated chips also result in a lower power density than 3D-integrated chips, thereby resulting in lower chip temperatures (Stow et al., 2016). 2.5D integration further decouples the design of compute cores, accelerators (GPUs, APUs, etc.) and the memory systems (Kannan et al., 2015). Such an approach enables flexible integration of homogeneous or heterogeneous dies. 2.5D integration has, therefore, become prominent in commercial chips such as Xilinx Vertex 7 (Saban, 2011), Nvidia Tesla and Pascal GPUs (Hu et al., 2018), ARM CoWoS (Lin et al., 2020) and AMD Fiji GPU series (Lee et al., 2016a). Intel has also developed a 2.5D stacking technology called Embedded Multi-die Interconnect Bridge for their FPGA products (Mahajan et al., 2016) and Foveros for their LakeField CPU (Ingerly et al., 2019).

2.1.1 Chip-scale Networks in 2.5D Manycore Systems

A critical performance bottleneck with integrating higher number of cores and chiplets in 2.5D manycore systems arises from the data movement in the network. Such a bottleneck could crop up due to several factors ranging from many-to-few network patterns blocking critical packets (Li and Chen, 2020), non-uniformity of the transmission data sizes (Shamim et al., 2019), redundancy of transmitted data, or local congestions in the network blocking other packets (Liu et al., 2018). Increasing the

number of cores and chiplets in 2.5D manycore systems, therefore, demands an efficient design of the chip-scale network. Vivet *et al.* design flexible and scalable system network topologies between the chiplets using an active interposer (Vivet et al., 2020). *NoD* is an independent network chiplet for 2.5D manycore chips that is responsible for routing packets from a source router to a destination router (Ebrahimi et al., 2017). Jerger *et al.* develop an asymmetric network-on-chip organization that accounts for the various network attributes (Jerger et al., 2014). Though these works implement efficient communication network designs, the basic fabric underneath such designs uses electrical link technology that underperforms severely due to its constrained bandwidth and long latencies.

2.1.2 Main Memory in 2.5D Manycore Systems

A primary benefit with 2.5D manycore chips is the integration of memory modules on the same interposer as compute chiplets in contrast to 2D manycore chips that have processors and memory as separate dies. An interposer-based design in 2.5D chips also decouples the size of the processor chip from the memory stacks, which is an issue with 3D-integrated memory-processor chips (Loh et al., 2015). As a result, a larger size of memory chips can be integrated on the interposer. Current memory chips provide fixed bandwidth per stack. Integrating multiple such memory modules on an interposer increases the overall memory capacity as well as the peak memory bandwidth of the system. As an example, the HBM has a data transfer rate of 1024 bits operating at $1GT/s$, yielding a memory bandwidth of $128GB/s$ (JEDEC, 2013). With 8 HBM stacks integrated on the interposer and exposed to the compute chiplets, the total available bandwidth grows to $1TB/s$ (Loh et al., 2015).

2.2 Silicon-Photonic Link Technology

Silicon-photonic links enable data transfers at the speed of light. Compared to conventional electrical links, silicon-photonic links provide high-bandwidth-density at lower latencies and negligible data-dependent power. Silicon-photonic links are, therefore, emerging as a promising network solution in 2.5D manycore systems for data-centric applications. The feasibility of on-chip integration of optical devices such as photodiodes, low-loss waveguides, grating couplers, and MRR modulators and filters through slightly adapted or unmodified CMOS process has revolutionized the design of silicon-photonic links (Virot et al., 2014; Cardenas et al., 2009; Wade et al., 2015; Bogaerts et al., 2012).

With the maturity of silicon-photonic links for chip-scale communication, several industrial and academic efforts have focused on designing 2.5D manycore systems with such links. Oracle developed the *Macrochip* (Koka et al., 2010), which integrates multiple manycore processors in a single package with silicon-photonic links, yielding high inter-die communication bandwidth. *Galaxy* is a multi-chip architecture that integrates multiple small chiplets through optical fibers and incorporates local electrical signaling for near-communication and photonic waveguides for distant intra-chiplet communication (Demir et al., 2014). Grani *et al.* implement a crossbar-based PNoC using arrayed waveguide grating router on a silicon interposer and demonstrate high bisection bandwidth at low energy-per-bit values (Grani et al., 2017).

2.2.1 Operation of WDM Silicon-Photonic Link

Figure 2·1 illustrates communication via a silicon-photonic link. A laser source emits multiple optical signals with n different resonant wavelengths λ_1, λ_2, ..., λ_n. [1] An optical fiber carries these n signals from the laser source to an on-chip waveguide,

[1] The laser source can be either off-chip or integrated on-chip. In our work, we consider off-chip laser sources to simplify thermal management (Werner et al., 2017).

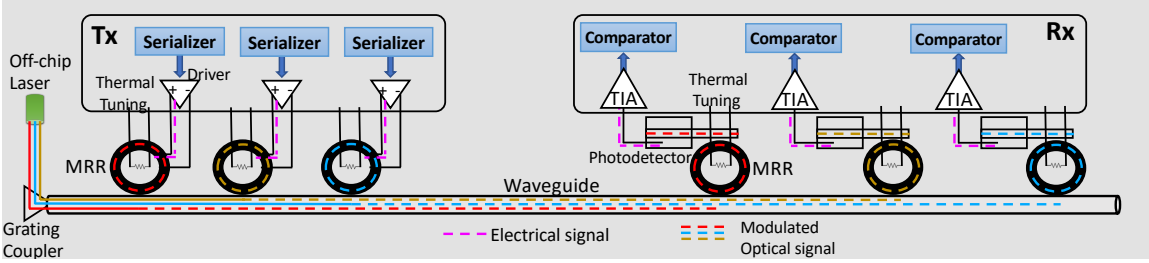


Figure 2·1: An example WDM silicon-photonic link. An off-chip laser emits 3 different optical signals. 3 MRRs at Tx modulate the data onto these 3 optical signals, and 3 MRRs at Rx filter out the data from these 3 optical signals.

where optical coupling is achieved using grating couplers. Owing to WDM, multiple optical signals, each with a distinct resonant wavelength, can coexist in the same waveguide with minimal crosstalk. Prior works have demonstrated up to 32 optical signals in a single waveguide, resulting in dense WDM and, consequently, higher bandwidth density for on-chip communication (Lee et al., 2008).

In Figure 2·1, data is sent over the silicon-photonic link from Tx to Rx. MRRs are used for data modulation at Tx and data filtering at Rx. An MRR utilizes a coupling mechanism to access the optical signal in a waveguide. When the coupled optical wave in an MRR builds up a round trip phase that is an integral multiple of 2π, the MRR is in resonance with it and most of the optical power is diverted from the waveguide to the MRR. A cascade of n MRRs are placed at Tx, each with a resonant wavelength matching one of the n optical signals from the laser. A data packet is first serialized and modulated by an MRR on to one of the optical signals. Similarly, another data packet is serialized and modulated by another MRR on to a second optical signal. The modulated optical signals traverse the silicon-photonic link to Rx. At Rx, another set of n MRRs are placed, each of which resonates as the n optical signals. Each MRR can filter out a modulated optical signal with matching resonant wavelength from the waveguide. The filtered optical signal is then captured by a

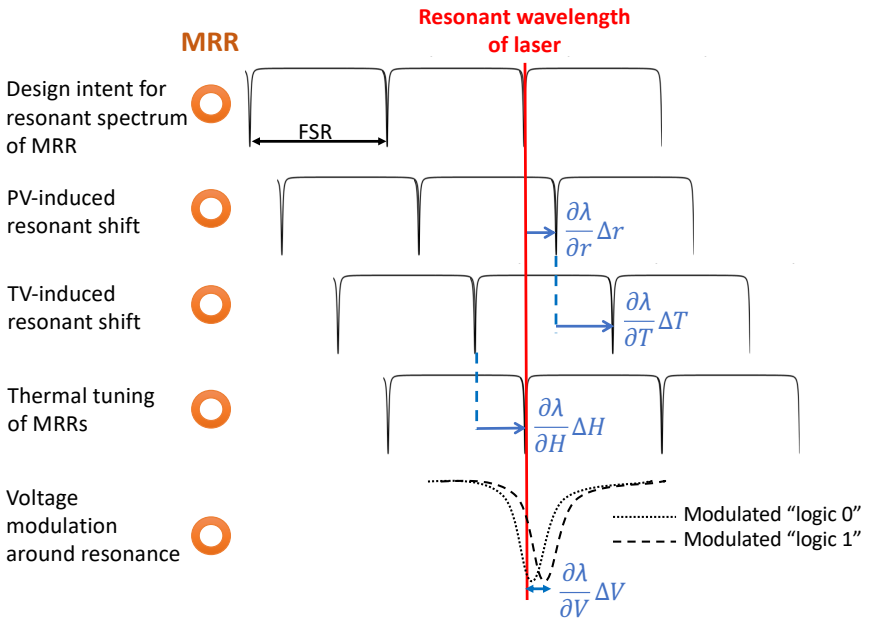


Figure 2·2: MRR sensitivity to TV and PV. MRRs are designed to resonate at peak resonant wavelength of an optical signal. TV and PV induces shifts in the MRR resonant wavelength. The MRRs are supplied with heating power to tune back to laser wavelength.

photodetector (Lischke et al., 2015) that converts the optical signal into an electrical signal. This electrical signal is amplified by a TIA, and read by a set of comparators as either logic 0 or logic 1.

2.2.2 Device-level Characteristics

MRRs are typically fabricated using silicon, which has a high thermo-optic coefficient ($1.86 \times 10^{-4} K^{-1}$) (Densmore et al., 2009). With changes in temperature, the high thermo-optic coefficient induces variations in the refractive index of the MRR, which in turn shifts the MRR resonant wavelength to a higher value, as shown in Figure 2·2. As a result, the MRR moves out of resonance with its coupled optical signal's wavelength. Silicon MRRs have been demonstrated to have a high sensitivity to TV, about $70 - 100 pm/K$ (Padmaraju and Bergman, 2014). With chip temperature gradients rising as high as $20 - 25K$, the MRR resonant wavelength shift due to

TV becomes critically high. Moreover, the MRRs at Tx and Rx experience different resonant wavelength shifts due to the on-chip thermal gradients. The resulting mismatches in the MRR resonant wavelengths at Tx and Rx, therefore, impact the link integrity during data transmission in the silicon-photonic link.

Furthermore, the non-idealities associated with CMOS fabrication process introduce variations in the thickness, width and roughness of the MRRs (Chen et al., 2013). Krishnamoorthy *et al.* quantify the intrawafer and interwafer variations on the resonant wavelengths of MRRs (Krishnamoorthy et al., 2011). Their study shows that absolute resonances of MRRs cannot be controlled across the wafers or even across reticles within a wafer. Due to variations in waveguide width, silicon thickness and etch-depth non-uniformities, the effective refractive index of silicon changes. As a result, the resonant wavelengths of MRRs shift significantly from the design intent, as shown in Figure 2·2. Therefore, during the fabrication process of a die reticle, two distant MRRs in the same die experience completely different shifts in their resonant wavelengths. These PV-induced shifts further add to the TV-induced mismatches in Tx and Rx MRR resonant wavelengths.

It is, therefore, critical to mitigate the effect of TV- and PV-induced resonant wavelength shifts to ensure reliable communication using silicon-photonic links.

2.2.3 Thermal Management in Silicon-Photonic Links

Active control of MRR resonant wavelengths is carried out by thermally tuning an MRR to the higher order resonant wavelength of an optical signal. Figure 2·2 shows the thermal tuning of an MRR, which is achieved by controlled local heat injection using resistive heaters inside the MRRs. These heaters supply energy to the MRRs using Joule effect, thereby increasing the MRR temperature and right-shifting the MRR resonant wavelength (Bahadori et al., 2017a). The MRR, thus, locks on to the higher order wavelength of the optical signal in the wavelength spectrum.

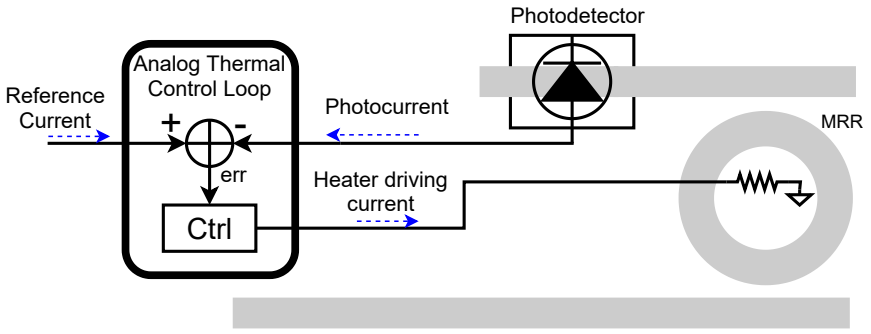


Figure 2·3: An analog thermal control loop compares the photocurrent with a reference current and drives a heater current to thermally tune an MRR.

Thermal Control Loop

Thermal tuning with controlled local heat injection requires a closed-loop feedback system that monitors the MRR resonance shift and the tuning required for an MRR to lock on to an optical signal. As shown in Figure 2·3, this is done by measuring the optical power on the drop port of the MRR with a photodetector. An analog control compares the photocurrent to a reference current that is set based on the MRR resonance. The error signal drives a heater current to thermally tune the MRR using Joule heating. The heater maintains a fixed temperature within the MRR, so that the MRR resonance remains fixed to the resonant wavelength of the optical signal. Several design techniques exist for analog thermal control to close the feedback loop and derive a heating level from the optical monitoring of the drop port (Rakowski et al., 2015; Yu et al., 2015; Sun et al., 2016; Thonnart et al., 2018; Li et al., 2015).

Additionally, a second level of control is required to handle the large temporal TV occurring at runtime. When the large TV introduces an increased shift in the MRR resonant wavelength, thermally tuning the MRR to its original optical channel requires a high heater power. Fortunately, with WDM, the resonant wavelengths of optical channels are evenly spaced in the FSR as shown in Figure 2·4. It is, therefore, possible to thermally tune MRR_0 and lock it to λ_1 instead of λ_0. This

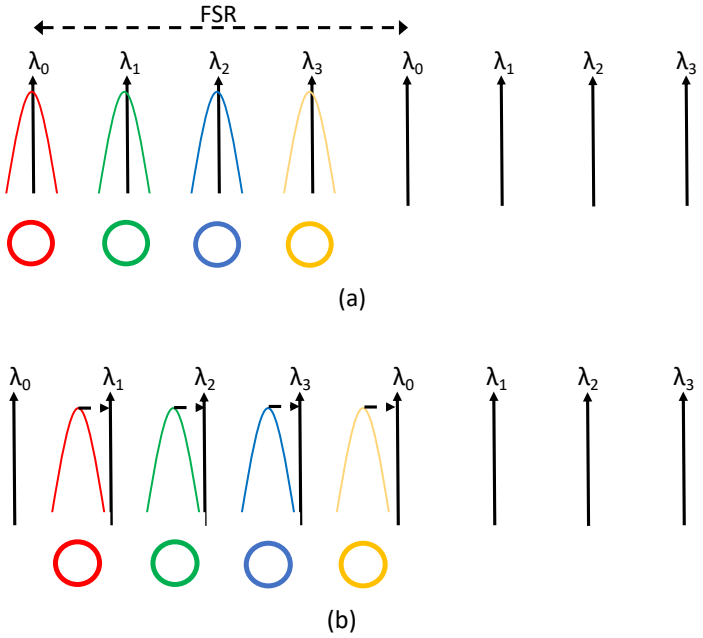


Figure 2·4: (a) The thermal control loop maintains the 4 MRRs at the resonant wavelength of 4 optical signals. (b) A large TV induces a high MRR resonance shift, and the thermal control loop performs thermal remapping to a new set of optical signals.

additional level of control enables the wavelength remapping of MRRs to a different set of optical channels. During an application execution, when the chip temperature increases close to the target MRR temperature, the analog thermal control forces a remapping of MRRs to a different set of optical channels. These remappings between n wavelengths are only possible if the heater efficiency is high enough that it can shift by more than FSR/n with some margin. As remapping requires larger amounts of thermally-controlled shifts, it is a relatively slow process of about $100\mu s$, but occurs less than once per second due to the thermal inertia of chips (Thonnart et al., 2018).

System-level Management

The analog thermal control to thermally tune the MRRs is an effective thermal management technique in silicon-photonic links. However, such device-level techniques do not account for the runtime characteristics of workloads. There is a strong diver-

sity in workloads' runtime bandwidth and resource utilization that result in highly workload-specific power and thermal profiles. The heating power for MRR thermal tuning is, therefore, a strong function of the system architecture and runtime application behavior. Prior work on system-level management focuses on cross-layer optimization methodologies that model device and design-level thermal management strategies under different system-level constraints.

RingAware (Zhang et al., 2014), *Therma* (Beigi and Memik, 2016) and *FreqAlign* (Abellán et al., 2017) employ thread allocation and migration to reduce the thermal variations around communicating MRRs. *Aurora* (Li et al., 2015) encompasses a cross-layer approach at the device, system and OS-level to control the thermal tuning power. *LIBRA* (Thakkar and Pasricha, 2018) uses a reactive technique at device-level and a proactive thread migration policy at system-level to reduce the impact of TV- and PV-induced MRR resonant shifts.

2.2.4 Bandwidth Allocation in Silicon-Photonic Links

In addition to the heating power for MRR thermal tuning, the power consumed in the laser sources and the circuitry for E-O and O-E conversion form a major portion of the overall photonic power. A high density of multiplexed optical signals is used in WDM silicon-photonic link to deliver increased bandwidth for data-centric applications. Consequently, the photonic power increases linearly with the number of optical signals in the silicon-photonic link (Bahadori et al., 2016). It is, therefore, critical to address the trade-off between achieving high bandwidth and reducing photonic power consumption and ensure sub-pJ operation at $> 1Tbps$ on-chip bandwidths.

System-level bandwidth allocation techniques are implemented by assigning optical channels depending on the bandwidth requirements of applications. Several studies perform bandwidth allocation in different contexts by enabling a higher number of channels for maximum aggregated bandwidth (Bahadori et al., 2016), or via

optimized wavelength allocation in silicon-photonic links based on application task graph (Luo et al., 2018), or using an arbitration-free shared-channel silicon-photonic link (Zulfiqar et al., 2013), among others. Winkle *et al.* design a learning-based technique using silicon-photonic link utilization to determine the optimal number of channels (Van Winkle et al., 2018). Chen *et al.* perform runtime bandwidth allocation on clos and butterfly network topologies based on latency at each application phase (Chen and Joshi, 2013). *R-3PO* is a reconfigurable 3D-integrated silicon-photonic network that monitors the bandwidth availability and performs runtime reconfiguration of network bandwidth (Morris et al., 2012).

A key missing aspect in these bandwidth allocation policies arises from a lack of characterization models of MRR device-level sensitivities. The thermal control loop uses a continuous mechanism to monitor the TV and PV sensitivities of on-chip MRRs. A large temperature drift during an application execution results in a major shift in MRR resonant wavelength and, therefore, requires MRR remapping to a new set of optical channels. This remapping provides the opportunity to remap to a new set of optical channels that result in minimal thermal tuning power. The prior works for bandwidth allocation do not model the thermal control loop, leaving an open opportunity to incorporate MRR remapping to reduce the thermal tuning power.

2.3 Optically-controlled Phase Change Memory

Though silicon-photonic links enable high-bandwidth and low-latency chip-scale network designs, the system performance is still hampered by internal bandwidth and latency of main memory systems. With DRAM technology facing critical scaling challenges at lower technology nodes (Kim et al., 2010; Kim and Popovici, 2018), memory vendors and academic researchers are focusing their efforts towards developing non-volatile and scalable memory systems. Non-volatile memories such as memresistor

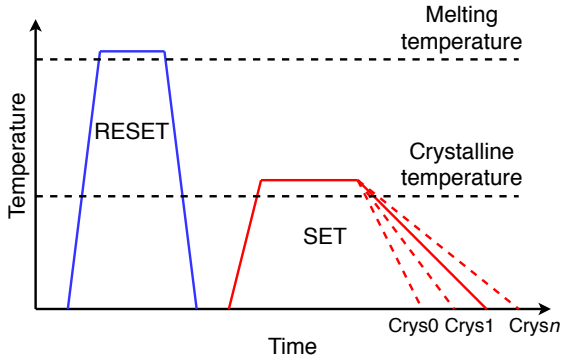


Figure 2·5: Operating principle of a GST element. **RESET:** The GST element is heated to its melting temperature and rapidly cooled to change to a-GST. **SET:** The GST element is heated to its crystalline temperature and gradually cooled to change to c-GST.

arrays, spin-transfer torque magnetic RAM, NAND Flash memory, and PCM have emerged as promising non-volatile alternatives to DRAM (Rho et al., 2017; Kwon et al., 2015; Lee et al., 2009; Kim et al., 2019; Zhang et al., 2019; Bhattacharjee et al., 2017). PCMs outperform many other NVM candidates owing to their higher reliability, increased bit density, and better write endurance (Bedeschi et al., 2008; Burr et al., 2010; Wuttig et al., 2017; Nirschl et al., 2007). In this section, we first study the properties of phase change materials. We then look at the operation and challenges of conventional electrically-controlled PCM cells. We contrast OPCM cells to EPCM cells and present promising opportunities to design high-throughput and scalable main memory systems using OPCM technology.

2.3.1 Properties of Phase Change Materials

Phase change materials typically exist in a fully-amorphous state or a fully-crystalline state with high stability. These states have distinct electrical (resistance) and optical properties (refractive index), which can be used to map the data bit to the state of the material, i.e. logic 0 to amorphous and logic 1 to crystalline state (Ovshinsky, 1968; Wuttig and Yamada, 2007; Burr et al., 2010). We can rapidly switch between

the two states using electrical heating (Raoux et al., 2014; Choi et al., 2012) or high intensity optical pulses (Zhang et al., 2017; Tanaka et al., 2012; Ríos et al., 2015).

Chalcogenides (e.g. S, Se or Te) are well-known phase change materials that exhibit high contrast in electrical properties and optical properties of the two states (Wuttig and Yamada, 2007). $Ge_2Sb_2Te_5$ (GST) is a chalcogenide that has been widely explored due to its long data retention time (up to years), nearly zero leakage power, and nanoscale size (Wuttig and Yamada, 2007; Lyeo et al., 2006; Rios et al., 2014). Moreover, it is also possible to partially crystallize GST to an intermediate state between the fully-amorphous and the fully-crystalline state with high reproducibility. These partially crystalline states have unique distinguishable electrical and optical properties, enabling multi-level storage capabilities (Bedeschi et al., 2008; Nirschl et al., 2007).

The operating principle of a GST element is shown in Figure 2·5. RESET operation results in the amorphization of the GST element, and SET operation results in crystallizing the GST element. We refer to GST in amorphous state as a-GST and GST in crystalline state as c-GST. The GST element is RESET by heating it above its melting temperature ($\sim 600^oC$ (Yamada et al., 1991)), where the material loses its crystalline state and transforms into an unordered state. The material is then rapidly cooled to retain its amorphous state. The GST element is SET by heating it to the crystallization temperature ($100 - 150^oC$ (Yamada et al., 1991)). The material is maintained at this temperature to enable atomic reordering. The heating energy applied to the GST material is slowly released to gradually cool down the material and induce crystal growth. The rate of energy release determines the partial crystallization state of the GST material.

2.3.2 Issues with an Electrically-controlled PCM Cell

The structure of an EPCM cell is similar to that of a DRAM cell (Lee et al., 2009). The cell consists of 1 access transistor and 1 GST element. The read and write (SET or RESET) operations are performed by passing different currents through the cell. EPCM cells, however, suffer from critical issues that limit the adoption of EPCM systems as a main memory alternative to DRAM systems. (1) EPCM cells utilize the resistance values of GST element at different states to distinguish the states. However, the resistance of a-GST and c-GST drifts over time (Li et al., 2012; Karpov et al., 2007). Due to this drift, we need a larger noise margin, which limits the MLC capability to $2bits/cell$. (2) EPCM cells have longer SET latencies, which increases the write latency compared to DRAM cells. The long write latencies slow down the critical read requests by $2-3\times$, thereby impacting the performance of data-centric applications. (3) Furthermore, RESET operation of EPCM cells requires large drive current to amorphize the GST element (Lee et al., 2009). The power consumed in the charge pumps to supply the current is $5\times$ higher than DRAM cells, which severely impacts the energy efficiency of EPCM systems (Kim et al., 2019). (4) Phase change materials such as GST suffer from wearout due to repeated switching of states. Typically, EPCM cells can endure about $10^7 - 10^8$ (Qureshi et al., 2009a) writes before wearing out, compared to DRAM cells ($> 10^{15}$ (Chang et al., 2016b)). Hence the average lifetime of EPCM systems is a critical concern.

Prior works on designing EPCM systems have focused on addressing the long write latencies, high RESET energies and low write endurance. Some of these techniques to hide long write latencies and reduce RESET energy include fine-grained power budgeting (Jiang et al., 2012a), write truncation (Jiang et al., 2012b), dynamic write consolidation (Xia et al., 2014), logical decoupling and mapping (Yoon et al., 2014), proactive SET (Qureshi et al., 2012), partition-aware scheduling (Song et al., 2019),

double-XOR mapping (Du et al., 2013) and boosting rank parallelism (Arjomand et al., 2016). Promising techniques to enhance the write endurance include rotation-based wear leveling (Qureshi et al., 2009a), process variation-aware leveling (Dong et al., 2011; Zhao et al., 2014), and writeback minimization and endurance management (Ferreira et al., 2010). Despite these promising design strategies, memory systems designed using EPCM cells are still not a viable alternative to DRAM due to their lower system performance and increased energy-per-bit.

2.3.3 Operation of an Optically-controlled PCM Cell

Figure 2·6 shows the structure of an OPCM cell, where the GST is integrated on a waveguide (Ríos et al., 2015; Li et al., 2019). An OPCM cell consists of only a GST element, and does not use a separate access transistor like in an EPCM cell. The waveguides are fabricated using a Si_3N_4 layer deposited over a SiO_2 layer (Li et al., 2020). The GST layer is covered with a layer of Indium-Tin-Oxide to prevent oxidation. The optical signals to read and write the OPCM cell lie in the C band ($1530nm - 1565nm$) and L band ($1565nm - 1625nm$) of the telecommunication spectrum (Li et al., 2020).

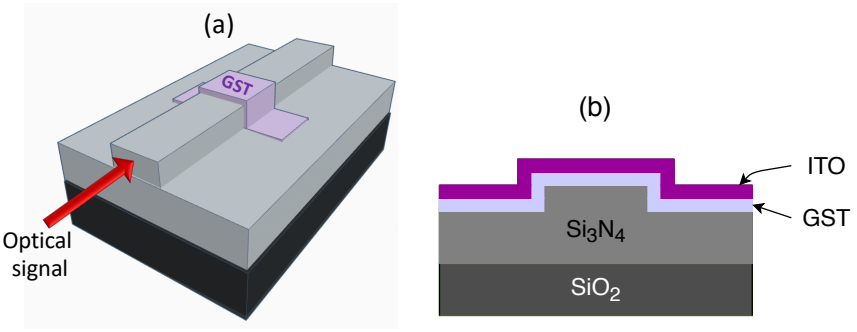


Figure 2·6: (a) 3D view of a GST-based PCM cell that is controlled using optical signals. (b) Cross-sectional view of OPCM cell, where the GST deposited on a Si_3N_4 waveguide.

Write and Read Operation of OPCM Cell

The write operation of an OPCM cell, i.e., either SET or RESET, is performed by passing an optical signal along the waveguide. The optical signal is coupled to the GST element and the energy of this signal triggers a state transition. For RESET operation, an optical pulse of $180pJ$ is passed through the GST element for $25ns$ (Li et al., 2019). For SET operation, an optical pulse of $130pJ$ is passed through the GST element for $250ns$ (Li et al., 2019). Optical pulses of varying energies between $60 - 130pJ$ can be applied to transition the GST element to a desired partially crystalline states (Li et al., 2019).

The contrast in the refractive indices of a-GST (3.56) and c-GST (6.33) enables readout of stored data in the GST element (Michel et al., 2014). When an optical signal is passed through the GST element, the higher refractive index of c-GST results in an increased optical absorption by the GST element. Rios *et al.* demonstrate that c-GST absorbs 79% of the input optical signal and allows transmission of only 21% of the optical signal (Ríos et al., 2015). In contrast, a-GST transmits 100% of the optical signal. The partial crystalline states allow transmission between 100% and 21% (Ríos et al., 2015). The data is, therefore, read out by sending sub-ns optical pulses through the GST element and measuring the transmitted optical intensity.

MLC Capability of OPCM Cells

In OPCM cells, the read operation uses the distinct refractive index of each state to determine the stored value. Unlike the resistance value used in EPCM cells, refractive index experiences minimal or no drift over time (Li et al., 2019; Ríos et al., 2015). This enables designing OPCM cells with higher number of stable partially crystalline states having unique refractive indices. Thus, each OPCM cell supports higher $bits/cell$ than an EPCM cell. Prior works have demonstrated that it is possible to reliably program the GST element using optical signals to contain more than 34

unique partially crystalline states (Li et al., 2019; Youngblood et al., 2019), which corresponds to 5 $bits/cell$. Using a higher capacity MLC enables the read and write operation of a higher number of bits per access for the same number of processor-to-memory links, thereby increasing the memory throughput. Theoretically, an OPCM with 4 $bits/cell$ provides 2× higher peak memory throughput than a typical EPCM with 2 $bits/cell$. With OPCM cells projected to support 8 $bits/cell$ in the near future, we can obtain 4× higher peak memory throughput than an EPCM with 2 $bits/cell$.

2.3.4 High-throughput Access with Silicon-Photonic Links

Section 2.2 introduces the promise of silicon-photonic links as high-bandwidth-density networks in manycore systems. Silicon-photonic links have been extensively explored for high-bandwidth and low-energy communication between processor and memory in prior works. Beamer *et al.* design a joint silicon-photonic link and electro-photonic DRAM design to provide high internal bandwidth (Beamer et al., 2009). However, the O-E-O conversion in such DRAM designs adds to the latency overhead. An OPCM system with silicon-photonic links presents the opportunity for optical signals to directly read/write the OPCM cells, eliminating the O-E-O conversion overhead. Despite OPCM cells suffering from long SET latencies similar to EPCM cells, the increased MLC capability and direct access using dense WDM silicon-photonic links increase the peak memory throughput.

2.4 Memory Management in Heterogeneous Memory Systems

Traditionally, computing systems consist of homogeneous memory modules as the primary main memory. The key attributes of main memory are its power, bandwidth, latency, non-volatility, scalability and area density. An ideal main memory system should be highly scalable and non-volatile, provide high bandwidth ($> 1TBps$) at low latency ($< 10ns$), low energy-per-bit ($< 10pJ/bit$) and high area density

($> 50MB/mm^2$). Unfortunately, due to the trade-off between these metrics, there is not a single memory module that can provide all the above features. Heterogeneous memory systems consisting of multiple memory modules are, therefore, becoming prominent in a wide variety of systems from embedded systems to modern data centers and cloud systems (Phadke and Narayanasamy, 2011; Chatterjee et al., 2012; Sodani, 2015; Kannan et al., 2017; Olarig et al., 2003; Avissar et al., 2001; Gai et al., 2016). This section introduces the academic works and industrial products using heterogeneous memory systems and then discusses the state-of-the-art techniques for memory management in such systems.

2.4.1 Heterogeneous Memory Systems

To cater to the wide diversity in application memory requirements, memory vendors provide memory modules with different performance and power characteristics. For example, RLDRAM is a memory type optimized for low access latency, which makes it ideal for switch and router applications (Toal et al., 2007). However, the static and dynamic power consumption of RLDRAM is $4-5\times$ higher than a DDR3/DDR4 module, and the bandwidth is lower. On the other hand, LPDDR reduces power consumption substantially, but has higher access latency and lower bandwidth; thus, it is attractive for mobile platforms (Rho et al., 2017). HBM is an innovative memory technology that stacks multiple DRAM layers vertically, where layers are connected by TSVs. HBM2.0 boasts of more channels per device, smaller page sizes per bank, wider activation windows and a dual command line for simultaneous read and write (Lee et al., 2016b). These features distinguish HBMs to provide performance and power improvements in case of bandwidth-sensitive workloads. Moreover, as we saw from Section 2.3.2, NVM systems such as EPCM are superior over DRAM-based systems in terms of higher scalability, lower leakage power and non-volatility. However, they suffer from long write latencies and high write energies, which renders them impractical

as primary main memory modules.

In spite of the promising memory technologies, there is no single memory module that can provide the the ideal benefits of non-volatility, high area density, lowest latency, highest bandwidth, and lowest power consumption at the same time. Therefore, homogeneous memory systems are often not sufficient in an era of diverse compute- and memory-intensive workloads. Motivated by performance-power trade-off among various memory modules for diverse workloads, heterogeneous memory systems have been proposed to improve performance and energy efficiency of computing systems. Several prior works design heterogeneous memory systems consisting of either on-chip scratchpad memory (Shen et al., 2016; Peón-quirós et al., 2015) or 3D-stacked memory (Meswani et al., 2015; Tran et al., 2013; Dong et al., 2010). Heterogeneous memory systems have also been proposed for reduced data processing in cloud computing (Gai et al., 2016; Kannan et al., 2017). Hybrid memory systems with DRAM and NVM systems have been widely explored due to the additional benefits of scalability and data persistence obtained using these NVM modules (Dulloor et al., 2016; Pavlovic et al., 2013; Khouzani et al., 2016). Commercial products such as Intel's Knights Landing processor (Sodani, 2015) and Knights Mill processor (Bradford et al., 2017) have an on-chip HBM together with an off-chip DDR4, and AMD Radeon Fury X (Macri, 2015) consists of an interposer with an HBM stack along with DDR3 memory. In 2015, Intel unveiled its Optane memory line based on 3D-Xpoint technology, which has high density similar to DRAM, but with additional benefits of scalability and data persistence (Hady et al., 2017).

2.4.2 Page Allocation in Heterogeneous Memory Systems

With multiple memory modules in a heterogeneous memory system offering varied power-performance benefits, it is critical to develop a memory management policy that can best utilize the benefits of different modules in such a system. This policy

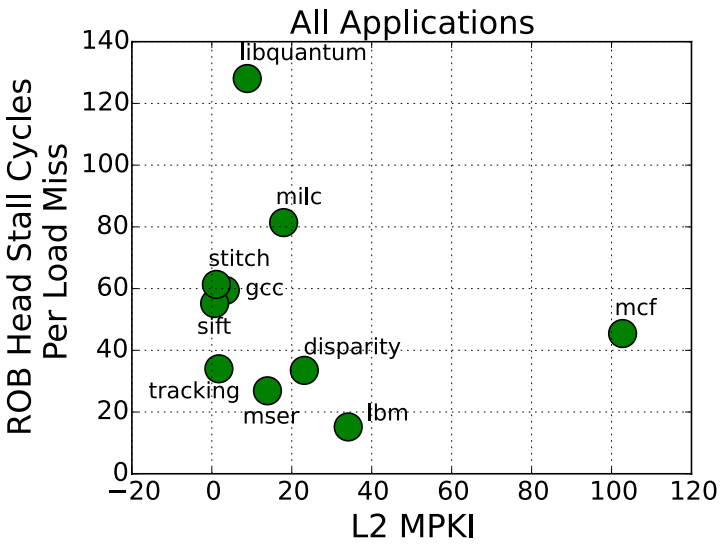


Figure 2·7: Memory access behavior of selected applications from SPEC CPU2006 and SDVBS benchmarks. A high L2 MPKI indicates that the application is memory-intensive. A low number of ROB stall cycles for a memory-intensive application implies high memory-level parallelism.

needs to account for the application's memory requirement and allocate memory pages from a module that is best suited to the application. A wide diversity in applications' memory characteristics necessitates a robust and systematic page allocation policy. Figure 2·7 plots the memory access behavior of applications from SPEC CPU2006 benchmark (Henning, 2006) and SDVBS benchmark (Venkata et al., 2009). The L2 MPKI and ROB head stall time specifies the memory intensity and the MLP of applications, respectively. From this figure, we observe that applications have diverse memory intensities and exhibit different MLP. Achieving high energy efficiency and system performance is, therefore, contingent upon placing an application's data in the right memory module.

Phadke *et al.* introduce an application-level allocation policy for heterogeneous memory systems (Phadke and Narayanasamy, 2011). They profile the memory access behavior of every application as a whole and allocate the entire application to

the best-fit memory module. Other works employ optimal page-level allocation policies to utilize the lowest latency memory module by tracking frequently accessed pages (Meswani et al., 2015; Dong et al., 2010; Pavlovic et al., 2013), or controlling the amount of memory-mapped based on bandwidth utilization (Tran et al., 2013). Chatterjee *et al.* place critical words in a cache line in latency-optimized memory module and rest of the cache line in power-optimized modules (Chatterjee et al., 2012). Shen *et al.* use PIN-based profiling to track array allocations for placing frequently accessed and low-locality arrays in the on-chip scratchpad (Shen et al., 2016). Dulloor *et al.* profile memory access patterns of data structures as either sequential, random, or involving pointer chasing to place either in DRAM or PCM (Dulloor et al., 2016). Peon-Quiros *et al.* track the access frequency and changing memory footprint over time of dynamically allocated data structures to place them in either on-chip SRAM or off-chip DRAM modules (Peon-Quiros et al., 2015). Intel's Knights Landing processor enables the programmer to explicitly allocate workloads' critical data in HBM using built-in APIs or compiler annotations (Sodani, 2015).

These memory management policies, though promising, perform page allocation at a much coarser granularity. As the datasets of application are growing at an exponential rate, there exists increased diversity in memory characteristics even within a single application. Memory management at a coarser granularity such as application-level, cache-line or highly-accessed pages leaves the benefits of the heterogeneous memory system under-utilized. Furthermore, built-in APIs for memory management puts the burden on the programmer to explicitly define these APIs in the application source code.

2.5 Distinguishing Aspects of this Thesis

This thesis identifies key gaps in architectural design and system management in leveraging the promising benefits of silicon-photonics technology and optical phase change memory. The major distinguishing aspects of this thesis in contrast to prior state-of-the-art works are as follows:

- We design a simulation framework that characterizes the PV and TV sensitivities of MRRs and includes a model of the thermal control loop for thermal tuning of MRRs. We are the first to model this thermal control loop, which enables runtime thermal remapping of MRRs to the nearest resonant peak of an optical signal. With the help of this thermal control model, the bandwidth allocation policies developed in this thesis, *SO-WAVES* (Narayan et al., 2019) and *PROWAVES* (Narayan et al., 2020b) reduces the photonic power substantially, thus increasing the energy efficiency of silicon-photonic links. To further improve these power management policies, we perform application instrumentation on the software stack that can assist *SO-WAVES* and *PROWAVES* (Narayan et al., 2020a). We also present the efficacy of silicon-photonic links for graph workloads and discuss the architectural considerations of systems with silicon-photonic links (Narayan et al., 2020c). These system management policies enable an energy-efficient design of chip-scale communication networks as well as processor-to-memory networks.

- We are the first to design an optically-controlled non-volatile main memory that can be directly accessed with silicon-photonic links. Our Co-designed Optical phaSe change Memory and Optical link System, *COSMOS*, provides increased memory throughput due to silicon-photonic links and increased bit density per cell. *COSMOS* includes a hierarchical design of OPCM array microarchitecture

with novel read/write access protocol. The design of the OPCM array combines WDM and mode-division-multiplexing properties of optical signals to deliver high internal memory bandwidth. We design an E-O-E control unit for seamless integration of *COSMOS* with the processor. This E-O-E control unit receives standard DRAM protocol commands from the processor, and converts them into OPCM address, data, and control signals that are mapped onto optical signals.

- With workloads exhibiting strong diversity in their memory characteristics, a single memory module such as DDR4, LPDDR, RLDRAM, *COSMOS*, etc. fails to sufficiently satisfy a wide range of diverse memory needs. With heterogeneous memory systems gaining popularity in a variety of computing systems, we present a case for developing memory management policies at a fine granularity. Our proposed framework, Memory Object Classification and Allocation, *MOCA* (Narayan et al., 2018), performs memory managemenet at the granularity of memory objects that are allocated in the heap space. *MOCA* profiles an application and allocates memory objects within an application to memory modules that are best suited to the objects' memory characteristics.

Chapter 3

System-level Management of Silicon-Photonic Links

With the emergence of CMOS-integrated silicon-photonic technology, photonic links are being widely adopted as chip-scale networks since they provide high bandwidth density and low latencies at minimal energy-per-bit communication. However, a limiting factor towards the wide-scale adoption of silicon-photonic links arises from the high power overhead in the laser, the circuitry for electrical-optical conversion and thermal tuning. In this chapter, we characterize the network bandwidth requirements in applications and present system-level policies that limit the photonic power with minimal impact on application performance.

The chapter begins with the architectural description of a 2.5D manycore chip that uses silicon-photonic links. We describe our cross-layer framework that models the device-level sensitivities of silicon-photonic devices and the architectural details of the 2.5D manycore chip, and evaluates the system-level performance, power and thermal profile for different workloads. We present our system-level management policy called wavelength selection (*WAVES*) to provide a power-efficient operation of silicon-photonic links for a range of communication-intensive big data workloads. We then demonstrate the efficacy of silicon-photonic links for graph workloads and present the architectural opportunities in redesigning memory hierarchies with photonic links. Next, we demonstrate the benefits of software-level application instrumentation on top of wavelength selection policies for further reducing the photonic power.

3.1 2.5D Manycore System with Silicon-Photonic Links

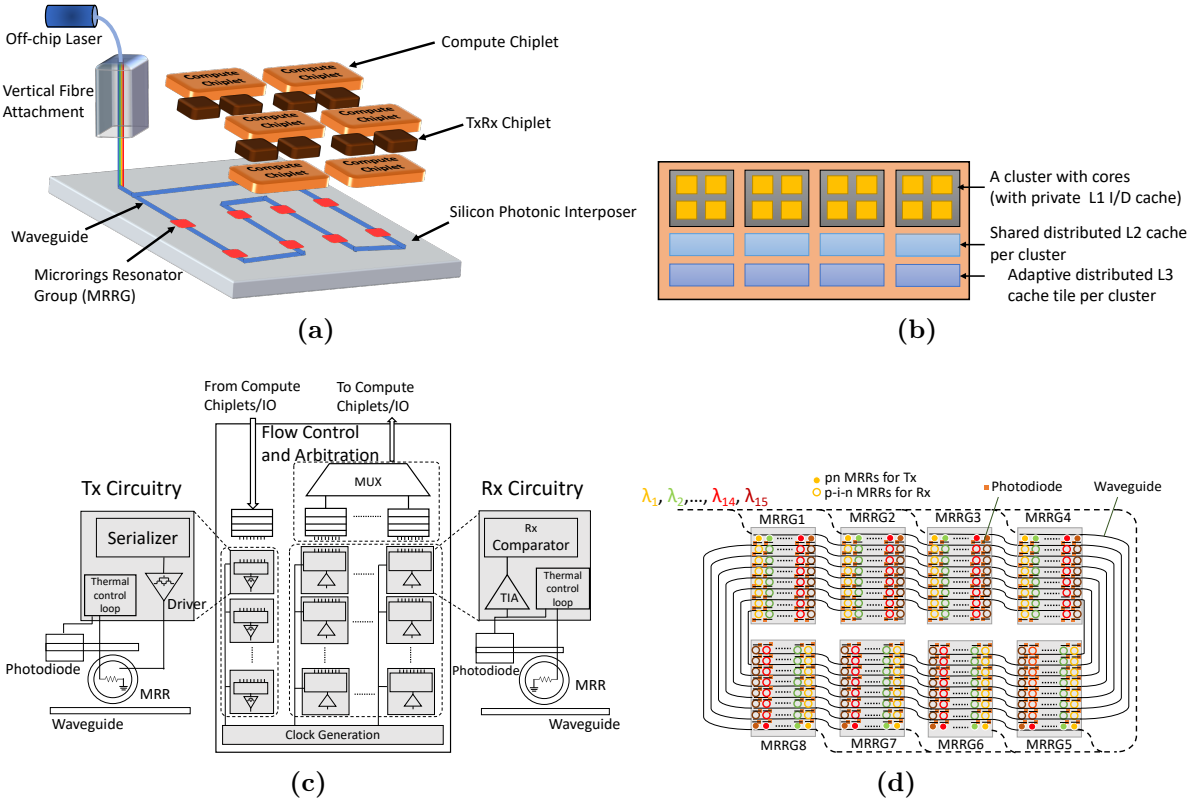


Figure 3·1: *POPSTAR* architecture. (a) 2.5D manycore chip with six compute chiplets and eight TxRx chiplets that are integrated on a photonic interposer. (b) Architecture of a compute chiplet, with four clusters and four cores per cluster. (c) Architecture of a TxRx chiplet, with circuitry for data modulation, data filtering, flow control and arbitration. (d) SWMR routing of optical channels and MRRG assignment.

In our work, we use a homogeneous 2.5D manycore chip with integrated silicon-photonic interconnect on the interposer, called Processor On Photonic Silicon in-Terposer ARchitecture (*POPSTAR*) (Thonnart et al., 2020). Figure 3·1 shows the complete architecture and organization of *POPSTAR*. *POPSTAR* consists of 96 cores that are organized into six compute chiplets. The analog and digital circuitry that handle the photonic communication are organized into eight TxRx chiplets. The

compute and the TxRx chiplets are integrated on a photonic interposer. In *POP-STAR*, we consider off-chip laser sources that emit optical signals onto the photonic interposer through a fiber attachment. Vertical grating couplers couple these optical signals between the waveguides on the interposer and the fiber attachment.

Compute Chiplets

POPSTAR comprises of 96 IA-32 cores from SCC (Howard et al., 2010). These cores are organized into six compute chiplets, each containing 16 cores. Within a chiplet, the 16 cores are further organized into four clusters of 4 cores each, as shown in Figure 3·1b. Each core has a private L1 I/D cache of $16KB$. There is a shared distributed L2-cache with $256KB$ per cluster, and a distributed L3-cache, with 4 L3 tiles ($4 \times 1MB$) per compute chiplet.

TxRx Chiplets

A compute chiplet accesses the silicon-photonic link on the interposer via a TxRx chiplet. The TxRx chiplets are composed of analog and digital circuitry required for modulating digital data on optical signals, and converting the data received on optical signals back into digital data. Figure 3·1c shows the architecture of a TxRx chiplet. There are six TxRx chiplets that are connected to the six compute chiplets, and two TxRx chiplets are connected to the external peripherals, IOs, and memory controllers. For data modulation, the TxRx chiplet uses a serializer and a modulation driver for every wavelength of optical signals in the system. Similarly, for data filtering, there is a TIA and a comparator circuit for every wavelength of optical signals. An analog thermal loop (Thonnart et al., 2018) detects the photodiode current, compares it with a reference bias current and supplies heating power to thermally tune the MRRs so that the detected photocurrent is equal to the reference current. The TxRx chiplet uses FIFO queues and multiplexers to handle the flow control.

Network Architecture

The silicon-photonic network on the interposer handles the data and coherence traffic between the chiplets and main memory. The global network topology connecting the TxRx chiplets is a Single-Writer Multiple Reader (SWMR) topology. The optical channels are mapped onto a U-shaped spiral of waveguides on the photonic interposer, as shown in Figure 3·1d. Each TxRx chiplet can send data over any of the optical signals in the system. The data passes through the appropriate waveguide and is routed to the destination TxRx chiplet, where the data is filtered out by the photodetector. The data rate of each optical channel is $12Gbps$, resulting in a peak aggregate bandwidth of $1.5Tbps$ on the interposer with 16 optical channels.

MRRG Architecture

Each of the 8 TxRx chiplets has a set of MRRs organized underneath into an MRRG. An MRRG consists of 16 WDM bundle of MRRs, with each WDM bundle operating at a different optical channel. For each optical channel, an MRRG consists of a single Tx MRR for data transmission to seven other TxRx chiplets and seven Rx MRRs for receiving data from seven other TxRx chiplets. The Tx MRR in an MRRG modulates data on one optical channel, which traverses through the silicon-photonic link to the other MRRGs. The seven Rx MRRs in an MRRG are utilized to receive data from MRRGs in other seven TxRx chiplets. The MRRs have a radius of $10\mu m$, and designed around a center wavelength of $1310nm$ with an FSR of $10.8nm$. Thermal tuning of MRRs is achieved via dedicated local heaters.

Table 3.1: Notations used in modeling of silicon-photonic links.

Notation	Description
C	Number of TxRx chiplets (and waveguides)
λ_{tot}	# available optical channels in the system
λ_{act}	# activated optical channels
λ_{min}	Minimum # optical channels required for an application
$\Delta\lambda_{shift}$	MRR wavelength shift due to PV and TV
$\Delta\lambda_{heat}$	Thermal tuning shift required for an MRR
$\frac{d\lambda}{dH}$	Heater efficiency
P_{heat}	Heating power for MRR thermal tuning
P_{laser}	Overall laser power in the system

3.2 Cross-layer Simulation Framework for Silicon-Photonic Links

We design a simulation framework to evaluate the runtime characteristics of workloads on *POPSTAR*. Our simulation framework is a cross-layer approach that models the impact at different levels in the computing stack. (a) At the device-level, we model the impact of TV and PV on the MRR resonant wavelength, and the analog thermal control loop that enables thermal remapping during different application phases, (b) at the architectural-level, we model the processor architecture, communication traffic arising in the silicon-photonic link and power consumed in different circuit elements in the TxRx chiplet, and (c) at the system-level, we model performance, power and thermal profile of workloads when executed on *POPSTAR*, and implement our system-level policies to reduce the photonic power. Table 3.1 lists the notations used in modeling the different parameters in our framework.

3.2.1 Device-level Modeling

As explained in Section 2.2.2, the resonant wavelength of MRRs shifts due to variations in temperature and fabrication process imperfections. In large 2.5D systems,

the high compute activity across the chip introduces thermal hot spots and large thermal gradients on the chip, which can reach temperatures $> 85^oC$ for compute-intensive applications (Abellán et al., 2017). These temperature variations are not only temporal, but also spatially lateral, since heat is not uniformly spread across the interposer. Additionally, the process variations are mostly geometric and introduce a random component. During the fabrication process of a die reticle, two distant MRRs in the same die may experience completely different variations in their resonant wavelength. To ensure reliable on-chip communication in the photonic link, it is, therefore, essential to mitigate the impact of thermal and process variations on the resonant wavelength of the MRRs.

For modeling the thermal-variation-induced resonance shifts in MRRs, we consider MRR thermal sensitivity of $78pm/K$ (Thonnart et al., 2018). Given the small area footprint of an MRRG, we assume that all the MRRs within an MRRG are at the same temperature at a given time. As a consequence, all the MRRs within an MRRG undergo the same resonance shift due to thermal variations. Characterization studies at the die and wafer level show that the process variations of MRRs can be modeled as a gaussian distribution (Thonnart et al., 2020). We, therefore, model the local MRRG process variations as a gaussian distribution with a standard deviation of $100pm$. The overall wavelength shift ($\Delta\lambda_{shift}$) for an MRR can be expressed as follows:

$$\Delta\lambda_{shift} = \frac{d\lambda}{dT} \cdot \Delta T + \Delta\lambda_{shift,PV} \ . \tag{3.1}$$

From Section 2.2.2, we saw that active control of resonant wavelength of MRRs is performed by supplying heat to thermally tune the MRR to a higher order resonant wavelength. The heating power to thermally tune the MRR depends on the overall resonant wavelength shift of the MRR, the FSR and the heater sensitivity. Since two adjacent resonant peaks are separated by FSR, the maximum wavelength shift

required for an MRR is one FSR. With WDM, it is possible to multiplex multiple optical signals in a waveguide, with the peak resonant wavelength of each signal evenly spaced in the FSR. Therefore, an MRR can now be tuned to the nearest resonant peak in the FSR. With a total of λ_{tot} optical signals, the maximum wavelength shift required for an MRR is FSR/λ_{tot}. The analog control loop in the TxRx chiplet detects the aggregate resonant shift of each MRR and supplies appropriate heating current to lock the MRR to its nearest resonant peak (Thonnart et al., 2018). This tuning range can be expressed as follows:

$$\Delta\lambda_{heat} = \frac{FSR}{\lambda_{tot}} - (\Delta\lambda_{shift} \ \ mod \ \ \frac{FSR}{\lambda_{tot}}) \tag{3.2}$$

The total heating power in *POPSTAR* can be calculated by aggregating the heating power across all Tx and Rx MRRs. With a heater efficiency ($d\lambda/dH$) of $100pm/mW$, the total heating (P_{heat}) is calculated as follows:

$$P_{heat} = \sum_{i=1}^{C} \sum_{r=1}^{C\cdot\lambda_{act}} \frac{\Delta\lambda_{heat_{ir}}}{\frac{d\lambda}{dH}}. \tag{3.3}$$

3.2.2 Architecture-level Modeling

The system performance and the energy of the silicon-photonic links is highly impacted by its microarchitectural details. We model the core microarchitecture of *POPSTAR*, which is described in Section 3.1. The silicon-photonic link is modeled as SWMR topology with a point-to-point latency of one cycle and a data rate of $12Gbps$. The packets sent on the silicon-photonic link consist of data and coherence accesses by a core to an LLC on a separate chiplet in addition to the main memory accesses.

The network traffic in the silicon-photonic link impacts the laser power and the active power in the circuit elements in the TxRx chiplet. These powers are, therefore, a strong function of the number of optical channels in the system. The laser source power of a single wavelength (P_L) should be higher than the sum of the worst-case

Table 3.2: Power consumption of a laser source and the different active elements in TxRx chiplet for E-O-E conversion (Polster et al., 2016)

Component	Active Power		Idle Power	
	Notation	Value (mW)	Notation	Value (mW)
Laser (wall-plug)	P_L	30		0
Serializer	$P_{srl,a}$	3	$P_{srl,i}$	1
Driver	P_{drv}	3		0
Rx Comparator	$P_{cmp,a}$	1	$P_{cmp,i}$	0.33
TIA	P_{TIA}	2		0
Arbitration and Flow Control	$P_{arb,a}$	32	$P_{arb,i}$	10

power loss in the silicon-photonic link and the photodetector sensitivity. We calculate this value as $30mW$. The overall laser power, P_{laser}, for λ_{act} laser wavelengths can then be expressed as:

$$P_{laser} = P_L \cdot C \cdot \lambda_{act} \ . \tag{3.4}$$

To calculate the power consumed in the EOE circuit elements in the TxRx chiplet, we consider the active and idle power of each element. Table 3.2 displays the active and idle power that are determined from the post-layout simulations using Prime-Time power analysis (Polster et al., 2016). We break down the overall EOE power consumption into the power consumed by the Tx circuitry, Rx circuitry and the logic for arbitration and flow control. Depending on the number of active optical channels in the silicon-photonic link, we express the overall EOE power as follows:

$$P_{Tx} = P_{drv} \cdot \lambda_{act} + P_{srl,a} \cdot \lambda_{act} + P_{srl,i} \cdot (\lambda_{tot} - \lambda_{act}) \ , \tag{3.5}$$

$$P_{Rx} = P_{TIA} \cdot \lambda_{act} + P_{cmp,a} \cdot \lambda_{act} + P_{cmp,i} \cdot (\lambda_{tot} \cdot C - \lambda_{act}) \ , \tag{3.6}$$

$$P_{arb} = P_{arb,a} \cdot \tfrac{\lambda_{act}}{\lambda_{tot}} + P_{arb,i} \cdot \tfrac{\lambda_{tot} - \lambda_{act}}{\lambda_{tot}} \ , \tag{3.7}$$

$$P_{EOE} = C \cdot (P_{Tx} + P_{Rx} + P_{arb}) \ . \tag{3.8}$$

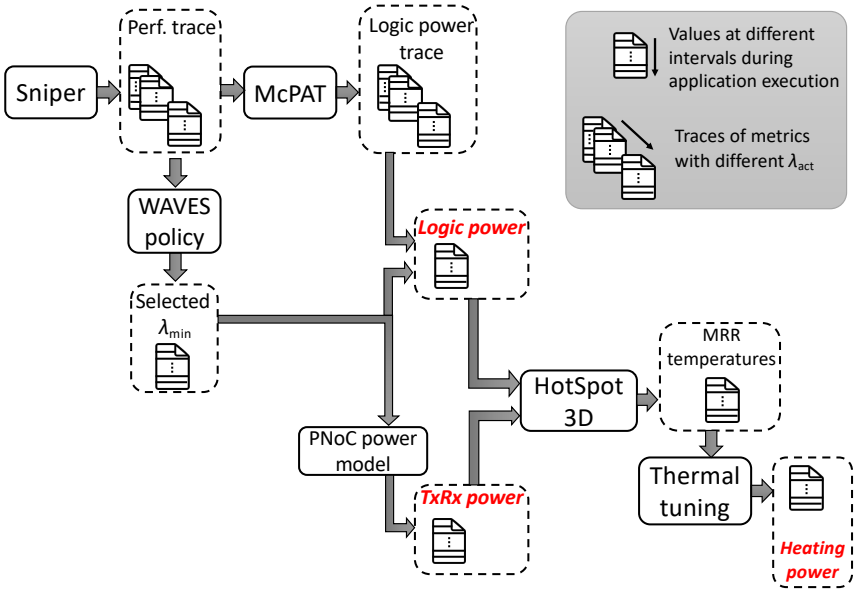


Figure 3·2: Simulation framework for modeling performance, power and temperature of *POPSTAR*.

3.2.3 System-level Performance, Power and Thermal Modeling

To evaluate the performance and power consumption of *POPSTAR* with different wavelength selection policies, we set up a simulation framework that is composed of a performance simulator, a logic core power calculator, a PNoC power model and a thermal simulator. Figure 3·2 depicts our system-level toolflow. For our experiments, we use a diverse set of HPC applications from PARSEC (Bienia et al., 2008), SPLASH-2 (Woo et al., 1995), UHPC (Campbell et al., 2012), HPCCG (Heroux, 2007) and NAS Parallel Benchmark (Bailey et al., 1991). We also conduct experiments on large scale graph processing algorithms from the GAP-BS benchmark (Beamer et al., 2015). For graph applications, we use real-world datasets from the Stanford Large Network Dataset Collection (Leskovec and Krevl, 2014). Table 3.3 details the HPC and graph workloads from these benchmarks.

We model the architectural details of *POPSTAR* in Sniper (Carlson et al., 2011) for simulating the performance of applications. In our simulations, we fast forward the

initial phase of the application execution to the ROI. For each application, we execute 10 billion instructions in the ROI unless the application ROI finishes earlier. During this execution, we collect the performance traces pertaining to the PNoC activity for every interval. The interval size in our experiments is 100 million instructions, unless otherwise stated. To understand the impact of our system-level policies for different system utilization, we run each application with varying number of thread counts.

We use McPAT (Li et al., 2009) for calculating the core and cache power at every interval. We feed the performance statistics from Sniper as input to McPAT. McPAT calculates the dynamic power of all the active elements in the core based on their activity. We calibrate the McPAT power numbers by scaling these numbers to the published average power of the IA-32 core (Howard et al., 2010). We assume that the idle cores are put to sleep and consume negligible power. We calculate the leakage power in the cores using a linear temperature-dependent model. The power consumed in the silicon-photonic link comes from the laser source, the EOE circuit elements in the TxRx chiplet and the heating power to thermally tune the MRRs. We use our analytical model to calculate the laser power (Equation 3.4), EOE power (Equation 3.5-3.8) and the heating power (Equation 3.3) for each interval based on the number of active optical channels at that interval.

Table 3.3: Description of applications from the HPC and Graph Benchmarks used in system simulations of silicon-photonic links.

Application	**Descripion**
mg	Multi-grid on a sequence of meshes
sp	Scalar Penta-diagonal solver
bt	Block Tri-diagonal solver
is	Integer Sort
ft	Discrete 3D Fast Fourier Transform
lu	Lower-Upper Gauss-Siedel Solver
$hpccg$	High Performance Computing Conjugate Gradients
pr	PageRank
bfs	Breadth-First Search
bc	Betweenness Centrality
$sssp$	Single-source Shortest Paths

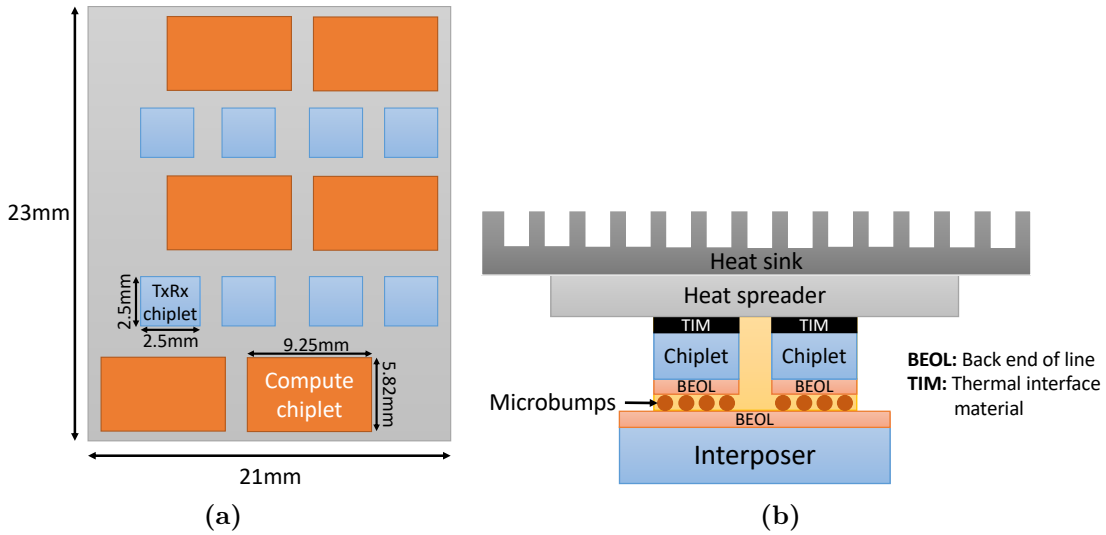
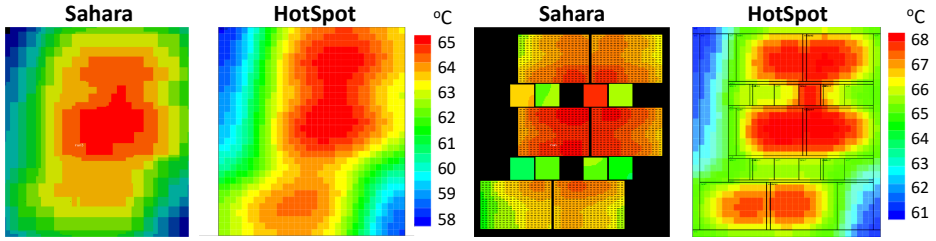


Figure 3·3: (a) Layout of *POPSTAR* along with the dimensions of compute and TxRx chiplets, (b) Cross-sectional view of *POPSTAR* with the different layers in 2.5D integration

We use the 3D extension of HotSpot (Skadron et al., 2003; Meng et al., 2012) to determine the transient temperatures of MRRs. HotSpot uses the power traces for core and caches from McPAT and the power traces for the TxRx chiplet from Equation 3.5-3.8. We model the layout of the compute and TxRx chiplets as shown in Figure 3·3a. The 3D cross-section of *POPSTAR* is shown in Figure 3·3b, where the chiplets are integrated on the interposer via microbumps using BEOL integration technology. For efficient vertical heat dissipation, there is a heat spreader and heat sink over the *POPSTAR* chip. Table 3.4 shows the material properties of different layers. We calibrate the HotSpot temperatures to the temperatures obtained from Project Sahara, which is a signoff thermal tool from Mentor. We obtain HotSpot temperatures within 2% error margin of Project Sahara on average. Figure 3·4 illustrates the thermal map of *POPSTAR* in Project Sahara and HotSpot.

Table 3.4: Material properties and dimensions of different layers in *POPSTAR*

Layer	Thickness (μm)	Thermal conductivity ($W/m.K$)	Specific heat ($J/kg.K$)	Density (kg/m^3)
Heat sink	6900	400	396	8960
Heat spreader	1000	400	396	8960
TIM	10	6.8	900	1300
Chiplets	750	150	700	2330
BEOL	10	145	612	4237
Microbump	Pitch=40, diameter=20	0.86	846	2689
Interposer	750	150	700	2330



(a) Interposer layer. **(b)** Chiplet layer.

Figure 3·4: Thermal map of *POPSTAR* in Sahara tool (Parry and Wang, 2018) and HotSpot tool (Skadron et al., 2003)

3.3 Wavelength Selection for Energy-efficient Silicon-Photonic Links

The high data footprints and the growing on-chip communication traffic in data-centric applications necessitate the design of silicon-photonic links with increased peak bandwidth. The peak aggregate bandwidth of a silicon-photonic link is the product of λ_{act} and the modulation bit rate of an optical channel. For applications with high inter-chiplet communication, a higher λ_{act} provides increased communication bandwidth, and therefore, is desirable for higher performance. Figure 3·5a shows an improvement in application performance as the number of optical channels in the silicon-photonic link increases. However, the overall photonic power consumed in the laser source, EOE circuitry in TxRx chiplet and the MRR thermal tuning also in-

creases with an increase in optical channels. Equations 3.3, 3.4-3.7 and 3.2 show the dependence of laser power, E-O-E power and thermal tuning power, respectively, on the number of active optical channels (λ_{act}) in the silicon-photonic link. Figure 3·5b also shows the rise in overall system power with increasing optical channels. This increased photonic power consumption limits the ability to provide high bandwidth density for applications. From Figure 3·5a, we observe a general trend that the system performance tends to saturate at a particular λ_{act}. As a result, we can activate the minimum number of optical channels, λ_{min}, that sufficiently caters to the required bandwidth needs of an application. We now present our proposed wavelength selection policy (*WAVES*), which can be performed either statically before the application execution (*SO-WAVES*), or dynamically during the application execution (*PROWAVES*).

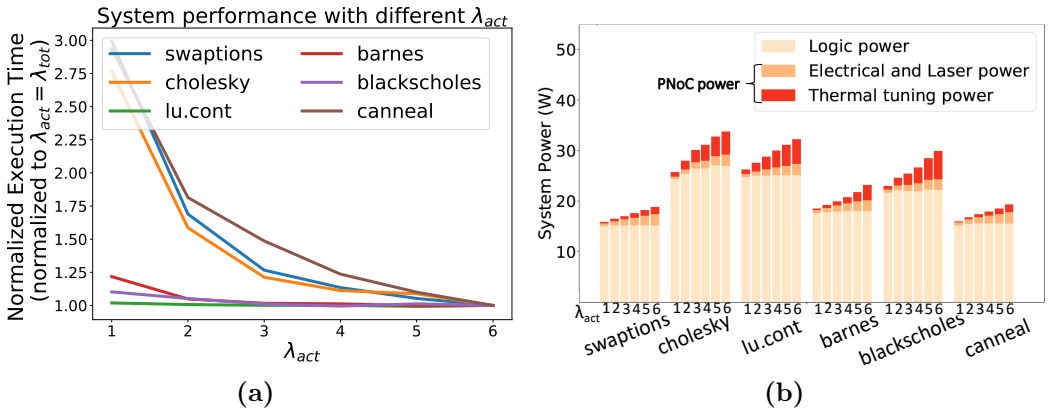


Figure 3·5: (a) Normalized execution time and (b) system power breakdown with different number of active optical channels (λ_{act}) in the silicon-photonic link.

3.3.1 Static Policy: *SO-WAVES*

In *SO-WAVES* policy, we determine the minimum number optical channels that can satisfy the average bandwidth needs of an application. An application has the highest performance when run with λ_{tot} optical channels. We set a performance loss thresh-

old, L_{thr}, from the maximum performance that is deemed accepted for the system. We reduce the optical channels from λ_{tot} and determine the minimum number of optical channels, λ_{min}, that provides a performance within the set L_{thr}. At runtime, we execute the application with λ_{min} optical channels out of a total of λ_{tot} optical channels. Section 3.3.3 explains the selection of the best combination of λ_{min} from λ_{tot} considering the process variations of MRRs, thermal profile of the chip and the MRR locking mechanism.

3.3.2 Dynamic Policy with Time-series Prediction: *PROWAVES*

Figure 3·6 illustrates that the transfer of network packets in silicon-photonic links is highly dynamic and periodic during the application execution. The plot shows that applications have varying trends in bandwidth requirements. Since *SO-WAVES* selects a single λ_{min} for the entire application execution, much of the power benefits from wavelength selection remain under-utilized. Therefore, a dynamic policy that can select the minimum optical channel at each application phase is desirable. To this end, we consider a Dynamic Oracle WAVES (*DO-WAVES*), which selects $\lambda_{act} = \lambda_{min}$ within the set L_{thr} at each application phase. It is imperative to note that *DO-WAVES* policy is not practically realistic as it assumes accurate knowledge of the future execution trends to select the optimal λ_{min} at each application phase. Our goal is to design a proactive policy that can closely match the λ_{min} of *DO-WAVES*.

PROWAVES is a dynamic policy that predicts the network activity for an application phase using time-series forecasting, and proactively determines the λ_{min} for that phase. The network activity in the silicon-photonic link is characterized by the average packet latency, which is expressed as follows:

$$Lat_{avg_i} = \frac{T_{queue_i}}{N_{p_i}}, \tag{3.9}$$

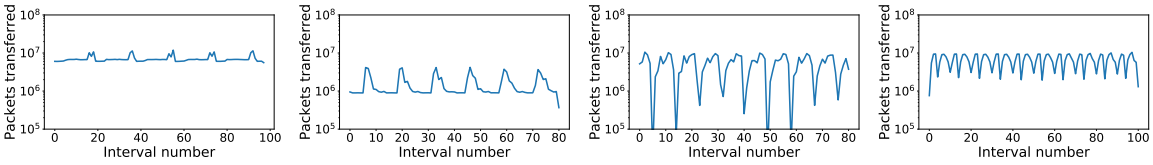


Figure 3·6: Inter-chiplet packets transferred during application execution for (a) *bt*, (b) *ep*, (c) *shock*, and (d) *lu*. Applications have phases where a higher number of packets are transferred compared to other phases and these phases exhibit periodic behavior.

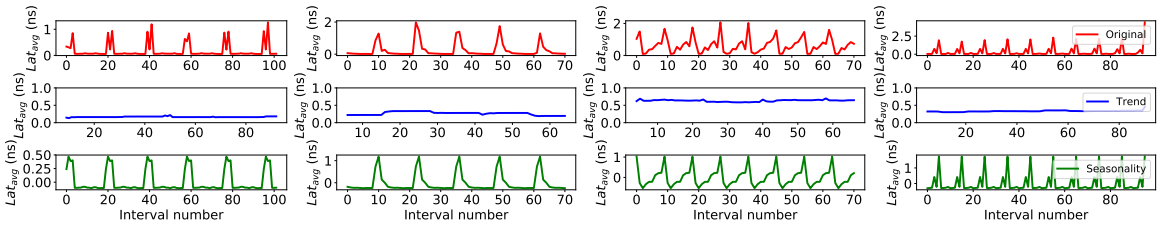


Figure 3·7: Trends and seasonality in the Lat_{avg} time series for (a) *bt*, (b) *ep*, (c) *shock*, and (d) lu.

where T_{queue_i} is the aggregate queue latency of all packets, and N_{p_i} is the total number of packets transferred during an application interval i.

We utilize an ARIMA (Box et al., 2015) predictor to forecast Lat_{avg} for each application interval by utilizing past trends in Lat_{avg}. ARIMA model requires the time series to be stationary i.e., the time series should be devoid of trends and/or seasonality. The average packet latency experiences minimal trends but strong seasonality during the application execution as depicted in Figure 3·7. We convert this time series to stationary by computing the difference between consecutive time intervals, a process known as differencing.

The ARIMA (p, d, q) forecasting model consists of: (1) an autoregression model that forecasts a variable using the relationship between an observation and p prior observations, (2) differencing of raw observations d times to make the time-series sta-

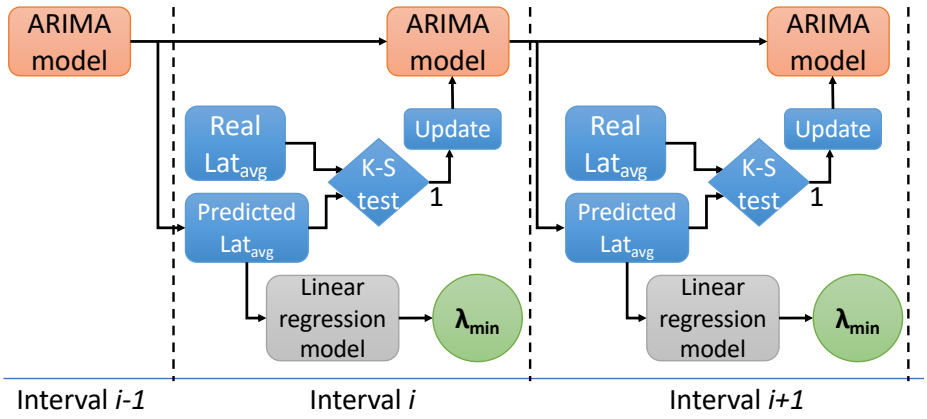


Figure 3·8: Flow of *PROWAVES*. Every interval, the ARIMA model forecasts the Lat_{avg}. The linear regression model selects the λ_{min} from the forecasted Lat_{avg}. A K-S test is applied to update the ARIMA model in case of divergence.

tionary, and ③ a moving average model applied to q prior observations to extract the dependency between an observation and its residual error. We build the best-fitting $ARIMA(p, d, q)$ model using the Akaike information criterion (AIC) (Akaike, 1969). The AIC estimates the goodness of fit of the model on the dataset, by determining the relative information lost by the ARIMA model. The less information the model loses, the higher the quality of that model. We start with an $ARIMA(1, 0, 0)$ model and perform a grid search for a range of p, d and q parameters. We increment these parameters and determine the values that yield the lowest AIC value.

Figure 3·8 shows the operational flow of *PROWAVES* policy. During each interval, we use the $ARIMA(p, d, q)$ model to forecast the average packet latency for the following interval. A time series, however, may diverge from the initial training dataset and result in inaccurate forecasting. We incorporate a goodness-of-fit test to detect the divergence of the real data from the ARIMA predicted data. We integrate Kolmogorov-Smirnov (K-S) (Massey Jr, 1951) test into the ARIMA model to run every interval. If the K-S test fails during an interval, i.e., marked by 1 in Figure 3·8, we rebuild the ARIMA model by grid-searching again over the range of p,d and q

parameters. An updated $ARIMA(p, d, q)$ model is then utilized to forecast for the following intervals.

With the predicted Lat_{avg}, we determine the λ_{min} required for the next interval. We devise a methodology to correlate the Lat_{avg} to the optimal λ_{min}, which is selected by *DO-WAVES*. For different intervals of our training applications, we determine the λ_{min} that provides a performance within the set L_{thr} at those intervals. Figure 3·9 shows a plot of λ_{min} against the log of Lat_{avg} at those intervals. We fit a line through these points, such that 90% of the points are above this line to ensure that the bandwidth needs at an interval is always satisfied. We store the parameters of this linear regression model on-chip. At runtime, *PROWAVES* determines the λ_{min} for the next interval based on the forecasted Lat_{avg} using the linear regression model.

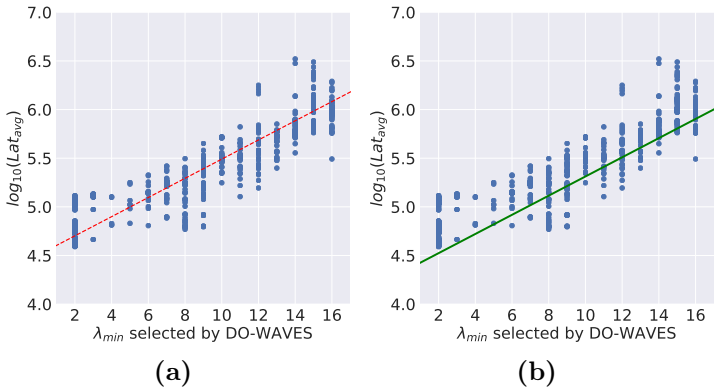


Figure 3·9: Scatterplots of Lat_{avg} vs λ_{min} selected by *DO-WAVES* for $L_{thr} = 5\%$. (a) shows the line with least mean square error, (b) shows the line such that 90% of the points are above the line.

3.3.3 MRR Locking with Wavelength Selection

During application execution with *PROWAVES*, due to resonance shift, the MRRs need to be locked to the activated optical channels under three conditions: (1) when *PROWAVES* increases the number of optical channels due to increased bandwidth demand, (2) when *PROWAVES* reduces the number of optical channels due to lower

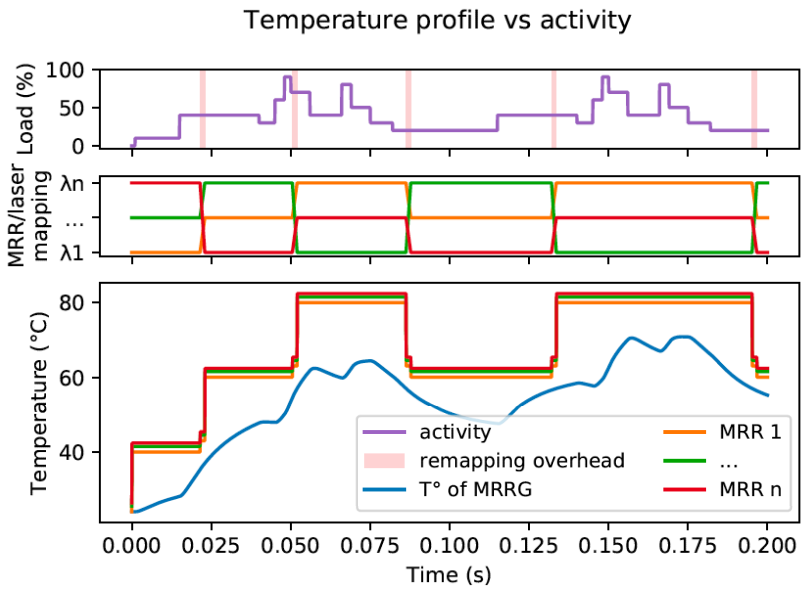


Figure 3·10: Thermal remapping of MRRs to λ_{act}. As chip activity varies during execution, the thermal profile of MRRGs varies, causing MRRs within an MRRG to map to different optical channels.

bandwidth demand, and ③ a large temperature drift introduces a resonance shift greater than FSR/λ_{tot}. For each of these three scenarios, the analog thermal control loop in the TxRx chiplet supplies heating power to remap these MRRs to the nearest laser wavelength in the spectrum.

Figure 3·10 shows an example of the thermal remapping of MRRs. When the MRR shift increases over the tuning range of the heaters, the computation is temporarily halted. An on-chip LUT is polled to determine the set of λ_{act} laser wavelengths that result in the lowest thermal tuning power. The thermal control loop then supplies the heating power to lock the MRRs to these new λ_{act} optical channels. This is shown in Fig. 3·10, where the MRRs resonate at different laser wavelengths after remapping. Similarly, when *PROWAVES* increases or decreases the λ_{act} during application execution, the on-chip LUT is polled to identify the new set of MRRs that needs to be mapped to the selected λ_{act}.

3.3.4 Hardware Cost of Wavelength Selection

Implementation of *SO-WAVES* or *PROWAVES* policy on a 2.5D system comes at a minimal hardware cost. In *SO-WAVES*, an offline analysis determines the λ_{min} for the entire application execution. Thus, *SO-WAVES* does not incur any runtime hardware overhead for determining λ_{min}. In *PROWAVES*, the hardware performance counters are polled at the end of every interval to read out network activity statistics, i.e., the number of inter-chiplet packets transferred and overall queue time. An initial ARIMA model is created using these statistics from the training interval and the model parameters (p, d, q) are stored. On average, this ARIMA model is created in $72ms$ for an application. This ARIMA model is utilized to determine the λ_{min} for the next interval in parallel with the execution of the current interval. We observe that ARIMA forecasting takes less than 0.1% of the execution time of an interval and, therefore, is always hidden in the execution time of the current interval.

Once λ_{min} for the next interval is determined, we need to activate the best combination of λ_{min} among a total of $\binom{\lambda_{tot}}{\lambda_{min}}$ combinations. An LUT holds floating point values of heating power for all the MRRGs, λ_{tot} wavelengths and temperature range of $300 - 380K$ ($0.5K$ precision). The memory footprint of this LUT is estimated as $400kB$, and can be stored on-chip. At runtime, depending on the thermal profile at the end of an interval, we poll this LUT and exhaustively search across all the laser combinations to determine the best combination of λ_{min}. As the worst-case LUT access time is $\leq \binom{\lambda_{tot}}{\lambda_{tot}/2} \cdot C$ lookups and additions, this latency is hidden within the thermal remapping latency ($100\mu s$).

A major factor contributing to the performance overhead in *PROWAVES* comes from the latency associated with increasing and decreasing the optical channels. Figure 3·11 illustrates the latencies of different components in *PROWAVES* during an application execution. The ARIMA model predicts λ_{min} for the next interval in par-

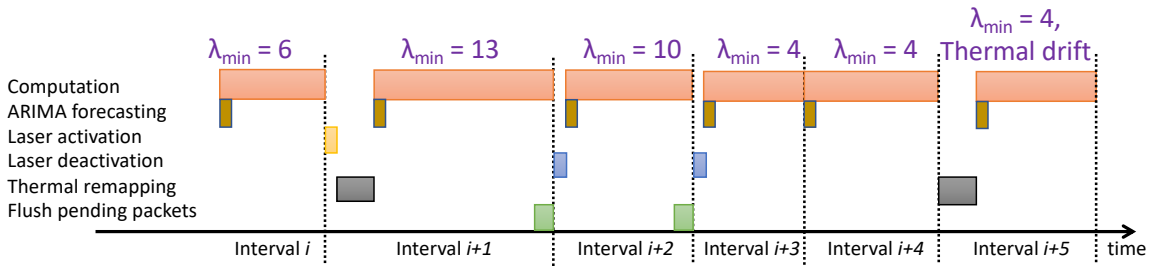


Figure 3·11: Latency overhead of *PROWAVES*. Increasing λ_{min} involves laser activation ($2ns$) and thermal remapping ($100\mu s$). Decreasing λ_{min} involves laser deactivation ($2ns$) and flushing pending packets ($100ns - 1\mu s$), both of which are hidden in the computation time.

allel with the execution of current interval. MRR remapping, if required for the next interval, begins at the completion of the current interval. When λ_{min} is increased, the latency comprises of the laser power-on latency and the thermal remapping of the new group of MRRs to the activated optical channels. Laser power-on takes about $2ns$ with relatively low drift (Simon et al., 2016). Once the laser wavelengths are activated, the thermal control loop remaps the MRRs to the activated laser wavelengths in $100\mu s$ (Thonnart et al., 2018). Therefore, activating additional laser wavelengths during an application execution introduces a latency overhead of $100\mu s$.

When λ_{min} is decreased, the next application interval requires deactivation of certain laser sources ($2ns$ (Simon et al., 2016)). We observe that there is no additional WDM group of MRRs that needs to be tuned to the new set of laser wavelengths, therefore, the MRR thermal remapping during laser deactivation is not necessary. We simply release the heating power on the MRRs that were communicating via the deactivated laser wavelengths, and maintain the heating power on the remaining MRRs [1]. However, decreasing λ_{min} at runtime requires flushing the pending packets on deactivated laser wavelengths. We measure the worst-case completion of pending

[1]Note that when deactivating laser wavelengths, we do not perform LUT lookup to select λ_{min}, as the LUT lookup requires MRR thermal mapping to a new set of λ_{min} with a remapping cost of $100\mu s$. So the activated λ_{min} may not be the best combination that result in lowest thermal tuning power.

packets in the PNoC to be $100ns - 1\mu s$, and this latency is hidden in the next application interval. Hence, the overall latency of decreasing λ_{min} is negligible.

3.3.5 Experimental Results and Analysis

We evaluate the power benefits of performing wavelength selection by comparing *SO-WAVES* and *PROWAVES* against a baseline policy that activates all the optical channels throughout the application execution. Moreover, we contrast the benefits of thermal remapping obtained with the modeling of analog thermal control loop by comparing *PROWAVES* to a prior power scaling technique (Van Winkle et al., 2018). We also quantify the performance overheads of *SO-WAVES* and *PROWAVES*. We conduct our experiments with varying system utilization by running the application with different thread counts. In all experiments, we use L_{thr} values of 1% and 5% to demonstrate the user flexibility of setting the performance loss threshold and exploring the bandwidth-power tradeoffs in silicon-photonic links.

Power Benefits of *SO-WAVES* with Varying System Utilization

We evaluate the power benefits of *SO-WAVES* for different system utilizations by varying the thread count in applications. Each application is run with L_{thr} values of 1%, 5% and 10%. Figures 3·12a-d shows the power savings for different applications running on *POPSTAR* with 24, 48, 72 and 96 threads, respectively. In most applications, larger thread counts result in increased inter-chiplet network traffic among the communicating threads. Consequently, larger thread counts require higher λ_{min}. This is evident in *canneal* and *cholesky* applications running 96 threads, which require all the optical channels to be activated, even for an L_{thr} of 10%. Moreover, for applications with lower communication traffic (e.g., *blackscholes*, *barnes*, and *lu.cont*), the system performance saturates for a lower λ_{act} compared to other applications. As a consequence, even for a 1% L_{thr}, a lower λ_{min} is activated and we observe aver-

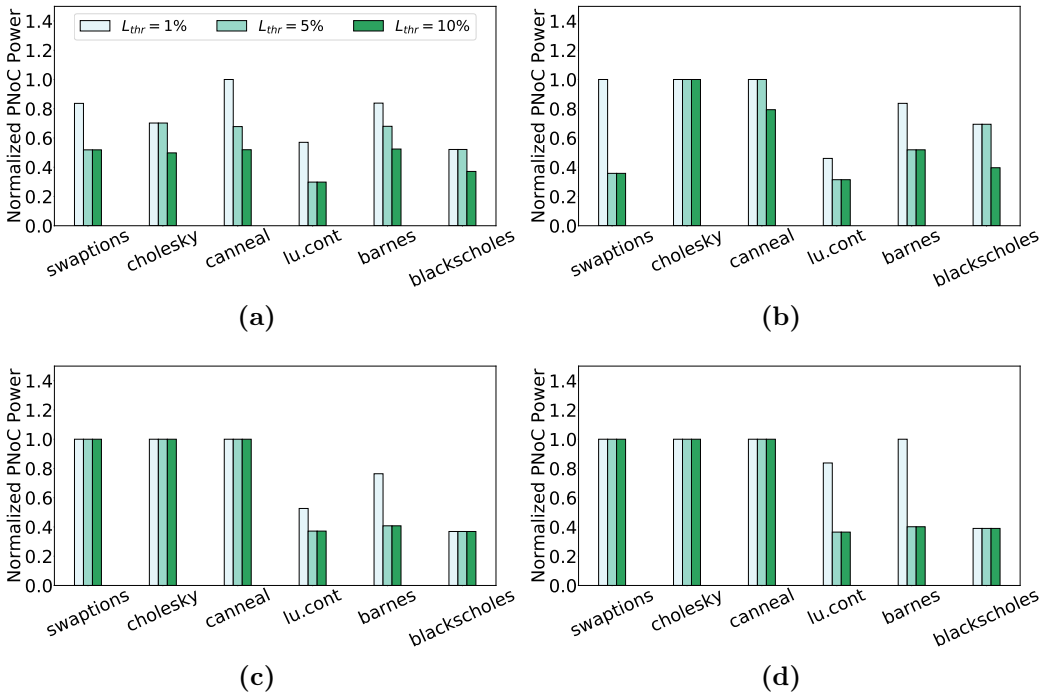


Figure 3·12: Photonic power consumption of *POPSTAR* with *SO-WAVES* for (a) 25% system utilization, (b) 50% system utilization, (c) 75% system utilization, and (d) 100% system utilization.

age power savings of 38%. For applications with higher communication traffic (e.g., *canneal*, *swaptions*, and *cholesky*), the high network traffic demands higher λ_{min}, resulting in average power savings of only 8% for $L_{thr} = 1\%$. On average across all applications, $SO - WAVES$ achieves 23%, 38%, and 42% average photonic power savings with 1%, 5%, and 10% performance loss, respectively.

Power Benefits of *PROWAVES* with Varying System Utilization

We next study the power benefits of the proactive dynamic *WAVES* policy, *PROWAVES*, in contrast to *SO-WAVES*. We also compare the power benefits of *PROWAVES* to *DO-WAVES*, which selects the theoretical minimum number of optical channels for an interval. Our goal with *PROWAVES* is to select a λ_{min} that is as close as possible to the λ_{min} selected by *DO-WAVES*. Our baseline case activates all the optical channels

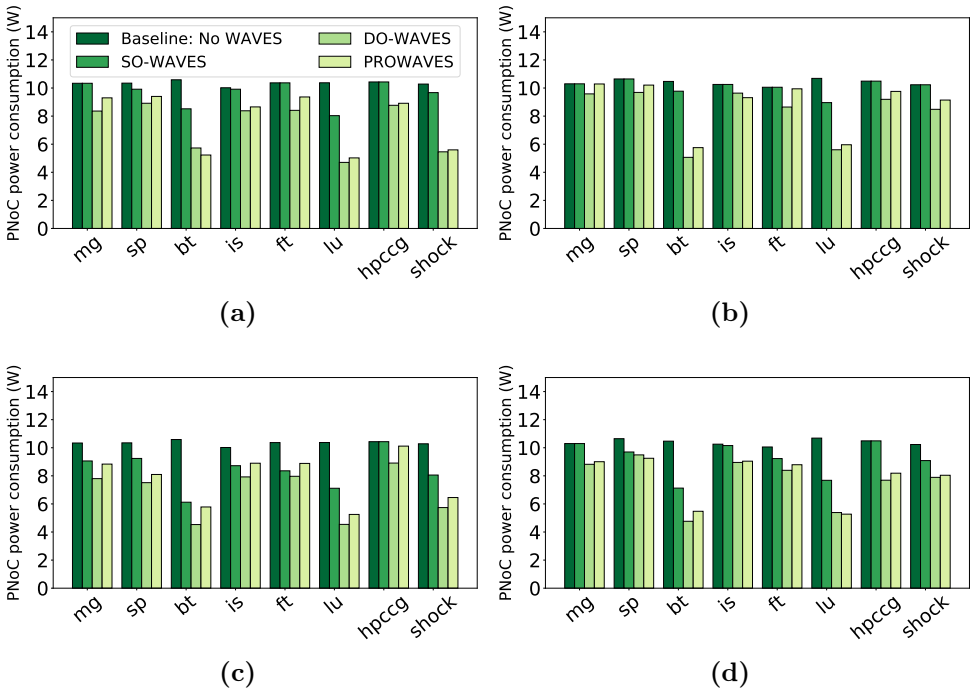


Figure 3·13: Photonic power consumption of *POPSTAR* with different WAVES policies for (a) 50% system utilization, $L_{thr} = 1\%$, (b) 100% system utilization, $L_{thr} = 1\%$, (c) 50% system utilization, $L_{thr} = 5\%$, and (d) 100% system utilization, $L_{thr} = 5\%$.

in the systems, i.e., ($\lambda_{act} = \lambda_{min}$). Figure 3·13 shows the photonic power consumption with different *WAVES* policies under varying system utilization.

SO-WAVES consumes 8.6% and 21% lower PNoC power on average than the baseline case for an L_{thr} of 1% and 5%, respectively. *DO-WAVES* is able to uncover additional photonic power savings in the system by activating lower λ_{min} during phases of low bandwidth needs. This is in contrast to *SO-WAVES* that selects and activates a single λ_{min} during the entire application execution. As a result, for L_{thr} of 1% and 5%, *DO-WAVES* provides 34.4% and 40.7% reduction in photonic power than the baseline. In comparison, for L_{thr} of 1% and 5%, the system with *PROWAVES* consumes 18% and 33% lower photonic power than the baseline and 10.2% and 16.4% lower photonic power than *SO-WAVES* respectively. The power savings with *PROWAVES*

is within 12% of the theoretical minimum, which is achieved by *DO-WAVES*. The power savings obtained from *PROWAVES* lowers with increasing system utilization. With a higher number of threads per chiplet, there is an increased inter-chiplet network traffic, resulting in higher bandwidth requirements. Consequently, a higher λ_{min} is selected to satisfy the high bandwidth needs.

Thermal Tuning Power Savings with *PROWAVES*

A primary benefit of our cross-layer modeling in *SO-WAVES* and *PROWAVES* is obtained with the modeling of the thermal control loop that enables runtime MRR locking. We, therefore, compare *PROWAVES* against a power scaling technique based on a ridge regression model (*RR-PS*) (Van Winkle et al., 2018). With a feature set consisting of network metrics and L1/L2 cache misses, *RR-PS* predicts the number of packets transferred in the PNoC. Using the predicted number of packets, *RR-PS* calculates the minimum number of optical channels that can support the network packets. A major limitation of *RR-PS* comes from the lack of TV and PV modeling, and the resulting thermal tuning power. Here, we evaluate the two major benefits of *PROWAVES*: (1) the modeling of the low-level thermal control loop that enables MRR thermal remapping, and (2) the selection of best λ_{min} for every interval.

Figure 3·14a shows the thermal tuning power in *POPSTAR* with *PROWAVES* or *RR-PS*. The impact of TV and PV-induced resonance shift is prominently observed in *RR-PS* compared to *PROWAVES*. Due to the lack of modeling a control loop for thermal tuning in *RR-PS*, all the MRRs need to be tuned to the designated laser wavelengths. Therefore, the average case tuning range for a random PV distribution across MRRG in *RR-PS* is $FSR/2$. In contrast, the presence of a control loop for thermal tuning in *PROWAVES* enables thermal remapping to the nearest activated laser wavelength, resulting in a worst-case tuning range of FSR/λ_{act}. Since we model the low-level thermal control loop at the system-level, we are able to capture the

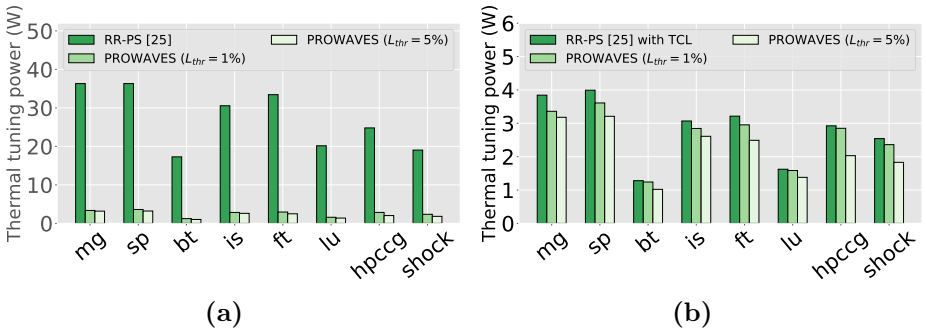


Figure 3·14: Thermal tuning power comparison between *RR-PS* and *PROWAVES*. In (a), *RR-PS* does not model thermal control loop that enables thermal remapping, as initially proposed in (Van Winkle et al., 2018). In (b), *RR-PS* is updated to include a thermal control loop model as *PROWAVES*, but does not select best λ_{min} that accounts for PV.

benefits of thermal remapping and significantly reduce the overall thermal tuning power. Compared to *RR-PS*, *PROWAVES* consumes $24.6W$ and $26.3W$ lower thermal tuning power with an L_{thr} of 1% and 5% respectively. Thus, modeling of the thermal control loop is essential to evaluate system-level power benefits.

Since thermal control loop is essential for thermal remapping and significantly reduces the thermal tuning power, we incorporate its modeling in *RR-PS* in Figure 3·14b. This modeling enables us to isolate the specific benefits of wavelength selection. *PROWAVES* accounts for the impact of PV-induced resonance shift, which varies across MRRs in an MRRG, and across different MRRGs. *PROWAVES* activates the best combination of laser wavelengths to reduce the impact of PV-induced resonance shifts as opposed to *RR-PS with TCL*, which always selects a fixed set of laser wavelengths for an interval. This finer level of wavelength selection in *PROWAVES* reduces the thermal tuning power by 7.1% and 22.01% for L_{thr} of 1% and 5% respectively, as compared to *RR-PS*.

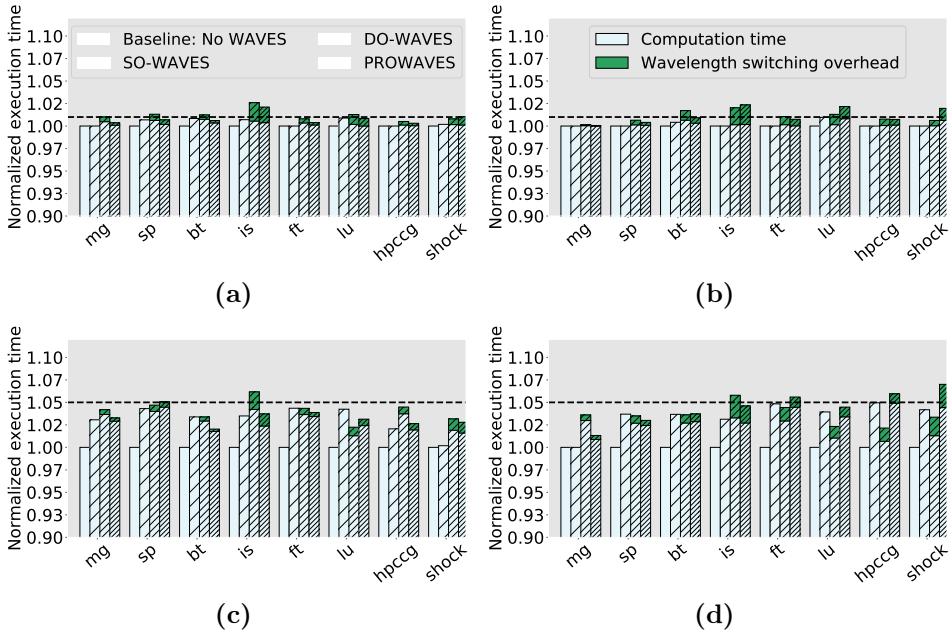


Figure 3·15: Normalized execution time and wavelength switching overhead with different WAVES policies for (a) 50% utilization, $L_{thr} = 1\%$, (b) 100% utilization, $L_{thr} = 1\%$, (c) 50% utilization, $L_{thr} = 5\%$, and (d) 100% utilization, $L_{thr} = 5\%$. The dotted line indicates L_{thr}.

Performance Overhead in Wavelength Selection

Figure 3·15 shows the execution time of applications with the baseline case ($\lambda_{act} = \lambda_{tot}$) and under different policies, normalized to the baseline case. For each application, we calculate the wavelength switching overhead of *PROWAVES* by determining the count of thermal remappings arising due to laser activation or a large thermal drift during the execution. On average, this switching overhead is computed to be only 0.73% of the overall execution time for *PROWAVES*. Since we calculate the λ_{min} for *PROWAVES* by comparing only the computation time with the performance loss threshold, the overall execution time including the wavelength selection overhead occasionally violates the set L_{thr}.

Compared to the execution time of *SO-WAVES*, the dynamic selection in *PROWAVES* is able to provide better performance at lower PNoC power, leading

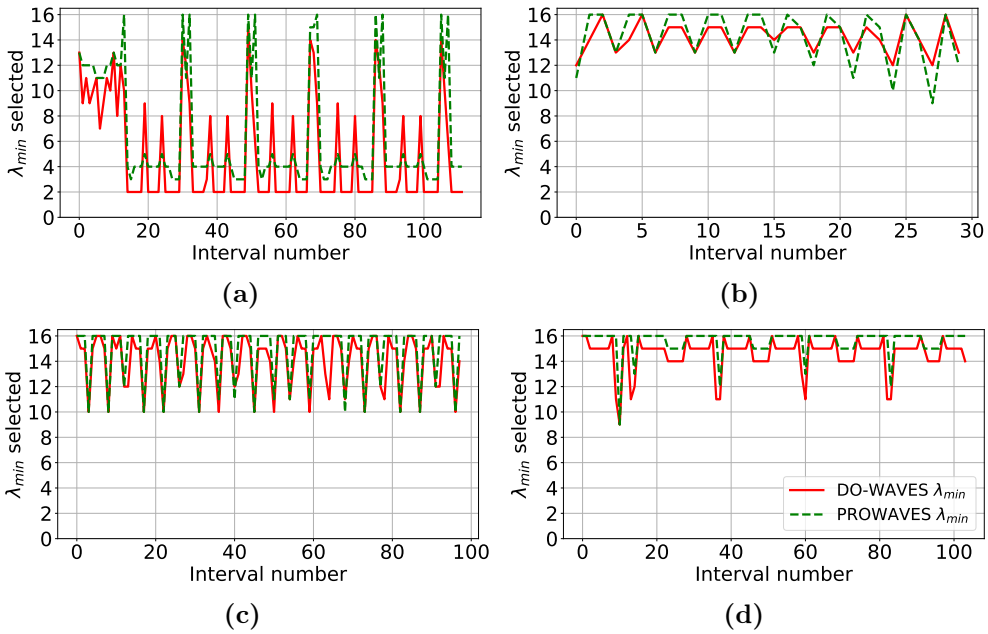


Figure 3·16: Comparison of λ_{min} selected by *DO-WAVES* and *PROWAVES* with $L_{thr} = 5\%$ for applications (a) *bt*, (b) *is*, (c) *sp*, and (d) *mg*. During periods of high bandwidth needs, a higher λ_{min} is activated, and during periods of lower bandwidth needs, a lower λ_{min} is activated.

to much lower PNoC energy compared to *SO-WAVES*. In *PROWAVES*, higher λ_{min} is selected during periods of high bandwidth needs and a lower λ_{min} is selected during periods of lower bandwidth needs. In contrast, since *SO-WAVES* only selects a single λ_{min} throughout the application execution, this λ_{min} is roughly averaged. Therefore, during periods of high bandwidth needs, *SO-WAVES* falls short of selecting the optimal λ_{min}. Similarly, during periods of low bandwidth needs, *SO-WAVES* overestimates and selects a higher λ_{min} than required.

Forecasting Accuracy of *PROWAVES* Policy

Figure 3·16 illustrates the deviation of λ_{min} selected by our proposed *PROWAVES* policy from *DO-WAVES*. This deviation in the selected λ_{min} and the resultant lower power savings in *PROWAVES* can be attributed primarily to two major reasons.

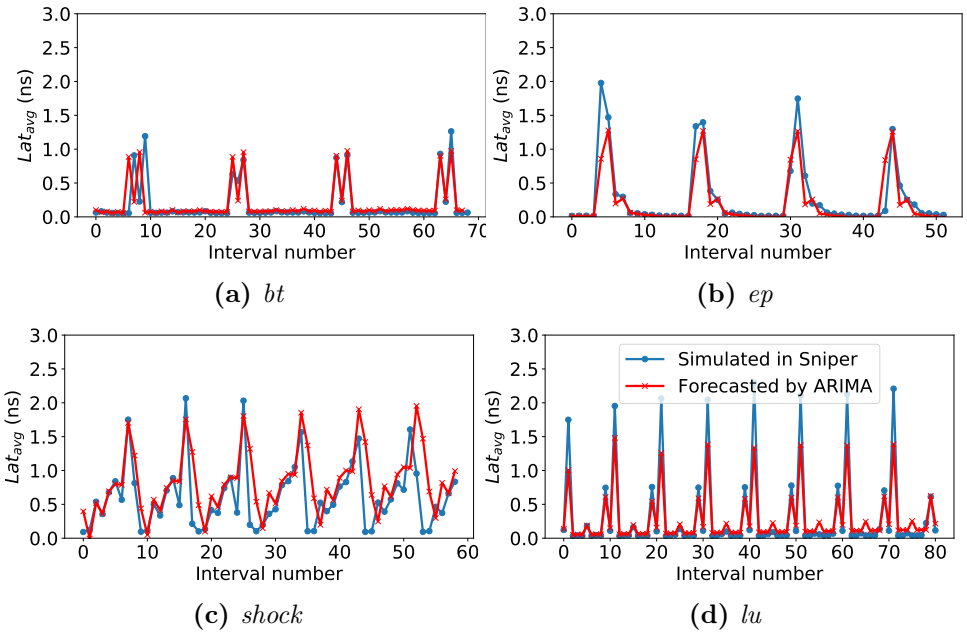


Figure 3·17: Lat_{avg} values from Sniper simulations and forecasted Lat_{avg} values using ARIMA model for applications (a) *bt*, (b) *ep*, (c) *shock*, and (d) *lu* running 96 threads.

First, the Lat_{avg} predicted by the ARIMA model does not have a 100% forecasting accuracy. Second, the linear regression model used to correlate the predicted Lat_{avg} to the *DO-WAVES* λ_{min} has inaccuracies that further contribute to a slightly different λ_{min}. Figure 3·17 illustrates the simulated values of Lat_{avg} on Sniper and the predicted Lat_{avg} values by ARIMA. We calculate the mean squared error of predicted Lat_{avg} as $0.019ns^2$. Thus, our ARIMA predictor with K-S test has an automated process of forming the model with 96.3% accuracy. Figure 3·17 depicts the Lat_{avg} values from our Sniper simulations and the forecasted Lat_{avg} values from our ARIMA model. We observe that the ARIMA model with K-S test captures the seasonality in the Lat_{avg} time series with high precision.

We analyze the selected λ_{min} against the training data in the linear regression model. We obtain an R-squared value of 0.916 with a low p-value, strongly suggesting that changes in the predictor's value (Lat_{avg}) are related to changes in the

Table 3.5: Summary of modeling parameters and results of different wavelength selection policies

	RR-PS [25]	**SO-WAVES [18]**	**PROWAVES**
Static/dynamic policy	Dynamic	Static	Dynamic
Model	Ridge regression	Offline	ARIMA
Thermal control loop modeling	No	Yes	Yes
Process variation modeling	No	Yes	Yes
Power savings over baseline	13%	8.6%[a] 21%[b]	18%[a] 33%[b]

Latency overhead [a]1% L_{thr}, [b]5% L_{thr}

response variable (λ_{min}). This shows that Lat_{avg} is statistically significant in predicting λ_{min}. We observe from Fig. 3·16 that during each phase of an application run, *PROWAVES* selects a λ_{min} that is equal to or higher than the λ_{min} selected by *DO-WAVES*. Therefore, at the cost of slightly lower power savings, the performance with *PROWAVES* is always better than *DO-WAVES*.

Summary of wavelength selection policies

The model for thermal control loop and the resultant MRR remapping enables *SO-WAVES* and *PROWAVES* to reduce higher photonic power when compared to *RR-PS*. Moreover, the dynamic wavelength selection in *PROWAVES* when helps uncover additional photonic power savings compared to *SO-WAVES*, while staying within the performance threshold. Table 3.5 summarizes the results and modeling parameters of *PROWAVES* compared to *RR-PS* (Van Winkle et al., 2018) and *SO-WAVES*.

3.4 Silicon-Photonic Links for Graph Workloads

Graphs represent the basic relationship between two vertices. With data in several application domains becoming increasingly connected, graphs are rather ubiquitous in social networks, financial sectors, transportation representations and webpages.

A primary bottleneck in graph applications arises from the highly irregular memory access patterns resulting in poor spatial and temporal locality (Ahn et al., 2015). These irregular access patterns often result in high and frequent memory accesses. In 2.5D systems, when the LLCs are spread across multiple chiplets, the memory accesses constitute a major fraction of the application execution time (Wang et al., 2019). These graph processing applications require more than $1Tbps$ bandwidth in the communication network. Though silicon-photonic networks are able to meet the high bandwidth density demands of graph applications, the high photonic power limits the energy efficiency of the overall system. In this section, we demonstrate the efficacy of *WAVES* in reducing the photonic power when running graph applications.

3.4.1 Evaluation of Wavelength Selection for Graph Workloads

We simulate the execution of graph applications from GAP-BS (Beamer et al., 2015) benchmark when run on *POPSTAR* using our cross-layer simulation framework. Table 3.3 shows the description of the graph applications used in these simulations. We evaluate these applications on three datasets, two Kronecker graphs with 2^{18} and 2^{20} nodes and a real-world dataset from Google web graph (|V|=875713, |E|=5105039) (Leskovec and Krevl, 2014).

Figure 3·18 illustrates the normalized execution time of graph applications as we increase the peak aggregate bandwidth in the interposer by activating more optical channels. Even for graph applications, we observe that the performance saturates at different bandwidth values for different applications. We apply our wavelength selection policy, *SO-WAVES*, on graph applications using a performance loss threshold of $L_{thr} = 1\%$. We determine the best combination of λ_{min} that result in the lowest thermal tuning range. Figure. 3·19 shows the normalized photonic power with λ_{min}, compared to the power with the highest bandwidth, i.e. λ_{tot}. *SO-WAVES* provides 36% average reduction in power with λ_{min} than when using the peak aggregate

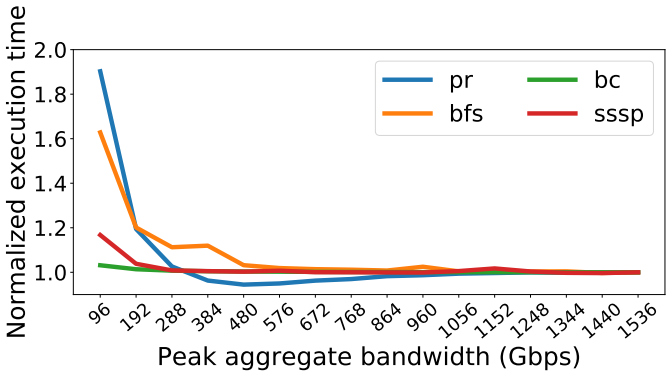


Figure 3·18: Normalized performance with increasing inter-chiplet bandwidth for graph applications on Google web graph. The performance is normalized to the performance with peak bandwidth of $1.536 Tbps$.

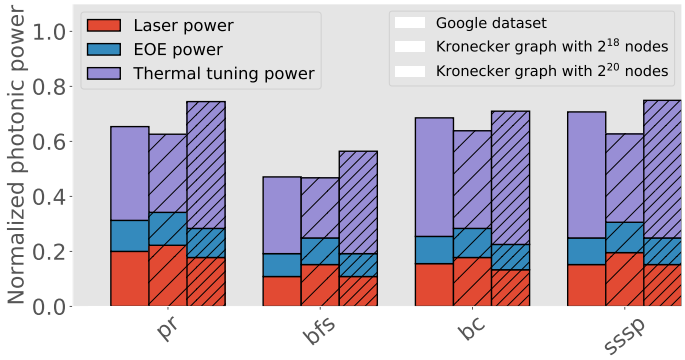


Figure 3·19: Photonic power consumption with *SO-WAVES* for graph applications on three different datasets. Power numbers are normalized to baseline case where all laser wavelengths are activated.

bandwidth with λ_{tot}. We observe that graphs with larger datasets consume higher photonic power. This is due to the increased bandwidth needs and higher inter-chiplet communication traffic as the scale of input dataset increases.

3.4.2 Architectural Exploration for Graph Workloads

Silicon-photonic links provide higher orders of bandwidth density compared to electrical links and meet the bandwidth demands of graph applications. Thus, there is an opportunity to rethink the design of conventional memory hierarchy for graph

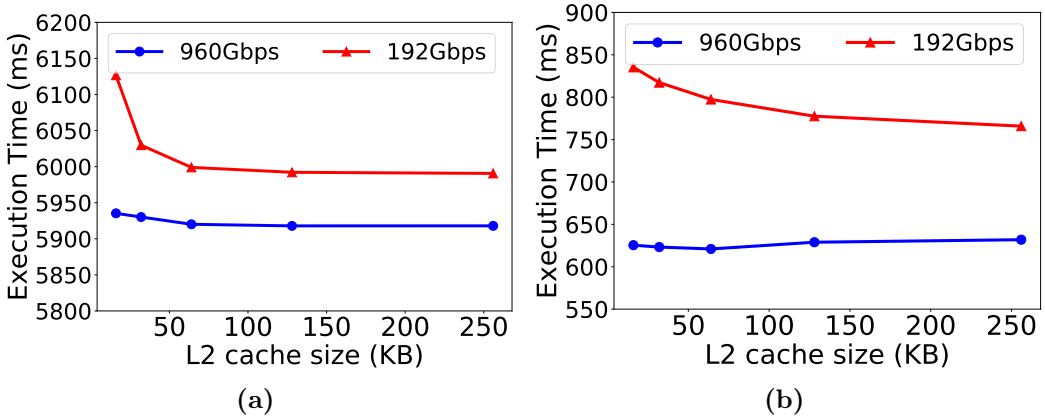


Figure 3·20: Performance of (a) *bc* and (b) *pr* with different inter-chiplet bandwidth, when executed on 2 systems with different L2 cache sizes.

applications, which do not utilize the cache hierarchy effectively. We first evaluate the performance of graph applications with varying private L2 cache sizes for two different inter-chiplet bandwidth. For this experiment, we use the Google web graph dataset from SNAP (Leskovec and Krevl, 2014). Figure 3·20 shows the application performance with increasing L2 cache size. We observe that the application performance improves as we increase the L2 cache size for a low inter-chiplet bandwidth of 192*Gbps*. However, a higher inter-chiplet bandwidth of 960*Gbps* shows minimal execution time variations with increasing L2 cache size.

For lower inter-chiplet bandwidth and smaller L2 cache sizes, the execution time due to L2 misses also includes the high fraction of queue latency in the photonic link. Increasing the L2 cache size improves the hit rate and we observe a speedup in performance. However, the L2 miss latency is still dominated by the queue latency in the photonic link. When we increase the inter-chiplet bandwidth to meet the bandwidth requirements of graph applications, we significantly reduce the queue latency. As a result, the L2 cache misses for the same L2 cache size is serviced faster with a high-bandwidth link. Due to irregular memory accesses in graph applications, we do

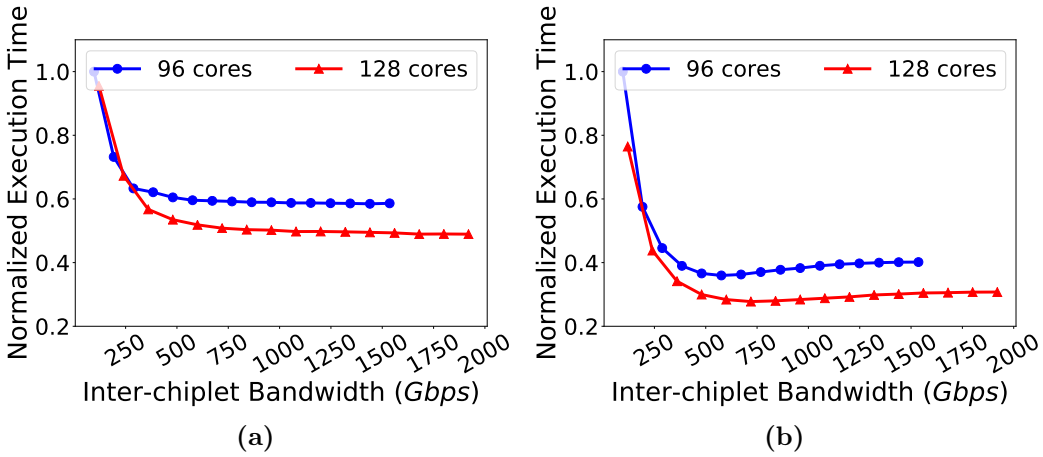


Figure 3·21: Performance of (a) *bfs* and (b) *pr* with different inter-chiplet bandwidth, when executed on 2 systems with different core counts.

not observe performance improvement with increasing L2 cache when the bandwidth requirements are met. As silicon-photonic links are able to meet the high bandwidth demands of applications, there is an opportunity to incorporate a smaller L2 cache per core and per chiplet.

We next evaluate the performance scaling of graph applications with increasing core counts. As 2.5D systems enable modularity, we integrate more chiplets on the interposer, keeping the same number of cores per chiplet. For this experiment, we use our largest data graph, the Kronecker graph with 2^{20} vertices. The maximum bandwidth with $\lambda_{act} = 16$ increases from $1.5Tbps$ in a 96-core system to $1.9Tbps$ in a 128-core system. From Figure 3·21, we observe a performance improvement of 21% on average for a 128-core system compared to a 96-core system for the same number of activated laser wavelengths. It is interesting to note that the system performance saturates at a higher inter-chiplet bandwidth for the 128-core system than the 96-core system. For example, in *bfs*, we obtain a system performance within 1% of peak performance for an inter-chiplet bandwidth of $864Gbps$ ($\lambda_{act} = 9$) in a 96-core system,

while in the 128-core system, we obtain 1% of peak performance for an inter-chiplet bandwidth of $1.56Tbps$ ($\lambda_{act} = 13$). Similarly, in *pr*, we obtain the peak performance for $\lambda_{act} = 6$ for both systems. However, the aggregate bandwidth corresponds to $576Gbps$ in a 96-core system and $720Gbps$ in a 128-core system.

These observations enforce the scalability of graph applications with number of coures due to their inherent parallelism. There is a significant increase in inter-chiplet traffic with increasing LLC and memory accesses with higher chiplet counts. Therefore, 2.5D manycore systems with silicon-photonic links are able to meet the required bandwidths for graph applications.

3.5 Wavelength Selection using Application Instrumentation

Our wavelength selection policies provide the required bandwidth for an application by activating the appropriate number of optical channels. However, most bandwidth allocation policies including *SO-WAVES* and *PROWAVES* are typically implemented at the system-level and have minimal to no exposure to the application source code. Our proposed wavelength selection characterizes the bandwidth requirement using offline analysis as in *SO-WAVES* or forecasts network activity at runtime as in *PROWAVES*. However, the chip-scale communication traffic also depends on the software implementation of the application algorithm. This dependence provides an opportunity to develop a generalized software-level approach for performing wavelength selection at the system-level.

This section introduces the software framework to instrument an application and guide wavelength selection at the system-level. We instrument data structures or privileged instructions in the application source code to provide information regarding the communication traffic during the application execution. This information can then be utilized at the system level to perform wavelength selection.

3.5.1 Application Instrumentation

As an example, we consider PageRank, an iterative graph algorithm that has extremely high parallelism. We demonstrate that using appropriate instrumentation of PageRank source code, we pass additional information regarding active vertices that can be used to reduce the network bandwidth density.

Motivational Example: PageRank

The PageRank algorithm begins with equal ranks assigned to each vertex in the input graph. Depending on the number of vertices connected to a vertex v, ($g.out\_degree(v)$), the rank of v is updated. At the end of every iteration, the rank of each vertex is compared with an error threshold. The algorithm iterates until all vertices converge.

A key characteristic of PageRank is the varying number of iterations required to converge the vertices, which result in asymmetric convergence (Ozdal et al., 2015). We demonstrate this characteristic by running PageRank on a Google webgraph from SNAP (Leskovec and Krevl, 2014) and a Kronecker graph with 2^{18} vertices. Figure 3·22 shows the fraction of vertices that have not yet converged at the end of each iteration. On a Google webgraph, 21.77% of vertices converge in a single iteration,

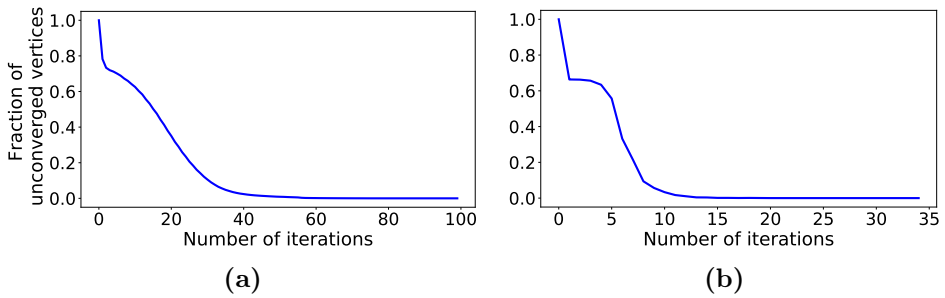


Figure 3·22: Number of unconverged vertices with iterations for PageRank on (a) Google webgraph, (b) Kronecker graph with 2^{18} vertices.

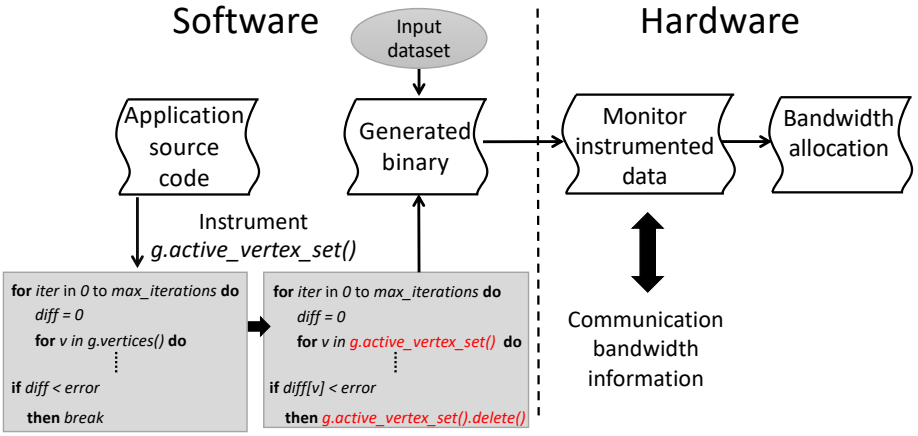


Figure 3·23: Framework of bandwidth allocation using application instrumentation.

another 75.8% of vertices converge in the next 40 iterations, and less than 3% of vertices converge in the last 60 iterations. It can be noted that a significantly high fraction of vertices converge in the first few iterations, leaving a very low fraction of unconverged vertices in later iterations. This observation implies reduced memory accesses in later iterations. Thus, there is an opportunity to reduce the network bandwidth between memory and LLCs by deactivating certain photonic links in later iterations and save photonic network power.

Framework for application-instrumentation-assisted wavelength selection

Figure 3·23 shows the framework of application instrumentation-assisted bandwidth allocation. We instrument the PageRank source code to maintain a data structure called active vertex set (Ozdal et al., 2015). The active vertex set maintains a list of all unconverged vertices. During each iteration, PageRank algorithm operates only on vertices in the active vertex set. At the end of each iteration, we update this active vertex set by deleting vertices that converge during the current iteration.

We study the network characteristics when an instrumented PageRank is executed on *POPSTAR*. Figure 3·24 illustrates the network packets transferred in the

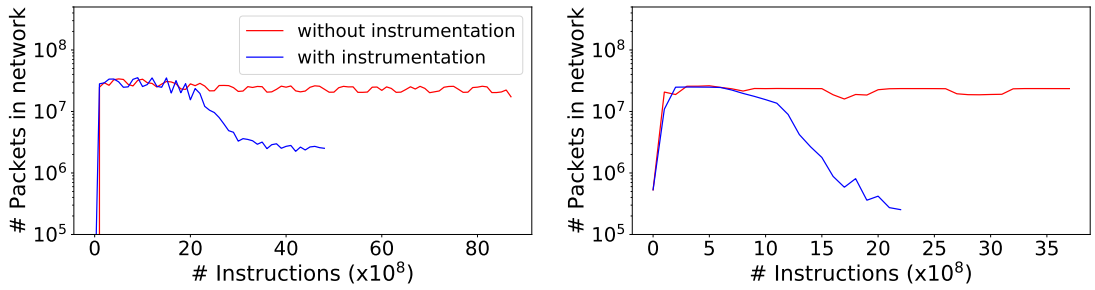


Figure 3·24: Number of packets transferred in the photonic network during application execution for (a) Google webgraph and (b) Kronecker graph with 2^{18} vertices

silicon-photonic link during application execution. Instrumenting the PageRank algorithm with the active vertex set enables the algorithm to execute on lower number of vertices every iteration. This results in an overall decrease in the LLC and main memory traffic, resulting in a lower number of packets in the silicon-photonic link as the application progresses. At the system-level, the wavelength selection policy monitors the number of active vertices in PageRank as the application progresses. This instrumented information is utilized in addition to other network parameters to determine the minimum number of optical channels.

3.5.2 Simulation Results and Analysis

We evaluate *SO-WAVES* to demonstrate the benefits of application-instrumentation-assisted bandwidth allocation. We utilize our simulation framework as described in Section 3.2. We modify the source code in PageRank to maintain the active vertex set that is updated every iteration with unconverged vertices. We model different network bandwidths in Sniper for instrumented and uninstrumented PageRank and determine the number of optical channels that satisfy the L_{thr} for SO-WAVES. Figure 3·25 shows the photonic power savings with SO-WAVES that uses application instrumentation in contrast to SO-WAVES without instrumentation.

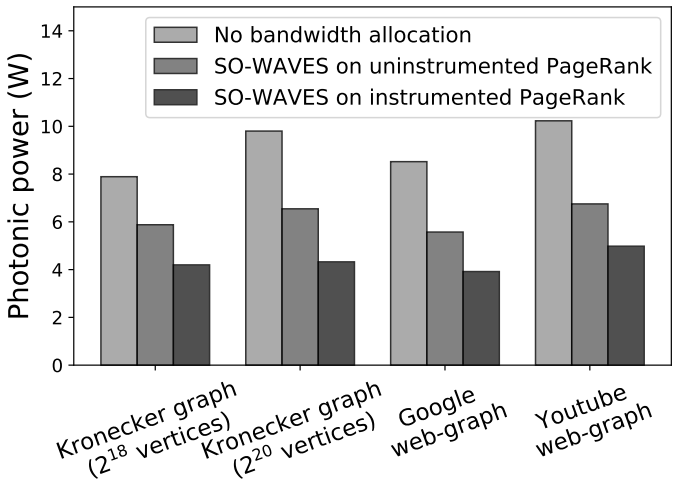


Figure 3·25: Photonic power savings using application instrumentation-assisted bandwidth allocation

For instrumented PageRank, a higher number of vertices converge in the initial interations. *SO-WAVES*, therefore, allocates a higher bandwidth by activating a higher number of optical channels during the initial iterations. For later intervals, *SO-WAVES* activates a lower number of optical channels as the bandwidth demand is reduced with fewer unconverged vertices. Using our instrumentation-assisted bandwidth allocation across four datasets, on average, we reduce 35.13% of photonic network power compared to bandwidth allocation on an uninstrumented PageRank algorithm.

3.6 Chapter Summary

Silicon-photonic links are an effective alternative to electrical links as a high-bandwidth and low-latency chip-scale networks in large manycore systems. However, a cause of concern arises from the device sensitivity towards TV and PV, and the high power overhead in the laser sources, electrical circuitry for E-O-E conversion and the heating power for MRR thermal tuning. This high power overhead limits the

energy-per-bit in silicon-photonic links at high bandwidth operations. Though device-level strategies and architectural designs address these limitations in various degrees, the growing diversity in applications' network requirements demands a system-level solution.

This chapter presents bandwidth allocation policies called wavelength selection that also includes a cross-layer model of device sensitivities and solutions, architectural designs and the system policies. We show that a static wavelength selection policy, *SO-WAVES*, is effective in limiting the photonic power by activating the optimal set of optical channels for an application. To further address the dynamic changes in application's bandwidth requirements, this chapter presents *PROWAVES* that uses a time forecasting model to proactively activate the best set of optical channels for the next phase. We evaluate a diverse set of data-centric HPC and graph applications to demonstrate the potential of wavelength selection. We then present the efficacy of application instrumentation that can assist these wavelength selection policies to further reduce the photonic power. These system-level policies, in tandem with architectural designs and device-level solutions, are promising towards achieving a sub-pJ operation of silicon-photonic links at $> TBps$ on-chip bandwidths.

Chapter 4

Architecting Optically-controlled Phase Change Memory

Recent demonstrations of PCM prototypes that can be optically-controlled have invigorated the concept of an optical memory system. Further merit to such memory systems lies in their ability to directly interface with silicon-photonic links. The non-volatility and the high bit density offered by such an optically-controlled PCM, called OPCM, promises a high-throughput and scalable main memory system. Unfortunately, the architecture and the access protocol used in current DRAM and EPCM systems are designed to align with the properties of electrical addressing. The design of a memory system using OPCM cells, therefore, requires a complete redesign of the microarchitecture and access protocol tailored to the properties of OPCM technology and silicon-photonic links.

The chapter begins with a discussion on design challenges in adapting the current DRAM architecture for OPCM, rendering such a design impractical. We then introduce our proposed Co-designed Optically-controlled phaSe change Memory and Optical link System, *COSMOS*, that includes a hierarchical multi-banked OPCM array, WDM silicon-photonic links to access the OPCM cells, laser sources and an E-O-E control unit that maps the standard memory protocol from processor to OPCM-specific commands. We study data-centric graph and HPC workloads and evaluate their performance and energy consumption when run on a computing system that uses *COSMOS* as the main memory in contrast to an EPCM system.

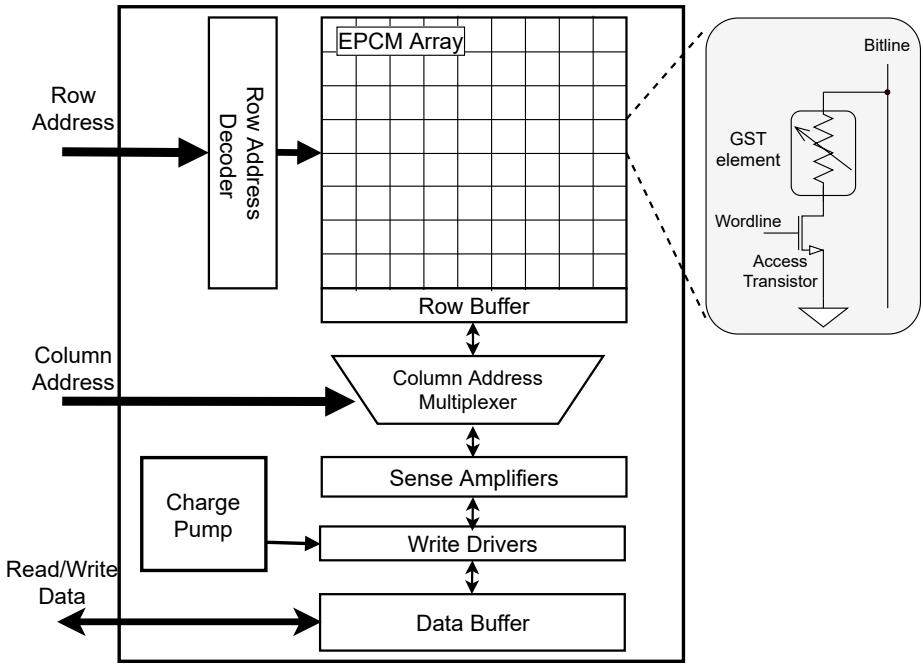


Figure 4·1: A typical EPCM architecture (Lee et al., 2009).

4.1 Challenges with Adapting DRAM Architecture for OPCM

In this section, we first describe a typical EPCM architecture and then explain why such an architectural design is impractical for OPCM. Figure 4·1 shows the architecture of EPCM (Lee et al., 2009). An EPCM cell consists of an access transistor and a GST element. The EPCM array is a hierarchical organization of banks, blocks and sub-blocks, as proposed by Lee *et al.* (Lee et al., 2009). During read or write operation, the EPCM first receives a row address. The row address decoder reads the appropriate row from the EPCM array into a row buffer. The EPCM next receives the column address, and the column address multiplexer selects the appropriate data block from the row buffer. The bitlines of the selected data block are connected to the write drivers for write operation or to the sense amplifiers for read operation. The write operation of an EPCM cell is performed by passing particular current values to SET/RESET the GST element. The charge pumps supply the required drive voltage

corresponding to SET/RESET operation to the write drivers. For read operation, a read current is passed through the GST (Lee et al., 2009) and sense amplifiers determine the voltage on the bitline to read out logic 0 or logic 1.

Naively adapting the EPCM architecture for OPCM, where we just replace the EPCM cells with OPCM cells, raises latency, energy and thermal concerns, thereby rendering such a design impractical. To understand these concerns, let us consider an OPCM array that uses the EPCM architecture from Figure 4·1 with either an optical row buffer or an electrical row buffer. Such an OPCM architecture has the following limitations:

Limitations with optical row buffer: An optical row buffer can be designed using a row of GST elements, whose states are controlled using optical signals. When a row is read from the OPCM array using an optical signal, the data is encoded in the signal's intensity. This intensity is not large enough to update the state of the GST elements in the optical row buffer. So we need to first convert the read value into the electrical domain. Based on this value, we then generate a new optical signal with the appropriate intensity to write the value into the optical row buffer. Therefore, even though the optical signals contain an intensity corresponding to the data, this intensity does not correlate directly to the optical energy required to write that data to the optical data buffer. Essentially we perform an extra O-E and E-O conversion. This necessitates the use of photodetectors, receiver, buffers, transmitter and optical pulse generators, which unnecessarily add to the energy and latency of a memory access. Hence, an optical row buffer is not a viable option.

Limitations with electrical row buffer: An electrical row buffer can be designed either using capacitor cells as in DRAM or using phase change materials that are controlled using electrical current as in EPCM. In both these cases, the row buffer is accessed using electrical addressing.

1. **Impact on read latency:** With an electrical row buffer, the row address is read from the OPCM array into the row buffer, upon receiving the row address. The row address first needs to be converted to an optical pulse, which is applied to OPCM cells in the array to read the data. After optical readout of the entire row, the data needs to be converted back into electrical domain to store it in the row buffer. This requires E-O and O-E conversions on the memory side (irrespective of whether we use electrical or silicon-photonic links for processor-to-memory communication). These conversions increase latency for each read access.

2. **Impact on write latency:** When writing data from the row buffer to the OPCM array, a set of sense amplifiers reads the data from the electrical row buffer. This row buffer data is then converted into an appropriate optical signal intensity using pulse generation circuitry within memory. The optical signals can then be used to write the data to the OPCM cells. Therefore, the write operation too requires additional E-O and O-E conversion (irrespective of electrical or silicon-photonic links for processor-to-memory communication), thereby increasing the latency for each write access.

3. **Impact on read/write energy:** The energy spent in the peripheral circuitry for optical signal generation and readout, as well as in the circuitry for E-O-E conversion increases the active power within memory (Notomi et al., 2014; Bahadori et al., 2016). Given that each read/write operation encounters multiple E-O-E conversions, the energy per read and write access increases.

4. **Thermal issues:** Optical devices such as MRRs are highly sensitive to thermal variations (Padmaraju and Bergman, 2014). The increased power density in OPCM due to additional circuitry for E-O and O-E conversions causes potential thermal violations. The thermal variations due to varied power distribution

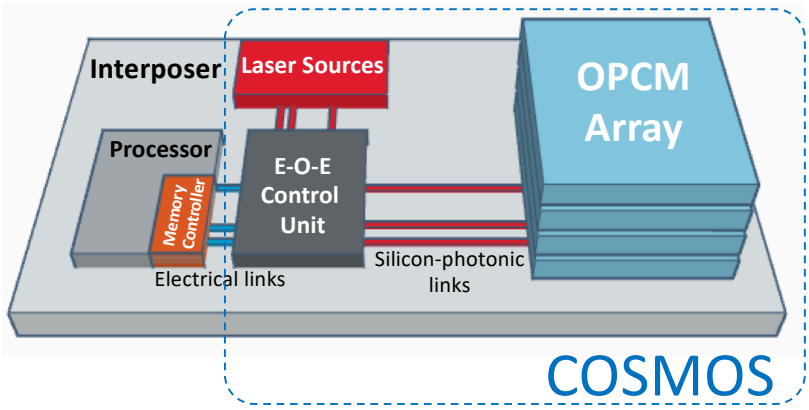


Figure 4·2: Overview of a 2.5D integrated computing system with *COSMOS* as the main memory.

within memory lowers the reliability of the MRR operation. Such a design, therefore, calls for active thermal and power management in OPCM, which further adds to the access latency and energy.

Hence, we argue for the need to redesign the microarchitecture and the read/write access protocol for OPCM in a way that is tailored to the properties of the OPCM cell technology.

4.2 *COSMOS*: OPCM Memory System with Silicon-Photonic links

We introduce Combined Optical phaSe change Memory and Optical link System, *COSMOS*, that provides high read/write throughput and consumes low read/write energy when combined with high-bandwidth-density silicon-photonic links connecting the processor and OPCM array. Figure 4·2 shows a high-level system overview of a 2.5D integrated computing system that uses *COSMOS* as the main memory.[1] *COSMOS* includes a hierarchical design of a multi-banked OPCM array microarchitecture

[1] *COSMOS*-based main memory is agnostic of the integration technology. Since 3D-integrated systems raises thermal concerns and 2D-systems result in large system footprints, we use a 2.5D-integrated system with *COSMOS*.

and a novel read/write access protocol, which are customized to the properties of the OPCM cell. The optical signals in the silicon-photonic links directly access the OPCM cells, eliminating the need for row buffers for intermediate storage. These optical signals are generated by an E-O-E control unit that serves as an intermediary between the memory controller in the processor and the OPCM array. This E-O-E control unit is responsible for mapping the standard DRAM protocol commands sent by the memory controller to optical signals, and then sending these optical signals to the OPCM array. The distinguishing features of *COSMOS* as as follows:

1. The design of the OPCM array in *COSMOS* combines WDM and mode division multiplexing properties of optical signals to deliver high memory bandwidth.

2. The OPCM array is only composed of passive optical components such as MRRs, GST elements and waveguides. As a result, the OPCM array does not dissipate active power during its operation, eliminating the need of thermal management policies.

3. *COSMOS* uses a novel protocol for performing the read and write operations of a cache line in the OPCM array. A cache line is interleaved across multiple banks in the OPCM array to enable high-throughput access. The write data to an OPCM cell is encoded in the intensity of optical signals that uniquely address the cell. The readout of the OPCM cell uses a 3-step operation that measures the attenuation of the optical signal transmitted through the cell, where the attenuation corresponds to a predetermined bit pattern. Since the read operation is destructive, *COSMOS* uses an opportunistic writeback operation of the read data to restore the OPCM cell state.

4. *COSMOS* consists of an E-O-E control unit for seamless integration of the OPCM array with the processor. This E-O-E control unit receives standard

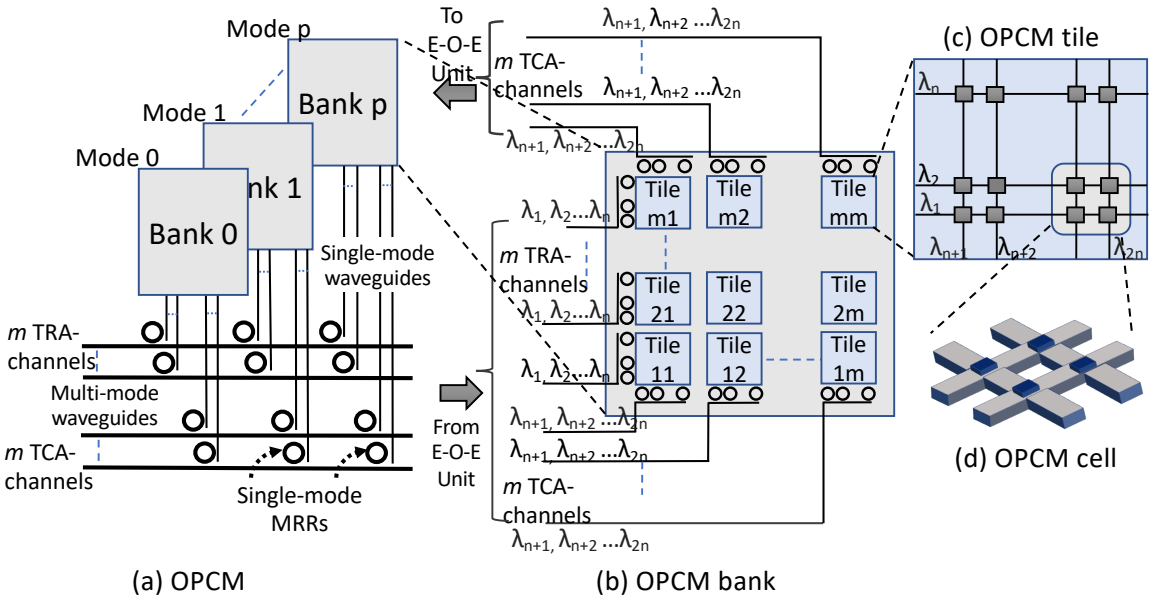


Figure 4·3: *COSMOS* architecture. (a) A multibanked-OPCM uses p optical modes to access p banks. (b) An OPCM bank is an array of $m \times m$ tiles. Every tile is accessed by a TRA-channel and a TCA-channel, each channel containing n optical signals. (c) An OPCM tile is an array of $n \times n$ cells. Every cell is accessed by a unique pair of optical signals. (d) OPCM cells are placed at every waveguide crossing.

DRAM protocol commands from the processor, and converts them into the OPCM address, data, and control signals that are mapped onto optical signals. These optical signals are then used to read/write data from/to the OPCM array. The responses from the OPCM array are converted by the E-O-E control unit back into standard DRAM protocol commands that are sent to the processor.

4.3 OPCM Array Microarchitecture in *COSMOS*

In this section, we describe the microarchitecture of the high-throughput OPCM array in *COSMOS*. The key innovation of the proposed microarchitecture is enabling direct access of OPCM cells by the optical signals in the silicon-photonic links. This direct access avoids the extra E-O and O-E conversions that are required while adapting an EPCM architecture for *COSMOS*. Our OPCM array microarchitecture is a hierarchi-

cal multi-banked design that maximizes the degree of parallelism for read and write accesses within the array. A distinguishing feature of our OPCM array design is that it does not contain any active circuits that consume power; that is, it only contains passive optical devices. To enable high-throughput access of OPCM cells within this array, we propose a novel read and write access protocol for *COSMOS*. Figure 4·3 illustrates the detailed microarchitecture of our proposed OPCM array in *COSMOS* that uses GST as the phase change material. Next, we describe each component of the proposed architecture, particularly focusing on how to access an OPCM cell in the optical domain with minimal E-O and O-E conversions, how to maximize parallelism in our OPCM microarchitecture, and how to perform low-latency read and write operations within the OPCM array.

4.3.1 OPCM Tile

An OPCM tile (see Figure 4·3c) consists of an $n \times n$ array of GST elements, i.e., OPCM cells. The GST elements are placed on top of waveguide crossings as shown in Figure 4·3d. This organization enables every OPCM cell to be accessed using a unique pair of optical signals: one on the associated row and one on the associated column. We need a total of n unique optical signals with wavelengths λ_1, λ_2,..., λ_n that are routed in the rows (one per row waveguide), and n unique optical signals with wavelengths λ_{n+1}, λ_{n+2},..., λ_{2n} that are routed in the columns (one per column waveguide). Wavelengths λ_1 to λ_n together form the Tile Row Access (TRA)-channel, and wavelengths λ_{n+1} to λ_{2n} together form the Tile Column Access (TCA)-channel. A TRA-channel (and similarly each TCA-channel) is mapped to one or more waveguides depending on the number of wavelengths that can be multiplexed in a waveguide. Owing to MLC, each OPCM cell stores b_{cell} bits. The total capacity of an OPCM tile is $n^2.b_{cell}$. A maximum of n cells can be read/written in parallel from a single tile, which gives us a peak throughput of $n.b_{cell}$ bits per read/write access for a tile. We use

a Gray Coding scheme when mapping bit patterns to the states of the GST material. Due to only one-bit difference in bit patterns between adjacent states, there is a very low probability of multi-bit errors during read/write operations. These single-bit errors can be corrected using standard single-bit error correction techniques (Spica and Mak, 2004).

4.3.2 OPCM Bank

Figure 4·3b shows the organization of an OPCM bank. The OPCM bank is composed of an array of $m \times m$ OPCM tiles, and has a total capacity of $m^2.n^2.b_{cell}$ bits. The OPCM bank uses m TRA-channels, one for each row in the bank, and m TCA-channels, one for each column in the bank to communicate with the E-O-E control unit. Each TRA-channel uses λ_1 to λ_n, and each TCA-channel uses λ_{n+1} to λ_{2n}. We design a hierarchical array of OPCM cells (m^2 tiles with n^2 OPCM cells per tile) instead of a large monolithic array ($m^2.n^2$ OPCM cells), as designed by Feldman *et al.* (Feldmann et al., 2017; Feldmann et al., 2019) to decrease the laser power by the optical signals. With our proposed design, the laser sources only need to support $2n$ unique optical signals (in the range of λ_1 to λ_{2n}) instead of the $m.2n$ unique optical signals that would be required in a large monolithic array. We utilize MRRs to couple the optical signals of each TRA-channel and TCA-channel to its corresponding tile. We need n MRRs that are tuned to λ_1 to λ_n in each of the m TRA-channels and n MRRs that are tuned to λ_{n+1} to λ_{2n} in each of the m TCA-channels.

4.3.3 Multi-banked OPCM Array

We interleave a cache-line across multiple banks using mode-division multiplexing. The spatial mode of electromagnetic radiation describes the field pattern of the propagating waves. An optical signal can propagate in several spatial modes. A waveguide can carry a single or multiple such modes of the optical signal. Several prior works

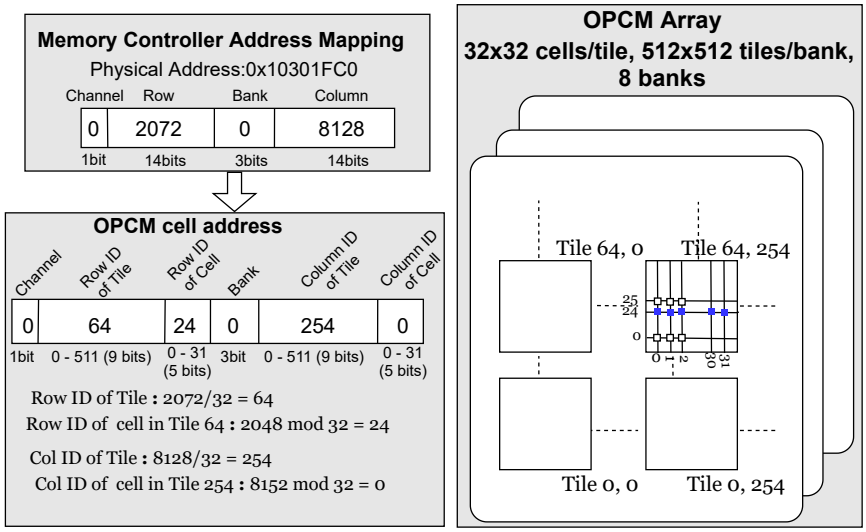


Figure 4·4: Mapping of the physical address in the memory controller to the physical location of the OPCM cell in the OPCM array.

have exploited the multiple spatial modes of optical signals coupled with wavelength-division multiplexing to design high-bandwidth-density silicon-photonic links (Luo et al., 2014; Wu et al., 2017). Figure 4·3a shows the proposed multi-banked organization of the OPCM array using mode-division multiplexing. There are p banks, each supporting one of the p spatial modes of the $2n$ optical signals. For example, Bank 0 only supports mode 0 of all optical signals λ_1,.. λ_n and λ_{n+1},.. λ_{2n}, Bank 1 only supports mode 1 of all optical signals, and so on. The waveguides connecting the OPCM to the E-O-E control unit are multi-mode waveguides, which carry all the p spatial modes of optical signals. We employ single-mode MRRs (Yang et al., 2014; Wang et al., 2017) that couple a single spatial mode of optical signals from the multimode waveguide to a bank.

4.3.4 Address Mapping in *COSMOS*

Figure 4·3e shows an example mapping of the physical address received by the MC to the physical location of cells within the OPCM array in *COSMOS*. A cache line of $64B$ is stored in a total of 128 OPCM cells with $4bits/cell$. We interleave the cache

line across 4 different banks. Within a bank, we map the 128-bit chunk of a cache line to a tile. The tile has 32×32 cells, and so we map that 128-bit chunk to an entire row within a tile. The row (column) field of physical address in the MC is mapped to the row ID of tile (column ID of tile) field and the row ID of cell (column ID of cell) field. In Figure 4·3e, we show how the different fields of the physical address `0x10301FC0` are mapped to bank ID, row ID of tile, column ID of tile, row ID of cell, and column ID of cell.

4.4 Access Protocol in *COSMOS*

To enable high-throughput access of OPCM cells within the OPCM array, we propose a novel read and write access protocol for *COSMOS*. When the MC issues a read or write operation, the row address and column address are entered into the Row Address Queue and Column Address Queue, respectively, and the write data is entered into the Data Buffer in the E-O-E control unit.

4.4.1 Writing a Cache Line to OPCM Array

To write a cache line to the OPCM array, the E-O-E control unit identifies the bank ID, the row ID and column ID of the tile, and the row ID and column ID of the cell within a tile using the address mapping. In our example with 32×32 array of cells in a tile, when writing 128-bit chunk of a cache line, we end up updating all the cells in a row (any misaligned accesses are handled on the processor side). Hence, for writes at cache line granularity, the column ID within a tile is not used. The E-O-E control unit determines the optical intensity that is required at each OPCM cell in the row to write the 128-bit chunk of the cache line. It then breaks down the optical intensity into two signals, one with a constant intensity of I_0 and the other with a data-dependent intensity of I_i, where $i = 1, 2, ..., 128$. The E-O-E control unit modulates the constant intensity I_0 onto the optical signal corresponding to the row (selected by the row ID

of cell) within a tile. The E-O-E control unit then modulates the data-dependent optical intensities (i.e., I_1, I_2, ...,I_{128}) onto the optical signals corresponding to the columns within the tile. The E-O-E control unit transmits the row signal I_0, and the column optical signals $I_1, I_2, ..., I_{128}$ in parallel to write the cache line in the OPCM array. The superposition of the optical signals, i.e., I_0+I_1, I_0+I_2, ..., I_0+I_{128} updates the state of the OPCM cells. Note that since a cache line is spread across 4 banks, the E-O-E control unit modulates data on optical signals to write to an OPCM tile in each of these 4 banks. None of the optical signals individually carries sufficient intensity to trigger a state transition at any cell, so none of the other cells along the row or column are affected.

4.4.2 Reading a Cache Line from OPCM Array

To read a cache line from OPCM array, the E-O-E control unit transmits sub-ns optical pulses along all the columns in a tile that contain the cache line and measures the pulse attenuation. However, there are multiple OPCM cells along each column and so the output intensity of optical signals will be attenuated by all cells in that column. It is, therefore, not possible to determine the OPCM cell values using a one-pulse readout. Hence, we use a three-step process for read operation of OPCM array in *COSMOS*. (1) To read a cache line, the E-O-E control unit first determines the bank ID, row ID and column ID of tile, row ID and column ID of cell. The E-O-E control unit transmits a read pulse RD_1 through all the columns in a tile containing the cache line. Note that since a cache line is spread across 4 banks, the E-O-E control unit transmits RD_1 on the 4 different optical modes corresponding to the 4 banks. Each read pulse is attenuated by all the OPCM cells in the column. The attenuated pulses are received by the E-O-E control unit, which records the intensities of these attenuated pulses as $I_{1,1}$, $I_{2,1}$, ..., $I_{128,1}$. These intensities are converted into electrical voltage and stored as $V_{1,1}$, $V_{2,1}$, ..., $V_{128,1}$. (2) The E-O-E control unit then

transmits a RESET pulse to the OPCM cells of the cache line, i.e., all the cells along a row within a tile. All the cells along the row are now amorphized and have 100% optical transmission. ③ The E-O-E control unit then sends a second read pulse RD_2 through all the columns of a tile containing the cache line. Each read pulse is again attenuated by all OPCM cells in the column. Given that step 2 amorphized all OPCM cells of the cache line, the output pulse intensities are different from those in step 1. The attenuated pulses are received by the E-O-E control unit, which records the intensities of these attenuated pulses as $I_{1,2}$, $I_{2,2}$, ..., $I_{128,2}$. These intensities are converted into electrical voltage and stored as $V_{1,2}$, $V_{2,2}$, ..., $V_{128,2}$. The E-O-E control unit computes the difference of the stored voltages of steps 1 and 3, i.e., $V_{1,1} - V_{1,2}$, $V_{2,1} - V_{2,2}$, ..., $V_{128,1} - V_{128,2}$. This difference is used to determine the cache line data stored in the OPCM cells.

4.4.3 Opportunistic Writeback for Read Operation

The RESET operation in step 2 of the read operation destructs the original data in the OPCM cells. We, therefore, perform an opportunistic writeback of the cache line to the OPCM cells. After completing the 3 steps of the read operation, the read data and the address are saved into a *holding* buffer in the E-O-E control unit. When there are no pending read or write operations from the MC, the E-O-E control unit reads the data and its address from the *holding* buffer and writes the data back to the OPCM array. This writeback operation does not block any critical pending read and write operations coming from the MC. The dependencies in read and write requests between the *holding buffer* and the data buffer is handled in the E-O-E control unit. For a Read-After-Read case, the second read operation reads the data from the *holding buffer* if present. If the data is not in the *holding buffer* then the second read operation just uses the 3-step process + writeback (described above) to complete the read operation. For a Write-After-Read case, if the write address

matches the read address and there is an entry for that read in the *holding buffer*, then the corresponding entry in the *holding buffer* is invalidated. The write data is then entered into the data buffer and then written into the appropriate OPCM array. The Write-After-Write and Read-After-Write are not an issue as the E-O-E control unit processes them in order.

4.5 E-O-E control Unit Architecture

We design the E-O-E control unit as an interface between the processor and the OPCM array. An LLC miss in the processor leads to a memory access request being sent to the MC. The MC sends standard DRAM access protocol commands to the E-O-E control unit.[2] The E-O-E control unit maps these commands onto optical signals that read/write the data from/to OPCM array.

For write operation, depending on the write address, the E-O-E control unit selects specific optical signals in TRA-channel and TCA-channel, and maps the write data onto an appropriate pulse intensity of these optical signals. For read operation, using the 3-step process explained in Section 4.4.2, the E-O-E control unit filters the optical signals received from the OPCM cell and determines the value stored in the cell. The E-O-E control unit consists of five sub-units: a data modulation unit (DMU), an address mapping unit (AMU), a pulse selector unit (PSU), a pulse amplification unit (PAU), and a pulse filtering unit (PFU). Each bank has a dedicated set of these five sub-units. Figure 4·5 shows the various sub-units in the E-O-E control unit.

4.5.1 Data Modulation Unit (DMU)

The DMU generates the modulation voltage and bias currents corresponding to the write data. For write operation, we divide the *k-bit* write data into k/b_{cell} entries and

[2]Given that OPCM cells do not require Activate/Precharge/Refresh operations, the E-O-E control unit does not take any action for these commands. Though we can design an OPCM-specific MC, our goal is to enable the OPCM operation with a standard MC in any processor.

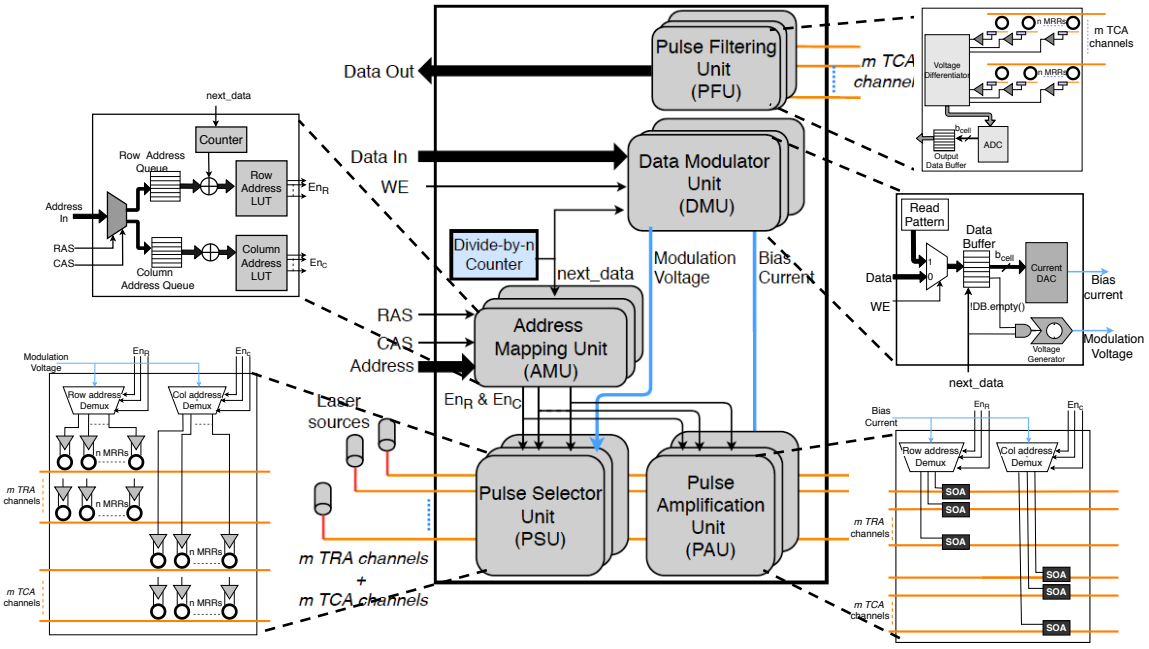


Figure 4·5: (a) E-O-E control unit design. DMU: Generates the modulation voltage and the bias current corresponding to read/write data. AMU: Determines optical signals that correspond to read/write address. PSU: Selects the optical signals. PAU: Amplifies the optical signals using the bias current. PFU: Filters the optical signals to read cell data.

store them in the Data Buffer (DB) (one entry per cell) of the DMU. For each entry in DB, the DMU generates a modulation voltage (same fixed value for all possible values that can be written to a cell) and a bias current depending on the exact value that needs to be written. The DMU uses a voltage generator for generating the modulation voltage and a current DAC for generating the bias current. It takes T_{EO} cycles to map each entry from the DB to the appropriate optical signals. So the DMU generates the modulation voltage and bias current every T_{EO} cycles, if the DB is not empty, giving a write throughput of $1/T_{EO}$. The modulation voltage is input to the PSU, and the bias current is input to the PAU.

In our 3-step read operation (described in Section 4.4.2), the DMU generates a modulation voltage to select the optical signal. For step 2 in the read operation,

it generates a modulation voltage and a bias current that corresponds to a RESET operation.

4.5.2 Address Mapping Unit (AMU)

The AMU receives the address bits from MC in parallel with RAS and CAS signals, and maps them to appropriate row and column optical signals. For a given read/write address, the AMU generates two enable signals, En_R and En_C. The En_R signal is used in the PSU to select the appropriate optical signal in the TRA-channel and to select the associated silicon-photonic link driver. The En_R signal is also used in the PAU to select the amplifier associated with the optical signal. Similarly, En_C signal is used in the PSU to select the optical signal in the TCA-channel and the associated silicon-photonic link driver, and to select the amplifier in the PAU. The AMU is synchronized with DMU such that the enable signals from the AMU, and the modulation voltage and bias current from the DMU reach the PSU and PAU at the same time.

4.5.3 Pulse Selector Unit (PSU)

The PSU uses En_R and En_C to select the appropriate optical signals in TRA-channel and TCA-channel, respectively, for the read/write operation. It also uses the En_R and En_C signals to route the modulation voltage (received from the DMU) to the silicon-photonic link drivers associated with the optical signals selected in the TRA-channel and TCA-channel, respectively. The driver uses this modulation voltage to detune the MRRs corresponding to the selected optical signals, and allows the optical signals to continue to the PAU. The remaining MRRs filter out and block the other optical signals from reaching the PAU.

4.5.4 Pulse Amplification Unit (PAU)

The PAU amplifies the optical signals (received from PSU) in the TRA-channel and the TCA-channel using the bias current received from the DMU. The PAU uses semiconductor optical amplifiers (SOA) for amplification, where the gain is a function of the input bias current (Connelly, 2007). The amplified optical signals traverse through the silicon-photonic link to the target OPCM cell.

4.5.5 Pulse Filtering Unit (PFU)

The PFU is only involved in the OPCM read operation. For step 1 and step 3 of the read operation (described in Section 4.4.2), the PFU receives an optical signal back from OPCM. The PFU uses MRRs for filtering the optical signals, and photodetectors and transimpedance amplifiers to generate V_{1e} during step 1 and V_{2e} during step 3. A voltage differentiator calculates V_{1e} - V_{2e}, and this difference is input to an ADC to get the digital value stored in the OPCM cell. The PFU aggregates the values from the 128 OPCM cells and then send a $64B$ cache link back to the processor.

4.6 Experimental Evaluation and Analysis

4.6.1 Evaluation Methodology

Computing System with *COSMOS*

We use an 8-core processor with fully-coherent LLC for our evaluation. We primarily evaluate OPCM with 4-bit MLC (given that OPCM cells with 5 $bits/cell$ has been prototyped (Li et al., 2019)) against an EPCM with 2-bit MLC. For processor-memory interconnects, we consider electrical as well as silicon-photonic links, with $1GT/s$ transfer rate. Table 4.1 provides details of the processor and memory configurations.

The OPCM is organized as a single rank connected to a memory channel on the MC via the E-O-E control unit. Each of the 8 OPCM banks has a set of dedicated

Table 4.1: Architectural details of the computing system for *COSMOS* evaluation

Processor, On-chip caches	
Cores	8-core, 1GHz x86 ISA, our-of-order
L1 caches	32kB split L1 I$ and D$, 2-way, 2-cycle
L2 cache	Shared L2$, 2MB, 8-way, 20-cycles, 64B line size
Main memory ($2GB$)	
EPCM (Choi et al., 2012)	4 banks, 8 devices/rank, 1 rank/channel, bus width=64, burst length=4 $t_{SET} = 120ns$, $t_{RESET} = 50ns$, $t_{read} = 60ns$, $t_{BURST} = 4ns$
OPCM (Ríos et al., 2015; Li et al., 2019)	8 banks, 1 rank/channel, 1 device/rank bus width=$32 \times b_{cell}$, burst length=8 $t_{SET} = 160ns$, $t_{RESET} = 25ns$, $t_{read} = 25ns$, $t_{BURST} = 1ns$, $t_{EOE} = 5ns$

DMU, ATU, PSU, PAU, and PFU sub-units in the E-O-E control unit. The average SET latency is t_{SET} + t_{EOE}, $165ns$, the RESET latency is t_{RESET} + t_{EOE}, $30ns$, and the read latency is t_{read} (time for 3-step read operation) + t_{EOE}, i.e. $30ns$. A maximum of $t_{SET}/t_{EOE} = 32$ writes can be issued from the E-O-E control unit to OPCM in parallel. The effective bus width between E-O-E control and OPCM for write operation is, therefore, $32 \times b_{cell}$. So, we can write $32 \times b_{cell}$ in parallel. A maximum of $t_{read}/t_{EOE} = 5$ reads is issued from the E-O-E control unit to OPCM in parallel. So, we can read $5 \times b_{cell}$ in parallel.

Simulation Framework

We model the architectural specifications of the system in Gem5 (Binkert et al., 2011). We conduct full-system simulations in Gem5 with Ubuntu 12.04 OS and Linux kernel v4.8.13. We fast-forward to the end of Linux boot and execute each workload for 10 billion instructions. The main memory models for DDR4 are based on DRAMSim2 (Rosenfeld et al., 2011). For modeling EPCM and OPCM, we integrate NVMain2.0 (Poremba et al., 2015) in Gem5. NVMain2.0 provides support for modeling MLC cells, and variable SET and RESET latencies.

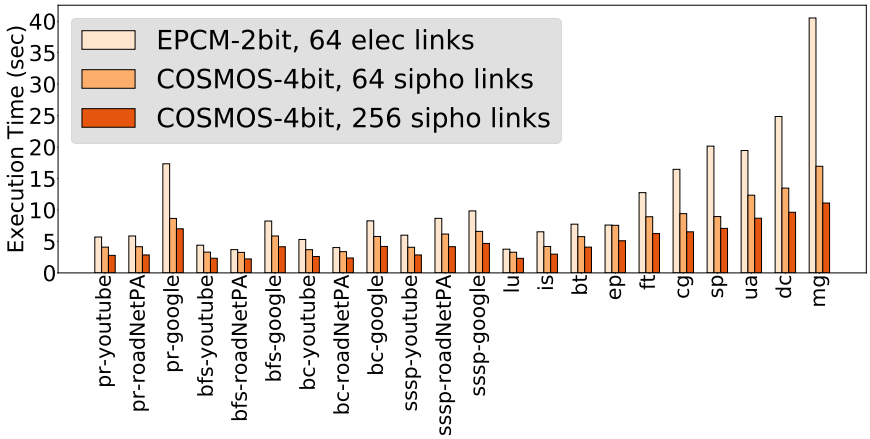


Figure 4·6: Performance comparison of *COSMOS* with EPCM.

Workloads

We simulate graph applications from GAP-BS benchmark (Beamer et al., 2015) and HPC applications from NAS-PB benchmark (Bailey et al., 1991). We evaluate the graph applications on three different input datasets from SNAP repository (Leskovec and Krevl, 2014): Google web graph (*google*), road network graph of Pennsylvania (*roadNetPA*) and Youtube online social network (*youtube*). For HPC applications from the NAS-PB benchmark, we use the large dataset. We execute 8 threads of an application in a workload, with each thread running on a dedicated core.

4.6.2 Performance Comparison with EPCM

We first compare EPCM (2bit MLC) that uses 64 processor-to-memory electrical links with a *COSMOS* system (4bit MLC) that also uses 64 processor-to-memory silicon-photonic links, and a *COSMOS* system (4bit MLC) that uses 256 processor-to-memory silicon-photonic links. Figure 4·6 shows the overall performance (execution time in seconds) for systems with these three configurations. Compared to the EPCM-2bit with 64 electrical links, the OPCM-4bit with 64 silicon-photonic links has on average 1.52× better performance across all workloads.

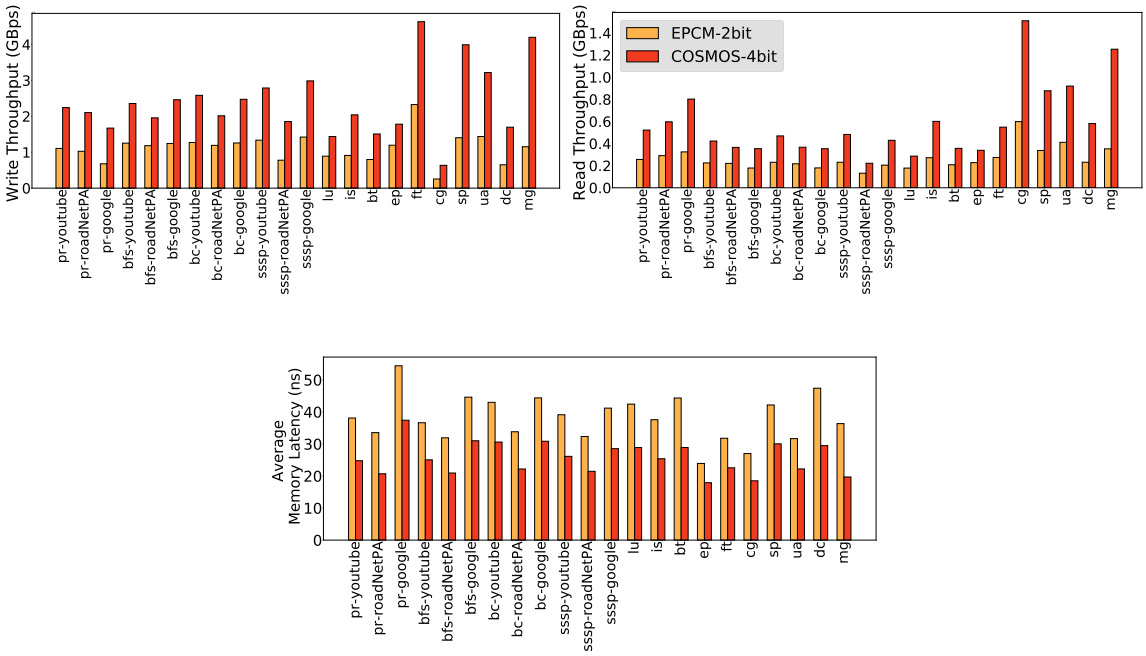


Figure 4·7: Comparison of EPCM-2bit with 64 electrical links and *COSMOS*-4bit in terms of (a) write throughput, (b) read throughput, (c) average memory latency

This performance improvement is due to the higher *bits/access* throughput of *COSMOS* resulting from higher MLC capability of OPCM cells and the single-cycle latency in silicon-photonic links. Increasing the number of silicon-photonic links from 64 to 256 further improves the system performance. Compared to EPCM-2bit using 64 electrical links, we observe performance improvement of 2.14× on average for graph and HPC workloads with *COSMOS*-4bit using 256 silicon-photonic links. These performance benefits are due to denser WDM capacity in silicon-photonic links.

We next study the increased throughput in *COSMOS* in contrast with an EPCM system. Figures 4·7a and 4·7b show the read and write throughput, respectively, of *COSMOS*-4bit with 256 silicon-photonic links and EPCM-2bit with 64 electrical links. Compared to EPCM-2bit with 64 electrical links, *COSMOS*-4bit with 256 silicon-photonic links theoretically has 8× higher peak bandwidth, i.e., 2× due to

Table 4.2: Optical power budget for $2GB$ *COSMOS*. The table shows optical power losses and SOA gain along the optical path from laser source to OPCM cells.

Loss/gain component	**Single**	**Total**
Coupling loss	$-1dB$	$-1dB$
MRR drop loss (E-O-E control)	$-0.5dB$	$-0.5dB$
MRR through loss (E-O-E control)	$-0.05dB$	$-3.2dB$
Propagation loss (Laser to SOA)	$-0.3dB/cm$	$-0.09dB$
SOA gain	$+20dB$	$+20dB$
Propagation loss (SOA to OPCM)	$-0.3dB/cm$	$-0.09dB$
Bending loss	$-0.167dB$	$-0.167dB$
MRR drop loss (OPCM)	$-0.5dB$	$-0.5dB$
MRR through loss (OPCM)	$-0.05dB$	$-3.2dB$
Propagation loss (in OPCM)	$-0.03dB/cm$	$-4.91dB$
Max. power required to SET the GST	$\frac{135pJ}{250ns}$	$-2.67dBm$
Power per optical signal		$-7.22dBm = 0.19mW$
Laser wall-plug efficiency		20%
Total laser power		$16.38W$

higher MLC capability and the 4× due to the increased number of processor-to-memory links. Therefore, it is possible to issue increased number of parallel read and write operations in *COSMOS*-4bit. From figure 4·7a and figure 4·7b we observe that *COSMOS*-4bit has 2.09× higher read throughput and 2.15× higher write throughput, respectively, than EPCM-2bit for graph and HPC workloads. This increased read and write throughput of *COSMOS*-4bit hides the long write latencies. Figure 4·7c shows that the average memory latency (read+write) of *COSMOS*-4bit is 33% lower than EPCM-2bit across all workloads. The key insight from this study is that the increased read and write throughput provided by the higher MLC capability and the silicon-photonic links hide the long write latencies in *COSMOS*.

Energy Consumption of *COSMOS*

The primary contributors to the overall power consumption during the read and write operations are the different active components in the E-O-E control unit and the laser

sources that drive the silicon-photonic links. The OPCM array in *COSMOS* consists of only passive optical devices, so it does not consume any active or idle power. The electrical power consumed in the laser source is proportional to its optical output power, which in turn depends on the optical losses in the path of the optical signal and the minimum power required to switch the farthest GST element. Table 4.2 lists the optical losses in the various components and the maximum switching power required at the GST element in decibels (dB). The various optical losses and SOA gains are obtained from prior characterization works (Batten et al., 2008; Grani and Bartolini, 2014; Shang et al., 2015; Li et al., 2019). By accounting for the wall-plug efficiency, we calculate the minimum required laser power per optical signal as $0.95mW$. Aggregating the laser power for all optical signals required in a $2GB$ *COSMOS* system, we get a total laser power of $16.38W$.

In the E-O-E control unit, the current-DAC in DMU and the ADC in PFU consume $0.3mW$ each (Rekhi et al., 2019). For OPCM-4bit, 32 write operations can be issued in parallel per bank, i.e., we can write $32 \times b_{cell} \times 8 = 128B$ in parallel with an average write latency of $160ns$. That aggregates to writing 2 cache lines of $64B$ each in parallel. A cache line is interleaved across 4 banks and is row aligned in an OPCM tile. Therefore, we need 4 row optical signals and 4×32 column optical signals to write a cache line. Therefore, the total power of the laser, SOAs and DACs in the E-O-E control unit for writing 2 cache lines in parallel aggregates to $334.8mW$. This equates to $40.68pJ/bit$ for writing to *COSMOS*-4bit.

For read operation, up to 5 read operations can be issued in parallel per bank, i.e., $5 \times b_{cell} \times 8 = 20B$ bits in parallel, with a read latency of $25ns$. The total power of the laser, SOA, DAC, and ADC in E-O-E control for 5 parallel read operations is $9.3mW$, resulting in a read energy of $11.6pJ/bit$ for *COSMOS*-4bit. The energy consumed in the electrical links connecting the processor and the E-O-E control unit

Table 4.3: Energy-per-bit for read and write operations in EPCM and *COSMOS* with 4-bit OPCM cells

Energy per bit (pJ/bit)	**EPCM-2bit**	**OPCM-4bit**
Write	243	40.68
Read	44.5	11.6
Opportunistic Writeback	NA	40.68

is $< 1pJ/bit$ (Coskun et al., 2020). For EPCM, we use NVSim (Dong et al., 2012) to compute the energy-per-bit for read and write operations. The opportunistic write-back operation in *COSMOS* uses the same energy as that required for write operation. Table 4.3 shows the energy-per-bit for EPCM-2bit and *COSMOS*-4bit. The read and write energy-per-bit of *COSMOS*-4bit are 3.8× and 5.97× lower, respectively, than that of EPCM-2bit.

4.6.3 Sensitivity Analysis with Optical Parameters

In this section, we evaluate the sensitivity of *COSMOS* performance with respect to several design variables.

MLC capacity

Rios *et al.* gave the first demonstration of a 2-bit OPCM cell operation (Ríos et al., 2015). Advances in optical signaling and control have resulted in the demonstration of denser multilevel OPCM cells. Li *et al.* demonstrated 5-6 bits per OPCM cell (Li et al., 2019). Further prototypes have demonstrated scalable integration of OPCM cell arrays in silicon and silicon nitride platforms (Li et al., 2020; Feldmann et al., 2019). With the maturity in optical integration technologies, OPCM technology with 8 *bits/cell* is expected in the near future (Li et al., 2019). We compare the performance of systems having OPCM with different MLC capabilities, ranging from 2 *bits/cell* to 8 *bits/cell*, for the same number of silicon-photonic links (see Fig-

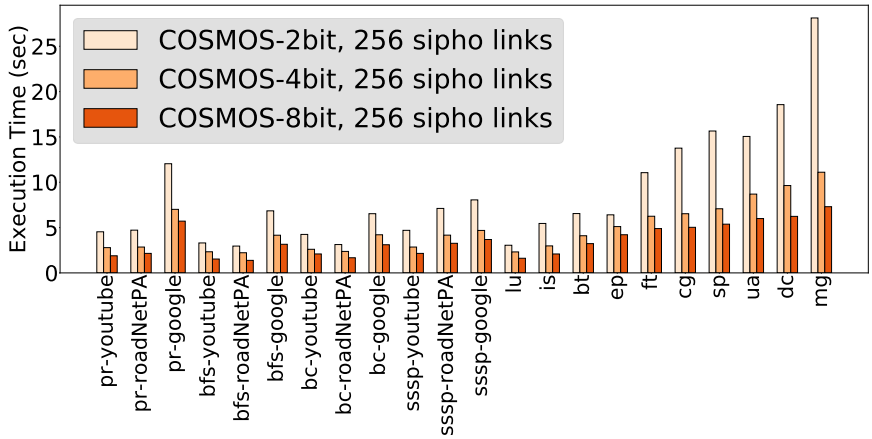


Figure 4·8: Performance comparison of *COSMOS* with different MLC OPCM cells.

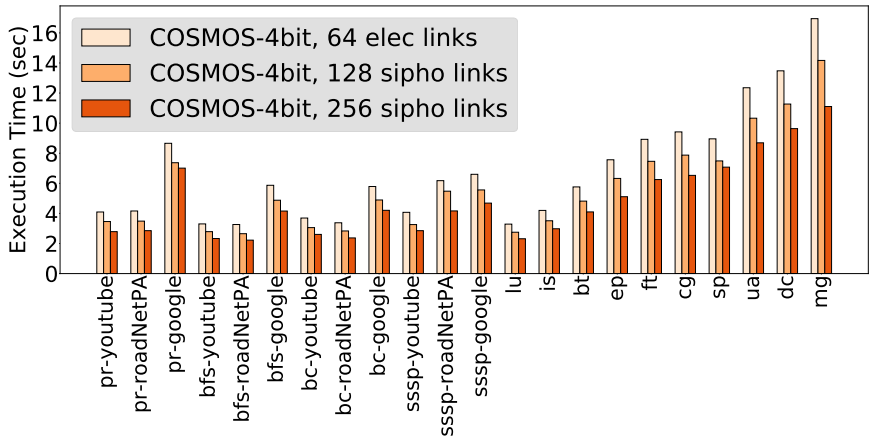


Figure 4·9: Performance comparison of *COSMOS* with different number of optical channels in the silicon-photonic link.

ure 4·8). The performance across applications increases by 39.2% on average as the MLC capacity increases from 2 *bits/cell* to 4 *bits/cell* and by 26.4% as the MLC capacity increases from 4 *bits/cell* to 8 *bits/cell*. As the MLC capability of OPCM cell increases, the *bits/access* number increases for the same number of processor-memory links, thereby increasing the memory throughput.

Silicon-photonic links

We compare the performance of systems with OPCM-4bit having different number of

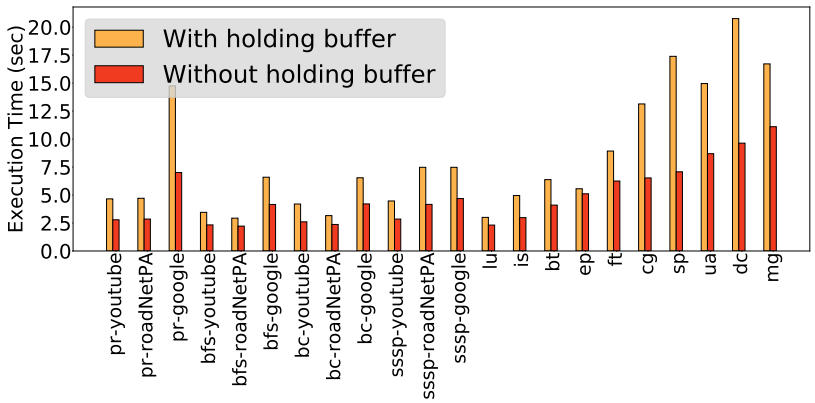


Figure 4·10: Performance comparison of *COSMOS* with and without holding buffer for opportunistic writeback in read operation.

silicon-photonic links. This corresponds to the increasing number of optical channels in the silicon-photonic links. Figure 4·9 shows the system performance of OPCM-4bit with increasing optical channels. Increasing the number of optical channels in silicon-photonic link enables parallel read and write accesses to a higher number of OPCM cells. Due to this higher read and write throughput, as the number of optical channels increases, the overall system performance improves. We observe a performance improvement of 29.3% (on average) for OPCM-4bit with 256 silicon-photonic links over OPCM-4bit with 64 links. A higher number of densely multiplexed optical signals in the silicon-photonic link increases the peak memory bandwidth, and therefore, improves the overall system performance.

Holding Buffer

Figure 4·10 shows the system performance comparison with and without the *holding buffer*. In absence of the *holding buffer*, the read data needs to be written back to the OPCM cells immediately after readout because the read operation is destructive. Therefore, the complete read operation incurs a total latency of readout latency ($25ns$) + writeback latency ($160ns$). In contrast, when the E-O-E control unit consists of a *holding buffer*, the read data is stored in the *holding buffer* at the end of read operation.

The data from the *holding buffer* is written back to the OPCM cells only when the DB in the E-O-E control unit is empty, ensuring that the writeback operation does not stall any critical read and write operations. Using the highest read and write rate of the workloads that we evaluated, we determine that a *holding buffer* with 16 cache line slots, i.e., $1KB$, is enough to avoid any memory read/write stalls. The *holding buffer* occupies less than 1000 μm^2 area and can be integrated into the E-O-E control unit with minimal overhead.

4.6.4 OPCM Endurance Analysis

Similar to EPCM, OPCM cells have lower write endurance due to cell wearout. The OPCM cell endurance depends on how often we write to that cell (Qureshi et al., 2009a). Given that the read operation in OPCM also includes a write (RESET) in step 2, the read rate also needs to be accounted for in the endurance analysis. We calculate the average read and write rate across all the graph and HPC workloads and then estimate the OPCM lifetime using the following equation proposed by Qureshi *et al.* (Qureshi et al., 2009b):

$$Y = \frac{S.W_m}{B.F.2^{25}}$$

where, Y is lifetime in years, W_m is maximum allowable writes per cell, B is write rate in bytes/cycle, F is processor frequency in Hz, and S is OPCM capacity in bytes.

Figure 4·11 plots the average lifetime for OPCM with different MLC capabilities. Here we assume that for a given memory size, all MLC options use the same number of silicon-photonic links. Hence, the OPCM with 8-bit MLC has higher effective throughput than the OPCM with 4-bit MLC. As a result, an application running on OPCM-8bit runs faster than an application running on OPCM-4bit. Hence, for an application, even if the absolute number of memory writes is same for both OPCM-

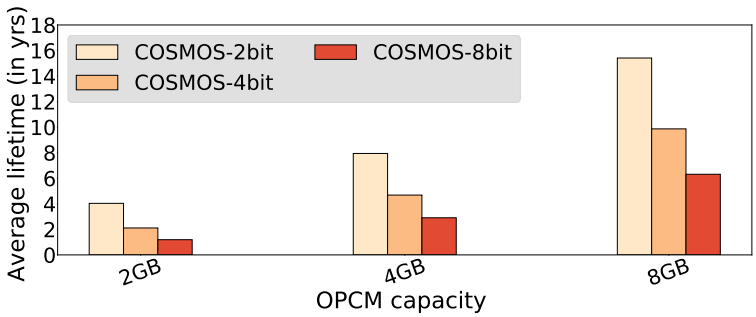


Figure 4·11: Average lifetime (in years) of *COSMOS* with different MLC capabilities for different memory capacities.

8bit and OPCM-4bit, the number of *writes/second* to OPCM-8bit is higher than the number of *writes/second* to OPCM-4bit. As a result, the lifetime of OPCM-8bit is lower than that of the OPCM-4bit and OPCM-2bit.

4.6.5 Area Efficiency of *COSMOS*

To design the OPCM array in *COSMOS*, we use the prototype of a GST element developed by Rios *et al.* (Rios et al., 2014; Ríos et al., 2015). This prototype demonstrates the MLC characteristics in $500nm \times 500nm$ GST element with $500nm$ separation between adjacent GST elements. We use 3D stacking for the OPCM array, with different banks stacked vertically (one bank per layer). The multi-mode waveguides are routed vertically, and in each layer single-mode MRRs filter out the mode of all optical signals that belong to its corresponding bank. We calculate the area of a bank as a function of the number of tiles in a bank, number of cells per tile, spacing between two cells, size of each cell, and the size of MRRs required in a bank.[3] We calculate the bit density of *COSMOS* as a function of the number of OPCM bank layers in the stack, the area of each OPCM bank, and the capacity of each bank.

We compare the area and bit density of the 3D-stacked OPCM array in *COSMOS*

[3]The tile size is limited by the number of unique optical signals in C and L bands with sufficient guardbands (32 in our case). The number of banks depends on the number of unique electromagnetic modes that can be supported (8 in our case).

Table 4.4: Bit density ($bits/mm^2$) of different memory technologies.

Memory technology	**Area of** $2GB$ **memory**	**Bit density** ($bits/mm^2$)
DDR4	$224mm^2$	$9.14MB/mm^2$
HBM2.0	$91.99mm^2$	$22.26MB/mm^2$
EPCM-2bit	$336mm^2$	$6.095MB/mm^2$
3D OPCM-4bit array in *COSMOS*	$268.43mm^2$	$7.63MB/mm^2$
3D OPCM-8bit array in *COSMOS*	$67.1mm^2$	$30.52MB/mm^2$

with DDR4, 3D-stacked HBM2.0 and EPCM-2bit memory system (see Table 4.4). With current OPCM cell footprints, 3D-stacked OPCM-4bit has 1.2× and 2.9× lower bit density than DDR4 and HBM2.0, respectively, and 1.25× higher bit density than EPCM-2bit. Due to its higher MLC capacity, 3D OPCM-8bit has 3.4×, 1.4× and 5× higher bit density than DDR4, HBM2.0 and EPCM-2bit, respectively. Nevertheless, device-level research efforts have demonstrated that GST elements are highly scalable and can retain the electrical and optical characteristics at amorphous and crystalline states (Raoux et al., 2008; Wang et al., 2020). An aggressive chip prototype with $200nm \times 200nm$ GST element with $50nm$ separation has been recently fabricated (Hosseini et al., 2014). These aggressive optical fabrication technologies promise achieving several orders higher densities for OPCM arrays than current DRAM technologies.

4.6.6 Performance and Energy Comparison with DRAM

The overarching goal of *COSMOS* is to replace DRAM systems that are used widely in computing systems. We noted that though all other NVM systems (in their current form) provide non-volatility, data persistence and high scalability, their poor performance negates their benefits and makes them impractical to replace DRAM systems. We, therefore, compare the performance and energy of a DDR4 system with 64 electrical links, a DDR4 system with 256 silicon-photonic links (Beamer et al.,

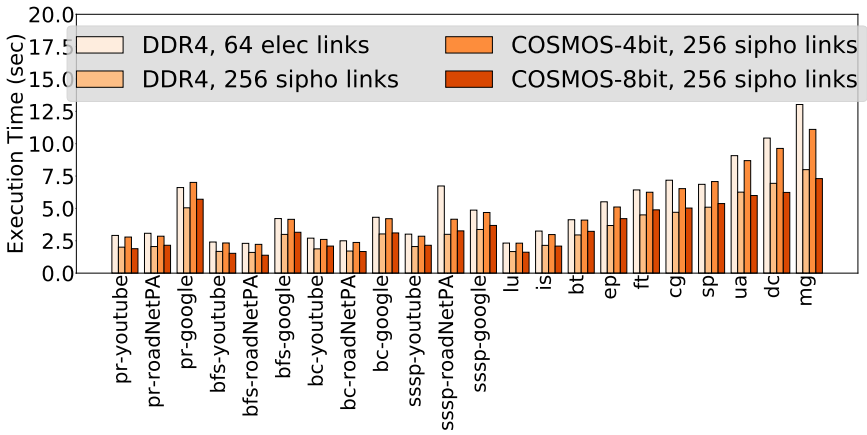


Figure 4·12: Performance comparison of DDR4 and *COSMOS* with OPCM-4bit array.

2009), *COSMOS*-4bit with 256 silicon-photonic links, and *COSMOS*-8bit with 256 silicon-photonic links. Figure 4·12 shows the overall system performance across the four configurations. For DDR4, replacing 64 electrical links with 256 silicon-photonic links provides 32% average performance improvement. This improvement results from the higher throughput due to dense WDM and single-cycle latency of silicon-photonic links. With *COSMOS*-4bit, we obtain 5.6% improvement in performance compared to DDR4 with 64 electrical links. This is in stark contrast to EPCM-2bit, which performs $3-4\times$ worse than DDR4. *COSMOS*-8bit with 256 silicon-photonic links performs 30.6% better than DDR4 with 64 electrical links and 2.1% better than DDR4 with 256 silicon-photonic links. The increased read and write throughput due to the higher MLC capacity and dense WDM silicon-photonic links reduces the average memory access latency of *COSMOS*. Figure 4·7c shows the the average memory latency in *COSMOS* is $33.64ns$ across all workloads, which is lower than DDR4 DRAM ($40ns$). Moreover, from Table 4.3 we observe that energy-per-access for write operation in *COSMOS*-4bit is similar to that of DDR4 DRAM ($40pJ/bit$) and the energy-per-access for read operation in *COSMOS*-4bit is $3.45\times$ lower than DDR4 DRAM ($40pJ/bit$).

Though we evaluate DDR4 memory with silicon-photonic links, such a system encounters several design challenges. To support silicon-photonic links in DDR4, memory requests from MC require an E-O conversion in MC and an O-E conversion in memory, and memory responses from DDR4 require an E-O conversion in memory and an O-E conversion in MC. Effectively, we need two extra conversions on the memory side. The active peripheral circuitry to support E-O-E conversions within memory increases the power density and raises thermal concerns. Due to the high thermal sensitivity of MRRs, there is a need for active thermal management. The power and resulting thermal concerns affect the reliability of optical communication in DRAM systems.

We observe that *COSMOS* with 4 *bits/cell* OPCM array demonstrates similar performance and energy characteristics as current DDR4 systems, while *COSMOS* with 8 bits/cell OPCM array improves performance. This is particularly exciting as *COSMOS* can be scaled further, and unlike DRAM it has zero leakage power and non-volatility, making it a viable replacement for DRAM in the near future.

4.7 Chapter Summary

With DRAM technologies facing critical scaling challenges, the scalability of memory systems to meet the ever-increasing capacity and bandwidth requirements of applications is causing a major concern. In contrast, non-volatile memory systems including EPCM systems suffer from long write latencies and high write energies, yielding poor performance and high energy consumption for data-centric applications. This chapter presents a disruptive memory system, *COSMOS* that is based on the concept of PCM cells with optical control. OPCM cells have already shown tremendous promise due to their higher bit density owing to increased MLC capacity. They also present an opportunity to interface with high-bandwidth-density silicon-photonic links.

COSMOS is a first-of-its-kind memory architecture consisting of a dense OPCM array that provides a high read and write throughput when combined with silicon-photonic links. This chapter also presents the design of an E-O-E control unit that acts as an intermediary between any off-the-shelf processor and the OPCM array. We demonstrate that a computing system with *COSMOS* delivers high performance and low energy consumption, which are comparable to DRAM systems, at the same time providing non-volatily, higher bit density and zero leakage power.

Chapter 5

Memory Management in Heterogeneous Memory Systems

In chapter 4, we presented *COSMOS* as a main memory module providing increased bandwidth than DDR4, but with long write latency. Workloads with high memory parallelism yield increased performance in *COSMOS*. However, workloads that are more sensitive to memory latency would result in sub-par performance with *COSMOS* as the main memory. Due to the inherent tradeoff of bandwidth-power in memory modules, a single memory module can never provide the lowest latency, highest bandwidth and consume lowest power at the same time. Heterogeneous memory systems are effective in catering to a diverse range of memory access characteristics across workloads. A computing system with heterogeneous memory uses multiple memory modules, each of which are optimized to either provide high internal memory bandwidth or low memory access latency or low memory access power. The power-performance benefits of such a system is contingent upon a memory management policy that is aware of the access characteristics in applications.

In this chapter, we characterize the access patterns in applications at a fine granularity of memory objects that are allocated in the heap space. These memory objects exhibit vastly diverse memory access behavior, which are often significantly different from the application's aggregate memory access behavior. We, therefore, present our argument for object-level page allocation and introduce our memory management framework for heterogeneous memory systems.

5.1 Memory Access Characteristics of Heap Objects

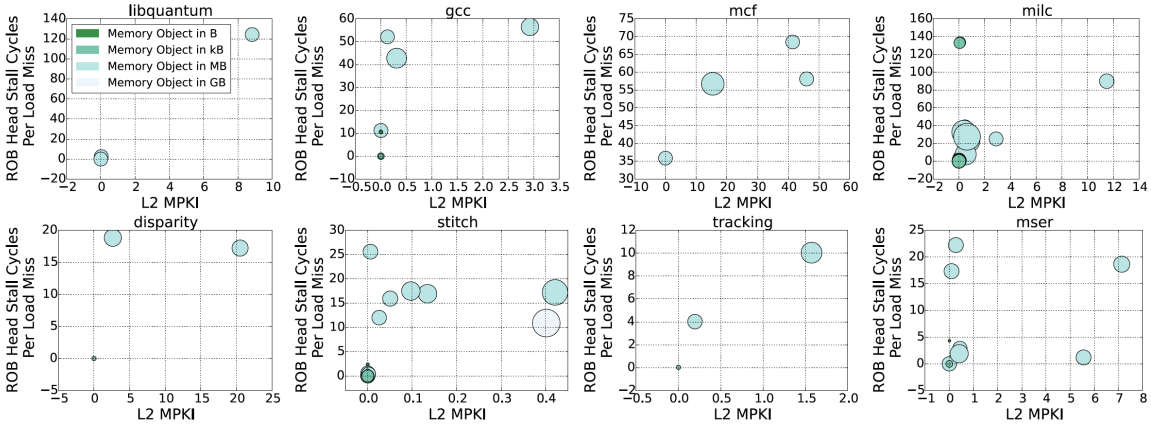


Figure 5·1: Access intensity and memory-level parallelism of heap memory objects for applications from SPEC CPU2006 and SDVBS benchmarks.

Section 2.4.2 in Chapter 2 presented the diverse memory access characteristics of different applications. An application-level page allocation policy profiles the access patterns of applications and feeds this information to runtime page allocation (Phadke and Narayanasamy, 2011). However, such a policy operates at a coarser-level granularity and fails to distinguish the diversity in memory access characteristics that exists within an application. Many applications are composed of a number of heap memory objects that are dynamically allocated at runtime. We study the memory access characteristics of these memory objects since they are often accessed periodically.

Figure 5·1 shows the distribution of memory objects within selected applications from SPEC CPU2006 (Henning, 2006) and SDVBS (Venkata et al., 2009) benchmark suites. The L2 MPKI and ROB head stall time specify the memory intensity and the MLP of objects, respectively. The size of a circle indicates the relative size of that object. Figure 5·1 shows a wide distribution across both of these metrics for memory objects within the same application. Therefore, an application-level allocation that uses the memory access characteristics of the application as a whole may not yield the

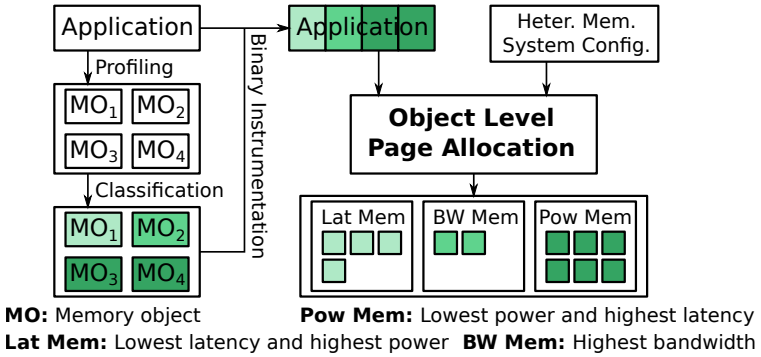


Figure 5·2: The flow of *MOCA*. The profiling stage uniquely names memory objects and characterizes the memory intensity and memory level parallelism. Classification stage uses this information to classify objects. At runtime, each memory object is allocated with pages from the best-fitting memory module based on object's type.

full benefits in a heterogeneous memory system. As an example, memory-intensive applications such as *milc* and *mser* have only a few memory objects with high L2 MPKI. In contrast to an application-level allocation, which would place all the objects into an RLDRAM module for these applications, a finer-level allocation could place the objects with low L2 MPKI into an LPDDR module, thereby improving the memory energy efficiency.

5.2 *MOCA*: Memory Object Classification and Allocation

In order to tap into the heterogeneity in memory access characteristics of objects within an application, we develop the *MOCA* framework. *MOCA* consists of a profiler that first uniquely names all the memory objects allocated in the heap address space. For each memory object in the application, *MOCA* collects metrics that characterize the memory intensity and the memory-level parallelism of that object. We then classify these memory objects as either latency-sensitive, bandwidth-sensitive or non-memory-intensive using predetermined thresholds on these metrics. We instrument the application binary, where each memory object is tagged with the corresponding

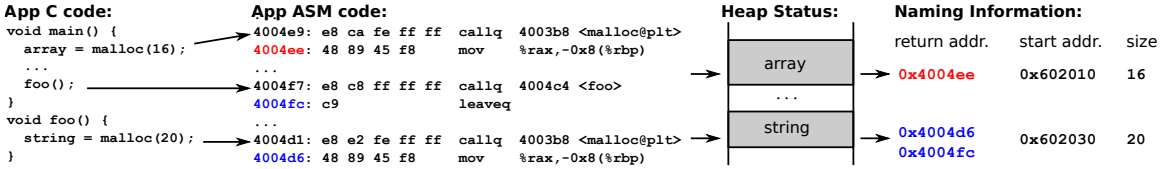


Figure 5·3: An example of memory object naming convention.

type. The page allocation algorithm in the OS is modified to track the type of each memory object and allocate the object to the corresponding memory module in the heterogeneous memory system. The profiling and classification of memory objects are conducted offline, and the page allocation of memory objects happens at runtime during application execution. Figure 5·2 shows the different steps involved in *MOCA*.

5.2.1 Memory Object Naming

The profiling stage uniquely names memory objects and collects metrics that characterize the memory access behavior of each object. To name memory objects, we use the return address of each dynamic memory allocation function (e.g., malloc, calloc, etc. in C) and record the virtual address of its caller function in the stack. These two addresses are unique to every object. Our naming convention for an example C code is shown in Figure 5·3. When the memory object “array" is initialized, we first record the virtual address of the caller function and the size of the object. The return address stack of “array" consists of only one return address in the *main* function. When the memory object “string" is initialized from inside the foo function, we again record the virtual address of its caller function and its size. However, in contrast to “array", the return address stack of “string" will consist of two return addresses, i.e., the return address of malloc in foo and the return address of foo in the *main* function.

5.2.2 Statistics Collection

Once we name all memory objects within an application, we utilize metrics that characterize their memory access behavior. We record the LLC MPKI for each object, which provides an indication of how frequently the memory is accessed. In addition, we collect average ROB head stall cycles per load miss (Mutlu et al., 2006) for each object. ROB head stall time is computed as the average cycles spent waiting at the head of the ROB for load misses and This has been used as an effective measure for memory level parallelism in prior works (Mutlu et al., 2006; Phadke and Narayanasamy, 2011).

5.2.3 Memory Object Classification

We use the collected statistics from profiling (memory objects, their LLC MPKI and ROB head stall times) to classify objects as being either latency-sensitive, bandwidth-sensitive, or neither. A memory object with high LLC MPKI implies increased main memory accesses. Such memory objects are classified as memory-intensive. Among the memory-intensive objects, the ones exhibiting low ROB head stall time imply that the memory latencies of objects are largely hidden in the latency of prior objects. Such objects, therefore, exhibit high MLP and are classified as bandwidth-sensitive memory objects. The remaining memory-intensive objects with high ROB stall time are more sensitive to the memory access latency. Such objects are classified as latency-sensitive memory objects. The objects with low LLC MPKI have minimal main memory accesses and are classified as non-memory-intensive objects. Such objects can be placed in low-power memory modules without affecting the system performance, thereby reducing memory power consumption.

Figure 5·4 depicts this classification where *Lat Mem* is a latency-optimized memory module, *BW Mem* is an bandwidth-optimized memory module and *Pow Mem* is a

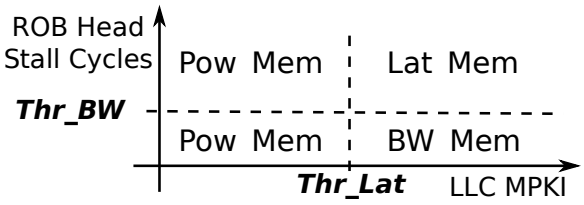


Figure 5·4: Classification of memory objects into different types based on latency and bandwidth thresholds.

power-optimized memory module. We classify objects with LLC MPKI greater than Thr_{Lat} as memory-intensive objects. Among these objects, the ones with ROB head stalls higher than Thr_{BW} are allocated to *Lat Mem* module. The memory-intensive objects with ROB head stalls lower than Thr_{BW} are allocated to *BW Mem* module. The rest of the objects with LLC MPKI lower than Thr_{Lat} are allocated to *Pow Mem* module.

5.2.4 Binary Instrumentation

The offline profiling and classification stages collectively identifies the type of each memory object in the application. We then instrument the memory object classification information into application binaries. We modify the standard memory allocation function (e.g. *malloc*, *calloc*, *realloc*) to enable an additional “type" field specifier. This field specifier can be 0 to represent a latency-sensitive object, 1 to represent a bandwidth-sensitive object, or 2 to represent a non-memory-intensive object. For each object, we update the “type" field specifier in the application binary.

5.2.5 Page Allocation

At runtime, *MOCA* uses the object-level information to perform page allocation. The heap memory address space in the virtual memory is divided into three regions as shown in Figure 5·5. Similarly, the physical address space is also divided into regions pertaining to the available memory modules in the system. The OS maintains the

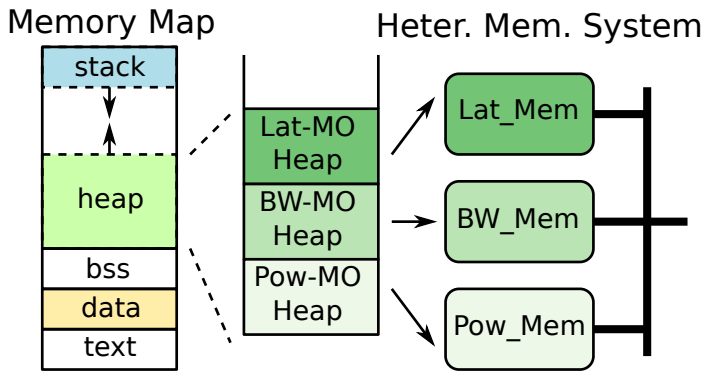


Figure 5·5: Mapping of virtual pages in the heap space to multiple memory modules in physical memory in *MOCA*.

mapping of virtual memory pages of a particular memory type (e.g., latency-sensitive) to the physical frames of the corresponding memory module (e.g., RLDRAM).

When a memory object is instantiated through the modified memory allocator (including the extra “type" field specifier), that object is allocated with virtual pages from the heap space based on its type. In the page translation process, based on the memory object’s virtual page number, the OS identifies the type of the memory object and maps a physical frame from the memory module corresponding to its type, as shown in Figure 5·5.

5.3 Implementation of *MOCA*

MOCA targets applications that run repeatedly on a servers and data centers. Therefore, in a real system, it MOCA uses representative training inputs for offline profiling and classification of memory objects. Once the binary of the application is instrumented using the memory characteristics of objects, consequent runs of the applications can be executed seamlessly. We implemented a simulation framework in Gem5 (Binkert et al., 2011) to conduct full-system architectural simulations. We use a Linux 2.6.32 disk image as the host operating system. We track memory objects allocated using the memory allocation library of C language (e.g., malloc()).

5.3.1 Offline Profiling and Classification

To implement the naming process, we modify the memory allocation functions to get the return addresses of each memory allocation function and its caller function using a built-in function $__builtin\_return\_address()$. We create a shared library of the modified memory allocation functions and preload this library while executing an application. We add a profiler flag to our compiler to maintain all the objects within an application in a LUT. This LUT contains all the information of every object (call stack, size, start address, LLC MPKI, ROB head stall cycles per load miss). *MOCA* uses the hardware performance counters of the processor to record the LLC misses and the ROB head stall cycles for each memory object. Each time an object is read/written to, if the ROB stalls for a memory read or if there is an LLC miss, we identify the accessed memory object (based on the requested address) and increment the corresponding counter for that memory object in the LUT. We also update the object's size as needed.

For classifying the memory objects as either latency-sensitive, bandwidth-sensitive or non-memory-intensive, we empirically set the Thr_{Lat} and Thr_{BW}. For our target heterogeneous system, we set Thr_{Lat} as 1 and Thr_{BW} as 20. Thr_{Lat} and Thr_{BW} need to be customized for a given system, as memory, cache, and core microarchitectural parameters significantly impact memory performance and energy efficiency.

5.3.2 Runtime Page Allocation

MOCA's runtime page allocation algorithm runs on top of the existing OS memory management. As noted earlier, we use the classification information of objects to instrument the application binary by specifying the "type" in the modified memory allocation function. [1] When the CPU issues a memory request, it goes to the L1 cache.

[1] Alternatively, one could instrument the application binary with object statistics (LLC MPKI and ROB stalls) and pass the Thr_Lat and Thr_BW thresholds to the OS.

In parallel, the CPU searches TLB for the physical page number of this memory request. On a hit, the TLB sends the physical page corresponding to the requested virtual page. Otherwise, there is a page fault, and the OS searches through page table to find the required virtual-physical page translation. The requested PTE is returned and inserted into TLB. The OS maintains the starting, ending, and the next available page number of each memory module in a heterogeneous memory system. The OS is also given the priorities of memory modules for different memory object types in case the most desired memory module is full (i.e., next best module if the ideal one is full).

5.3.3 Overheads of *MOCA*

The profiling and classification of memory objects are conducted offline and do not impact system performance at runtime. We measure the performance overhead of running our applications with profiling turned on, and observe only 0.59% slowdown on average. At runtime, the OS performs page allocation for memory objects only when they are instantiated. Therefore, the page allocation overhead is negligible in contrast to page migration policies that need to monitor runtime information.

5.4 Experimental Evaluation and Analysis

5.4.1 Simulation Framework

Computing System with Homogeneous/Heterogeneous Memory

We use the AMD Magny Cours processor (Conway et al., 2009) for demonstrating the benefits of *MOCA*. Table 5.1 shows the microarchitectural details of the AMD Magny Cours processor. We conduct simulations on both a single-core system and a 4-core multicore system. We consider a computing system with $2GB$ DDR3 memory module as the baseline for all simulations, since most high-end servers and data centers

Table 5.1: Microarchitectural details of AMD Magny Cours processor used in Gem5 simulations.

Execution Core	1GHz x86 ISA with out-of-order execution Fetch/Decode/Dispatch/Issue/Commit width 3, 84-entry ROB, 32-entry LQ, tournament branch predictor with 4K BTB entries
On-chip caches	64KB split L1 I and D cache, 2-way, 2 cycle , 64B line size, 4 MSHR Unified L2, 512KB, 16-way, 20 cycles, 64B line size, 20 MSHR
Memory Controller	Address mapping RoRaBaChCo, 4 channels, FR-FCFS scheduling

employ this memory module. In addition, we consider 3 computing systems with homogeneous memory, one with $2GB$ RLDRAM as the latency-optimized memory, one with $2GB$ HBM as the bandwidth-optimized memory and one with $2GB$ LPDDR2 as the power-optimized memory. We denote system with DDR3 memory as *Homogen-DDR3*, system with LPDDR2 memory as *Homogen-LPDDR2*, system with RLDRAM memory as *Homogen-RDLRAM* and system with HBM memory as *Homogen-HBM*.

Our target computing system with heterogeneous memory consists of four memory channels and each channel is connected to a type of memory module. We model this memory system to consist of a $768MB$ HBM module, a $256MB$ RLDRAM module, and two $512MB$ LPDDR2 modules. We use a dedicated memory controller for each memory channel as the device timing parameters differ for different memory modules. We compare our proposed *MOCA*, which is an object-level page allocation in heterogeneous memory system, with an application-level allocation (Phadke and Narayanasamy, 2011), where all the memory objects in one application are allocated to that application's best-fit memory module. We denote the heterogeneous memory system with application-level allocation as *Heter-App*.

Performance and Power Simulation

We conduct full-system simulations in Gem5 (Binkert et al., 2011). We model the microarchitectural parameters of AMD Magny Cours processor in Gem5. We use a Linux 2.6.32 disk image as the host operating system. For each application, we run the applications for 100 million instructions at each simpoint (Hamerly et al., 2005) to collect memory object statistics for each application.

We feed the Gem5 output statistics to McPAT (Li et al., 2009) for core and cache power calculation. We calibrate the runtime dynamic core power values using measurements collected on the AMD Magny Cours processor (Kumar et al., 2003). We model performance characteristics of our memory system designs in Gem5 and use MICRON's DRAM power calculators for DDR3 (MICRON, 2011), RLDRAM (MICRON, 2016) and LPDDR2 (MICRON, 2013) to calculate memory power consumption. This calculator takes in memory read and write access rates as inputs and provides detailed DRAM power traces for each banks. For HBM, we scale down the DDR3 precharge and power-down current (Li et al., 2016a), and then estimate memory power from SDRAM power calculator (MICRON, 2011). We assume that the I/O power and the on-chip bus power are negligible compared to total chip power.

Workloads

We run selected C-based applications from SPEC CPU2006 (Henning, 2006) and SD-VBS (Venkata et al., 2009). For SPEC benchmarks, we conduct profiling using the training input sets and perform allocation on reference input sets. In case of SD-VBS benchmarks, we select two different images from MIT-Adobe fivek dataset (Bychkovsky et al., 2011) for profiling and allocation. We classify the applications as a whole to be either latency-sensitive (L), bandwidth-sensitive (B) or non-memory-intensive (N). To run workloads on a multicore system, we create multi-program workload sets consisting of a diverse mix of these applications. As an example, *2L1B1N*

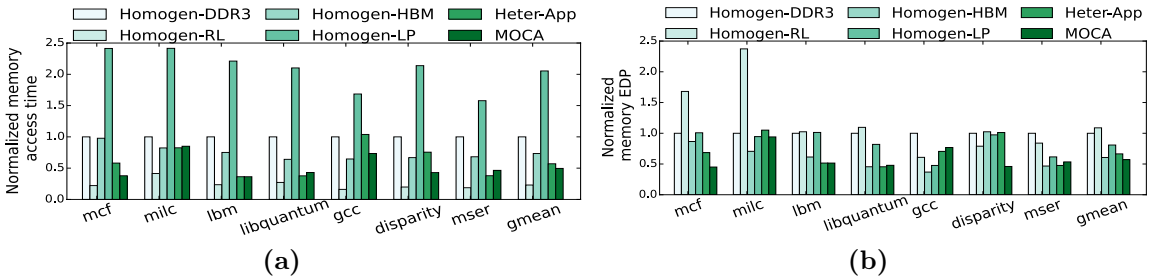


Figure 5·6: (a) Memory performance in access time, and (b) memory energy efficiency in EDP of homogeneous and heterogeneous memory systems for single-program workloads

represent a workload set with two latency-sensitive applications, one bandwidth-sensitive application and one non-memory-intensive application.

5.4.2 Performance and Energy Benefits for Single-core Systems

We demonstrate the benefits of heterogeneous memory over homogeneous memory systems, and the benefits of object-level page allocation with *MOCA* over application-level page allocation in heterogeneous memory systems. Figure 5·6a and Figure 5·6b shows the memory performance and energy efficiency in EDP, respectively, across different memory configurations for a single-core computing system. The memory access times and memory EDP are normalized to that of an homogeneous memory system with on DDR3 memory module, i.e. *Homogen-DDR3*.

For a single-core system, *MOCA* reduces the memory access time by 51% and the memory EDP by 43% over *Homogen-DDR3*. *Homogen-RL* unsurprisingly has the lowest memory access time whilst the worst energy efficiency. On the other hand, *Homogen-LP* has the worst performance among all memory systems, but due to its low power cost, it still has better EDP than *Homogen-RL* and Homogen-DDR3. *MOCA* achieves the best energy efficiency among all experimented memory systems and stays closest to *Homogen-RL*'s performance.

Compared to *Heter-App*, *MOCA* outperforms in memory performance by 14%

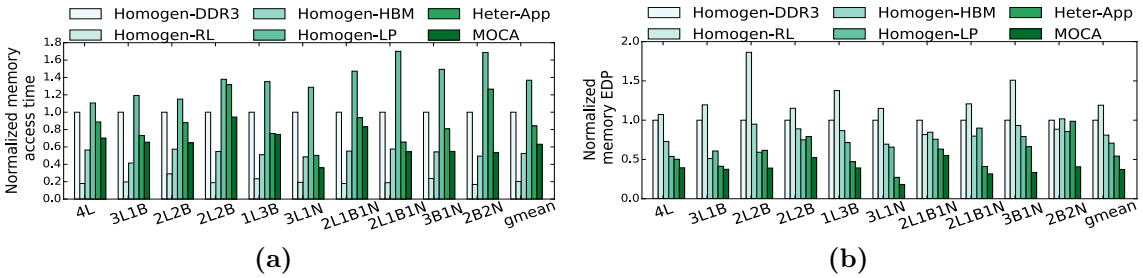


Figure 5·7: (a) Memory performance in access time, and (b) memory energy efficiency in EDP of homogeneous and heterogeneous memory systems for multi-program workloads

and in energy efficiency by 15% for single-core systems. In particular, *MOCA* provides more benefits in performance and energy efficiency for latency-sensitive applications, such as *disparity*. *disparity* has two major memory objects, one with a high L2MPKI and the other with a relatively low L2MPKI. *Heter-App* first allocates the lower-L2MPKI object in RLDRAM module since it is the first one identified during runtime. Since RLDRAM module capacity is used up by this object, the higher-L2MPKI object is allocated in HBM module. In contrast, *MOCA* is aware of both objects' characteristics, and thus, allocates the higher-L2MPKI object in RLDRAM and the lower-L2MPKI one in HBM, which improves the memory performance and reduces the memory EDP. Therefore, object-level page allocation in *MOCA* is able to unearth more of heterogeneous memory systems' potential than an application-level page allocation.

5.4.3 Performance and Energy Benefits for Multicore Systems

Figure 5·7a and Figure 5·7b shows the memory performance and energy efficiency in EDP, respectively, for a multicore system. The memory EDP with *MOCA* is 63% higher than *Homogen-DDR3* and 40% higher than *Homogen-LP*, which makes *MOCA* the most energy-efficient one among all tested memory systems. In addition, *MOCA* reduces the memory access time by 26% and the memory EDP by 33% over *Heter-App*.

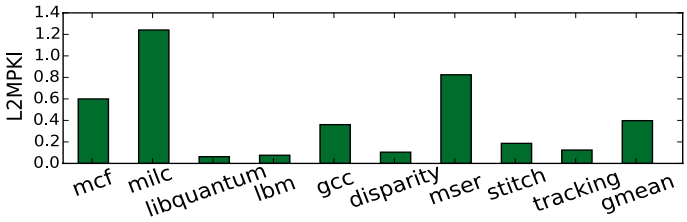


Figure 5·8: L2 MPKI of stack and code segment for all applications

MOCA prioritizes the high-L2MPKI objects to RLDRAM and the high-MLP objects to HBM, thereby reducing overall memory access time. In addition, *MOCA* also places the non-memory-intensive objects to LPDDR modules, thereby reducing the memory power consumption significantly. Thus, we see energy efficiency improvement from *MOCA* over *Heter-App*, which tries to place all objects in RLDRAM module.

5.4.4 Classifying Stack Data and Code Segment

In *MOCA*, we mainly profile and allocate memory objects allocated in the heap space. In addition, there are also memory accesses to the code segment as well as the stack space. However, the memory access intensity of these segments is considerably lower than that of the heap objects. Figure 5·8 shows the L2MPKI for stack and code segments of the target applications. These segments exhibit lower L2MPKI values due to the higher locality of code segment and lower data size of the stack segment. Therefore, we allocate pages from LPDDR module for these segments in *MOCA*.

5.5 Chapter Summary

Heterogeneous memory systems are very effective in catering to a wide diversity of workloads with varied memory characteristics. Although, such systems need a system memory management policies to leverage their full potential in delivering high system energy efficiency. In contrast to coarser-level page allocation policies in prior work, this chapter points out that memory objects within an application exhibit substantial

diversity in memory characteristics. Our proposed framework, *MOCA* exploits this observation to design an intelligent data placement, which profiles an application and places each object in a memory module that best suits that object's memory access behavior.

This chapter demonstrates that heterogeneous memory systems with *MOCA* outperform current homogeneous systems composed of DDR3 modules with 63% improved energy efficiency and 30% higher performance. *MOCA* also enables an efficient framework for memory management by providing 26% higher performance and 33% improved energy efficiency compared to an application-level page allocation.

Chapter 6

Conclusions and Future Directions

As we usher in an era of extreme data-centric computing, the primary focus has shifted towards data movement and data access in current and future manycore systems. With silicon-photonic links and optical phase change materials undergoing major breakthroughs in device research, they present a fascinating platform for designing energy-efficient manycore systems. This thesis has presented architectural designs and system management policies for chip-scale networks and main memory using silicon-photonics technology. This chapter summarizes the important findings of the thesis and discusses open problems for future research directions.

6.1 Summary of Thesis Contributions

This thesis addresses the energy efficiency concerns in data movement and data access on two major fronts: developing system management policies for power-efficient utilization of silicon-photonic links and designing an optically-controlled memory system that is interfaced using silicon-photonic links.

Supporting the high-*Tbps* demands of data-centric applications on manycore chips requires the design of dense WDM silicon-photonic links with increased optical channels. The consequent increase in the photonic power impacts the energy-per-bit budget of chip-scale networks. We, therefore, postulate that it is sufficient to activate the minimum number of optical channels that satisfies the application bandwidth requirements. To this end, we model the different components of the power consump-

tion in the silicon-photonic links, i.e., laser sources, electrical circuitry in the E-O-E conversion and thermal tuning of MRRs. We identify that thermal tuning is a major contributor to the photonic power, and leverage the analog thermal control loop at device-level to enable MRR remapping at runtime. Our cross-layer simulation framework accounts for the MRR sensitivities to PV and TV, architectural parameters of the 2.5D manycore system and the system-level resource utilization, power and thermal profile during an application execution. We propose a bandwidth allocation policy, *SO-WAVES* (Narayan et al., 2019), that is effective in limiting the photonic power by 38% compared to activating all the optical channels with only 5% loss in performance.

A limiting factor of our *SO-WAVES* policy stems from the fact that it only accounts for the averaged bandwidth needs of an application. An application has some phases with high bandwidth utilization and some phases with low bandwidth utilization. Since *SO-WAVES* activates optical channels based on averaged bandwidth needs, this may under or over provision the dynamic bandwidth needs during an application execution. To address this shortcoming, we propose a dynamic bandwidth allocation policy called *PROWAVES* (Narayan et al., 2020b). *PROWAVES* consists of a time-series forecasting that uses an ARIMA model to predict the bandwidth requirement for the next phase and proactively activate the optical channels. We observe that *PROWAVES* consumes 16.4% lower photonic power than *SO-WAVES* for the same performance loss threshold, i.e., 5%. We also compare *PROWAVES* with a prior bandwidth allocation technique that uses ridge regression model for bandwidth prediction, RR-PS (Van Winkle et al., 2018). Owing to model of the device-level thermal control loop, *PROWAVES* results in $26.3W$ lower thermal tuning power than RR-PS for the same performance loss threshold, i.e., 5%.

We evaluate a diverse range of workloads from standard HPC and graph benchmarks that form the basis of a majority of data-centric applications across several domains. We particularly investigate graph benchmarks as they exhibit random memory accesses, resulting in increased network traffic. We demonstrate that silicon-photonic links, owing to their high bandwidth-density, are able to meet the high bandwidth demands of graph applications (Narayan et al., 2020a). Moreover, we performed several architectural sensitivity analyses using silicon-photonic links for graph applications that present promising opportunities for redesigning future systems.

On top of our system-level bandwidth allocation policies, we implemented a software-level instrumentation that reduces the network traffic in silicon-photonic links during application execution (Narayan et al., 2020a). Using this application-instrumentation approach, we observe a reduced number of inter-chiplet transferred packets, thus reducing the application's bandwidth requirements. As a result, we are able to save 35.13% higher photonic power using instrumentation-assisted *SO-WAVES* compared to *SO-WAVES* without instrumentation.

The performance of computing systems, despite using silicon-photonic links as chip-scale networks, are still bottlenecked due to constrained bandwidth and long access latency of the main memory. Our goal is to design a main memory system that can be directly accessed by optical signals in the silicon-photonic links. This thesis proposes *COSMOS*, a main memory system that combines optically-controlled phase change materials with silicon-photonic links. The increased bit density per cell and the high-bandwidth-density access of memory cells using silicon-photonic links in *COSMOS* deliver a high memory throughput for read and write. *COSMOS* uses a novel read and write access protocol that is tailored to the properties of OPCM cells and the optical constraints of silicon-photonic links. Moreover, we design an E-O-E control unit to enable interfacing *COSMOS* with current processors. The E-

O-E control unit is responsible for mapping the DRAM protocol commands, data and addresses to OPCM-specific optical signals.

Evaluation of a 2.5D computing system with *COSMOS* demonstrates 2.15× higher write throughput and 2.09× higher read throughput compared to an equivalent computing system with EPCM. This increased memory throughput in *COSMOS* reduces the memory latency by 33%. Overall, when compared to EPCM, *COSMOS* has 2.14× better performance, 1.24× lower read energy-per-bit, and 4.06× lower write energy-per-bit for graph and HPC workloads. *COSMOS* provides a scalable and non-volatile alternative to DDR4 DRAM memory, with 5.6% higher performance and similar energy-per-bit for read and write accesses. With DRAM technology undergoing critical scaling challenges, *COSMOS* presents the first non-volatile main memory system with improved scalability, increased bit density, high area efficiency and comparable performance and energy as DRAM. Our promising initial demonstration of *COSMOS* architecture can open the doors for interesting architectural, design and system-level directions that enable the feasibility of OPCM-based main memory in future manycore systems.

This thesis finally addresses the shortcomings of homogeneous memory systems in manycore chips in the era of workloads with diverse memory characteristics. Heterogeneous memory systems, with their potential to cater to a diverse range of workloads with varying memory characteristics, still need a systematic memory management policy. We present *MOCA*, a framework for page allocation in heterogeneous memory systems at the granularity of heap memory objects. *MOCA* first profiles an application, collects statistics of different memory objects to classify them into different categories based their memory characteristics, and finally allocates them at runtime to the best-fit memory module. Our evaluation of *MOCA* provides 63% energy improvement compared to a homogeneous DDR3 memory system, and 33% energy improvement

compared to an application-level allocation in heterogeneous memory system.

6.2 Future Research Directions

The architectural designs and system-level management presented in this thesis open up interesting research directions in designing energy-efficient chip-scale networks and memory systems.

6.2.1 Designing Efficient Silicon-Photonic Links

Bandwidth Allocation for Heterogeneous 2.5D Systems

This thesis presents *WAVES* as a system-level bandwidth allocation policy to address the high photonic power overhead at increased network bandwidth. Our evaluations demonstrated the power benefits of *SO-WAVES* and *PROWAVES* on a 96-core homogeneous 2.5D system, *POPSTAR*. A homogeneous 2.5D system consists of the same compute chiplets integrated on an interposer. In such a system, the inter-chiplet network traffic remains evenly distributed when a multi-threaded application is executed with equal threads/chiplet. In contrast, a heterogeneous 2.5D system may experience highly uneven network traffic distribution, as chiplets differ in their compute ability. Such uneven network distribution has been shown in heterogeneous manycore systems consisting of CPU and GPU chiplets (Mirhosseini et al., 2017; Zhan et al., 2016). Our proposed *WAVES* and *PROWAVES* policy activates minimum number of optical channels for the entire system. As a result, some of the inter-chiplet network traffic may be under-provisioned than the required bandwidth and yield lower performance. To address this limitation, studying the network traffic patterns in heterogeneous 2.5D systems is an interesting research direction.

- In a heterogeneous 2.5D system consisting of CPU chiplets, GPU or accelerator chiplets and memory chiplets, the network traffic may be higher to and from the

GPU chiplet due to its increased capability for parallel processing. Therefore, a bandwidth allocation policy in such a system needs to account for uneven network traffic between certain set of chiplets. Such a policy should also ensure fairness to the network packets transferred so that none of the chiplets starve for a long time.

- A heterogeneous 2.5D system also presents interesting design considerations in chiplet placement and routing of silicon-photonic links to ensure thermal reliability. Chiplets with higher compute activity result in increased temperatures, demanding higher need for MRR thermal management. The placement of compute chiplets, memory chiplets and the TxRx chiplets along with the routing of the silicon-photonic links can be formulated as an optimization problem constrained by the thermal thresholds, the bandwidth required for the different chiplets and the power budget.

Software Frameworks for Silicon-Photonic Links

This thesis presents the efficacy of bandwidth selection using application instrumentation on PageRank algorithm. The benefits of such an approach opens up interesting opportunities in designing more generalized software frameworks that can assist bandwidth allocation policies. Such a framework can be implemented in several ways.

- One potential approach is to enable programmers with higher capability to design software at the function-level to minimize the data or cache coherency traffic in the chip-scale network. Similar to our design, the programmer can instrument the application using privileged instructions. Another powerful design can enable the programmer to define a specific flow of the application task graph, which contains information about data dependencies. The goal of these designs is to embed information in the source code to reduce network traffic. A

dedicated core on chip can monitor the hardware performance counters to record the network metrics, which can guide the system-level management policies.

- Another design approach is to enable the compilers to provide network information to system management policies. Such an approach reduces the burden on programmers to design optimized codes targeted towards network power optimization. The compiler can use the code's intermediate representation to implement code-improving transformations to generate the object or machine code. These transformations can leverage the data dependencies in the source code and potential data coherency to minimize the chip-scale network traffic in the silicon-photonic links.

6.2.2 Architectural Opportunities with *COSMOS*

This thesis proposes *COSMOS* as the first novel memory architecture using OPCM cells. There are promising avenues to explore in the design architecture and the software stack to maximize the potential of OPCM arrays.

OPCM-aware Scheduling and Application Mapping in *COSMOS*

The primary goal of *COSMOS* architecture is to ensure its compatibility with current off-the-shelf processors. The E-O-E control unit is designed to interface with memory controllers and map the standard DRAM protocol commands to OPCM-specific optical signals. The *COSMOS* system, therefore, uses the same memory scheduling policies, virtual-physical address mapping, and error detection and correction mechanisms that are used in current memory controllers. It will be interesting to investigate whether the above mechanisms are indeed suitable for *COSMOS*.

- Most current memory controllers use FR-FCFS policy for memory scheduling (Valsan and Yun, 2015; Martinez and Ipek, 2009). FR-FCFS policy is conventionally designed to maximize the row-buffer locality in DRAM systems by

prioritizing memory commands that hit in the row buffer. However, since *COSMOS* does not use a row-buffer in its design and directly accesses the cells in the OPCM array, such a policy may result in sub-optimal performance. Therefore, it is essential to identify the key bottlenecks with FR-FCFS policy in *COSMOS*, determine the extent of its impact and implement a memory scheduling algorithm that maximizes the specific address mapping used in *COSMOS*.

- *COSMOS* relies on the E-O-E control unit to map the memory controller commands to optical signals. Therefore, *COSMOS* still uses conventional electrical links for memory controller to E-O-E control unit communication. Despite silicon-photonic links + OPCM array delivering high throughput, the memory controller can develop as a bottleneck for applications with high read/write rates. The buffering of read/write operations may potentially stall the processor from issuing memory instructions even though the OPCM array is capable of delivering the required throughput. Moreover, the memory controller still issues commands for Precharge, Activate and Refresh, which are required for DRAM systems, but are redudant for *COSMOS*. Therefore, it would be beneficial to design an OPCM-specific memory controller in the processor. Such a design also shifts the different sub-units in the E-O-E control unit inside the processor or the memory controller as a separate chiplet.

- Application mapping to OPCM array is another interesting aspect to explore as a potential direction. The high throughput obtained using *COSMOS* is contingent on data accesses that are independent, which allows the E-O-E control unit to pipeline the read and write accesses. However, in applications with dependent accesses (e.g., pointer chasing, iterative algorithms), *COSMOS* can yield suboptimal performance due to inefficient data mapping to OPCM cells. Therefore, it is essential to consider data mapping policies in cases of graph

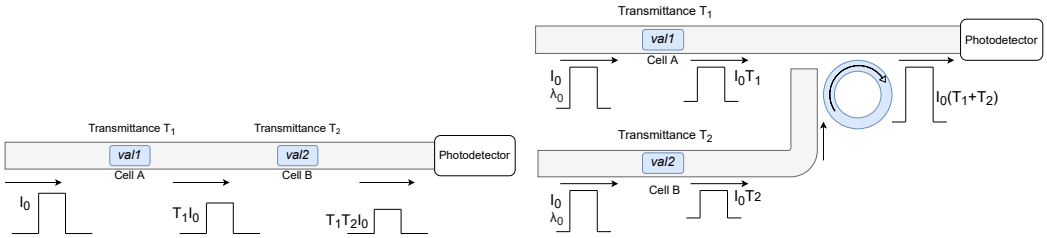


Figure 6·1: (a) Multiplication and (b) Addition operation of two values stored in two OPCM cells in *COSMOS*.

workloads and data analytics to utilize the internal OPCM organization more efficiently.

Processing-in-Memory in *COSMOS*

The current computing systems are mostly *processor-centric*, i.e., data is moved from memory to processor for computation and then written back to memory. With the data explosion in today's workloads, most of this data movement leads to unnecessary resource wastage and increases computation time and energy. Processing-in-Memory (PIM) architectures have emerged as a *memory-centric* design that provide support for computational elements or mechanisms inside the memory chips. These PIM designs are based on analog or digital principles with the software stack providing new primitives to support certain operations. PIM architectures have been extensively studied in DRAM systems (Seshadri et al., 2013; Chang et al., 2016a; He et al., 2020; Seshadri et al., 2017) as well as NVM systems (Li et al., 2016b; Angizi et al., 2017; Angizi et al., 2018).

Owing to the higher *bits/cell* capability and high-throughput access, the OPCM array in *COSMOS* provides an excellent opportunity for designing PIM architectures. The associative property of optical signals enable us to perform addition and multiplication operation of two values stored in two separate OPCM cells. Figures 6·1a and 6·1b illustrate the multiplication and addition operation of two values stored in

two different OPCM cells. During multiplication operation, an optical signal with intensity I_0 is passed through the two OPCM cells that hold the operands for multiplication. The output optical signal has an intensity of $T_1.T_2.I_0$, where T_1 and T_2 are the transmittance of the two OPCM cells based on the stored data. For addition operation, two optical signals, both with intensity I_0, are passed in parallel through the two OPCM cells. The output intensities from the cells, $T_1.I_0$ and $T_2.I_0$ are aggregated into the same waveguide. As a result, the final intensity of the optical signal is $I_0.(T_1 + T_2)$.

Due to the high MLC capacity of OPCM cells, we can store n bits per cell, which enables us to directly perform addition and multiplication of $n - bit$ operands. Currently, OPCM cells store 4 bits, with future projections forecasting up to 8 bits per cell (Li et al., 2019). Most machine learning models operate on int4 and in8 operands (Martinez and Ipek, 2009; Fu et al., 2016; Zhu et al., 2020), which makes OPCM-based array with PIM capabilities particularly attractive for machine learning applications. Unlike DRAM, which can take up to 8+ cycles for a single multiplication or addition of int8 operands, and EPCM, which can take up to 4+ cycles and 3× latency to write back the result, OPCM-based PIM systems can perform these operations in a single cycle. Using dense WDM silicon-photonic links, we can also perform multiple such independent operations in parallel, thereby increasing the compute throughput substantially compared to DRAM-based PIM designs. Furthermore, neural network inference models are primarily composed of matrix-vector multiplication operations. Therefore, a memory system capable of performing high-throughput addition and multiplication operations can perform matrix-vector multiplication operations at much lower J/ops. Feldman *et al.* demonstrated a photonic hardware accelerator that is capable of operating at tera-MAC operations/sec using optically-controlled GST elements for data-heavy AI applications (Feldmann et al., 2021). All

these above factors open up interesting future research directions as follows:

- Designing a PIM system using OPCM-based arrays requires enabling architectural support and defining PIM primitives in the programming model. These primitives can either be defined at instruction-level by embedding PIM-specific instructions in the ISA or at the function-level using a pragma-based or directive-based approach to offload a kernel/function on to PIM.

- The PIM paradigm in OPCM-based arrays would call for a redesign of the E-O-E control unit as well. In addition to address and data mapping to optical signals, the E-O-E control unit would need to decode the PIM instructions and generate specific intensity optical signals corresponding to PIM operations. An interesting direction would be to explore optimization techniques on operation scheduling and pipelined execution by leveraging the high bandwidth density offered by silicon-photonic links.

- For data-centric applications such as graph algorithms and privacy-preserving workloads, the data operands are often longer than 8 bits, unlike machine learning applications. Such applications would, therefore, require novel memory organization and compute mechanisms for arithmetic operations, where operands are stored across multiple OPCM cells.

6.2.3 Memory Management in Heterogeneous Memory Systems

Designing heterogeneous memory systems with *COSMOS*

This thesis presents *COSMOS* as a high-throughput main memory candidate that is particularly beneficial in servicing high orders of parallel reads and writes. However, latency-sensitive workloads with highly dependent reads and writes will encounter the long write latency in the OPCM array. Similarly, workloads with significantly high

write rate will result in a faster wearout of OPCM cells in *COSMOS*. Therefore, an interesting direction is to design a heterogeneous memory system with *COSMOS* and DRAM systems that utilizes the non-volatily, high bit density and high bandwidth advantages of *COSMOS* and the latency and endurance benefits of DRAM.

- We can employ two strategies in designing a heterogeneous memory system with DRAM and *COSMOS*. One strategy can employ DRAM as an upper layer cache memory of *COSMOS*. Similar to the L1 and L2 caches, DRAM cache here is hidden from the OS and can be accessed only using hardware implementation. The DRAM cache can be implemented as a writeback cache to minimize the write frequency to *COSMOS*.

- Another strategy is to design DRAM and *COSMOS* at the same memory hierarchy level. The physical address is spread across both the DRAM and *COSMOS*, and the OS needs to manage the page translation and allocation. In such a scenario, an updated framework of *MOCA* that monitors the write rate and the required throughput for a memory object can be investigated.

Dynamic memory object migration in heterogeneous memory systems

In our proposed *MOCA* framework, we address the diversity in memory characteristics that exists in memory objects within an application. We profiled their characteristics and allocated them at runtime to the best-fit memory module. The runtime page allocation, therefore, was performed at the object initialization. However, objects may exhibit changing memory characteristics during an application execution. For example, an application may initialize a memory object in parallel, which makes the object bandwidth-intensive. During an initial phase of execution, the object may be accessed in parallel, which retains the object's bandwidth-intensive property. But, during a later phase of execution, the object may encounter dependent accesses, which

makes it latency-sensitive. As another example, a large object may be initialized and accessed in parallel during an early phase of the application execution. The object may not be accessed for the rest of the application, which makes the object non-memory-intensive. Such behaviors are common in graph and machine learning applications. Therefore, an object migration policy between memory modules needs to be explored in the *MOCA* framework.

- A potential approach to implement a page migration policy for memory objects can be a reactive one. We can monitor the object's memory characteristics during the application execution and when the characteristics drastically changes based on a predetermined threshold, the object's pages can be migrated to the target memory module. It is essential to account the performance and energy overhead of page migration in this approach.

- Another approach to implement page migration for memory objects is to use a proactive policy. We can design a time-forecasting model to learn the object's memory characteristics over time. Using this model, the object's pages can be proactively migrated to the destination memory module. With this approach, it is also possible to hide the page migration latency of an object during the application execution.

References

Abellán, J. L., Coskun, A. K., Gu, A., Jin, W., Joshi, A., Kahng, A. B., Klamkin, J., Morales, C., Recchio, J., Srinivas, V., and Zhang, T. (2017). Adaptive tuning of photonic devices in a photonic NoC through dynamic workload allocation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(5):801–814.

Ahn, J., Hong, S., Yoo, S., Mutlu, O., and Choi, K. (2015). A scalable processing-in-memory accelerator for parallel graph processing. In *Proceedings of International Symposium on Computer Architecture, Portland, OR, USA*, pages 105–117.

Akaike, H. (1969). Fitting autoregressive models for prediction. *Annals of the institute of Statistical Mathematics*, 21(1):243–247.

AMD (2019). AMD RDNA GPU Architecture. `https://www.amd.com/en/technologies/rdna`.

Angizi, S., He, Z., and Fan, D. (2018). PIMA-logic: A novel processing-in-memory architecture for highly flexible and energy-efficient logic computation. In *Proceedings of Design Automation Conference, San Francisco, CA, USA*, pages 1–6.

Angizi, S., He, Z., Parveen, F., and Fan, D. (2017). RIMPA: A new reconfigurable dual-mode in-memory processing architecture with spin hall effect-driven domain wall motion device. In *Proceedings of International Symposium on VLSI, Bochum, Germany*, pages 45–50. IEEE.

Arjomand, M., Kandemir, M. T., Sivasubramaniam, A., and Das, C. R. (2016). Boosting access parallelism to PCM-based main memory. In *Proceedings of International Symposium on Computer Architecture, Seoul, South Korea*, pages 695–706.

Avissar, O., Barua, R., and Stewart, D. (2001). Heterogeneous memory management for embedded systems. In *Proceedings of International. Conference on Compilers, Architecture, and Synthesis for Embedded Systems, Atlanta, GA, USA*, pages 34–43.

Bahadori, M., Gazman, A., Janosik, N., Rumley, S., Zhu, Z., Polster, R., Cheng, Q., and Bergman, K. (2017a). Thermal rectification of integrated microheaters for microring resonators in silicon photonics platform. *Journal of Lightwave Technology*, 36(3):773–788.

Bahadori, M., Polster, R., Rumley, S., Thonnart, Y., Gonzalez-Jimenez, J.-L., and Bergman, K. (2016). Energy-bandwidth design exploration of silicon photonic interconnects in 65nm CMOS. In *Proceedings of Optical Interconnects Conference, San Diego, CA, USA*, pages 2–3.

Bahadori, M., Rumley, S., Polster, R., Gazman, A., Traverso, M., Webster, M., Patel, K., and Bergman, K. (2017b). Energy-performance optimized design of silicon photonic interconnection networks for high-performance computing. In *Proceedings of Design, Automation & Test in Europe Conference & Exhibition, Lausanne, Switzerland*, pages 326–331. IEEE.

Bailey, D., Harris, T., Saphir, W., Van Der Wijngaart, R., Woo, A., and Yarrow, M. (1991). The NAS parallel benchmarks. *International Journal of Supercomputing Applications*, 5(3):63–73.

Batten, C., Joshi, A., Orcutt, J., Khilo, A., Moss, B., Holzwarth, C. W., Popovic, M. A., Li, H., Smith, H. I., Hoyt, J. L., Kartner, F., Ram, R. J., Stojanović, V., and Asanović, K. (2008). Building many-core processor-to-DRAM networks with monolithic CMOS silicon photonics. In *Proceedings of Symposium on High Performance Interconnects, Stanford, CA, USA*, pages 21–30.

Beamer, S., Asanović, K., and Patterson, D. (2015). The gap benchmark suite. *arXiv preprint arXiv:1508.03619*.

Beamer, S., Sun, C., Kwon, Y.-j., Joshi, A., Batten, C., Stojanovic, V., and Asanovi, K. (2009). Re-architecting DRAM with monolithically integrated silicon photonics. In *Proceedings of International Symposium on Computer Architecture, Saint-Malo, France*, pages 129–140.

Bedeschi, F., Fackenthal, R., Resta, C., Donze, E. M., Jagasivamani, M., Buda, E. C., Pellizzer, F., Chow, D. W., Cabrini, A., Calvi, G. M. A., Faravelli, R., Fantini, A., Torelli, G., Mills, D., Gastaldi, R., and Casagrande, G. (2008). A multi-level-cell bipolar-selected phase-change memory. In *Proceedings of International Solid-State Circuits Conference, San Francisco, CA*, pages 428–625.

Beigi, M. V. and Memik, G. (2016). Therma: Thermal-aware run-time thread migration for nanophotonic interconnects. In *Proceedings of International Symposium on Low Power Electronics and Design, San Francisco, CA, USA*, pages 230–235.

Bergal, A. (2019). Trends in DRAM price per gigabyte. `https://aiimpacts.org/trends-in-dram-price-per-gigabyte/#easy-footnote-bottom-8-2408`.

Bergman, K. (2018). Empowering flexible and scalable high performance architectures with embedded photonics. In *Proceedings of International Symposium of Parallel and Distributed Processing, Vancouver, BC, Canada*, page 378.

Bhattacharjee, D., Devadoss, R., and Chattopadhyay, A. (2017). ReVAMP: ReRAM based VLIW architecture for in-memory computing. In *Proceedings of Design, Automation & Test in Europe Conference & Exhibition, Lausanne, Switzerland*, pages 782–787. IEEE.

Bienia, C., Kumar, S., Singh, J. P., and Li, K. (2008). The PARSEC benchmark suite: Characterization and architectural implications. In *Proceedings of Parallel architectures and compilation techniques, Toronto, Canada*.

Binkert, N., Beckmann, B., Black, G., Reinhardt, S. K., Saidi, A., Basu, A., Hestness, J., Hower, D. R., Krishna, T., Sardashti, S., Sen, R., Sewell, K., Shoaib, M., Vaish, N., Hill, M. D., and Wood, D. A. (2011). The gem5 simulator. *ACM SIGARCH computer architecture news*, 39(2):1–7.

Bogaerts, W., De Heyn, P., Van Vaerenbergh, T., De Vos, K., Kumar Selvaraja, S., Claes, T., Dumon, P., Bienstman, P., Van Thourhout, D., and Baets, R. (2012). Silicon microring resonators. *Laser & Photonics Reviews*, 6(1):47–73.

Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: Forecasting and control*. John Wiley & Sons.

Bradford, D., Chinthamani, S., Corbal, J., Hassan, A., Janik, K., and Ali, N. (2017). Knights mill: New intel processor for machine learning. In *Proceedings of Hot Chips Symposium, Cupertino, CA, USA*, volume 29.

Burr, G., Breitwisch, M., Franceschini, M., Garetto, D., Gopalakrishnan, K., Jackson, B., Kurdi, B., Lam, C., Lastras, L., Padilla, A., and Rajendran, B. (2010). Phase change memory technology. *Journal of Vacuum Science & Technology B, Nanotechnology and Microelectronics: Materials, Processing, Measurement, and Phenomena*, 28(2):223–262.

Bychkovsky, V., Paris, S., Chan, E., and Durand, F. (2011). Learning photographic global tonal adjustment with a database of input/output image pairs. In *Proceedings of Computer Vision and Pattern Recognition, Providence, RI, USA*, pages 97–104.

Campbell, D., Bader, D., Brandt, S., Cook, D., Gokhale, M., Hornung, R., Keasler, J., LeBlanc, P., Marin, G., and Mulvaney, B. (2012). Ubiquitous high performance computing: Challenge problems specification. *Georgia Technical Research Institute, Atlanta, GA, USA, Tech. Rep. HR0011-10-C-0145*.

Cardenas, J., Poitras, C. B., Robinson, J. T., Preston, K., Chen, L., and Lipson, M. (2009). Low loss etchless silicon photonic waveguides. *Optics express*, 17(6):4752–4757.

Carlson, T. E., Heirman, W., and Eeckhout, L. (2011). Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation. In *Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis, Seatle, WA, USA*, page 52.

Chang, K. K., Nair, P. J., Lee, D., Ghose, S., Qureshi, M. K., and Mutlu, O. (2016a). Low-cost inter-linked subarrays (LISA): Enabling fast inter-subarray data movement in DRAM. In *Proceedings of International Symposium on High Performance Computer Architecture, Barcelona, Spain*, pages 568–580. IEEE.

Chang, Y.-M., Hsiu, P.-C., Chang, Y.-H., Chen, C.-H., Kuo, T.-W., and Wang, C.-Y. M. (2016b). Improving PCM endurance with a constant-cost wear leveling design. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 22(1):1–27.

Chatterjee, N., Shevgoor, M., Balasubramonian, R., Davis, A., Fang, Z., Illikkal, R., and Iyer, R. (2012). Leveraging heterogeneity in DRAM main memories to accelerate critical word access. In *Proceedings of International Symposium on Microarchitecture, Vancouver, BC, Canada*, pages 13–24.

Chatterjee, N., Shevgoor, M., Balasubramonian, R., Davis, A., Fang, Z., Illikkal, R., and Iyer, R. (2012). Leveraging heterogeneity in DRAM main memories to accelerate critical word access. In *Proceedings of International Symposium on Microarchitecture, Vancouver, BC, Canada*, pages 13–24.

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

Chen, X., Mohamed, M., Li, Z., Shang, L., and Mickelson, A. R. (2013). Process variation in silicon photonic devices. *Applied optics*, 52(31):7638–7647.

Choi, Y., Song, I., Park, M.-H., Hoeju Chung, S. C., Cho, B., Kim, J., Oh, Y., Kwon, D., Sunwoo, J., Shin, J., Rho, Y., Lee, C., Kang, M. G., Lee, J., Kwon, Y., Kim, S., Kim, J., jun Lee, Y., Wang, Q., Cha, S., Ahn, S., Horii, H., Lee, J., Kim, K., Joo, H.-S., Lee, K., Lee, Y.-T., Yoo, J.-H., and Jeong, G. (2012). A 20nm 1.8V 8Gb PRAM with 40MB/s program bandwidth. In *Proceedings of International Solid-State Circuits Conference, San Francisco, CA, USA*, pages 46–48.

Cirillo, D. and Valencia, A. (2019). Big data analytics for personalized medicine. *Current opinion in biotechnology*, 58:161–167.

Connelly, M. J. (2007). *Semiconductor optical amplifiers*. Springer Science & Business Media.

Conway, P., Kalyanasundharam, N., Donley, G., Lepak, K., and Hughes, B. (2009). Blade computing with the AMD Opteron™ processor (" magny-cours"). In *Proceedings of Hot Chips Symposium, Stanford, CA, USA*, pages 1–19. IEEE.

Coskun, A., Eris, F., Joshi, A., Kahng, A. B., Ma, Y., Narayan, A., and Srinivas, V. (2020). Cross-layer co-optimization of network design and chiplet placement in 2.5 D systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(12):5183–5196.

Demir, Y., Pan, Y., Song, S., Hardavellas, N., Kim, J., and Memik, G. (2014). Galaxy: A high-performance energy-efficient multi-chip architecture using photonic interconnects. In *Proceedings of International Conference on Supercomputing, Munich, Germany*, pages 303–312.

Densmore, A., Janz, S., Ma, R., Schmid, J. H., Xu, D.-X., Delâge, A., Lapointe, J., Vachon, M., and Cheben, P. (2009). Compact and low power thermo-optic switch using folded silicon waveguides. *Optics Express*, 17(13):10457–10465.

Dong, J., Zhang, L., Han, Y., Wang, Y., and Li, X. (2011). Wear rate leveling: Lifetime enhancement of PRAM with endurance variation. In *Proceedings of Design Automation Conference, New York, NY, USA*, pages 972–977.

Dong, X., Xie, Y., Muralimanohar, N., and Jouppi, N. P. (2010). Simple but effective heterogeneous main memory with on-chip memory controller support. In *Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis, New Orleans, LA, USA*, pages 1–11.

Dong, X., Xu, C., Xie, Y., and Jouppi, N. P. (2012). NVSim: A circuit-level performance, energy, and area model for emerging nonvolatile memory. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31(7):994–1007.

Du, Y., Zhou, M., Childers, B. R., Mossé, D., and Melhem, R. (2013). Bit mapping for balanced PCM cell programming. In *Proceedings of International Symposium on Computer Architecture, Tel Aviv, Israel*, page 428–439.

Dulloor, S. R., Roy, A., Zhao, Z., Sundaram, N., Satish, N., Sankaran, R., Jackson, J., and Schwan, K. (2016). Data tiering in heterogeneous memory systems. In *Proceedings of European Conference on Computer Systems, London, UK*, pages 1–16.

Ebrahimi, M., Weldezion, A. Y., and Daneshtalab, M. (2017). NoD: Network-on-Die as a standalone NoC for heterogeneous many-core systems in 2.5 D ICs. In *Proceedings of International Symposium on Computer Architecture and Digital Systems, Kish Island, Iran*, pages 1–6. IEEE.

Feldmann, J., Stegmaier, M., Gruhler, N., Ríos, C., Bhaskaran, H., Wright, C., and Pernice, W. (2017). Calculating with light using a chip-scale all-optical abacus. *Nature communications*, 8(1):1–8.

Feldmann, J., Youngblood, N., Karpov, M., Gehring, H., Li, X., Stappers, M., Le Gallo, M., Fu, X., Lukashchuk, A., Raja, A., et al. (2021). Parallel convolutional processing using an integrated photonic tensor core. *Nature*, 589(7840):52–58.

Feldmann, J., Youngblood, N., Li, X., Wright, C. D., Bhaskaran, H., and Pernice, W. H. (2019). Integrated 256 cell photonic phase-change memory with 512-bit capacity. *IEEE Journal of Selected Topics in Quantum Electronics*, 26(2):1–7.

Ferreira, A. P., Zhou, M., Bock, S., Childers, B., Melhem, R., and Mossé, D. (2010). Increasing PCM main memory lifetime. In *Proceedings of Design, Automation & Test in Europe Conference & Exhibition, Dresden, Germany*, pages 914–919.

Floridi, L. and Chiriatti, M. (2020). GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694.

Fu, Y., Wu, E., Sirasao, A., Attia, S., Khan, K., and Wittig, R. (2016). Deep learning with int8 optimization on xilinx devices. `https://www.xilinx.com/support/documentation/white_papers/wp486-deep-learning-int8.pdf`.

Gai, K., Qiu, M., and Zhao, H. (2016). Cost-aware multimedia data allocation for heterogeneous memory using genetic algorithm in cloud computing. *IEEE Transactions on Cloud Computing*, 8(4):1212–1222.

Gelsinger, P. P. (2001). Microprocessors for the new millennium: Challenges, opportunities, and new frontiers. In *International Solid-State Circuits Conference. Digest of Technical Papers, San Francisco, CA, USA*, pages 22–25.

Grani, P. and Bartolini, S. (2014). Design options for optical ring interconnect in future client devices. *ACM Journal on Emerging Technologies in Computing Systems*, 10(4):1–25.

Grani, P., Proietti, R., Akella, V., and Yoo, S. B. (2017). Design and evaluation of AWGR-based photonic NoC architectures for 2.5D integrated high performance computing systems. In *Proceedings of International Symposium on High Performance Computer Architecture, Austin, TX, USA*, pages 289–300.

Hady, F. T., Foong, A., Veal, B., and Williams, D. (2017). Platform storage performance with 3D XPoint technology. *Proceedings of the IEEE*, 105(9):1822–1833.

Hamerly, G., Perelman, E., Lau, J., and Calder, B. (2005). Simpoint 3.0: Faster and more flexible program phase analysis. *Journal of Instruction Level Parallelism*, 7(4):1–28.

He, M., Song, C., Kim, I., Jeong, C., Kim, S., Park, I., Thottethodi, M., and Vijaykumar, T. (2020). Newton: A DRAM-maker's accelerator-in-memory (AiM) architecture for machine learning. In *Proceedings of International Symposium on Microarchitecture*, pages 372–385. IEEE.

Henning, J. L. (2006). SPEC CPU2006 benchmark descriptions. *ACM SIGARCH Computer Architecture News*, 34(4):1–17.

Heroux, M. (2007). HPCCG microapp. `http://www.cs.sandia.gov/~maherou/HPCCG-0.3.tar.gz`.

Hertz, J. (2021). Micron unveils 1 DRAM process node — The highest-density DRAM to date. `https://www.allaboutcircuits.com/news/micron-unveils-1a-dram-process-node-highest-density-dram-to-date/`.

Hesla, L. (2012). Particle physics tames big data. `https://www.symmetrymagazine.org/article/august-2012/particle-physics-tames-big-data`.

Hosseini, P., Wright, C. D., and Bhaskaran, H. (2014). An optoelectronic framework enabled by low-dimensional phase-change films. *Nature*, 511(7508):206–211.

Howard, J., Dighe, S., Hoskote, Y., Vangal, S., Finan, D., Ruhl, G., Jenkins, D., Wilson, H., Borkar, N., Schrom, G., Pailet, F., Jain, S., Jacob, T., Yada, S., Marella, S., Salihundam, P., Erraguntla, V., Konow, M., Riepen, M., Droege, G., Lindemann, J., Gries, M., Apel, T., Henriss, K., Lund-Larsen, T., Steibl, S., Borkar, S., De, V., Wijngaart, R. V. D., and Mattson, T. (2010). A 48-core IA-32 message-passing processor with DVFS in 45nm CMOS. In *Proceedings of International Solid-State Circuits Conference, San Francisco, CA, USA*, pages 108–109.

Hu, X., Stow, D., and Xie, Y. (2018). Die stacking is happening. *IEEE Micro*, 38(1):22–28.

Ingerly, D. B., Amin, S., Aryasomayajula, L., Balankutty, A., Borst, D., Chandra, A., Cheemalapati, K., Cook, C. S., Criss, R., Enamul, K., Gomes, W., Jones, D., Kolluru, K. C., Kandas, A., Kim, G. ., Ma, H., Pantuso, D., Petersburg, C. F., Phen-givoni, M., Pillai, A. M., Sairam, A., Shekhar, P., Sinha, P., Stover, P., Telang, A., and Zell, Z. (2019). Foveros: 3D integration and the use of face-to-face chip stacking for logic devices. In *Proceedings of International Electron Devices Meeting, San Francisco, CA, USA*, pages 19.6.1–19.6.4.

JEDEC (2013). High bandwidth memory (HBM) DRAM. `https://www.jedec.org/standards-documents/docs/jesd235a`.

Jerger, N. E., Kannan, A., Li, Z., and Loh, G. H. (2014). NoC architectures for silicon interposer systems: Why pay for more wires when you can get them (from your interposer) for free? In *Proceedings of International Symposium on Microarchitecture, Cambridge, UK*, pages 458–470. IEEE.

Jiang, L., Zhang, Y., Childers, B. R., and Yang, J. (2012a). FPB: Fine-grained power budgeting to improve write throughput of multi-level cell phase change memory. In *Proceedings of International Symposium on Microarchitecture, Vancouver, BC, Canada*, pages 1–12.

Jiang, L., Zhao, B., Zhang, Y., Yang, J., and Childers, B. R. (2012b). Improving write operations in MLC phase change memory. In *Proceedings of International Symposium on High Performance Computer Architecture, New Orleans, LA, USA*, pages 1–10.

Kandlikar, S. G. (2014). Review and projections of integrated cooling systems for three-dimensional integrated circuits. *Journal of Electronic Packaging*, 136(2).

Kang, U., Yu, H.-S., Park, C., Zheng, H., Halbert, J., Bains, K., Jang, S., and Choi, J. S. (2014). Co-architecting controllers and DRAM to enhance DRAM process scaling. In *The memory forum*, volume 14.

Kannan, A., Jerger, N. E., and Loh, G. H. (2015). Enabling interposer-based disintegration of multi-core processors. In *Proceedings of International Symposium on Microarchitecture, Waikiki, HI, USA*, pages 546–558.

Kannan, S., Gavrilovska, A., Gupta, V., and Schwan, K. (2017). HeteroOS: OS design for heterogeneous memory management in datacenter. In *Proceedings of International Symposium of Computer Architecture, Toronto, ON, Canada*, pages 521–534.

Karpov, I., Mitra, M., Kau, D., Spadini, G., Kryukov, Y., and Karpov, V. (2007). Fundamental drift of parameters in chalcogenide phase change memory. *Journal of Applied Physics*, 102(12):124503.

Khouzani, H. A., Hosseini, F. S., and Yang, C. (2016). Segment and conflict aware page allocation and migration in DRAM-PCM hybrid main memory. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(9):1458–1470.

Kim, J. and Kim, Y. (2014). HBM: Memory solution for bandwidth-hungry processors. In *Proceedings of Hot Chips Symposium, Cupertino, CA, USA*, pages 1–24.

Kim, N. S., Song, C., Cho, W. Y., Huang, J., and Jung, M. (2019). LL-PCM: Low-latency phase change memory architecture. In *Proceedings of Design Automation Conference, Las Vegas, NV, USA*, pages 1–6.

Kim, S. K., Lee, S. W., Han, J. H., Lee, B., Han, S., and Hwang, C. S. (2010). Capacitors with an equivalent oxide thickness of $< 0.5nm$ for nanoscale electronic semiconductor memory. *Advanced Functional Materials*, 20(18):2989–3003.

Kim, S. K. and Popovici, M. (2018). Future of dynamic random-access memory as main memory. *MRS Bulletin*, 43(5):334.

Koka, P., McCracken, M. O., Schwetman, H., Zheng, X., Ho, R., and Krishnamoorthy, A. V. (2010). Silicon-photonic network architectures for scalable, power-efficient multi-chip systems. In *ACM SIGARCH Computer Architecture News*, volume 38, pages 117–128.

Krishnamoorthy, A. V., Zheng, X., Li, G., Yao, J., Pinguet, T., Mekis, A., Thacker, H., Shubin, I., Luo, Y., Raj, K., and Cunningham, J. E. (2011). Exploiting cmos manufacturing to reduce tuning requirements for resonant optical devices. *IEEE Photonics Journal*, 3(3):567–579.

Kumar, R., Farkas, K. I., Jouppi, N. P., Ranganathan, P., and Tullsen, D. M. (2003). Single-ISA heterogeneous multi-core architectures: the potential for processor power reduction. In *Proceedings of International Symposium on Microarchitecture, San Diego, CA, USA*, pages 81–92.

Kwon, K.-W., Fong, X., Wijesinghe, P., Panda, P., and Roy, K. (2015). High-density and robust STT-MRAM array through device/circuit/architecture interactions. *IEEE Transactions on Nanotechnology*, 14(6):1024–1034.

Lee, B. C., Ipek, E., Mutlu, O., and Burger, D. (2009). Architecting phase change memory as a scalable DRAM alternative. In *Proceedings of International Symposium on Computer Architecture, Austin, TX, USA*, pages 2–13.

Lee, B. G., Chen, X., Biberman, A., Liu, X., Hsieh, I., Chou, C., Dadap, J. I., Xia, F., Green, W. M. J., Sekaric, L., Vlasov, Y. A., Osgood, R. M., and Bergman, K. (2008). Ultrahigh-bandwidth silicon photonic nanowire waveguides for on-chip networks. *IEEE Photonics Technology Letters*, 20(6):398–400.

Lee, C., Hung, C., Cheung, C., Yang, P., Kao, C., Chen, D., Shih, M., Chien, C. C., Hsiao, Y., Chen, L., Su, M., Alfano, M., Siegel, J., Din, J., and Black, B. (2016a). An overview of the development of a GPU with integrated HBM on silicon interposer. In *Proceedings of Electronic Components and Technology Conference, Las Vegas, NV, USA*, pages 1439–1444.

Lee, J. C., Kim, J., Kim, K. W., Ku, Y. J., Kim, D. S., Jeong, C., Yun, T. S., Kim, H., Cho, H. S., Oh, S., Lee, H. S., Kwon, K. H., Lee, D. B., Choi, Y. J., Lee, J., Kim, H. G., Chun, J. H., Oh, J., and Lee, S. H. (2016b). High bandwidth memory(HBM) with TSV technique. In *Proceedings of International SoC Design Conference, Jeju, South Korea*, pages 181–182.

Lefurgy, C., Rajamani, K., Rawson, F., Felter, W., Kistler, M., and Keller, T. W. (2003). Energy management for commercial servers. *Computer*, 36(12):39–48.

Lepak, K., Talbot, G., White, S., Beck, N., and Naffziger, S. (2017). The next generation AMD enterprise server product architecture). In *Proceedings of Hot Chips Symposium, Cupertino, CA, USA*, pages 1–19. IEEE.

Leskovec, J. and Krevl, A. (2014). SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`.

Li, B., Song, C., Wei, J., Ahn, J. H., and Kim, N. S. (2016a). Exploring new features of high-bandwidth memory for GPUs. *IEICE Electronics Express*, 13(14):20160527–20160527.

Li, H., Xuan, Z., Titriku, A., Li, C., Yu, K., Wang, B., Shafik, A., Qi, N., Liu, Y., Ding, R., Baehr-Jones, T., Fiorentino, M., Hochberg, M., Palermo, S., and Chiang, P. Y. (2015). A $25Gb/s$ $4.4V$-swing AC-coupled Si-photonic microring transmitter with 2-tap asymmetric FFE and dynamic thermal tuning in $65nm$ CMOS. In *Proceedings of Solid-State Circuits Conference, San Francisco, CA, USA*, pages 1–3.

Li, J., Luan, B., and Lam, C. (2012). Resistance drift in phase change memory. In *Proceedings of International Reliability Physics Symposium, Anaheim, CA, USA*, pages 6C–1.

Li, S., Ahn, J. H., Strong, R. D., Brockman, J. B., Tullsen, D. M., and Jouppi, N. P. (2009). McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of International Symposium on Microarchitecture, New York, New York*, pages 469–480.

Li, S., Xu, C., Zou, Q., Zhao, J., Lu, Y., and Xie, Y. (2016b). Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories. In *Proceedings of Design Automation Conference, Austin, TX, USA*, pages 1–6.

Li, X., Youngblood, N., Cheng, Z., Carrillo, S. G.-C., Gemo, E., Pernice, W. H., Wright, C. D., and Bhaskaran, H. (2020). Experimental investigation of silicon and silicon nitride platforms for phase-change photonic in-memory computing. *Optica*, 7(3):218–225.

Li, X., Youngblood, N., Ríos, C., Cheng, Z., Wright, C. D., Pernice, W. H., and Bhaskaran, H. (2019). Fast and reliable storage using a 5 bit, nonvolatile photonic memory cell. *Optica*, 6(1):1–6.

Li, Y. and Chen, L. (2020). EquiNox: Equivalent NoC Injection Routers for Silicon Interposer-Based Throughput Processors. In *Proceedings of International Symposium on High Performance Computer Architecture, San Diego, CA*, pages 435–446.

Li, Z., Qouneh, A., Joshi, M., Zhang, W., Fu, X., and Li, T. (2015). Aurora: A cross-layer solution for thermally resilient photonic network-on-chip. *IEEE Transactions on Very Large Scale Integration Systems*, 23(1):170–183.

Lin, M., Huang, T., Tsai, C., Tam, K., Hsieh, K. C., Chen, C., Huang, W., Hu, C., Chen, Y., Goel, S. K., Fu, C., Rusu, S., Li, C., Yang, S., Wong, M., Yang, S., and Lee, F. (2020). A $7nm$ $4GHz$ Arm-core-based CoWoS chiplet design for high-performance computing. *IEEE Journal of Solid-State Circuits*, 55(4):956–966.

Lischke, S., Knoll, D., Mai, C., Zimmermann, L., Peczek, A., Kroh, M., Trusch, A., Krune, E., Voigt, K., and Mai, A. (2015). High bandwidth, high responsivity waveguide-coupled germanium p-i-n photodiode. *Optics express*, 23(21):27213–27220.

Liu, X., Wen, W., Qian, X., Li, H., and Chen, Y. (2018). Neu-NoC: A high-efficient interconnection network for accelerated neuromorphic systems. In *Proceedings of Asia and South Pacific Design Automation Conference, Jeju Island, South Korea*, pages 141–146. IEEE.

Loh, G. H., Jerger, N. E., Kannan, A., and Eckert, Y. (2015). Interconnect-memory challenges for multi-chip, silicon interposer systems. In *Proceedings of international symposium on Memory Systems, Washington, DC, USA*, pages 3–10.

Luo, J., Killian, C., Beux, S. L., Chillet, D., Sentieys, O., and O'connor, I. (2018). Offline optimization of wavelength allocation and laser power in nanophotonic interconnects. *ACM Journal on Emerging Technologies in Computing Systems*, 14(2):24.

Luo, L.-W., Ophir, N., Chen, C. P., Gabrielli, L. H., Poitras, C. B., Bergmen, K., and Lipson, M. (2014). WDM-compatible mode-division multiplexing on a silicon chip. *Nature communications*, 5(1):1–7.

Lyeo, H.-K., Cahill, D. G., Lee, B.-S., Abelson, J. R., Kwon, M.-H., Kim, K.-B., Bishop, S. G., and Cheong, B.-k. (2006). Thermal conductivity of phase-change material $Ge_2Sb_2Te_5$. *Applied Physics Letters*, 89(15):151904.

Macri, J. (2015). AMD's next generation GPU and high bandwidth memory architecture: FURY. In *Proceedings of Hot Chips Symposium, Cupertino, CA, USA*, pages 1–26.

Mahajan, R., Sankman, R., Patel, N., Kim, D., Aygun, K., Qian, Z., Mekonnen, Y., Salama, I., Sharan, S., Iyengar, D., and Mallik, D. (2016). Embedded Multi-die Interconnect Bridge (EMIB) – A High Density, High Bandwidth Packaging Interconnect. In *Proceedings of Electronic Components and Technology Conference, Las Vegas, Nevada, USA*, pages 557–565.

Martinez, J. F. and Ipek, E. (2009). Dynamic multicore resource management: A machine learning approach. *IEEE micro*, 29(5):8–17.

Massey Jr, F. J. (1951). The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78.

Mellanox (2015). Tile-gx72 processor. `http://www.mellanox.com/related-docs/prod_multi_core/PB_TILE-Gx72.pdf`.

Meng, J., Kawakami, K., and Coskun, A. K. (2012). Optimizing energy efficiency of 3D multicore systems with stacked DRAM under power and thermal constraints. In *Proceedings of Design Automation Conference, San Francisco, CA, USA*, pages 648–655.

Meswani, M. R., Blagodurov, S., Roberts, D., Slice, J., Ignatowski, M., and Loh, G. H. (2015). Heterogeneous memory architectures: A HW/SW approach for mixing die-stacked and off-package memories. In *Proceedings of International Symposium on High Performance Computer Architecture, Burlingame, CA, USA*, pages 126–136.

Michel, A.-K. U., Zalden, P., Chigrin, D. N., Wuttig, M., Lindenberg, A. M., and Taubner, T. (2014). Reversible optical switching of infrared antenna resonances with ultrathin phase-change layers using femtosecond laser pulses. *ACS Photonics*, 1(9):833–839.

MICRON (2011). DDR3 SDRAM power calculator. `https://www.micron.com/products/dram/ddr3-sdram`.

MICRON (2013). LPDDR2 SDRAM power calculator. `http://www.micron.com/products/dram/lpdram`.

MICRON (2016). RLDRAM3 power calculator. `http://www.micron.com/products/dram/rldram-memory`.

Microsoft (2019). Microsoft Turing NLG. `https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/`.

Mirhosseini, A., Sadrosadati, M., Soltani, B., Sarbazi-Azad, H., and Wenisch, T. F. (2017). BiNoCHS: Bimodal network-on-chip for CPU-GPU heterogeneous systems. In *Proceedings of International Symposium on Networks-on-Chip, Seoul, South Korea*, pages 1–8.

Morris, R., Kodi, A. K., and Louri, A. (2012). Dynamic reconfiguration of 3D photonic networks-on-chip for maximizing performance and improving fault tolerance. In *Proceedings of International Symposium on Microarchitecture, Vancouver, BC, Canada*, pages 282–293.

Mutlu, O. (2013). Memory scaling: A systems architecture perspective. In *International Memory Workshop*, pages 21–25.

Mutlu, O. (2018). Processing data where it makes sense in modern computing systems: Enabling in-memory computation. In *Proceedings of Mediterranean Conference on Embedded Computing*, pages 8–9.

Mutlu, O., Kim, H., and Patt, Y. N. (2006). Efficient runahead execution: Power-efficient memory latency tolerance. *IEEE Micro*, 26(1):10–20.

Narayan, A., Joshi, A., and Coskun, A. K. (2020a). Bandwidth allocation in silicon-photonic networks using application instrumentation. In *Proceedings of High Performance Extreme Computing Conference, Waltham, MA, USA*, pages 1–2.

Narayan, A., Thonnart, Y., Vivet, P., and Coskun, A. K. (2020b). PROWAVES: Proactive runtime wavelength selection for energy-efficient photonic NoCs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 1–1.

Narayan, A., Thonnart, Y., Vivet, P., Joshi, A., and Coskun, A. K. (2020c). System-level evaluation of chip-scale silicon photonic networks for emerging data-intensive applications. In *Proceedings of Design, Automation Test in Europe Conference Exhibition, Grenoble, France*, pages 1444–1449.

Narayan, A., Thonnart, Y., Vivet, P., Tortolero, C. F., and Coskun, A. K. (2019). WAVES: Wavelength selection for power-efficient 2.5D-integrated photonic NoCs. In *Proceedings of Design, Automation Test in Europe Conference Exhibition, Florence, Italy*, pages 516–521.

Narayan, A., Zhang, T., Aga, S., Narayanasamy, S., and Coskun, A. (2018). MOCA: Memory object classification and allocation in heterogeneous memory systems. In *Proceedings of International Parallel and Distributed Processing Symposium, Vancouver, BC, Canada*, pages 326–335.

Nirschl, T., Philipp, J., Happ, T., Burr, G., Rajendran, B., Lee, M.-H., Schrott, A., Yang, M., Breitwisch, M., Chen, C.-F., Joseph, E., Lamorey, M., Cheek, R., Chen, S.-H., Zaidi, S., Raoux, S., Chen, Y., Zhu, Y., Bergmann, R., Lung, H.-L., and Lam, C. (2007). Write strategies for 2 and 4-bit multi-level phase-change memory. In *Proceedings of International Electron Devices Meeting, Washington, DC, USA*, pages 461–464.

Notomi, M., Nozaki, K., Shinya, A., Matsuo, S., and Kuramochi, E. (2014). Toward fj/bit optical communication in a chip. *Optics Communications*, 314:3–17.

Nvidia (2018). NVidia Turing GPU Architecture. `https://www.nvidia.com/en-us/geforce/turing/`.

Nvidia (2019). Nvidia megatron. `https://github.com/NVIDIA/Megatron-LM`.

Olarig, S. P., Koenen, D. J., and Heng, C. S. (2003). Method and apparatus for supporting heterogeneous memory in computer systems. US Patent 6,530,007.

Ovshinsky, S. R. (1968). Reversible electrical switching phenomena in disordered structures. *Physical Review Letters*, 21(20):1450.

Ozdal, M. M., Yesil, S., Kim, T., Ayupov, A., Burns, S., and Ozturk, O. (2015). Architectural requirements for energy efficient execution of graph analytics applications. In *Proceedings of International Conference on Computer-Aided Design, Austin, TX, USA*, pages 676–681.

Padmaraju, K. and Bergman, K. (2014). Resolving the thermal challenges for silicon microring resonator devices. *Nanophotonics*, 3(4-5):269–281.

Parry, J. and Wang, L. (2018). A complete guide to 3D chip-package thermal co-design... 10 key considerations. `https://corner-stone.com.tw/wp-content/uploads/2018/08/A-COMPLETE-GUIDE-TO-3D-CHIP-PACKAGE.pdf`.

Pasricha, S. and Nikdast, M. (2020). A survey of silicon photonics for energy-efficient manycore computing. *IEEE Design & Test*, 37(4):60–81.

Paul, I., Huang, W., Arora, M., and Yalamanchili, S. (2015). Harmonia: Balancing Compute and Memory Power in High-Performance GPUs. In *Proceedings of International Symposium on Computer Architecture, Portland, Oregon, USA*, page 54–65.

Pavlovic, M., Puzovic, N., and Ramirez, A. (2013). Data placement in HPC architectures with heterogeneous off-chip memory. In *Proceedings of International Conference on Computer Design, Asheville, NC, USA*, pages 193–200.

Peon-Quiros, M., Bartzas, A., Mamagkakis, S., Catthoor, F., Mendías, J. M., and Soudris, D. (2015). Placement of linked dynamic data structures over heterogeneous memories in embedded systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 14(2):1–30.

Peón-quirós, M. et al. (2015). Placement of linked dynamic data structures over heterogeneous memories in embedded systems. *ACM Transactions on Embedded Computing Systems*, 14(2):37:1–37:30.

Phadke, S. and Narayanasamy, S. (2011). MLP aware heterogeneous memory system. In *Proceedings of Design, Automation and Test in Europe, Grenoble, France*, pages 1–6.

Polster, R., Thonnart, Y., Waltener, G., Gonzalez, J.-L., and Cassan, E. (2016). Efficiency optimization of silicon photonic links in 65-nm CMOS and 28-nm FDSOI technology nodes. *IEEE Transactions on Very Large Scale Integration Systems*, 24(12):3450–3459.

Poremba, M., Zhang, T., and Xie, Y. (2015). Nvmain 2.0: A user-friendly memory simulator to model (non-) volatile memory systems. *IEEE Computer Architecture Letters*, 14(2):140–143.

Qureshi, M. K., Franceschini, M. M., Jagmohan, A., and Lastras, L. A. (2012). PreSET: Improving performance of phase change memories by exploiting asymmetry in write times. In *Proc. International Symposium on Computer Architecture, Portland, Oregon, USA*.

Qureshi, M. K., Karidis, J., Franceschini, M., Srinivasan, V., Lastras, L., and Abali, B. (2009a). Enhancing lifetime and security of PCM-based main memory with start-gap wear leveling. In *Proceedings of international Symposium on Microarchitecture, New York, NY, USA*, pages 14–23.

Qureshi, M. K., Srinivasan, V., and Rivers, J. A. (2009b). Scalable high performance main memory system using phase-change memory technology. In *Proceedings of International Symposium on Computer Architecture, Austin, Texas, USA*, pages 24–33.

Rakowski, M., Pantouvaki, M., De Heyn, P., Verheyen, P., Ingels, M., Chen, H., De Coster, J., Lepage, G., Snyder, B., De Meyer, K., Steyaert, M., Pavarelli, N., Lee, J. S., O'Brien, P., Absil, P., and Van Campenhout, J. (2015). A $4 \times 20Gb/s$ WDM ring-based hybrid CMOS silicon photonics transceiver. In *Proceedings of Solid-State Circuits Conference, San Francisco, CA, USA*, pages 1–3.

Raoux, S., Burr, G. W., Breitwisch, M. J., Rettner, C. T., Chen, Y. ., Shelby, R. M., Salinga, M., Krebs, D., Chen, S. ., Lung, H. ., and Lam, C. H. (2008). Phase-change random access memory: A scalable technology. *IBM Journal of Research and Development*, 52(4.5):465–479.

Raoux, S., Xiong, F., Wuttig, M., and Pop, E. (2014). Phase change materials and phase change memory. *MRS bulletin*, 39(8):703–710.

Rekhi, A. S., Zimmer, B., Nedovic, N., Liu, N., Venkatesan, R., Wang, M., Khailany, B., Dally, W. J., and Gray, C. T. (2019). Analog/mixed-signal hardware error modeling for deep learning inference. In *Proceedings of Design Automation Conference, Las Vegas, NV, USA*, pages 1–6.

Rho, K., Tsuchida, K., Kim, D., Shirai, Y., Bae, J., Inaba, T., Noro, H., Moon, H., Chung, S., Sunouchi, K., Park, J., Park, K., Yamamoto, A., Chung, S., Kim, H., Oyamatsu, H., and Oh, J. (2017). A $4Gb$ LPDDR2 STT-MRAM with compact $9F2$ $1T1MTJ$ cell and hierarchical bitline architecture. In *Proceedings of International Solid-State Circuits Conference, San Francisco, CA, USA*, pages 396–397.

Rios, C., Hosseini, P., Wright, C. D., Bhaskaran, H., and Pernice, W. H. (2014). On-chip photonic memory elements employing phase-change materials. *Advanced Materials*, 26(9):1372–1377.

Rosenfeld, P., Cooper-Balis, E., and Jacob, B. (2011). DRAMSim2: A cycle accurate memory system simulator. *IEEE computer architecture letters*, 10(1):16–19.

Rumley, S., Bahadori, M., Polster, R., Hammond, S. D., Calhoun, D. M., Wen, K., Rodrigues, A., and Bergman, K. (2017). Optical interconnects for extreme scale computing systems. *Parallel Computing*, 64:65–80.

Ríos, C., Stegmaier, M., Hosseini, P., Wang, D., Scherer, T., Wright, C. D., Bhaskaran, H., and Pernice, W. H. (2015). Integrated all-photonic non-volatile multi-level memory. *Nature Photonics*, 9(11):725.

Saban, K. (2011). Xilinx stacked silicon interconnect technology delivers breakthrough FPGA capacity, bandwidth, and power efficiency. `https://www.xilinx.com/support/documentation/white_papers/wp380_Stacked_Silicon_Interconnect_Technology.pdf`.

Seshadri, V., Kim, Y., Fallin, C., Lee, D., Ausavarungnirun, R., Pekhimenko, G., Luo, Y., Mutlu, O., Gibbons, P. B., Kozuch, M. A., and Mowry, T. C. (2013). RowClone: Fast and energy-efficient in-DRAM bulk data copy and initialization. In *Procedings of International Symposium on Microarchitecture, Davis, CA, USA*, pages 185–197.

Seshadri, V., Lee, D., Mullins, T., Hassan, H., Boroumand, A., Kim, J., Kozuch, M. A., Mutlu, O., Gibbons, P. B., and Mowry, T. C. (2017). Ambit: In-memory accelerator for bulk bitwise operations using commodity DRAM technology. In *Proceedings of International Symposium on Microarchitecture, Boston, MA, USA*, pages 273–287.

Shamim, M. S., Narde, R. S., Gonzalez-Hernandez, J.-L., Ganguly, A., Venkatarman, J., and Kandlikar, S. G. (2019). Evaluation of wireless network-on-chip architectures with microchannel-based cooling in 3D multicore chips. *Sustainable Computing: Informatics and Systems*, 21:165–178.

Shang, K., Pathak, S., Guan, B., Liu, G., and Yoo, S. (2015). Low-loss compact multilayer silicon nitride platform for 3D photonic integrated circuits. *Optics Express*, 23(16):21334–21342.

Shen, D., Liu, X., and Lin, F. X. (2016). Characterizing emerging heterogeneous memory. In *Proceedings of International Symposium on Memory Management, Santa Barbara, CA, USA*, pages 13–23.

Simon, G., Saliou, F., Chanclou, P., Neto, L. A., and Erasme, D. (2016). Experimental demonstration of low cost wavelength drift mitigation for TWDM systems. In *Proceedings of European Conference on Optical Communication, Dusseldorf, Germany*, pages 1–3.

Skadron, K., Stan, M. R., Huang, W., Velusamy, S., Sankaranarayanan, K., and Tarjan, D. (2003). Temperature-aware microarchitecture. In *Proceedings of International Symposium on Computer Architecture, San Diego, CA, USA*, pages 2–13.

Sodani, A. (2015). Knights landing (knl): 2nd generation intel® xeon phi processor. In *Proceedings of Hot Chips Symposium, Cupertino, CA, USA*, pages 1–24.

Song, S., Das, A., Mutlu, O., and Kandasamy, N. (2019). Enabling and exploiting partition-level parallelism (PALP) in phase change memories. *ACM Transactions on Embedded Computing Systems*, 18(5s):1–25.

Spica, M. and Mak, T. (2004). Do we need anything more than single bit error correction (ECC)? In *International Workshop on Memory Technology, Design and Testing*, pages 111–116. IEEE.

Stow, D., Akgun, I., Barnes, R., Gu, P., and Xie, Y. (2016). Cost analysis and cost-driven IP reuse methodology for SoC design based on 2.5 D/3D integration. In *Proceedings of International Conference on Computer-Aided Design, Austin, TX, USA*, page 56.

Sun, C., Wade, M., Georgas, M., Lin, S., Alloatti, L., Moss, B., Kumar, R., Atabaki, A. H., Pavanello, F., Shainline, J. M., Orcutt, J. S., Ram, R. J., Popović, M., and Stojanović, V. (2016). A $45nm$ CMOS-SOI monolithic photonics platform with bit-statistics-based resonant microring thermal tuning. *IEEE Journal of Solid-State Circuits*, 51(4):893–907.

Tanaka, D., Shoji, Y., Kuwahara, M., Wang, X., Kintaka, K., Kawashima, H., Toyosaki, T., Ikuma, Y., and Tsuda, H. (2012). Ultra-small, self-holding, optical gate switch using $Ge_2Sb_2Te_5$ with a multi-mode Si waveguide. *Optics Express*, 20(9):10283–10294.

Thakkar, I. G. and Pasricha, S. (2018). Libra: Thermal and process variation aware reliability management in photonic networks-on-chip. *IEEE Transactions on Multi-Scale Computing Systems*, 4(4):758–772.

Thonnart, Y., Bernabé, S., Charbonnier, J., Bernard, C., Coriat, D., Fuguet, C., Tissier, P., Charbonnier, B., Malhouitre, S., Saint-Patrice, D., Assous, M., Narayan, A., Coskun, A., Dutoit, D., and Vivet, P. (2020). POPSTAR: A robust modular optical NoC architecture for chiplet-based 3D integrated systems. In *Proceedings of Design, Automation & Test in Europe Conference & Exhibition, Grenoble, France*.

Thonnart, Y., Zid, M., Gonzalez-Jimenez, J. L., Waltener, G., Polster, R., Dubray, O., Lepin, F., Bernabé, S., Menezo, S., Parès, G., Castany, O., Boutafa, L., Grosse, P., Charbonnier, B., and Baudot, C. (2018). A 10Gb/s Si-photonic transceiver with 150μW 120μs-lock-time digitally supervised analog microring wavelength stabilization for $1Tb/s/mm^2$ die-to-die optical networks. In *Proceedings of International Solid-State Circuits Conference, San Francisco, CA, USA*, pages 350–352.

Toal, C., Burns, D., McLaughlin, K., Sezer, S., and O'Kane, S. (2007). An RLDRAM II implementation of a 10Gbps shared packet buffer for network processing. In *Proceedings of Conference on Adaptive Hardware and Systems, Edinburgh, UK*, pages 613–618.

Tran, L., Kurdahi, F. J., Eltawil, A. M., and Homayoun, H. (2013). Heterogeneous memory management for 3D-DRAM and external DRAM with QoS. In *Proceedings of Asia and South Pacific Design Automation Conference, Yokohama, Japan*, pages 663–668.

Valsan, P. K. and Yun, H. (2015). MEDUSA: a predictable and high-performance DRAM controller for multicore based embedded systems. In *Proceedings of International Conference on Cyber-Physical Systems, Networks, and Applications, Seattle, WA, USA*, pages 86–93. IEEE.

Van Winkle, S., Kodi, A. K., Bunescu, R., and Louri, A. (2018). Extending the power-efficiency and performance of photonic interconnects for heterogeneous multicores with machine learning. In *Proceedings of International Symposium on High Performance Computer Architecture, Vienna, Austria*, pages 480–491.

Venkata, S. K., Ahn, I., Jeon, D., Gupta, A., Louie, C., Garcia, S., Belongie, S., and Taylor, M. B. (2009). SD-VBS: The San Diego vision benchmark suite. In *Proceedings of International Symposium on Workload Characterization, Austin, TX, USA*, pages 55–64.

Virot, L., Crozat, P., Fédéli, J.-M., Hartmann, J.-M., Marris-Morini, D., Cassan, E., Boeuf, F., and Vivien, L. (2014). Germanium avalanche receiver for low power interconnects. *Nature communications*, 5:4957.

Vivet, P., Guthmuller, E., Thonnart, Y., Pillonnet, G., Moritz, G., Miro-Panadès, I., Fuguet, C., Durupt, J., Bernard, C., Varreau, D., Pontes, J., Thuries, S., Coriat, D., Harrand, M., Dutoit, D., Lattard, D., Arnaud, L., Charbonnier, J., Coudrain, P., Garnier, A., Berger, F., Gueugnot, A., Greiner, A., Meunier, Q., Farcy, A., Arriordaz, A., Cheramy, S., and Clermidy, F. (2020). A $220GOPS$ 96-core processor with 6 chiplets 3D-stacked on an active interposer offering $0.6ns/mm$ latency, $3Tb/s/mm^2$ inter-chiplet interconnects and $156mW/mm^2$ at 82%-peak-efficiency DC-DC converters. In *Proceedings of International Solid- State Circuits Conference, San Francisco, CA, USA*, pages 46–48.

Wade, M., Anderson, E., Ardalan, S., Bhargava, P., Buchbinder, S., L. Davenport, M., Fini, J., Lu, H., Li, C., Meade, R., Ramamurthy, C., Rust, M., Sedgwick, F., Stojanovic, V., Van Orden, D., Zhang, C., Sun, C., Shumarayev, S. Y., O'Keeffe, C., Hoang, T. T., Kehlet, D., Mahajan, R. V., Guzy, M. T., Chan, A., and Tran, T. (2020). TeraPHY: A chiplet technology for low-power, high-bandwidth in-package optical I/O. *IEEE Micro*, 40(2):63–71.

Wade, M. T., Pavanello, F., Kumar, R., Gentry, C. M., Atabaki, A., Ram, R., Stojanović, V., and Popović, M. A. (2015). 75% efficient wide bandwidth grating couplers in a $45nm$ microelectronics CMOS process. In *Proceedings of Optical Interconnects Conference, San Diego, CA, USA*, pages 46–47.

Wang, J., Wang, L., and Liu, J. (2020). Overview of phase-change materials based photonic devices. *IEEE Access*, 8:121211–121245.

Wang, S., Feng, X., Gao, S., Shi, Y., Dai, T., Yu, H., Tsang, H.-K., and Dai, D. (2017). On-chip reconfigurable optical add-drop multiplexer for hybrid wavelength/mode-division-multiplexing systems. *Optics letters*, 42(14):2802–2805.

Wang, Z., Wang, Z., Xu, J., Chang, Y.-S., Feng, J., Chen, X., Chen, S., and Zhang, J. (2019). CAMON: Low-cost silicon photonic chiplet for manycore processors.

IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 39(9):1820–1833.

Werner, S., Navaridas, J., and Luján, M. (2017). A survey on optical network-on-chip architectures. *ACM Computing Surveys (CSUR)*, 50(6):1–37.

Woo, S. C., Ohara, M., Torrie, E., Singh, J. P., and Gupta, A. (1995). The SPLASH-2 programs: Characterization and methodological considerations. In *ACM SIGARCH computer architecture news*, volume 23, pages 24–36.

Wu, X., Huang, C., Xu, K., Shu, C., and Tsang, H. K. (2017). Mode-division multiplexing for silicon photonic network-on-chip. *Journal of Lightwave Technology*, 35(15):3223–3228.

Wuttig, M., Bhaskaran, H., and Taubner, T. (2017). Phase-change materials for non-volatile photonic applications. *Nature Photonics*, 11(8):465–476.

Wuttig, M. and Yamada, N. (2007). Phase-change materials for rewriteable data storage. *Nature materials*, 6(11):824–832.

Xia, F., Jiang, D., Xiong, J., Chen, M., Zhang, L., and Sun, N. (2014). DWC: Dynamic write consolidation for phase change memory systems. In *Proceedings of International conference on Supercomputing, Munich, Germany*, pages 211–220.

Yamada, N., Ohno, E., Nishiuchi, K., Akahira, N., and Takao, M. (1991). Rapid-phase transitions of $GeTe - Sb_2Te_3$ pseudobinary amorphous thin films for an optical disk memory. *Journal of Applied Physics*, 69(5):2849–2856.

Yang, Y.-D., Li, Y., Huang, Y.-Z., and Poon, A. W. (2014). Silicon nitride three-mode division multiplexing and wavelength-division multiplexing using asymmetrical directional couplers and microring resonators. *Optics express*, 22(18):22172–22183.

Yoon, H., Meza, J., Muralimanohar, N., Jouppi, N. P., and Mutlu, O. (2014). Efficient data mapping and buffering techniques for multilevel cell phase-change memories. *ACM Transactions on Architecture and Code Optimization*, 11(4):1–25.

Youngblood, N., Ríos, C., Gemo, E., Feldmann, J., Cheng, Z., Baldycheva, A., Pernice, W. H., Wright, C. D., and Bhaskaran, H. (2019). Tunable volatility of $Ge_2Sb_2Te_5$ in integrated photonics. *Advanced Functional Materials*, 29(11):1807571.

Yu, K., Li, H., Li, C., Titriku, A., Shafik, A., Wang, B., Wang, Z., Bai, R., Chen, C., Fiorentino, M., Chiang, P. Y., and Palermo, S. (2015). A $24Gb/s$ $0.71pJ/b$ Si-photonic source-synchronous receiver with adaptive equalization and microring wavelength stabilization. In *Proceedings of Solid-State Circuits Conference, San Francisco, CA, USA*, pages 1–3.

Zhan, J., Kayıran, O., Loh, G. H., Das, C. R., and Xie, Y. (2016). OSCAR: Orchestrating STT-RAM cache traffic for heterogeneous CPU-GPU architectures. In *Proceedings of International Symposium on Microarchitecture, Taipei, Taiwan*, pages 1–13. IEEE.

Zhang, H., Xu, L., Chen, J., Zhou, L., Rahman, B. M. A., Wu, X., Lu, L., Xu, Y., Xu, J., Song, J., and Hu, Z. (2017). Ultracompact Si-GST hybrid waveguides for nonvolatile light wave manipulation. *IEEE Photonics Journal*, 10(1):1–10.

Zhang, T., Abellán, J. L., Joshi, A., and Coskun, A. K. (2014). Thermal management of manycore systems with silicon-photonic networks. In *Proceedings of Design, Automation & Test in Europe Conference & Exhibition, Dresden, Germany*, page 307.

Zhang, Y., Feng, D., Tong, W., Liu, J., Wang, C., and Xu, J. (2019). Tiered-ReRAM: A low latency and energy efficient TLC crossbar ReRAM architecture. In *Proceedings of Symposium on Mass Storage Systems and Technologies, Santa Clara, CA, USA*, pages 92–102. IEEE.

Zhao, M., Jiang, L., Zhang, Y., and Xue, C. J. (2014). SLC-enabled wear leveling for MLC PCM considering process variation. In *Proceedings of Design Automation Conference, San Francisco, CA, USA*, pages 1–6.

Zhu, F., Gong, R., Yu, F., Liu, X., Wang, Y., Li, Z., Yang, X., and Yan, J. (2020). Towards unified int8 training for convolutional neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1969–1979.

Zulfiqar, A., Koka, P., Schwetman, H., Lipasti, M., Zheng, X., and Krishnamoorthy, A. (2013). Wavelength stealing: An opportunistic approach to channel sharing in multi-chip photonic interconnects. In *Proceedings of International Symposium on Microarchitecture, Davis, CA, USA*, pages 222–233.

CURRICULUM VITAE