University of Kentucky
UKnowledge

Theses and Dissertations--Electrical and Computer Engineering

Electrical and Computer Engineering

2023

A Flexible Photonic Reduction Network Architecture for Spatial GEMM Accelerators for Deep Learning

Bobby Bose
University of Kentucky, bobbybose19@gmail.com
Digital Object Identifier: https://doi.org/10.13023/etd.2023/478

Right click to open a feedback form in a new tab to let us know how this document benefits you.

Recommended Citation

Bose, Bobby, "A Flexible Photonic Reduction Network Architecture for Spatial GEMM Accelerators for Deep Learning" (2023). *Theses and Dissertations--Electrical and Computer Engineering*. 195.
https://uknowledge.uky.edu/ece_etds/195

This Master's Thesis is brought to you for free and open access by the Electrical and Computer Engineering at UKnowledge. It has been accepted for inclusion in Theses and Dissertations--Electrical and Computer Engineering by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Bobby Bose, Student

Dr. Ishan G Thakkar, Major Professor

Dr. Simone Silvestri, Director of Graduate Studies

A Flexible Photonic Reduction Network Architecture for Spatial GEMM Accelerators for Deep Learning

---

THESIS

A thesis submitted in partial
fulfillment of the requirements for
the degree of Master of Science in
Computer Engineering in the College
of Engineering at the University of
Kentucky

By
Bobby Bose
Lexington, Kentucky

Director: Dr. Ishan G Thakkar, Assistant Professor of Electrical Engineering
Lexington, Kentucky
2023

Copyright© Bobby Bose 2023

ABSTRACT OF THESIS

A Flexible Photonic Reduction Network Architecture for Spatial GEMM Accelerators for Deep Learning

As deep neural network (DNN) models increase significantly in complexity and size, it has become important to increase the computing capability of specialized hardware architectures typically used for DNN processing. The major linear operations of DNNs, which comprise the fully connected and convolution layers, are commonly converted into general matrix-matrix multiplication (GEMM) operations for acceleration. Specialized GEMM accelerators are typically employed to implement these GEMM operations, where a GEMM operation is decomposed into multiple vector-dot-product operations that run in parallel. A common challenge that arises in modern DNNs is the mismatch between the matrices used for GEMM operations and the hardware size of the GEMM accelerator. In case the matrices are smaller than the hardware size, some hardware resources go idle but still consume static power. This diminishes the energy efficiency. On the other hand, in case the matrices are larger than the hardware size, the many vector-dot-product operations involved in a GEMM operation cannot be fully mapped onto the hardware structure. As a result, the vector-dot-product operations need to be folded over time into multiple temporal frames. Each temporal frame generates a partial sum (psum) of the final output value of the corresponding dot-product operation. Consequently, to produce the final output matrix, these psums need to be stored in memory and redistributed back into the accelerator to be accumulated using a network of accumulators called a reduction network (RN). To efficiently accelerate modern DNNs with heterogeneous matrix sizes, customized spatial GEMM accelerators have been introduced in prior work. These accelerators employ flexible RNs to implement spatial and temporal reduction of psums of heterogeneous sizes. They create unique mappings of matrices depending on their sizes to compute multiple vector-dot-products in parallel while minimizing the number of computing resources remaining idle.

Despite their advantages, these flexible RNs from prior work are still limited due to their electronic design. A flexible RN typically comprises a network of accumulators that work together to collect and reduce psums. Every electronic accumulator has a limited fan-in, and therefore, a large number of accumulators need to be connected together. This increases the number of hardware components and network links required to achieve the desired reduction of psums, leading to a reduction in performance and energy efficiency. Nevertheless, to address this shortcoming, photonic devices and interconnects have been demonstrated. In this thesis, I present an innovative use of photonic devices and interconnects from the state-of-the-art to build a novel photonic RN architecture. Our photonic RN architecture substantially reduces the required counts of photonic accumulators and links to achieve the spatial and temporal reduction of psums of heterogeneous sizes with massive parallelism. We evaluate our photonic RN and compare it against the state-of-the-art electronic RN architectures from prior work for four modern DNN workloads. The evaluation results show a latency speed-up of up to 5.63× and energy efficiency improvement of up to 1.97× on average across the considered DNN workloads.

KEYWORDS: Reduction Network, Accumulation, Deep Learning, Photonic Architectures, Flexible Spatial Accelerators

Bobby Bose

December 11, 2023

A Flexible Photonic Reduction Network Architecture for Spatial GEMM Accelerators for Deep Learning

By
Bobby Bose

Dr. Ishan G Thakkar
Director of Thesis

Dr. Simone Silvestri
Director of Graduate Studies

December 11, 2023
Date

ACKNOWLEDGMENTS

I would firstly like to thank the members of my committee for taking the time to review my thesis and Masters exam and providing feedback. I would especially like to thank my advisor Dr. Thakkar for all his help and mentoring he has provided me over the years. His support has led to me being able to pursue my research interests in both my undergraduate and graduate college years. Without his guidance, I would not have been able to complete this thesis. He has taught and instilled in me great research techniques, valuable writing skills, and general life advice. I would also like to thank all my fellow research lab mates. Their support and advice has been very valuable and I am happy to have been able to work with and interact with them over the years. Lastly, I would like to thank my family and friends for always believing in me, and backing my research and other life goals. Without them and everyone's support, I would not have been able to reach this point.

TABLE OF CONTENTS

Acknowledgments . . . iii

List of Tables . . . v

List of Figures . . . vi

Chapter 1 Introduction . . . 1

Chapter 2 Background . . . 6
- 2.1 Introduction . . . 6
- 2.2 Convolutional Neural Networks (CNNs) . . . 6
- 2.3 Hardware Implementation of CNNs . . . 7
- 2.4 Flexible and Re-configurable Accelerator Networks . . . 8
- 2.5 Photonics Enabled DNN Accelerators . . . 11
- 2.6 Motivation . . . 11

Chapter 3 Photonic Reduction Network Architecture and Operation . . . 13
- 3.1 Introduction . . . 13
- 3.2 Network Overview . . . 13
- 3.3 Structure . . . 15
  - 3.3.1 Reduction Topology Assembler (RTA) . . . 15
  - 3.3.2 Photo-Charge Accumulator . . . 18
- 3.4 Operation and Example Mapping . . . 20
  - 3.4.1 Partial Sum Signal Conversion and Routing . . . 21
  - 3.4.2 Partial Sum Accumulation . . . 22
  - 3.4.3 Output Division and Reduction . . . 24
- 3.5 Reduction Complexity . . . 26

Chapter 4 Evaluation . . . 30
- 4.1 Evaluation Setup . . . 30
- 4.2 Latency . . . 30
- 4.3 Energy-efficiency . . . 32
- 4.4 Area . . . 33

Chapter 5 Conclusion and Future Work . . . 36
- 5.1 Summary of Results and Conclusion . . . 36
- 5.2 Future Work . . . 37

Bibliography . . . 38

Vita . . . 42

LIST OF TABLES

3.1 Reduction latency and hardware complexities of different RNs. n is the number of psums to accumulate, i is the number of folds, l is the number of levels in the adder tree network, b is bit precision, $T_e = 1.25$ns is electronic cycle time, and $T_p = 5$ps is photonic cycle time 27

4.1 Photonic RN Component Parameters 31
4.2 Losses incurred in the photonic RN 32

LIST OF FIGURES

2.1 Mapping of a convolution operation (a) to an equivalent general matrix multiplication (GEMM) (b) 7

2.2 Example of cluster mapping a GEMM operation (a) onto a flexible DNN accelerator (b) over multiple temporal folds 9

2.3 Typical architecture of a re-configurable DNN accelerator with a tree-based DN topology and a linear MN 10

3.1 Architecture of a 256 PE, photonic interconnect based reduction network 14

3.2 Reduction Topology Assembler used to select carrier wavelength, convert input into optical pulses, and route to an accumulation waveguide 17

3.3 Free Spectral Range of 4 MRs cascaded together to cover 128 wavelengths. Each MR can select from 32 possible wavelengths. 18

3.4 Photo-charge accumulator architecture 20

3.5 Example of mapping cluster size = 4 in a 256 PE photonic RN model . . 21

3.6 Example of the PCA accumulating psum signals from a 4 PE sized cluster over two total temporal frames. (a) Combined photodetector (PD) output current per PE for two temporal frames. Period X_1 captures the first pulse of a psum signal, and period X_2 captures the second pulse. Psum values are between 1-16 bits in example. (b) Voltage output of the two active TIRs during both temporal frames. TIR0 captures the pulse representing the first 8-bits, and TIR1 captures the pule representing the second 8-bits. (c) Total voltage output from the analog op-amp voltage adder after both temporal frames. Not to scale with TIR outputs. 23

3.7 Example operation of the voltage division and reduction circuit in the PCA. Circuit depicted is set up to split a 4-bit TIR output value into two 2-bit parts. (a) No current flow when both switches are open. (b) Voltage division of voltage adder output when S_3 closes. (c) Voltage reduction of the upper 2 bits' voltage when S_3 opens and S_4 closes. 25

3.8 Comparison of reduction latency to reduce different number of partial sums over different bit precisions 28

3.9 Comparison of reduction energy to reduce different number of partial sums over different bit precisions 29

4.1 DNN accelerator architecture used in evaluation with an example tree-based RN 31

4.2 Latency (ms) of the tested RNs to run different DNNs 32

4.3 FPS/W of the tested RNs to run different DNNs 33

4.4 FPS/mm^2 of the tested RNs to run different DNNs 34

Chapter 1 Introduction

In recent years, deep neural network (DNN) models have become more widespread and have significantly grown in complexity and size, with the aim of achieving higher accuracy [10,30]. In order to allow this growth, without driving computing metrics to unreasonable levels, a research motivation has been introduced to focus on improving the computational performance of these networks. One of the most widely seen DNN models is the convolutional neural network (CNN), which is commonly used for visual image analysis [26]. CNNs consist of many layers, such as the convolution, fully-connected, and pooling layers, among others. However, the convolution layers appear the most frequently and tend to consume a large portion of the CNN's overall computations. Therefore, as CNN models and the convolution layer increase in complexity, much of the research towards improving performance has been focused on improving the computation of these convolution layers.

The operation of DNN models, like the CNN, commonly occur on specially designed hardware accelerators, that have been introduced to improve performance while lowering computational latency and energy overhead [4,24]. When implementing the compute-heavy convolution layer onto hardware, it is common to convert it into a general matrix-matrix multiplication (GEMM) operation [30] for ease of computing. This GEMM operation is composed of multiple vector-dot-products that must be computed, which can then broken down into several multiply-and-accumulate (MAC) operations. Therefore, these hardware accelerators that implement the GEMM operation require multipliers and adder units in order to compute the MAC operations. However, it is common for there to be a mismatch between the required number of multiplications and additions for a dot-product, and the number of multipliers and adders that the hardware provides. This leads to one of two situations. If the required number of multiplications and additions is less than the number of multipliers and adders provided, then the accelerator sees an underutilization of devices, as some still consume static power but are not currently in use. On the other hand, if there are less hardware resources provided than what is required, the MAC operation is broken down and folded over time into multiple temporal frames. Each frame produces a partial sum (psum) of the final output value of the dot-product. The hardware then needs to collect and accumulate these psums together in some form of reduction hardware that can add psums spatially from different multiplier units and temporally from different temporal frames. Therefore, the hardware accelerators that compute the GEMM operations require multiplier units, adder units, and some form of reduction hardware.

In addition to the required and provided hardware size mismatch, the size of the inputs in a convolution layer vary heavily between different layers and CNNs. As a result, the size of the vector-dot-products and number of psums that need to be accumulated also vary between layers. Therefore, hardware accelerators must be flexible, in order to only engage as many multipliers and adders as needed for the vector-dot-product. To answer this challenge, flexible spatial accelerators have been introduced [14, 23]. These accelerators solve the issue of variable dot-product sizes, by logically dividing the hardware into clusters, where each cluster is responsible for the computation of one vector-dot-product. The size of the clusters is set with the goal of increasing utilization of hardware resources by computing several MAC operations in parallel. Thus, by creating a unique cluster mapping per layer, flexible spatial accelerators are able to handle variable sized inputs and appropriately increase utilization of computing resources.

Although flexible spatial accelerators have proven to answer the challenges presented by mapping the convolution layer and subsequently GEMM operations onto hardware, they still pose performance challenges of their own. The part of these accelerators that is responsible for collecting and reducing psums is commonly referred to as the reduction network (RN). Current state-of-the-art RNs are typically composed of a network of electrical links and adders [14, 20, 23]. These electrical adders are typically limited to a fan-in of two or three, while the electrical links have a limited fan-in of just one channel per link. Therefore, within the RN, each accumulator is only able to add a maximum of three psums, with each psum requiring a separate link to arrive on. This limitation typically results in a tree-based topology of electrical links and accumulators that make up the RN. However, this topology leads to a large latency and hardware reduction complexity, that scales logarithmically with the number of psums to accumulate. In order to overcome this limitation, prior works have attempted to modify this tree-based topology to improve reduction performance [14,20]. However, as long as these electrical links and accumulators have a low limited fan-in, the RN will still require this costly tree-based topology. Thus, there is a research motivation to create a RN that can overcome these limitations and reduce the latency and hardware complexity, so that the network can scale more reasonably and improve performance.

In this work, our approach to answering these challenges is to utilize photonic interconnect links and devices to replace the electrical links and adders. It has been shown that the fan-in of photonic interconnects can be increased far beyond the limitation of electrical links by utilizing wavelength-division multiplexing (WDM). This property of photonic interconnects allows for a superposition of multiple information signals to stream together on the same waveguide, using different wavelengths [19]. Thus, the spatial parallelism of each photonic interconnect is improved and it's fan-in increased. In addition, it is possible to spatially and temporally accumulate optical information through an electro-photonic accumulating device that utilizes a photodetector, capacitor, and time-integrating receiver [27,31]. With multiple streams of incoming optical information arriving on a photonic interconnect with WDM applied, this accumulator can have it's spatial accumulation fan-in increased beyond the low fan-ins seen in electrical accumulators. In addition, this device is capable of holding accumulated information within itself, allowing accumulation to happen temporally without the need to store the intermediate results in memory. Thus, by utilizing photonic interconnects and the electro-photonic accumulator, we can spatially and temporally accumulate psums, while also improving the fan-in of both the links and accumulator beyond what their electrical equivalents are capable of.

We implement the benefits of photonic interconnects and accumulators through a photonic RN that we introduce in this work. Our RN answers the challenges present in GEMM accelerators, including the mismatch of dedicated hardware and input sizes which results in temporal folding, flexibility needed to create clusters for variable-sized inputs, and collecting and reducing psums through accumulation hardware. The creation of clusters is handled through new circuits we introduce, the reduction topology assemblers (RTAs). The RTAs implement the flexibility to logically divide the photonic RN hardware into clusters. In addition, the RTA converts digital psum values into the analog domain by transforming each psum into a sequence of optical pulses, each representing an 8-bit slice of the psum value. These psum optical pulses are routed by the RTAs to the electro-photonic accumulators, which we have dubbed photo-charge accumulators (PCAs). The RTAs are able to imprint all optical pulses on just two photonic interconnect waveguides, due to WDM. Each cluster has one PCA, which can add the incoming pulses in just one cycle, thereby vastly improving the fan-in of both photonic interconnect links and accumulators while spatially accumulating all psums in a cluster. In addition, the PCA holds intermediate results within itself, thereby allowing for a temporal accumulation of psums as well. Thus, by implementing photonic interconnects and accumulators, the photonic RN is able to vastly improve latency and hardware reduction complexity while still flexibly accumulating psums spatially and temporally.

Output results from the PCA require conversion from the analog domain back into the digital domain before they can be sent to memory. However, an additional challenge arises when trying to perform this conversion through an ADC. The bit precision of psum values vary, and the total output result can be up to 32-bits long. No ADC is capable of a 32-bit value conversion. Therefore, the PCA needs to answer this challenge somehow. To do this, before converting into the digital domain, the analog output value is split into two parts, each representing 16-bits of the final output value. By carrying this out first, the PCA is able to instead use two 16-bit ADCs to convert the output result, and then populate a 32-bit register using their results. Thus, the analog output value is successfully converted into the digital domain to be sent to memory

To analyze how the reduction complexity improves due to the improved fan-in of the photonic interconnect links and electro-photonic accumulators, we compared the photonic RN against five other state-of-the-art electrical RNs, taken from [20]. We observed that in terms of time complexity, the photonic RN is primarily dependent on the bit precision of the psums, since each 8-bits in a psum requires an additional optical pulse. The electrical RNs time complexity on the other hand are independent of bit precision, but rather scale with the number of psums to accumulate due to the low limited fan-in of the electrical accumulators. In terms of hardware complexity, the photonic RN requires a different channel per psum for WDM, but each channel only needs to be 1 bit wide since it is in the analog domain. The electrical RNs instead require a much larger number of channels due to their limited fan-in, and each channel must be the size of the psums' bit precision since they are in the digital domain. The electrical RNs also require more adders than the photonic RN, again due to the difference in fan-ins between the electrical and electro-photonic accumulators. To see how the different dependencies affect the latency and hardware energy cost of reducing partial sums, we analyzed how the photonic RN performs against the best electrical RN, STIFT, as bit precision and number of psums both vary. For the latency, we observed that although the photonic RN has small bumps in latency every 8-bits as a new optical pulse is output, compared to the continuous increase of STIFT's latency with number of psums, the photonic RN performs vastly better overall. For the hardware energy cost, we observed that the photonic RN has a continuous increase with the number of psums, and therefore the number of channels required. On the other hand, STIFT's hardware energy cost comparatively continuously increases largely with both bit precision and number of psums, due to it's link channel size and number of required adders having dependencies on those metrics respectfully. Overall, we observe that the improved fan-in from the photonic interconnect links and electro-photonic accumulator lead to improved latency and hardware reduction complexities for the photonic RN, even compared to the best performing electrical RN.

In order to compare our photonic RN against electrical RNs in real applications, we evaluated our RN against the three best performing electrical RNs from [20] while varying cluster size in a transaction level, event drive simulator over four DNNs: GoogLeNet, ResNet50, DenseNet121, VGG16. In terms of latency, we observed that due to the improved fan-in of the photonic links and accumulator which leads to a much lower reduction time complexity, the photonic RN achieved the lowest latency across all clusters and all tested DNNs. Speed ups range from 1.98× at cluster size 2 to 5.63× at cluster size 128 when looking at the geometric mean of the DNN's results. We also measured reduction energy-efficiency (FPS/W) and observed that at lower cluster sizes (2-4), the photonic RN performed slightly worse than the electrical RNs. At its worst it is only 0.95× as energy-efficient as the lowest performing electrical RN. This can be attributed to the photonic RN's higher overall power cost, resulting from photonic devices generally requiring a larger power consumption. At the lower cluster sizes this factor plays a larger role than the improved time complexity seen by the photonic RN. However, as cluster size increases, the photonic RN starts to perform better in terms of energy-efficiency. By cluster size 16, it outperforms all the electrical RNs, as they start requiring both higher reduction times, and a larger number of links and accumulators due to their low limited fan-in. At cluster size 128, when comparing the geometric mean of the four DNN's results, the photonic RN is 1.27× to 1.97× more energy-efficient than the electrical RNs. We lastly calculated the area of our photonic RN and found it to be 2.63mm^2, which is considerably larger than the electrical RNs even at a similar technology node. However, we performed a footprint-efficiency analysis in terms of FPS/mm^2 and found that although the photonic RN performs worse at lower cluster sizes of 2 and 4, from cluster size 8 onwards, the improvements of the photonic RN's lower reduction time complexity overtakes the additional area overhead, resulting in better results compared to all electrical RNs. Thus, we see that in terms of latency, energy-efficiency, and footprint-efficiency, although the photonic RN may sometimes performs slightly worse at lower cluster sizes, the benefits of the improved link and accumulator fan-ins can be heavily seen in the reduction time complexity as cluster size increases.

The rest of this paper is organized into different chapters. Chapter 2 goes over important background information pertaining to CNNs and their implementation onto hardware, flexible accelerator structures and operation, current photonic-enabled accelerator, and the motivation for why we need this work. Chapter 3 details the photonic RN at various levels of abstraction, starting with a network overview, then moving onto the RN structure with a focus on the RTA and PCA, and then finally presenting an example operation and mapping to help explain end-to-end communication of psum collection and accumulation. This chapter also presents an analysis of the time and hardware reduction complexities for our photonic RN and some other electrical RNs. Chapter 4 explains how we evaluated our photonic RN against other electrical RNs over multiple DNNs. We present a latency, energy-efficiency, and footprint-efficiency analysis in order to compare the networks and explain how the benefits of the photonic RN show through. Lastly, in Chapter 5 we conclude our work, briefly going through everything that was discussed, and also present some ideas for future work that out photonic RN can be extended to.

Copyright© Bobby Bose, 2023.

Chapter 2 Background

2.1 Introduction

In this chapter, we provide helpful background information on the current state-of-the-art implementation of convolutional neural networks (CNNs) on hardware, discuss the difficulties involved with this, and then present our ideas on how to solve these challenges. We first go into detail on the structure of CNNs, focusing on the important convolution layer and how its input and output data can be represented (Section 2.2). We then discuss how the computations of CNNs are commonly implemented onto hardware accelerators, specifically focusing on the common method of converting the convolution layer into a general matrix-matrix multiplication (Section 2.3). In this section, we also detail the main challenges that the accelerator must answer, such as varying input data size, and a mismatch of hardware between what the convolution operation requires and what the accelerator provides. We then introduce flexible spatial accelerators, which have been developed to try and solve these challenges (Section 2.4). Photonic-enabled DNN accelerators are introduced (Section 2.5), which also aim to improve the acceleration of CNNs and other deep neural networks. Lastly, we explain the limitations of current state-of-the-art flexible spatial accelerators, specifically focusing on their reduction networks, and introduce the idea of how photonic interconnects and devices can be used to overcome these limitations (Section 2.6).

2.2 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a prevalent form of DNNs and were developed to primarily be used for image, video, and audio analysis [26, 37]. Like other types of neural networks, the operation of a CNN typically consists of two steps: training and inference [30]. Training involves the use of a set or multiple sets of data to determine the values of network parameters based on a target objective function and output accuracy. After training the network, running it on new data is referred to as inference. CNNs consist of many layers, such as the convolution, fully-connected, and pooling layers, among others. However, the convolution layers appear the most frequently and tend to consume a large portion of the CNN's overall computations [10, 26].

As seen in Fig. 2.1a, a convolution operation involves two important sets of data: the input feature map (ifmap) and the weight filters. Both sets of data can be represented as 3D tensors, with the ifmap having a height H, width W, and number of channels C, and the weight kernels having a height R, width S, and number of channels C. It is also common to have multiple sets of filters, M, being applied to an ifmap, in order to test for different features of the input. The output result of the convolution operation, the output feature map (ofmap), has a height E, width F, and number of channels M.

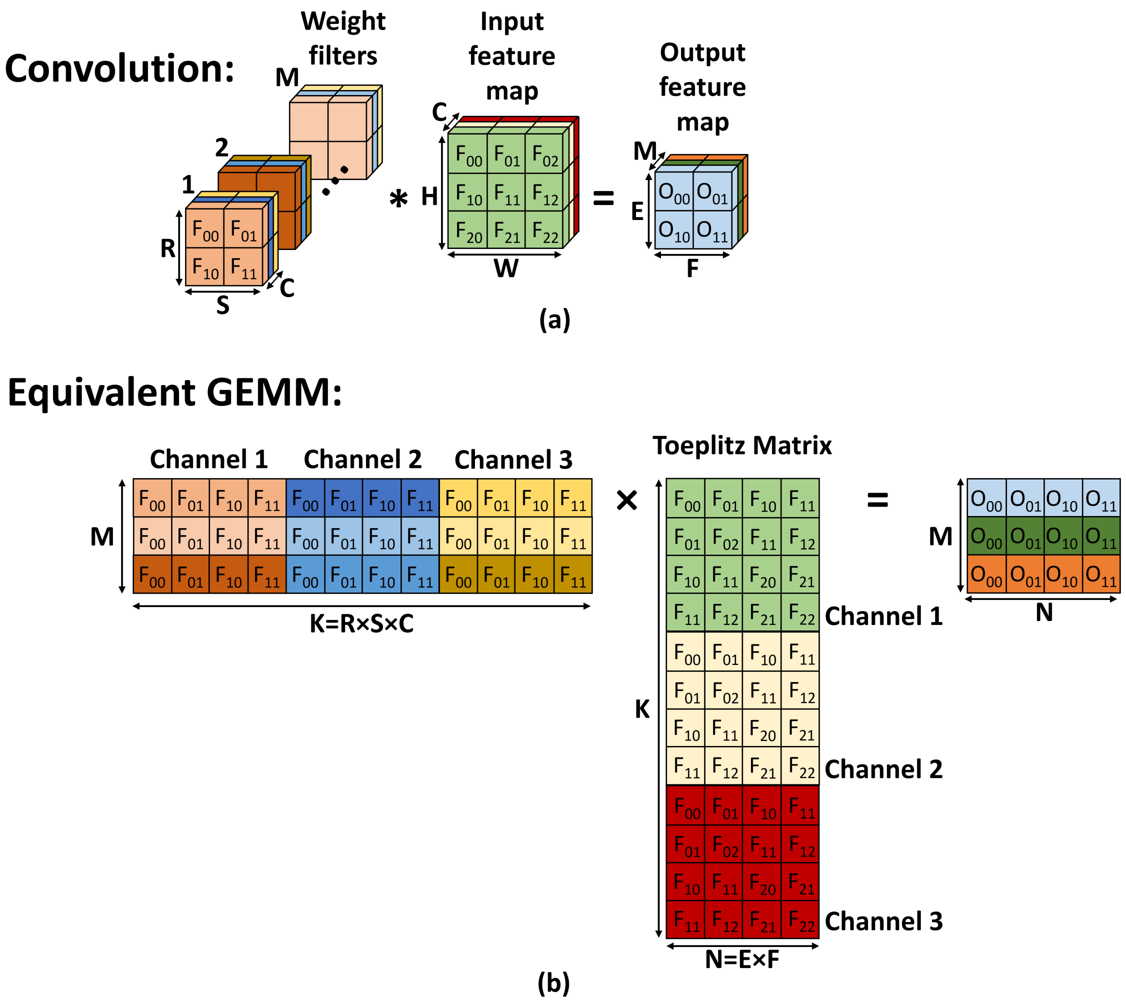

Figure 2.1: Mapping of a convolution operation (a) to an equivalent general matrix multiplication (GEMM) (b)

2.3 Hardware Implementation of CNNs

Since their inception, research has focused on improving the inference accuracy of CNNs/DNNs. However, this has come at the cost of CNN models increasing in size and complexity [30]. This cost has also caused the computational overhead of processing CNNs (in terms of computing latency, energy, and area) to increase to unsustainable levels. To continue to improve performance without also increasing computing latency, energy and area to unreasonable metrics, there has grown a research focus on improving the computational cost of these networks. Much of this on-going research has been focused on improving the performance of the compute-heavy convolution layer on hardware accelerators [18, 30].

The most common method to implementing the linear layers of computation in a CNN, like the fully-connected and convolution layers, is to convert them into a general matrix multiplication (GEMM) operation. To perform this conversion, both 3D input tensors are flattened into 2D matrices, as seen in Fig. 2.1b. Each channel of a weight filter is flattened into a vector and then appended together in order, producing a final weight matrix of width K=R×S×C. The height of the weight matrix, M, is equal to the number of weight filters being applied to the ifmap. The ifmap is converted into a Toeplitz matrix, whose values are set to reflect the multiplications and additions that occur in the convolution operation [30]. The input feature matrix has a width N=E×F, and height K, which matches the weight matrix width. As seen in Fig. 2.1b, in the equivalent GEMM form of the operation, each value of the output matrix is obtained through the vector-dot-product between a row of the weight matrix, and one column of the input feature matrix. This dot-product can be broken down into several multiply-and-accumulate (MAC) operations. The overall GEMM operation itself can then be thought of as a series of MAC operations that need to be computed.

To implement MAC operations, we require multipliers and accumulators in a hardware accelerator. A typical GEMM accelerator contains multiple multipliers and multiple adders connected together [4, 9, 11, 30]. To calculate all output features, we require $\mathrm{M} * \mathrm{N}$ dot-products of K size. To achieve this, the hardware needs to have $\mathrm{K} * \mathrm{M} * \mathrm{N}$ multipliers and accumulators. If the number of multipliers and number of adders available do not match with the multiplications and additions required for a vector-dot-product, then the accelerator needs to break down the MAC operation into partial sums (psums). For example, for a K sized dot-product, if we only had K/4 adders, we would need 4 psums to achieve the full dot-product. The psums need to be accumulated in an extra required structure that collects the psums and reduces them to the final output value. Thus, hardware accelerators for CNNs require multiple multipliers, adders, and reduction hardware that can reduce psums. The accelerator needs to support the flow of data through these employed multipliers, adders, and reduction hardware for psum accumulation.

2.4 Flexible and Re-configurable Accelerator Networks

Hardware accelerators typically implement multiple MAC operations by employing multiple multipliers and adders. The number of MAC operations that can be accelerated in parallel depends on how many multipliers and adders are available. Typically the number of multipliers and adders do not match the required amount. The MAC operations required to process one CNN layer often have to then be folded into multiple temporal frames. Each temporal frame creates a psum, which then need to be accumulated together to generate the final output. An example division of a vector-matrix multiplication into temporal folds is presented in Fig. 2.2a, where it is assumed the hardware dedicates only 2 multipliers and 2 adders for each MAC operation. Due to the mismatch of the number of dedicated multipliers and adders and the required amount for a MAC operation, the vector-dot-product that is needed to generate one output value in the output feature matrix needs to be folded into multiple temporal frames. Since there are twelve multiplications that need to occur, each MAC operation is folded into $12/2 = 6$ temporal frames, each generating a psum. The psums then need to be added together to generate the final output feature. The psums from each MAC operation in a temporal frame contribute to one partial output matrix. The sum of these matrices produces the final output matrix. Thus, to produce all output values, the multipliers and adders in the hardware need to be capable of computing multiple MAC operations in parallel, producing partial output matrices over multiple temporal folds which are added together to produce the output.

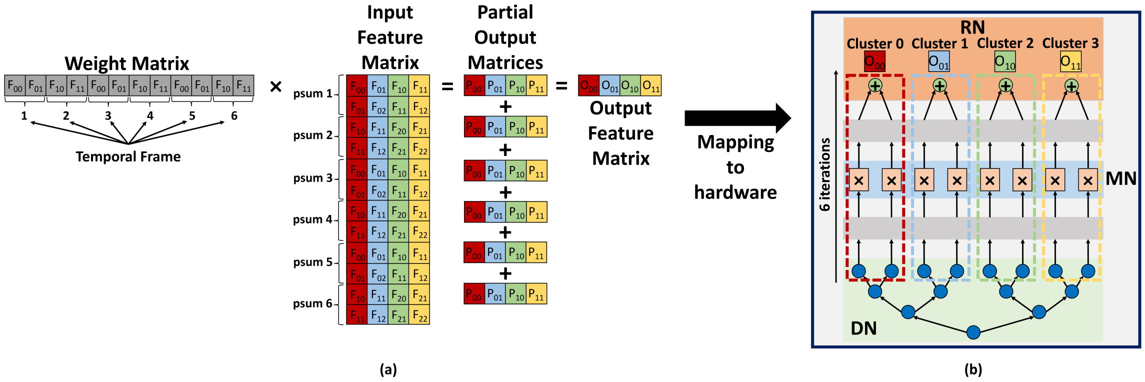

Figure 2.2: Example of cluster mapping a GEMM operation (a) onto a flexible DNN accelerator (b) over multiple temporal folds

The size of the matrices in the convolution operation vary across different layers of a CNN. As a result, the number of psums needed also change dynamically across different layers of the same CNN. It is imperative to design a flexible hardware that can support different sizes of matrices and a variable number of psums that must be accumulated together. This issue has been the subject of on-going research, with many flexible spatial accelerators having been proposed to answer this challenge [4, 14, 23, 24]. Flexible spatial accelerators are typically broken up into distinct networks, based on their focus of operation. As seen in Fig 2.3, these networks are normally referred to as the distribution network (DN), multiplier network (MN), and reduction network (RN). The DN is responsible for transmitting weights and input features stored in the global buffer to the MN. These values can be transmitted in a broadcast, multicast, or unicast manner, depending on how many multipliers need the same data. Typically, a tree-based network is employed for the DN to allow for the different types of data transmission. The MN is responsible for performing the individual MAC operations that make up the vector-dot-products. This network usually consists of a linear set of multiplier units, equivalent to the number of leaves in the DN tree-topology. The RN is responsible for collecting and accumulating the products of the MAC operations, or psums. As a whole, these three networks need to work together to fully compute the variable sized vector-dot-products needed to calculate output values across each layer of a CNN.

As the size of the matrices vary, and the number of psums needed change, the MN and RN need flexibility to only engage as many multipliers and adders as needed. When the size of the matrices is smaller than the number of multipliers, we see an underutilization as some devices go idle, resulting in a loss of parallelism and an

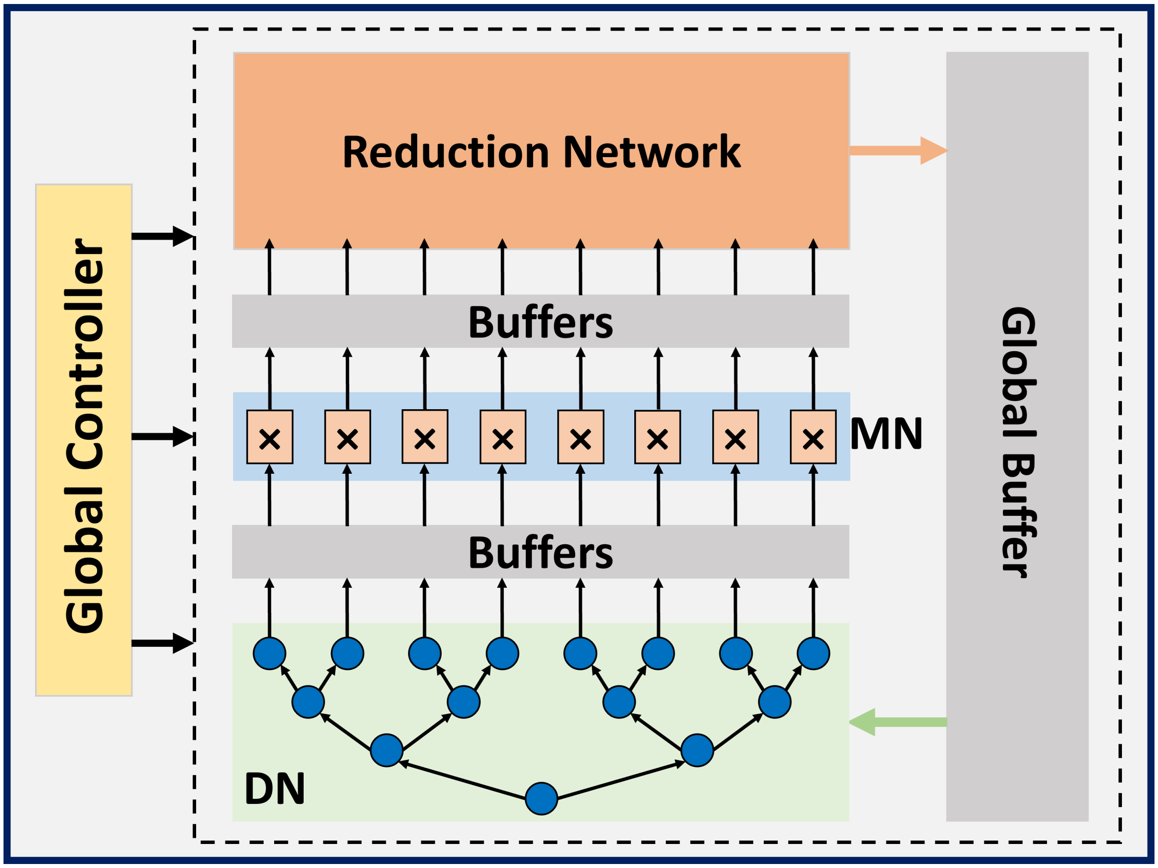

Figure 2.3: Typical architecture of a re-configurable DNN accelerator with a tree-based DN topology and a linear MN

inefficient static power consumption cost of the idle devices [4, 14, 23]. If the dot-product size is consistently larger than the multipliers, then the RN needs to be able to achieve temporal reduction of psums. Prior works have addressed these challenges by logically dividing the flexible spatial accelerator hardware into multiple clusters, with one dot-product mapped per cluster. Typically, clusters are made by creating unique memory footprints and communications patterns for each DNN layer, with the goal of increasing utilization of computing resources. An example of this is presented in Fig. 2.2b, where the vector-dot-product operations seen in Fig. 2.2a are mapped onto a flexible spatial accelerator hardware. The input and output matrices are colored by cluster to indicate how output values are calculated. As seen in Fig. 2.2b, in an accelerator with 8 multiplier units, a possible mapping is to dedicate two multipliers and their connected circuitry per cluster. In this setup, four dot-products are able to be computed in parallel. Since the number of multipliers does not match the required amount for each full MAC operation, the hardware will run for 12 multiplications / 2 multipliers = 6 iterations, creating 6 psums that are accumulated together to produce the final output value. Thus, the flexible spatial accelerator is able to utilize clustering to efficiently use all multipliers and compute multiple MAC operations in parallel with temporal folding. By achieving both spatial and temporal end-to-end flow of data, we see that flexible spatial accelerators are able to handle mixed sized GEMM operations.

2.5 Photonics Enabled DNN Accelerators

To continue to improve the performance of DNN accelerators and inference accuracy, various accelerators that utilize photonics have been proposed. Several photonic computing based accelerators have been introduced that handle the computation of the input and weight vector-dot-products with photonic devices. Albireo [25] is a photonic DNN accelerator that uses photonic devices for optical math in order to improve performance and broadcasting capabilities. SCONNA [31] aims to combine stochastic computing and photonics in order to improve the vector-dot-product operation. OXBNN [32], LightBulb [39], and ROBIN [29] are all photonic-based Binary Neural Network accelerators that convert CNN models to 1-bit precision in order to improve performance and inference accuracy.

In addition, there are also photonic-interconnect based accelerators that aim to overcome the high latency and energy limitations of metallic interconnects. Typically in these accelerators, the computation of psums is kept entirely in electrical processing elements, while the movement of data through the network is done through photonic interconnects. SPRINT [16], ASCEND [17], and SPACX [15] are all chiplet-based accelerators that utilize photonic interconnects in this way for it's distance-independent latency, and the ease of switching between different communication patterns.

2.6 Motivation

The RN it vital for the computation of output values, and consumes a large portion of computing energy and time [14, 20]. Due to this, there is strong research motivation to focus on improving performance of reduction in DNN accelerators [14, 20, 23]. Current state-of-the-art flexible spatial accelerator RNs are composed of a network of electrical links and adders, which handle spatial and temporal psum communication and accumulation from the multiplier units to memory [14, 20, 23]. In these flexible spatial accelerators, the electrical accumulators have a limited fan-in of up to two or three values maximum [14, 20]. In addition, electrical links are limited to one channel per physical link, resulting in a fan-in of only one. Due to these limitations, spatial reduction of only two or three psums is possible by one accumulator, with each psum needing to arrive on a separate physical link. The limited fan-in of electrical accumulators and links increases the required number of accumulators and links in the RN with logarithmic complexity, resulting in a tree-based topology of electrical links and accumulators. For example, the electrical RN $S-Tree$ from [20] requires $O(n-1)$ number of electrical accumulators and $O(2n-2+n/2-\log_2(n/2)-1 \times ReLU(1-\lfloor(n-1)/16\rfloor)$ number of electrical links to accumulate n psums. This large electrical accumulator and links complexity leads to a large energy, latency, and area overhead, furthering the motivation of on-going research to improve performance of the RN.

By studying photonic networks, it has been shown that photonic interconnects can serve as an alternative to electrical interconnects to overcome their inherent limitations. The fan-in of photonic interconnect links can be increased beyond the one channel per link limitation of electrical links, by utilizing wavelength-division multiplexing (WDM). WDM is a property of photonic interconnects that allows for a superposition of multiple optical data streams together on the same waveguide using different wavelengths. This enables spatial parallelism of data, increasing the fan-in to much larger metrics than possible in electrical links. In addition, it has been shown that we can spatially and temporally accumulate optical information through an integrated circuit utilizing a photodetector and time-integrating receiver (TIR) [27,32]. This photonic accumulator circuit is able to take advantage of the improved photonic link fan-in from WDM, increasing its own fan-in far beyond the small numbers electrical accumulators are capable of. Thus, by integrating both photonic interconnects and photodetector-TIR based accumulators in a flexible spatial accelerator, it is possible to improve performance beyond what electrical links and accumulators are currently capable of. In this work, we aim to leverage these benefits of photonic interconnects and devices to create a RN that does that.

Copyright© Bobby Bose, 2023.

Chapter 3 Photonic Reduction Network Architecture and Operation

3.1 Introduction

In this chapter, we discuss the photonic RN's architecture and overall operation of collecting and reducing psums from the MN to memory. First, we present a high-level overview of the whole network, detailing how our RN handles the challenges present in flexible spatial accelerator RNs (Section 3.2). Next, we go into further detail discussing the deeper structure of the overall photonic RN (Section 3.3). We focus on the two main circuits, the reduction topology assembler (Section 3.3.1) and photocharge accumulator (Section 3.3.2), that are used to answer the challenges presented in Section 3.2. After this, we explain the full psum collection and accumulation of the photonic RN, by presenting an example operation and cluster mapping (Section 3.4). We separate the description of the operation into three parts: psum signal conversion to the optical domain and routing to accumulators (Section 3.4.1), psum accumulation within the accumulators (Section 3.4.2), and conversion of the final output value back into the digital domain to send to memory (Section 3.4.3). Lastly, after the example we derive the O() complexities of our photonic RN, in terms of both psum reduction latency and hardware cost, and compare it to current state-of-the-art electrical RNs (Section 3.5). Both through the example and O() complexity presentations, we demonstrate how our photonic RN is able to collect and accumulate psums from the MN to memory, while displaying improvements over current state-of-the-art electrical RNs.

3.2 Network Overview

To address the shortcomings of current state-of-the-art flexible spatial accelerators, we propose a new photonic RN design, seen in Fig. 3.1, that implements the benefits of photonic interconnects and accumulators on top of existing flexible spatial accelerator structures. First, our RN needs to address the multiple challenges that flexible spatial accelerators already currently do. Since typically the available number of multipliers and adders for a cluster does not match the required amount to perform a MAC operation, our RN needs the ability to fold the MAC operations required for a CNN layer into multiple temporal frames. The psums generated from each frame need to then be summed together in some form of reduction hardware. In addition, as input matrix sizes change across CNN layers, our RN should only engage as many multipliers and adders as needed. To implement this, the RN needs to have capabilities to logically divide the network into clusters, allowing it to accelerate multiple MAC operations in parallel and efficiently utilize all computing resources. Therefore, for our RN to be able to compete with current state-of-the-art flexible spatial accelerators, we require it to be capable of temporally folding MAC operations in time, create clusters based on variable-sized matrices, and have hardware capable of accumulating the generated psums.

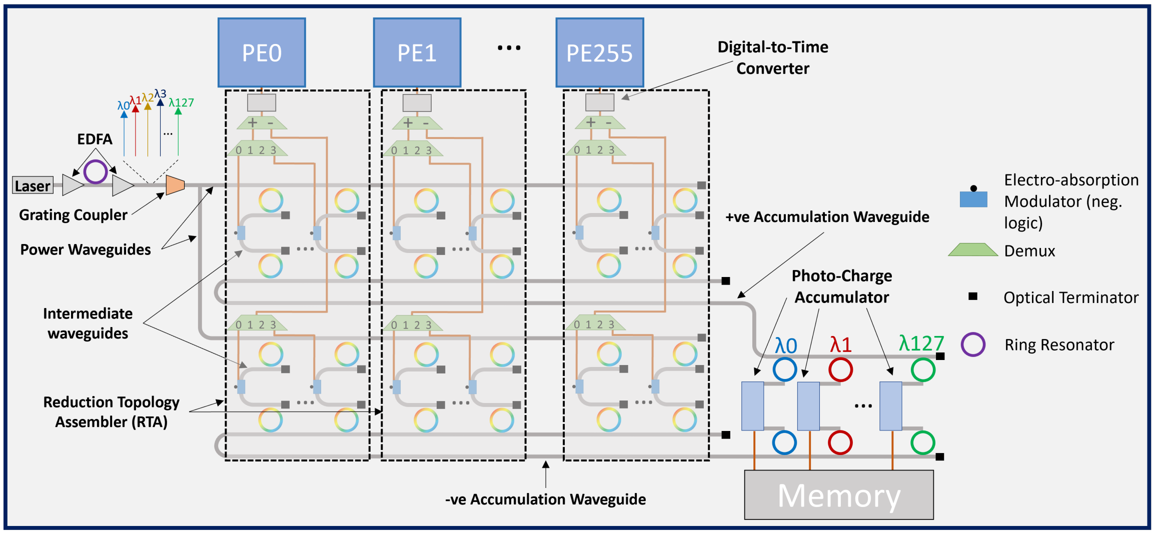

Figure 3.1: Architecture of a 256 PE, photonic interconnect based reduction network

Our photonic RN handles the required logical division of the network into clusters through a new circuit we introduce, the reduction topology assembler (RTA). Each processing element (PE) in the network (equivalent to multiplier units in the MN) requires an RTA, which all work together to create the logical clusters. Each cluster is assigned a unique wavelength, on which it's computed psums will be modulated onto. A comb laser source provides multiple wavelength carriers through a power waveguide which runs through the RTAs, providing multiple wavelengths to choose from for each cluster. The RTA converts the input digital psum value into 1 or more optical pulses, where the energy packetized in the pulse represents the psum value, and imprints this onto the selected carrier wavelength for the cluster. The optical pulses are routed onto a positive (+ve) or negative (-ve) accumulation waveguide, which both run through the RTAs. The accumulation waveguide the pulses are routed to is based on the sign of the psum value. Each active PE performs this task, resulting in a homodyne superposition of optical pulses representing all psums in the cluster for the current fold. We require only one +ve and -ve accumulation waveguide to accumulate all psums in all clusters due to WDM, which increases the fan-in of the accumulation waveguides far beyond the limited fan-in of only one that electrical links have. Thus, by selecting separate wavelengths per cluster and utilizing WDM, we are able to logically divide the network into clusters, while also keeping the number of required photonic interconnect links to a low minimum.

The photonic RN requires some reduction hardware that is able to accumulate psums spatially from different PEs, and also temporally over multiple temporal frames created by folding a MAC operation over time. We implement this through electro-photonic structures we introduce, named photo-charge accumulators (PCAs). Each PCA is an accumulator which has the capability to spatially accumulate the homodyne superposition of optical pulses representing psums in a cluster through a photodetector-TIR circuit [27, 32]. In addition, each PCA can temporally accumulate the psums that arrive in separate temporal frames by holding the partially calculated output value between folds. PCAs are able to overcome the limited fan-in of two or three seen in electrical adders due to each cluster only requiring one +ve and -ve accumulation waveguide to transfer all psums, and the PCA circuit being able to accumulate all incoming psums in one cycle. In short, together the RTAs and PCAs work together to provide end-to-end communication from the PEs to memory, are able to satisfy the requirements of flexible spatial accelerators to both spatially and temporally accumulate psums through multiple clusters and temporal frames, and improve on the shortcomings of current accelerators due to the limited fan-in of electrical links and adders.

3.3 Structure

In order to improve on the design of current state-of-the-art flexible spatial accelerators structures by integrating photonic interconnects and devices, we design our proposed photonic RN based on the interconnected accumulator structures seen in current flexible spatial accelerators [14, 20, 23]. Therefore, our RN is intended to be connectable with already existing spatial accelerator DNs and MNs. A model depicting how our proposed RN would look with 256 PEs can be seen in Fig. 3.1. As can be seen, our RN utilizes the RTAs and PCAs previously discussed, along with additional photonic interconnects (waveguides) and microrings (MRs), to create a end-to-end collection and accumulation network from the PEs to memory. By completing this path of data communication, our RN is able to perfectly fit in the typical flexible spatial accelerator structure seen in Fig. 2.3.

3.3.1 Reduction Topology Assembler (RTA)

The RTAs in our network serve three primary goals: transforming a digital psum signal into packetized optical energy, creating unique cluster mappings per CNN layer to parallelize MAC operations and efficiently use all computing resources, and routing signals from the PEs to the accumulators. The first task each RTA performs is the conversion of digital psum signals into an optical representation. To perform this, each active RTA reads in the digital psum value from its attached PE and sends it to a digital-to-time converter (DTC) [3], as seen in Fig. 3.2. This transforms the signal into a pulse train, where the total pulse length is proportional to the original psum value. Since DTCs have been shown to be able to operate in the low picosecond range [12, 13], with measured resolutions as low as 5 ps [3], the conversion from digital to time domain does not contribute significantly to overall network latency time. The generated pulse train feeds an electro-absorption modulator (EAM) [7,38], which is used to modulate optical power on a waveguide. The EAM uses negative logic and is by default powered on in order to not release and modulate any power onto the waveguide. The pulse train signal drives the EAM, allowing it to release power, modulating the optical power on the waveguide proportional to the time signal length. In order to keep the power released by the EAM to a minimum, we allow it to modulate power up to only 8-bit precision in a single modulated output pulse. To accommodate higher bit precisions of up to 32, which is seen with neural networks, the EAM outputs additional modulated pulses, each representing another 8 bits. Therefore for each psum, the EAM may output up to 4 packets of modulated optical energy to represent up to 32 bits. With this, the psum signal has been transformed from a digital value into packet(s) of optical energy.

In addition to converting psums into an optical representation, the RTAs are also responsible for logically dividing the network into clusters, where each cluster is responsible for the computation of one MAC operation. Cluster mappings are created uniquely for each CNN layer, in order to efficiently utilize all computing resources and minimize wasted static power consumption from unused devices. For a model with X PEs, cluster sizes can ranges from 2 PEs/cluster to X/2 PEs/cluster. For example, in the model depicted in Fig. 3.1 with 256 PEs, the cluster size can range from 2 to 128. To separate these clusters in the network, we utilize WDM to improve the fan-in of the photonic interconnect links. This allows us to provide all potentially needed wavelengths through one power waveguide from the off-chip comb laser. Hence, by logically dividing the network into clusters by using separate wavelengths and WDM, we can parallelize multiple MAC operations and efficiently use all computing resources, without needing to increase the number of required photonic interconnect links to the numbers seen in electrical RNs.

In order for the RTAs to separate clusters by wavelength, they need to also route psum signals through their structures and to the accumulators correctly, in order to properly utilize each cluster. There are many routing challenges that the RTA answers to accomplish this. Due to the inability to represent psum signage in packetized optical energy, the photonic RN requires two paths of psum flow to the accumulators, to carry positive (+ve) and negative (-ve) psums. Within the RTA, this means that the circuitry to convert the pulse train to optical energy needs to be doubled, as seen in Fig. 3.2. We refer to the two paths of psum flow as the +ve and -ve tracks. The output of the DTC feeds a demultiplexer, which routes the pulse train to either the +ve or -ve track of the network based on the psum signage. To ensure that optical power is only modulated on the correct wavelength for the psum's cluster, a MR first modulates out a portion of the optical power from the power waveguide at the specified wavelength onto an intermediate waveguide, which the EAM is placed on. All MRs have a limit to the number of wavelengths they can tune to. No MR currently proposed has the capabilities to tune to the possible 128 wavelengths required. To solve this issue, we cascade multiple MRs together, who in total can tune to 128 possible wavelengths. It has been shown that a channel gap of 0.2 nm is possible for MRs [22]. With a reasonable tuning range of 6.4 nm, each MR is able to be tuned to 32 unique wavelengths through thermal and electro-optic tuning [5], as seen in Fig 3.3. We would then need 4 MRs in a set to cover the necessary 128 wavelengths required for this model. In this case, the first MR would be tunable to $\lambda 0 - 31$, the second to $\lambda 32 - 63$, the third to $\lambda 64 - 95$, and the fourth to $\lambda 96 - 127$. In addition to four MRs, the RTA also requires four intermediate waveguides and four EAMs per track. A second demultiplexer is used to route the psum pulse train representation to the EAM, whose corresponding MR can be tuned to the current cluster's specified wavelength. Another MR couples the EAM output

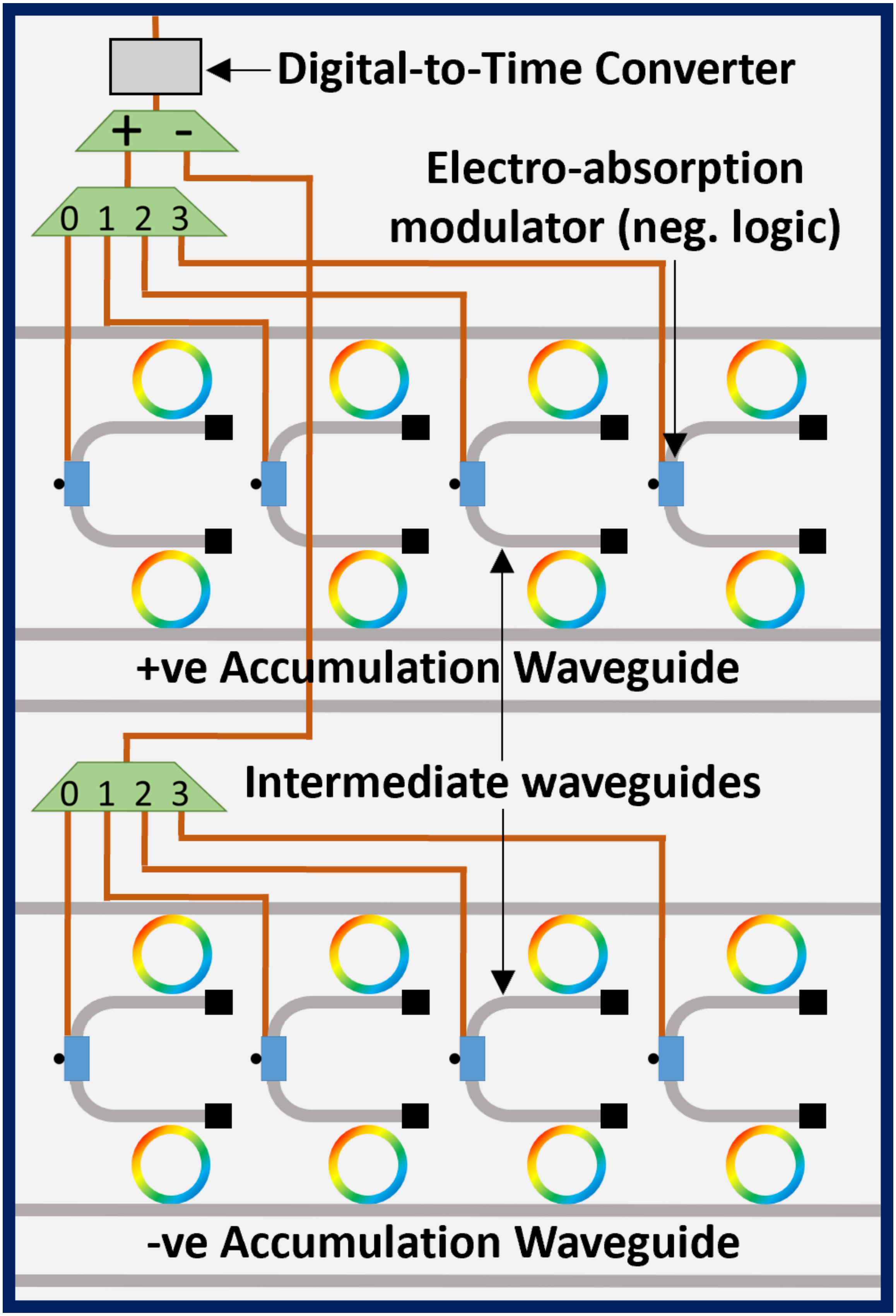
Digital-to-Time Converter
+
-
Electro-absorption modulator (neg. logic)
0 1 2 3
+ve Accumulation Waveguide
Intermediate waveguides
0 1 2 3
-ve Accumulation Waveguide

Figure 3.2: Reduction Topology Assembler used to select carrier wavelength, convert input into optical pulses, and route to an accumulation waveguide

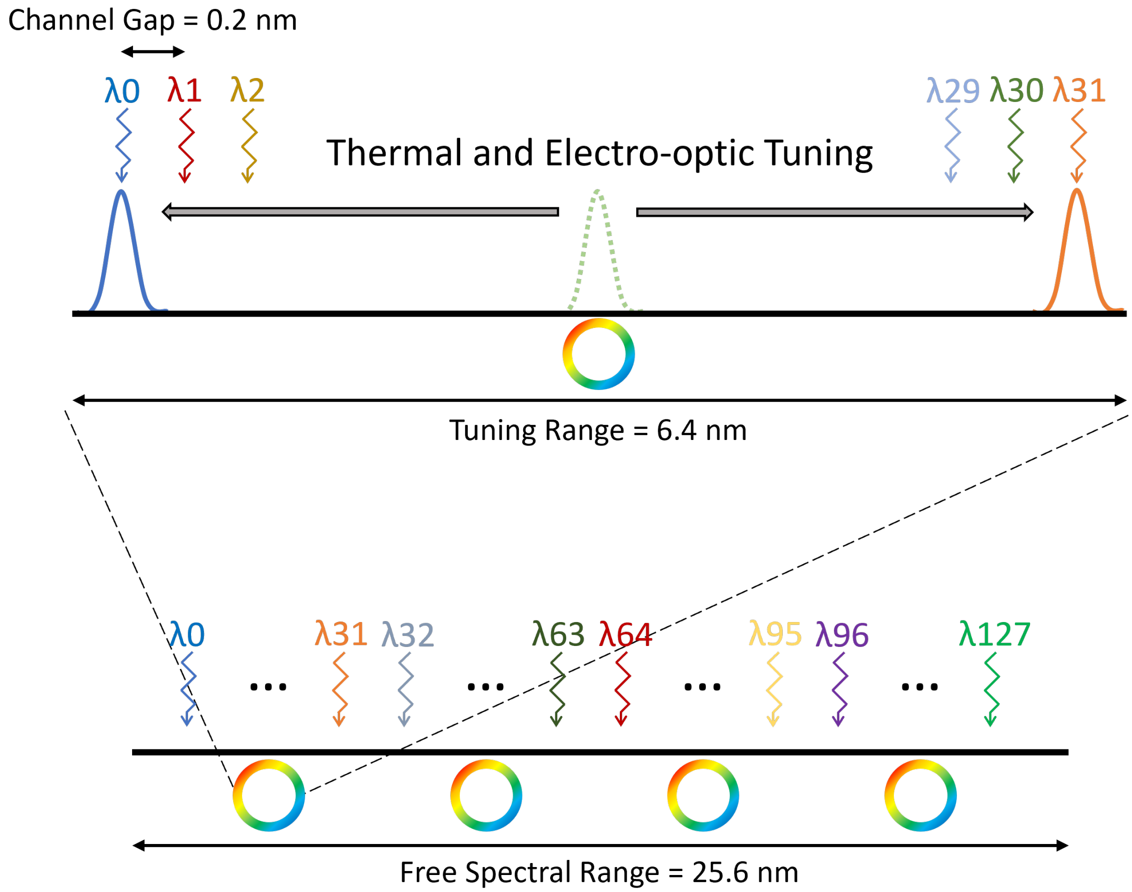

Figure 3.3: Free Spectral Range of 4 MRs cascaded together to cover 128 wavelengths. Each MR can select from 32 possible wavelengths.

onto an accumulation waveguide, which is used to finish routing the psum signal to the accumulators. Each PE in a cluster modulates its psum onto the accumulation waveguide, thereby separately accumulating the total positive and negative psums as a homodyne superposition of optical pulses. By utilizing WDM to separate clusters, the photonic RN requires only one accumulation waveguide for each track to carry all psums to the accumulators. Thus the RTAs are successfully able to convert all psums into packetized optical energy, separate them by clusters, and route all data from the PEs to the accumulators.

3.3.2 Photo-Charge Accumulator

To improve on the limited fan-in of electrical accumulators in typical flexible spatial accelerators and also incorporate temporal folding of MAC operations, our RN uses electro-photonic accumulator structures, called PCAs, to accumulate psum values. Each cluster in the network only requires one PCA to accumulate all psums in its MAC operation, due to the structure's increased fan-in. Therefore, there are a number of PCAs in the hardware equal to the maximum number of clusters, or potential carrier wavelengths. To collect the psums from a cluster on the accumulation, the total positive and negative psums are filtered out by ring resonators to the PCA corresponding with the specified wavelength for the cluster, as seen in Fig. 3.1. The PCA uses two photodetectors to detect the incoming homodyne superposition of optical pulses representing the psums on the +ve and -ve accumulation waveguides. During each pulse of incoming power, the photodetector outputs a current. The two photodetectors generate currents that are directed in opposing directions, allowing the positive and negative psums to accumulate together. As each psum may be represented as up to four pulses of packetized optical energy, the EAMs in the RTA are calibrated so that the same pulse of optical energy from each RTA in the cluster arrives at the PCA simultaneously, thereby accumulating photons on the photodetector during the same interval and contributing to the same current output. Each of the four current outputs is routed through a demultiplexer and drives a separate capacitor, storing a voltage proportional to the current pulse length. A time-impedance receiver (TIR) connected to each capacitor is used to fully capture the pulse if it's time length exceeds the inverse bandwidth of the photodetectors [27]. In addition, the stored partial output value can be held in this capacitor-TIR circuit, allowing the PCA to accumulate psums over multiple temporal frames [6]. Thus, the photonic RN requires only one PCA per cluster to accumulate all psums over all temporal folds of the MAC operation.

Although the incoming pulses of packetized optical energy for each psum represent different 8-bit parts of the whole psum representation, the bit position weight of the number is not present in the modulated power. For example, the second pulse represents bit positions 8-15 of the psum and should therefore have a weight of 2^8. To correct this, the PCA must apply these bit weights before the individual 8-bit voltage representations can be summed together. The four modulated pulses representing the psum have weights 2^0, 2^8, 2^{16}, and 2^{24} for the least to most significant 8 bits respectively. To apply this weight, the capacitance on the capacitors storing the 8-bit voltages is set based on the equation for standard parallel plate capacitors, $Q = CV$. Since capacitance and voltage have an inverse relationship, the weights to apply to the capacitance flip for the least to most significant 8 bits. Therefore, the capacitance for the capacitor for the least significant 8 bits is weighted $2^{24}\times$ more than the capacitance for the capacitor storing the most significant 8 bits. This is depicted in Fig. 3.4. Although not depicted in the figure, the capacitor storing voltage for bits 8-15 and bits 16-23 would have capacitance $2^{16}\times C_0$ and $2^8\times C_0$. Once all psums for the cluster's MAC operation are received, the PCA sums the individual voltages together using an analog op-amp voltage adder. Therefore, by manipulating the capacitors storing the psum voltages for a cluster, the PCA can create an accurate voltage representation of the final output value.

Before the total output voltage can be converted back into a digital value to be sent to memory, additional voltage manipulation must be performed. Currently, it is impossible to have an ADC with 32-bit precision on an integrated CMOS chip. To resolve this issue, the PCA breaks down the total accumulation result into its upper and lower 16 bits, reduces the weight of the voltage representing the upper 16

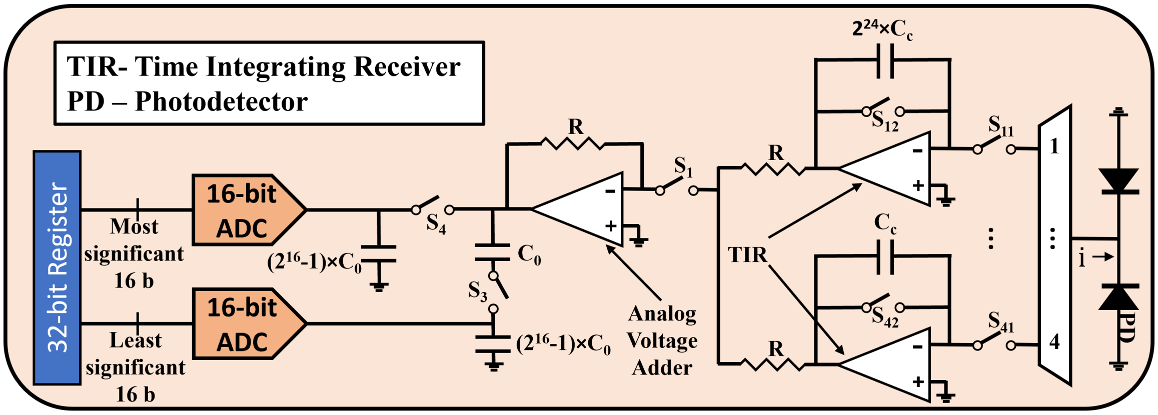

Figure 3.4: Photo-charge accumulator architecture

bits, and then uses two 16-bit ADCs to convert the result and populate one 32-bit register. A closer look at the voltage split and reduction circuit designed for a simpler 4-bit value is explained in more detail with a dedicated figure in Section 3.4.3. This circuit is driven by the output of the op-amp adder. The op-amp adder generates a constant short circuit current of 20 mA and is capable of driving a capacitive load of 500 pF. Designing a capacitor or capacitive circuit to be under 500 pF is within reason, and therefore no additional circuitry is required for the adder output. Once the voltage split and reduction is complete, two 16-bit ADCs then read in the upper and lower voltages, and populate the upper and lower 16-bits of a 32-bit register with their result respectively. Thus, the final output value produced from the summing of all psums in the cluster over all temporal folds is recorded in memory, and the computation of the MAC operation is over.

3.4 Operation and Example Mapping

To better explain the end-to-end communication and operation of the photonic RN, an example mapping and operation is presented. Consider the hardware setup seen in Fig 3.1, with 256 PEs. To create clusters, first the optimal cluster size must be determined in order to parallelize the proper amount of MAC operations to reduce the number of unused devices, which still consume static power. If 4 PEs per cluster is determined to be the optimal mapping, then we would have 256 total PEs / 4 PEs per cluster = 64 clusters mapped on the RN. As seen in Fig 3.5, the first four PEs (0-3) would belong to cluster 0, the second set of four PEs (4-7) would be mapped to cluster 1, etc. A unique wavelength would be assigned to each cluster ($\lambda 0$ for cluster 0, $\lambda 1$ for cluster 1, etc.). The first 64 PCAs, whose MRs are tuned to $\lambda 0 - 63$, are assigned to clusters 0-63 respectively. Together, these choices provide the RTAs and PCAs the necessary information to divide the network into the logical clusters.

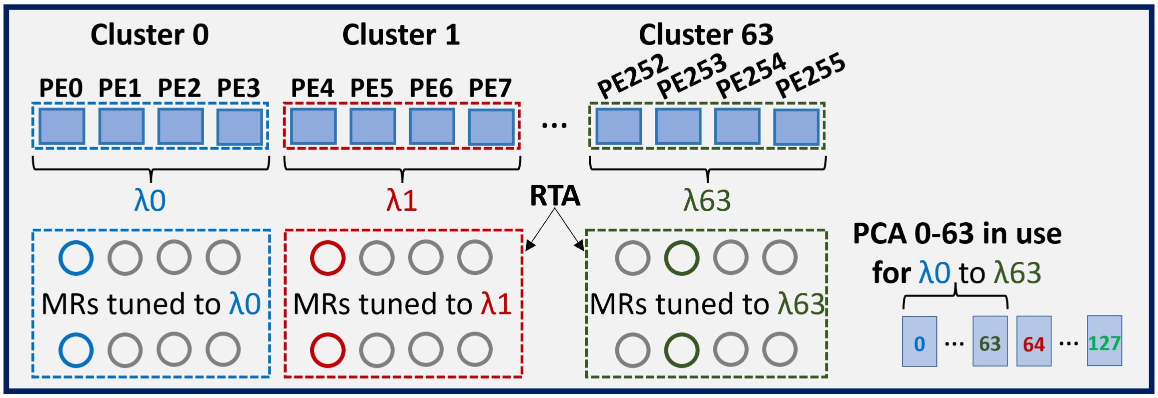

Figure 3.5: Example of mapping cluster size = 4 in a 256 PE photonic RN model

3.4.1 Partial Sum Signal Conversion and Routing

The psums output from the PEs first need to be sent through the RTAs to be converted into packetized optical pulses, and routed to the correct waveguides on the right wavelengths. For ease of explanation, we will focus on cluster 0. To accumulate the psums in this cluster, first PE0-3's MR that is tunable to $\lambda 0$ in the first MR set will modulate out a portion of the optical power from the power waveguide on that wavelength onto the MR's intermediate waveguides. This occurs in both the positive and negative tracks of the RN. Based off the device parameters discussed in Section 3.2.1, the first MR would be responsible for $\lambda 0$. Next PE0-3 sends their psums through their RTA DTCs to generate a pulse train each, where the number of pulses is proportional to the absolute value of the psum. For example, if PE0 had a psum of 6, PE1 had a psum of -4, PE2 had a psum of -3, and PE3 had a psum of 8, then their DTCs would generate pulse trains of 6, 4, 3, and 8 pulses respectively, with quantization value T_Q length per pulse. After the pulse trains are generated, the first demultiplexer routes the pulse trains to either the positive or negative path of the RTA, based on the psum polarity. For this example that means PE0 and PE3's pulse trains would be routed to the +ve track, and PE1-2's pulse trains would be routed to the -ve track. Another demultiplexer then routes this signal to the EAM on the intermediate waveguide connected to the active MRs for this mapping. In this example, the +ve track's demultiplexer routes PE0 and PE3's signals to the positive intermediate waveguide connected to the first MR, and the -ve track's demultiplexer routes PE1-2's signals to the negative intermediate waveguide connected to the first MR. The output signals from the DTCs drive the EAMs, modulating the power on $\lambda 0$ on the intermediate waveguides proportional to the time signal lengths. In this example, we will assume PE0 and PE2's psums are between 9-16 bits long, and PE1 and PE3's psums are between 1-8 bits long. That means PE0 and PE2's EAM will output two pulses of packetized optical energy to represent their psums. On the other hand, PE1 and PE3's EAM will only need to output one pulse of packetized optical energy to represent their psums. At the end of each intermediate waveguide is another tuned MR, from the second set of MRs, that transfers the power on $\lambda 0$

to the +ve or -ve accumulation waveguide. In this example, the modulated power representing PE0 and PE3's psums is coupled and superpositioned together on the +ve accumulation waveguide, and the modulated power for PE1-2 on the -ve accumulation waveguide. With this, the RTAs for this cluster have finished converting the input psums to packetized optical energies, and have routed these signals to the correct accumulation waveguides.

3.4.2 Partial Sum Accumulation

Now that all psums in the cluster for this temporal frame have been routed to the accumulation waveguides, the associated PCA for the cluster must accumulate them all together. To do this, first, further down the accumulation waveguides, the MRs tuned to $\lambda 0$ would pick up the optical power on the two accumulation waveguides and route it to a PCA. The photodetectors in the PCA are driven by the positive and negative accumulated psum signals, and output a current during each pulse. The positive and negative psum currents, which have opposing directions, combine, adding the total positive and negative psums together. As seen in Fig. 3.6a, the psum signals from each PE in the cluster contribute to the combined output current. Since PE0 and PE3 had positive psums, their combined photodetector currents are positive, whereas PE1-2's combined photodetector currents are negative due to their psums being negative. The incoming psums are represented as up to 16-bits in this temporal frame for this cluster, and therefore the PCA requires two periods X_1 and X_2 to read in both pulses, each representing 8-bits of the psums. The current output from each period drives a separate capacitor-TIR circuit. As seen in Fig. 3.4, the top two capacitor-TIR circuits are required to store voltages representing the first and second 8-bit parts of the accumulated psums. During X_1, the pulses representing the first 8-bits of the psums arrive, and the capacitor for TIR0 is charged, with TIR0 maintaining an output voltage representative of the accumulated 8-bit part of the psums, as seen in Fig. 3.6b. While accumulating, we close S_{11}, allowing the current to accumulate charge on the capacitor. The second pulse representing the next 8-bits arrives during X_2, and drives TIR1 and it's capacitor. While this is occurring, we reopen S_{11} and close S_{21}, allowing charge to accumulate on the next capacitor. The voltage maintained by TIR1 is weighted $2^8\times$ the voltage in TIR0, since this represents bits 8-15. Note that since PE1 and PE3's psums were less than 9 bits, they did not contribute to this second voltage. With this finished, the psums in this cluster for this temporal frame have now been successfully accumulated by the PCA.

Although temporal frame 1 has concluded, before the final output value can be achieved, the PCA still needs to accumulate psums from the remaining temporal frames for this cluster. After temporal frame 1, the cluster moves onto temporal frame 2. In between temporal frames we open S_{11} and S_{21}, and only close it when new psums arrive. This allows the capacitor-TIR circuits to hold the partially accumulated values. During temporal frame 2, PE0-3 potentially output new digital psums values to their RTAs, following the same procedure detailed in the previous section. In this example, let's assume PE0-2 all output new psums up to 16 bits long during temporal frame 2. These new psums eventually lead to new combined photodetector

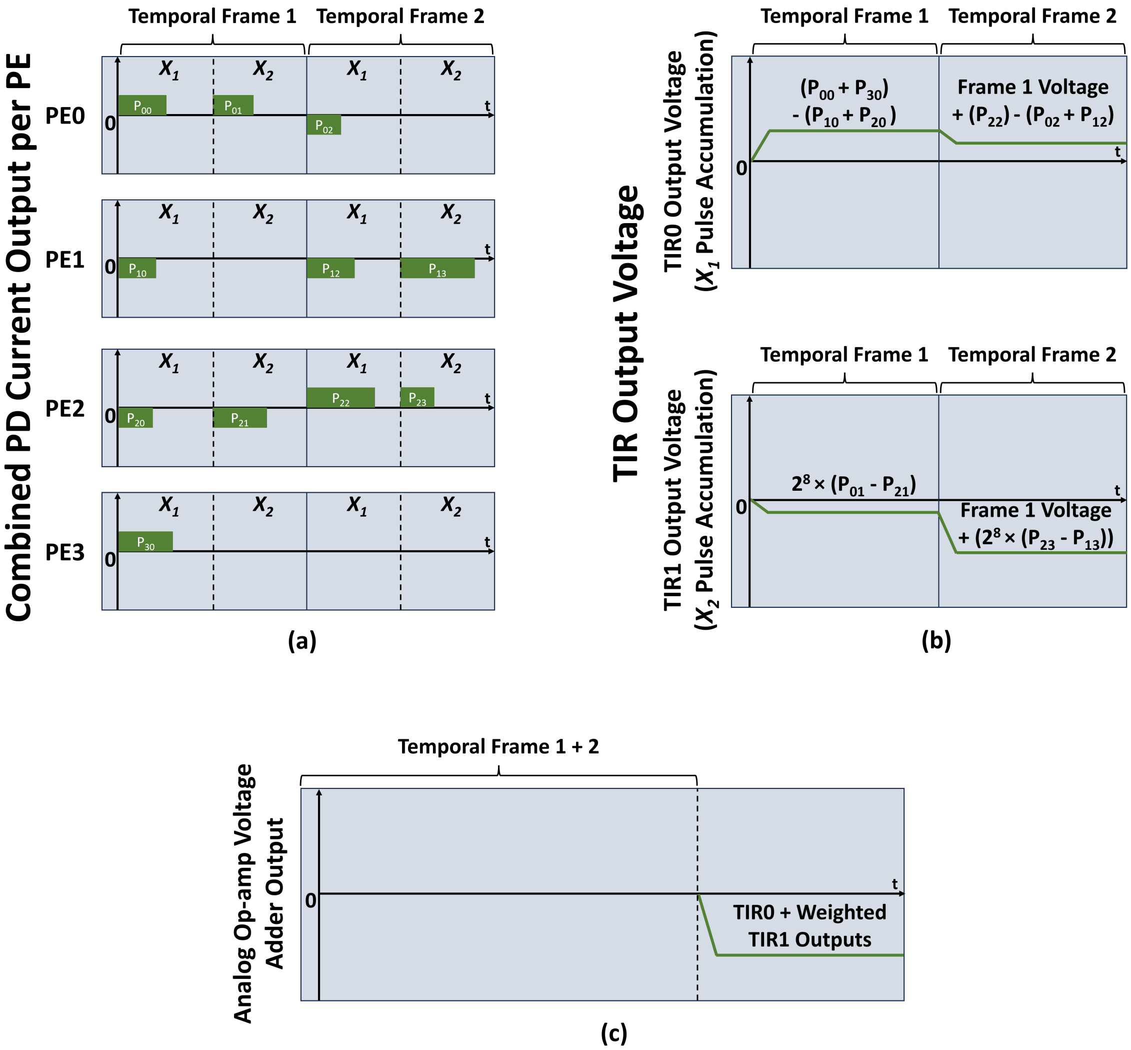
Combined PD Current Output per PE
Temporal Frame 1
Temporal Frame 2
X_1
X_2
PE0
PE1
PE2
PE3
0
t
P_{00}
P_{01}
P_{02}
P_{10}
P_{12}
P_{13}
P_{20}
P_{21}
P_{22}
P_{23}
P_{30}
(a)
TIR Output Voltage
TIR0 Output Voltage (X_1 Pulse Accumulation)
$(P_{00} + P_{30}) - (P_{10} + P_{20})$
Frame 1 Voltage $+ (P_{22}) - (P_{02} + P_{12})$
TIR1 Output Voltage (X_2 Pulse Accumulation)
$2^8 \times (P_{01} - P_{21})$
Frame 1 Voltage $+ (2^8 \times (P_{23} - P_{13}))$
(b)
Temporal Frame 1 + 2
Analog Op-amp Voltage Adder Output
TIR0 + Weighted TIR1 Outputs
(c)

Figure 3.6: Example of the PCA accumulating psum signals from a 4 PE sized cluster over two total temporal frames. (a) Combined photodetector (PD) output current per PE for two temporal frames. Period X_1 captures the first pulse of a psum signal, and period X_2 captures the second pulse. Psum values are between 1-16 bits in example. (b) Voltage output of the two active TIRs during both temporal frames. TIR0 captures the pulse representing the first 8-bits, and TIR1 captures the pule representing the second 8-bits. (c) Total voltage output from the analog op-amp voltage adder after both temporal frames. Not to scale with TIR outputs.

output currents, as seen in Fig. 3.6a. These currents will again drive TIR0 and TIR1, accumulating charge on their capacitors and modifying the held voltage, as seen in Fig. 3.6b. Once accumulation in this temporal frame is concluded, the cluster will move onto the next frame. In this example, we will assume their are only two temporal frames needed to fully compute the dot-product. In this case, once temporal frame 2 has concluded, the PCA moves onto summing the individual accumulated psum parts. To do this, first S_{12}, S_{22} and S_1 close, and we ensure that S_{11} and S_{21} are open. This connects the active capacitor-TIR circuits to the analog op-amp voltage adder, summing the weighted voltages together. As seen in Fig. 3.6c, the voltage stored in the adder represents the final output value summed from all psums over every temporal frame for this cluster. Thus, psum accumulation has concluded for this cluster.

3.4.3 Output Division and Reduction

Although the output of the op-amp adder represents the final output value, it must first go through the voltage split and reduction circuit to divide the output into its upper and lower 16 bits before it can be sent through ADCs and back to memory. An example operation of the split and reduction circuit can be seen in Fig. 3.7. For ease of demonstrating, let us initially assume the op-amp adder output actually only represents a 4-bit value instead of 32-bits, and that the ADCs can only handle 2-bit precision. To first create the split in voltage, we must determine how many voltage levels are present in a 4-bit value. To do this, we look at how many numbers can be represented with 4 bits, or simply take $2^4 = 16$. The lower 2 bits of this number represent $2^2 = 4$ of these voltage levels, and therefore the upper 2 bits represent the remaining $2^4 - 2^2 = 12$ voltage levels. To split the voltage, we need to divide it by this ratio of $12 : 4 = 3 : 1$. We use series capacitors to perform the split, as seen in Fig. 3.7 connected through S_3. Capacitors in series share the same charge Q, and split their voltage based on their capacitances. Since capacitance and voltage have an inverse relationship, the split ratio is flipped, becoming $1 : 3$ respectively. When splitting the voltage, we open S_1 and close S_2 and S_3, allowing the output of the adder to charge both capacitors. The top capacitor fills with a voltage representing the upper 2 bits, while the bottom capacitor's voltage represents the lower 2 bits. If we assume a 5V adder output, the series capacitors will split the voltage into 3.75V and 1.25V for the upper and lower 2 bits, as seen in Fig. 3.7b. These two voltages now proportionally represent the upper and lower 16-bits of the output value, completing the voltage division part of this circuit.

Before the upper 2 bit's voltage can be sent through an ADC, it's bit position weight needs to be removed and the voltage scaled down through the voltage reduction part of this circuit. For a 4-bit number, the upper 2 bits have a weight of $2^2 = 4$. We perform the voltage reduction through parallel capacitors, seen in Fig. 3.7 connected by S_4. When a capacitor is placed in parallel with an already existing capacitor, the shared voltage over them reduces by an amount equivalent to the sum of their capacitances. Therefore, to reduce the voltage representing the upper 2 bits by 2^2, we use parallel capacitors, whose added capacitance is $2^2\times$ the original capacitor. As

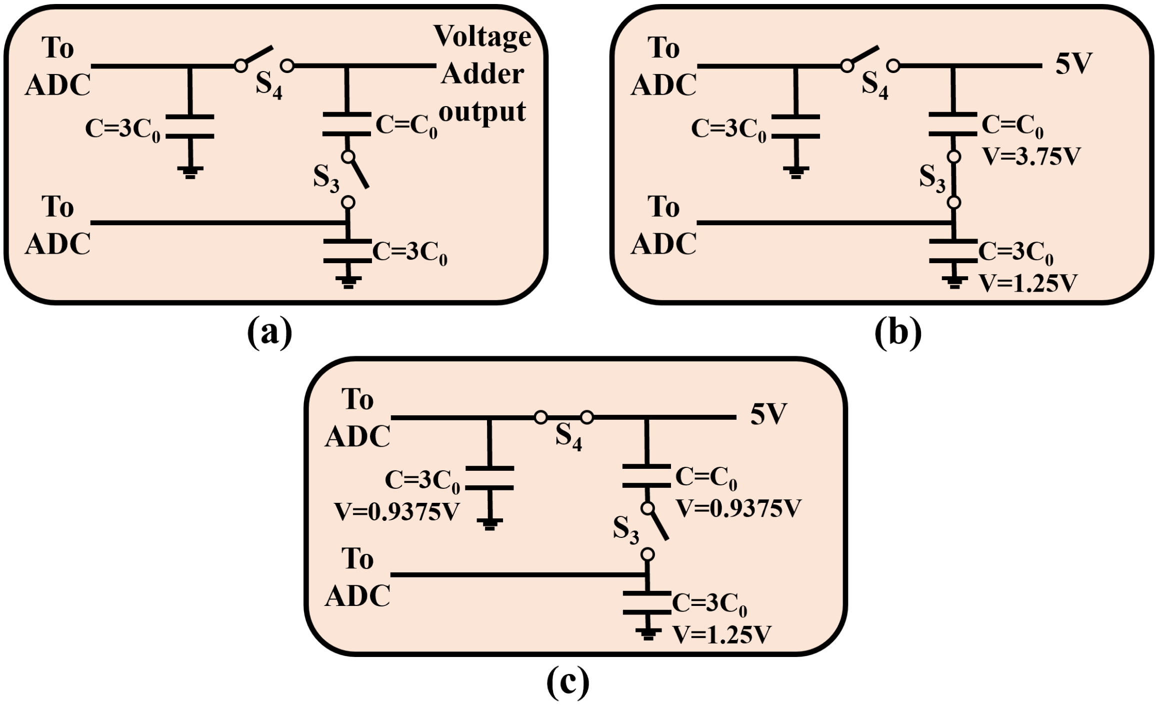

Figure 3.7: Example operation of the voltage division and reduction circuit in the PCA. Circuit depicted is set up to split a 4-bit TIR output value into two 2-bit parts. (a) No current flow when both switches are open. (b) Voltage division of voltage adder output when S_3 closes. (c) Voltage reduction of the upper 2 bits' voltage when S_3 opens and S_4 closes.

seen in Fig. 3.7c, this results in another capacitance ratio of 1 : 3 which reduces the voltage from 3.75V to 0.9375V. While this reduction is occurring, we open S_2 and S_3 and close S_4. Once the voltage reduction is done, both upper and lower voltages are read by ADCs, which then populate the upper and lower 2 bits of a 4-bit register with their outputs, thus completing the voltage division and reduction of the initial 4-bit value.

The same procedure used to calculate the 4-bit split and reduction can be extended to a 32-bit value. As seen in Fig. 3.4, this results in a split ratio of $1 : 2^{16} - 1$ for the upper and lower 16 bits, and a total reduction capacitance of 2^{16} over the parallel capacitors setup. After splitting and reducing the output value into it's upper and lower 16 bits, both ADCs will read in the value and convert it into a digital representation. These two 16-bit values will then be sent to memory, populating the upper and lower 16 bits of a 32-bit register. With this complete, the end-to-end communication and operation of psum collection and accumulation for cluster 0 is finished. While this is occurring, clusters 1-63 will also be undergoing the same process through the RN, using different wavelengths and PCAs as discussed. The same +ve and -ve accumulation waveguides are used for all parallel accumulations, since we employ WDM and each cluster uses a unique wavelength. Once computation of this CNN layer is complete, a new cluster mapping will be determined for the next layer.

3.5 Reduction Complexity

As seen in Table 3.1, the photonic RN exhibits a psum reduction time complexity of $O(T_p \times 2^8 \times \lceil b/8 \rceil \times i)$, where T_p is the photonic cycle time, b is the bit precision, and i is the number of folds needed to complete psum accumulation for the output value. T_p remains consistent throughout testing, at a set 5 ps. The primary contributing factor to changes in reduction time in the photonic RN is the bit precision of the psums. As discussed prior, the EAM is able to support up to 8-bit precision psums in one modulated output pulse. In order to support higher precision psums, the EAM outputs an additional modulated pulse, increasing the reduction time. The reduction complexities of several electrical RNs from [20] that we evaluate against can be seen in Table 3.1. In contrast to the photonic RN, the electrical RN's reduction complexities are independent of bit precision, but instead are heavily impacted by the number of psums to accumulate, n. In addition, as these RNs are not photonics-based, they operate at a different clock speed, with a electrical cycle time, $T_e = 1.25$ns.

To analyze the reduction time complexity, we compare our photonic RN against the best performing electrical RN from [20], STIFT. As seen in Fig. 3.8, we compare both networks latency to accumulate a number of psums (n) while sweeping the bit precision (b). We observe that the photonic RN remains consistent as n increases, which is as expected due to the reduction complexity having no direct relationship with this factor. However, we do see a small bump in the reduction time, as b increases from 8 to 9 and the EAM is required to output a second modulated pulse to support the higher bit precisions. STIFT experiences the opposite relationship, with it's reduction time remaining consistent as b changes, but increasing with n. Comparing the two RNs, we observe that the photonic RN performs better in terms of reduction time across all b and almost all n. The only situation in which STIFT performs better is when $n < 7$. However, in practice, n would rarely be this small, and tends to be on the larger end of the sweep.

When looking at the hardware complexity of the RNs, the benefits of the photonic RN again become clear. The number of channels needed to accumulate all psums is n in the photonic RN, since accumulation occurs in the PCA, and therefore we only require a channel per psum being sent. Aside from PT and ST-Linear which are fairly simple in hardware structure, due to the low fan-in of the electrical links and accumulators, the electrical RNs require a tree-based RN topology in order to fully collect and reduce the psums. This structure is very costly in terms of hardware complexity, and requires a sizable number of channels to build the tree. In addition, although the reduction time complexity of the electrical RNs is independent of the bit precision, the hardware complexity is not. The channels in the electrical RNs need to be at least b bits wide, in order to correctly accumulate two psums in an adder. In contrast, due to information in the photonic RN being represented as analog signals, the channels only need to be 1 bit wide. The photonic RN requires just $\lceil b/8 \rceil + 1$ adders (TIRs) to accumulate all psums for one output value. This is possible since

Table 3.1: Reduction latency and hardware complexities of different RNs. n is the number of psums to accumulate, i is the number of folds, l is the number of levels in the adder tree network, b is bit precision, $T_e = 1.25$ns is electronic cycle time, and $T_p = 5$ps is photonic cycle time

RN	Time	# Channels	Channel Size	# Adders
PT	$O(T_e \times n)$	$O(1)$	$O(b)$	$O(1)$
ST-Linear	$O(T_e \times n)$	$O(n/i)$	$O(b)$	$O(n/i + 1)$
S-Tree	$O(T_e \times (i + l) \times \log_2(n/i))$	$O(2n - 2 + n/2 - \log_2(n/2) - 1 \times ReLU(1 - \lfloor (n-1)/16 \rfloor)$	$O(b)$	$O(n - 1)$
$\mathrm{ST-Tree_{ac}}$	$O(T_e \times i \times \log_2(n/i))$	$O(2n + n/2 - \log_2(n/2) - 1 \times ReLU(1 - \lfloor (n-1)/16 \rfloor)$	$O(b)$	$O(n)$
STIFT	$O(T_e \times i \times \log_2(n/i))$	$O(2n + n/2 - \log_2(n/2) - 1 \times ReLU(1 - \lfloor (n-1)/16 \rfloor)$	$O(b)$	$O(n)$
Photonic RN	$O(T_p \times 2^8 \times \lceil b/8 \rceil \times i)$	$O(n)$	$O(1)$	$O(\lceil b/8 \rceil) + 1$

the PCA is capable of temporal accumulation, and therefore we only require a adder for each of the four 8-bit parts of the psum, as well as the analog op-amp voltage adder. S-Tree, $\mathrm{ST-Tree_{ac}}$, and STIFT instead all require an entire tree-based adder unit structure to accumulate all psums. Although PT and ST-Linear do not require this structure, the trade-off is that they have a much higher reduction latency, even greater than the other electrical RNs.

We compare the reduction hardware complexity by evaluating the total energy cost to accumulate all psums in the photonic RN and STIFT, as both b and n are swept. As seen in Fig. 3.9, the photonic RN maintains a consistent energy consumption as b changes, and increases slowly as n increases. This is as expected, after observing the relationships in the hardware complexity equations. In contrast, STIFT's energy consumption increases with both b and n. At lower b and n, both RN's energy is about equal. However, as b and n increase, STIFT's energy quickly exceeds the photonic RNs. Based off the hardware complexity equations, we can infer that this rapid increase is due to STIFT's positive relationship with both b and n, resulting in it's reduction energy increasing at a higher rate than the photonic RN's.

Copyright© Bobby Bose, 2023.

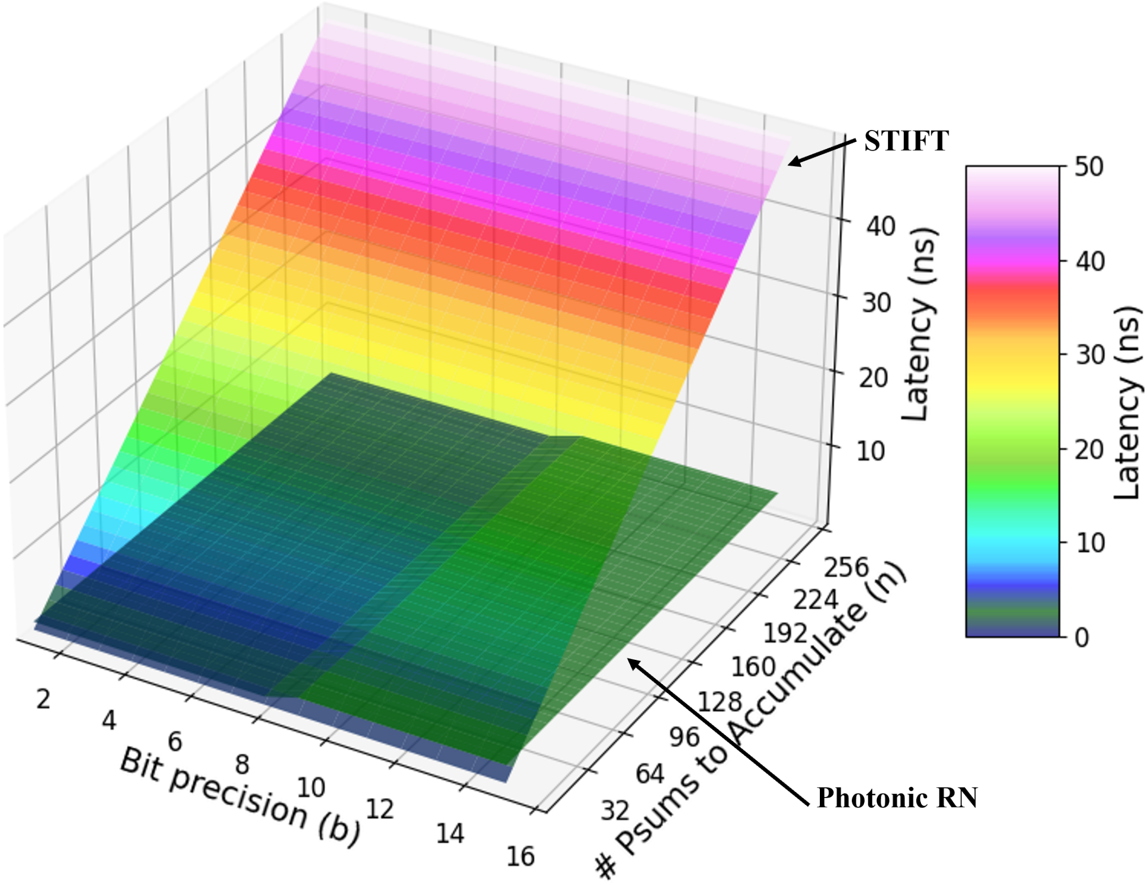

Figure 3.8: Comparison of reduction latency to reduce different number of partial sums over different bit precisions

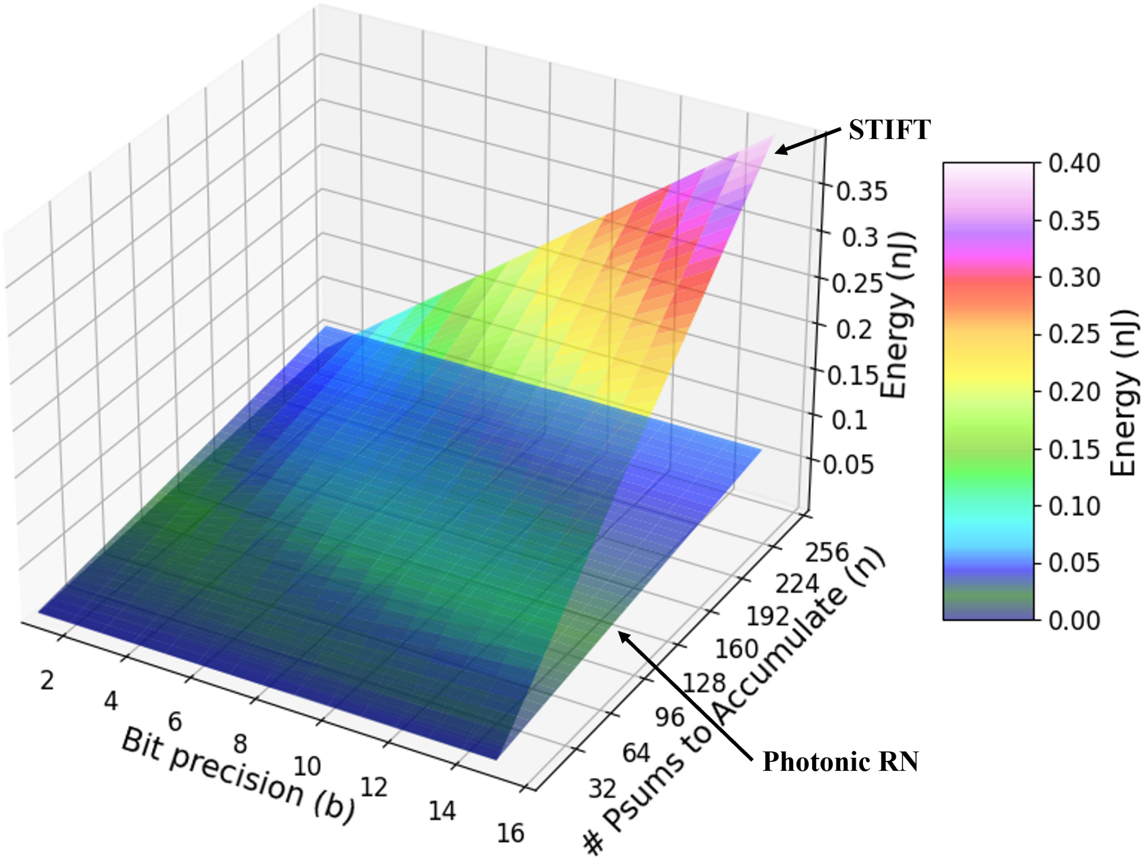

Figure 3.9: Comparison of reduction energy to reduce different number of partial sums over different bit precisions

Chapter 4 Evaluation

4.1 Evaluation Setup

We evaluate our photonic RN against the S-Tree, $\mathrm{ST-Tree_{ac}}$, and STIFT RNs described in [20], which are based on the MAERI [14] and SIGMA [23] RN structures. To model these networks, we extended the simulator developed for the SCONNA accelerator [31] to handle electrical networks. This is a transaction-level, event-driven python-based simulator originally designed to evaluate for SCONNA and other photonic accelerators. In addition, the simulator implements convolution to toeplitz matrix conversion and clustering.

We set hardware configuration parameters similar to [20], using two 512-MiB HBM2 DRAM modules, in order to maintain accuracy between tested RNs. We use an output-stationary dataflow for all evaluation and maintain a 8-bit precision for the psums. A pipe-lined cache is assumed to be used, resulting in a cache latency of only 1 cycle. As seen in Fig. 4.1, a tree-based DN and a linear MN are set and kept consistent throughout testing. It should be noted that the photonic RN does not require buffers between the MN and RN, and instead sends psums directly from the PEs (multipliers) to the RN. In addition, S-Tree requires an extra connection (not depicted in Fig. 4.1) from memory to the RN, in order to perform accumulation folding. $\mathrm{ST-Tree_{ac}}$, STIFT, and the photonic RN all perform accumulation folding within the RN itself.

We evaluate each RN for latency and energy-efficiency (FPS/W) over four different DNNs: GoogLeNet, ResNet50, DenseNet121, and VGG16. For each DNN, we test with set cluster sizes of 2, 4, 8, 16, 32, 64, and 128. In addition, the area for our photonic RN is calculated and compared to the other tested RNs through a footprint-efficiency analysis (FPS/mm^2), as well as other photonic interconnect based accelerators. Power and area metrics for the components in our photonic RN are presented in Table 4.1. The required laser power was calculated based off the insertion losses incurred through the path light travels in the network, as seen in Table 4.2. The calculated required laser power came out to be -23.154 dbm, which is equivalent to about 4.837μW.

4.2 Latency

As seen in Fig. 4.2, the photonic RN achieves the lowest latency across all tested DNNs. This is followed by $\mathrm{ST-Tree_{ac}}$ and STIFT, which exhibit similar latency's due to their identical reduction complexity. S-Tree performs the worst out of the tested RNs, which is as expected since it is unable to perform accumulation folding within the RN itself and requires reads and writes to memory. The photonic RN's low latency can be attributed to it's low reduction complexity which arises from being able to accumulate all psums in the network in just one cycle per 8-bit pulse. This is due to the improved fan-in seen by the photonic interconnect links and PCAs. Thus,

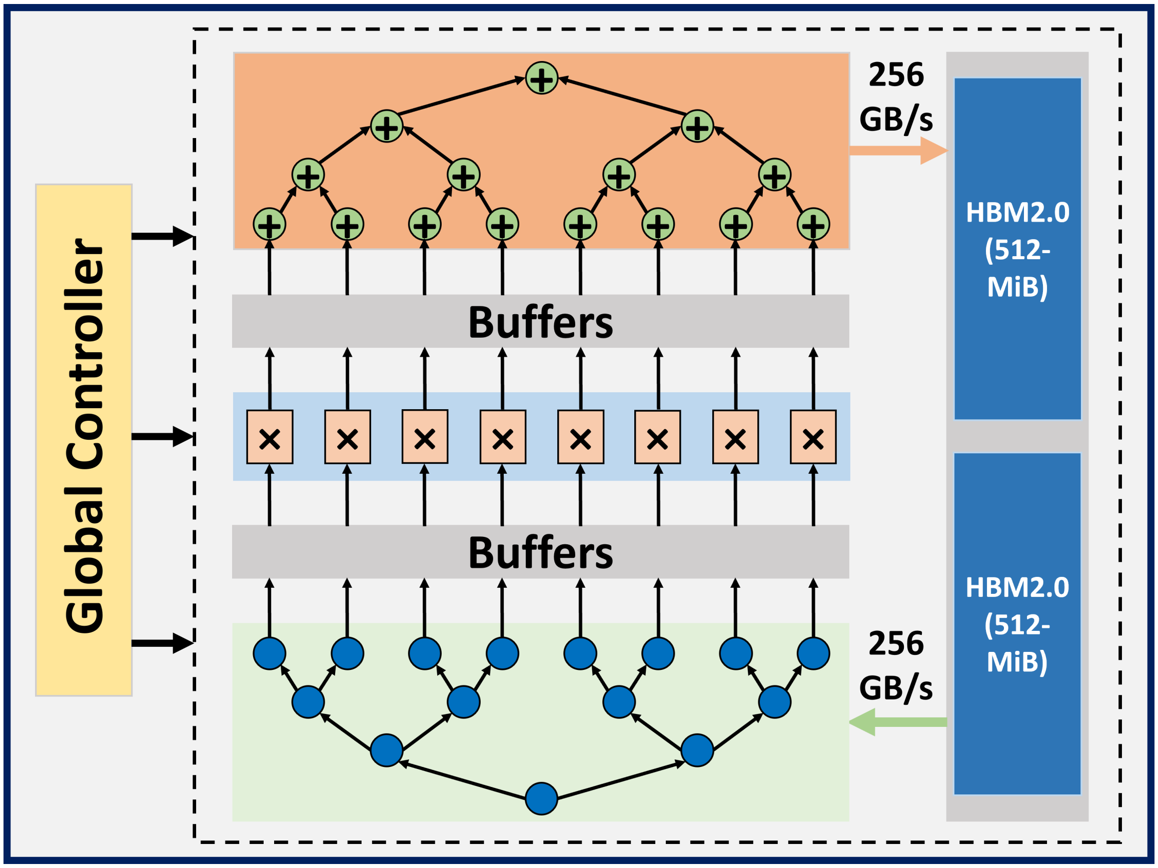

Figure 4.1: DNN accelerator architecture used in evaluation with an example tree-based RN

Table 4.1: Photonic RN Component Parameters

Component	Power (mW)	Area (mm^2)
DTC [3]	8	0.02
Demux (1 to 2)	0.042	1.59E-4
Demux (1 to 4)	0.125	4.76E-4
EAM [38]	1.5E-6	4E-7
TIR [8, 36]	1.158	5.2E-3
Photodetector [33, 34]	1.1	4.5E-6
Microring [28, 35]	0.8	9E-6
16-bit ADC [21]	2.55	2E-3

Table 4.2: Losses incurred in the photonic RN

Component	Loss (dB)
EAM [38]	0.25
Microring [1]	0.01
Waveguide propagation [1]	0.3/mm
Waveguide bend [2]	0.001

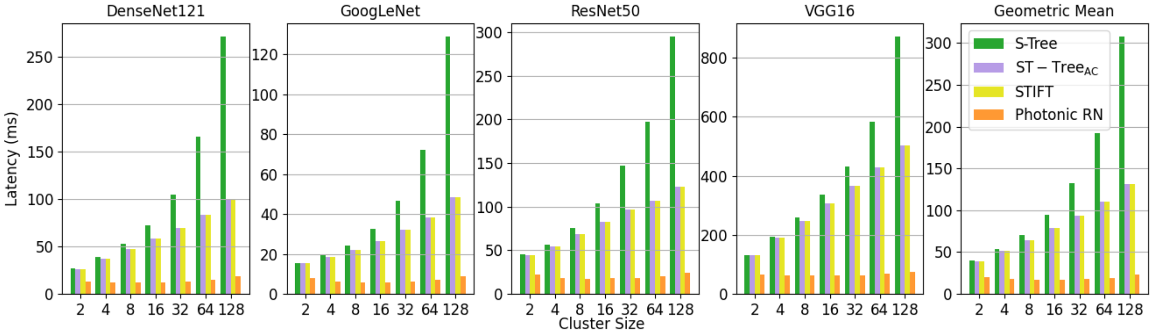

Figure 4.2: Latency (ms) of the tested RNs to run different DNNs

as a result we see that the improved fan-in has a direct positive relationship with the reduction complexity and the overall throughput of the network.

Across all tested DNNs, as cluster size increases, the latency for all the RNs also increases. This trend is the result of less output values being computed in parallel as cluster size increases and the total number of clusters decreases. In addition, as cluster size increases, the chance for components in a network to go unused both still consume static power also increases. Despite this, when looking at the geometric mean of the latency across the tested DNNs, we observe that as cluster size increases, the speed-up achieved by the photonic RN also increases. At cluster size 2, the photonic RN is 1.98× faster than the next fastest RN. At cluster size 128 however, this jumps up to 5.63× the next fastest RN. Therefore, we can infer that the photonic RN is affected by the increase in cluster size less due to less components in the network remaining unused.

4.3 Energy-efficiency

The energy-efficiency of the tested networks varies heavily at different cluster sizes. At the lower cluster sizes of 2 and 4, the photonic RN performs slightly worse than the electrical RNs, as seen in Fig. 4.3. At its worst, the photonic RN is only 0.95× as energy-efficient as the best performing electrical RN. This occurs at cluster size 4 when running GoogLeNet. The photonic RN's performance here can be attributed to the increased power cost it incurs compared to the electrical RNs due to the photonic interconnects and PCAs requiring higher power draws than their electrical equivalents. At the lower cluster sizes, the benefits of the improved fan-in of the

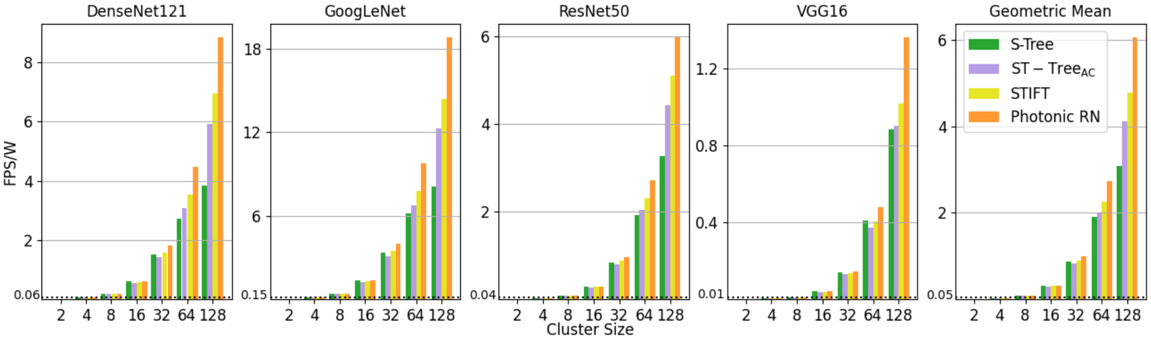

Figure 4.3: FPS/W of the tested RNs to run different DNNs

photonic RN are not as impactful in the throughput, and therefore the increased power draw has more of an effect on the results. However, from cluster size 8, we start to see an improvement of the photonic RN as it becomes more energy-efficient than $\mathrm{ST-Tree_{ac}}$. At cluster size 16, the photonic RN becomes more energy-efficient than all tested electrical RNs. This trend continues as cluster size increases, with the photonic RN's energy-efficiency improvement over the electrical RNs also increasing. This can be explained by the improved fan-in seen by the photonic interconnects and PCAs having a larger impact on throughput at higher cluster sizes and overcoming the increased power draw of the photonic RN. In addition, this energy-efficiency improvement at higher cluster sizes is magnitudes greater than the loss at the lower cluster sizes. When looking at the geometric mean, we can observe that at cluster size 128, the photonic RN is 1.97× more energy-efficient than S-Tree, 1.47× more than $\mathrm{ST-Tree_{ac}}$, and 1.27× more than STIFT.

4.4 Area

The total area of our 256 PE photonic RN, as seen in Fig. 3.1, was calculated to be 2.63 mm^2. Based on the results presented in [20] using TSMC 28nm GP standard LVT library, for a 256 PE architecture, the areas of the electrical RNs are 0.172 mm^2 for S-Tree, 0.32 mm^2 for $\mathrm{ST-Tree_{ac}}$, and 0.273 mm^2 for STIFT. It should be noted that S-Tree also requires a global buffer to store partially calculated output values, whereas the other two electrical RNs keep track of it within the network. Most photonic circuits cannot be implemented with technology nodes below 45nm. Therefore, to have an equivalent comparison based on similar technology nodes, if we implement the electrical RNs in 45nm technology, then the area values are scaled to 0.444 mm^2 for S-Tree, 0.827 mm^2 for $\mathrm{ST-Tree_{ac}}$, and 0.705 mm^2 for STIFT. Even when accounting for this, the photonic RN is still larger than the electrical RNs. This increase in area is expected however, as the electrical RNs are only made up of multiplier and adder units, which do not incur as much area overhead compared to the photonic and electro-optic devices and circuits used in the photonic RN.

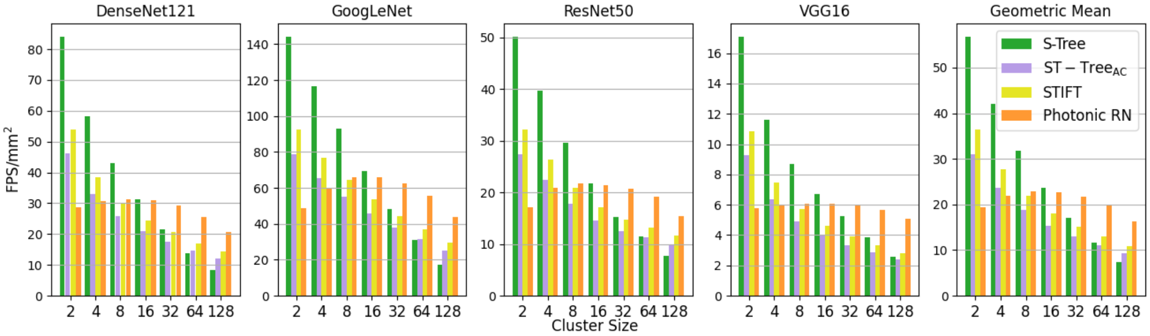

Figure 4.4: FPS/mm^2 of the tested RNs to run different DNNs

To get a better understanding of how the RNs perform when accounting for the difference in areas, we compare footprint-efficiency in terms of FPS/W of all RNs over the four tested DNNs while varying cluster size, as seen in Fig. 4.4. An important factor to note is that we perform this analysis at a similar technology node of 45nm, in order to keep results comparable. We observe that at the lower cluster sizes of 2 and 4, S-Tree tends to perform the best, followed by STIFT, $\mathrm{ST-Tree_{ac}}$, and then the photonic RN. This matches the order of RN size from smallest to largest. This relationship is understandable, as at these lower cluster sizes the benefits of the RNs are not as impactful, so their FPS/mm^2 performance is primarily dependent on the area. At cluster sizes 8 and 16, the photonic RN starts to outperform STIFT and $\mathrm{ST-Tree_{ac}}$ due to its improved reduction latency, but is still worse than S-Tree. However, the difference in performance between S-Tree and the other networks starts to significantly decrease as well. From cluster size 32 onwards, the photonic RN performs the best as its reduction latency benefits start to become even more impactful, and are able to overcome the increased area overhead cost. In addition, S-Tree's performance continuously becomes worse compared to the other RNs, as the benefits of its much lower area overhead lose importance, compared to the improved reduction latency benefits of the other RNs. Thus, overall we see that as cluster size increases, the FPS/mm^2 performance of the photonic RN continuously improves and eventually overcomes the increased area overhead, compared to the electrical RNs. One important caveat however, is that this analysis looks at a comparison between the RNs at a comparable technology node of 45nm. If we want to implement at a lower technology node, then area could end up being a greater challenge, and the electrical RNs might still be a better choice, despite the photonic RN's improved reduction latency.

In addition to comparing the photonic RN's area against the electrical RNs, we also compare it to other state-of-the-art photonic DNN accelerators. When compared to these accelerators, we observe that the photonic RN's area is very reasonable. The area of a single chiplet in ASCEND [17] is 24.07 mm^2. Albireo's [25] total architecture is 124.6 mm^2. Although these results are for an entire accelerator and not just the reduction portion, since the photonic RN is designed to work with electrical DNs and MNs, we can infer that it's total area would be smaller than these fully-photonic accelerators. Therefore, in general, in a purely area overhead comparison, the photonic RN might perform worse than its electrical counterparts, but is still reasonable in size when compared to fully photonic based DNN accelerators.

Copyright© Bobby Bose, 2023.

Chapter 5 Conclusion and Future Work

5.1 Summary of Results and Conclusion

Current state-of-the-art flexible spatial accelerators are limited by the low fan-ins seen by the electrical links and accumulators that typically compose the RN. This leads to high latency, energy, and area costs as the number of components required scales logarithmically with the number of psums that need to be accumulated in the RN. In order to overcome these limitations, in this work, we proposed a photonic RN for spatial GEMM accelerators that utilizes photonic interconnect links and an electro-photonic accumulator, the PCA. The photonic interconnects are able to overcome the low fan-in of one channel per link seen by electrical links, by utilizing WDM in order to improve spatial fan-in. The PCA is able to spatially accumulate all psums arriving on a photonic interconnect in just one cycle, heavily improving it's fan-in compared to electrical accumulators. In addition, the PCA can hold partially accumulated output information, allowing it to temporally accumulate psums as well, further reducing the total number of accumulators required in the RN. Therefore, by utilizing photonic interconnects and PCAs together, the photonic RN is able to improve on the low fan-ins seen by current flexible spatial accelerators RNs.

To evaluate our photonic RN, we first compared its latency and hardware reduction complexities against other state-of-the-art electrical RNs. We observed that due to the improved fan-in of the photonic interconnects and PCAs, the photonic RN achieves lower latency and hardware reduction complexities compared to its electrical counterparts. To test this, we performed an analysis of both latency and hardware energy cost, while varying bit precision and number of psums to accumulate, and found that the photonic RN consistently achieves better results compared to the best performing electrical RN. We also evaluated our photonic RN against the three best electrical RNs by running four DNNs over them while varying cluster size. Our results show that the photonic RN is able to achieves latency speed-ups ranging from 1.98× at cluster size 2, to 5.63× at cluster size 128. This is once again a result of the higher fan-ins leading to improved reduction complexities and therefore higher throughputs. In terms of energy-efficiency, we observed that although the photonic RN is slightly less energy-efficient at lower cluster sizes, as cluster size increases it becomes much more energy-efficient than the electrical RNs, achieving efficiency increases up to 1.97× at cluster size 128. This trend is a result of the improved fan-in and lower reduction complexity eventually being able to overcome the higher power consumption of the photonic RN. Lastly, we performed a footprint-efficiency analysis and found a similar trend. At lower cluster sizes the photonic RN performed worse than the electrical RNs. However, as the cluster size increases, the benefits of the improved reduction latency and throughput overcome the increased area overhead of the photonic RN. In conclusion, our results demonstrate that by using our proposed photonic RN for spatial GEMM accelerators, we can successfully overcome the limitations of the low fan-ins of electrical links and accumulators, and can further improve throughput, energy-efficiency, and footprint-efficiency due to the improved reduction complexities achieved by it.

5.2 Future Work

There are multiple areas of future research that the work presented in this paper can be applied to. Our photonic RN is currently only able to handle fixed-point psums. To handle the reduction of floating-point psums as well, our photonic RN could be extended to support representation of these values. This would allow the network to handle a larger coverage of DNN models. In addition, the full potential of photonic architecture reduction networks can be realized by integrating multi-chiplet systems with photonic distribution networks as well, like seen in [15–17]. This has the potential to improve the overall GEMM accelerator performance, by utilizing the benefits of photonic interconnects and devices across the entire architecture, and not just the RN. Lastly, our photonic RN does not need to be limited to CNN-specific applications. Rather, it can applied to any accelerator that is spatial and in which the main function to accelerate involves the reductions of psums. This includes any algorithm involving linear algebra sub-functions such as matrix-matrix multiplication or vector-matrix multiplication.

Copyright© Bobby Bose, 2023.

Bibliography

[1] M. A. Al-Qadasi, L. Chrostowski, B. J. Shastri, and S. Shekhar. Scaling up silicon photonic-based accelerators: Challenges and opportunities. *APL Photonics*, 7(2):020902, 02 2022.

[2] M. Bahadori, M. Nikdast, Q. Cheng, and K. Bergman. Universal design of waveguide bends in silicon-on-insulator photonics platform. *Journal of Lightwave Technology*, 37(13):3044–3054, 2019.

[3] C.-C. Chen, C.-S. Hwang, and K.-H. Chang. All-digital cost-efficient cmos digital-to-time converter using binary-weighted pulse expansion. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 28(4):1094–1098, 2020.

[4] Y.-H. Chen, T.-J. Yang, J. Emer, and V. Sze. Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(2):292–308, 2019.

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

[6] B.-B. Corporation. Precision switched integrator transimpedance amplifier. Technical report, 1996.

[7] Y. Gui, H. Wang, B. M. Nouri, H. Dalir, and V. J. Sorger. Broadband mach-zehnder interferometer modulator on indium tin oxide (ito) platform operating at 100 ghz with asymmetric power splitting. In *2022 IEEE Photonics Conference (IPC)*, pages 1–2, 2022.

[8] A. D. Gungordu, G. Dundar, and M. B. Yelten. A high performance tia design in 40 nm cmos. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2020.

[9] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally. Eie: Efficient inference engine on compressed deep neural network. In *Proceedings of the 43rd International Symposium on Computer Architecture*, ISCA '16, page 243–254. IEEE Press, 2016.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[11] R. L. R. Junior and P. Rech. Reliability of google's tensor processing units for convolutional neural networks. In *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S)*, pages 25–27, 2022.

[12] K. Kim, Y. Kim, W. Yu, and S. Cho. A 7b, 3.75ps resolution two-step time-to-digital converter in 65nm cmos using pulse-train time amplifier. In *2012 Symposium on VLSI Circuits (VLSIC)*, pages 192–193, 2012.

[13] K. Kim, W. Yu, and S. Cho. A 9 bit, 1.12 ps resolution 2.5 b/stage pipelined time-to-digital converter in 65 nm cmos using time-register. *IEEE Journal of Solid-State Circuits*, 49(4):1007–1016, 2014.

[14] H. Kwon, A. Samajdar, and T. Krishna. Maeri: Enabling flexible dataflow mapping over dnn accelerators via reconfigurable interconnects. page 461–475, 2018.

[15] Y. Li, A. Louri, and A. Karanth. Spacx: Silicon photonics-based scalable chiplet accelerator for dnn inference. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 831–845, 2022.

[16] Y. Li, A. Louri, and A. Karanth. Sprint: A high-performance, energy-efficient, and scalable chiplet-based accelerator with photonic interconnects for cnn inference. *IEEE Transactions on Parallel and Distributed Systems*, 33(10):2332–2345, 2022.

[17] Y. Li, K. Wang, H. Zheng, A. Louri, and A. Karanth. Ascend: A scalable and energy-efficient deep neural network accelerator with photonic interconnects. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 69(7):2730–2741, 2022.

[18] R. Mayer and H.-A. Jacobsen. Scalable deep learning on distributed infrastructures: Challenges, techniques, and tools. 53(1), 2020.

[19] D. A. B. Miller. Device requirements for optical interconnects to silicon chips. *Proceedings of the IEEE*, 97(7):1166–1185, 2009.

[20] F. Muñoz-Martínez, J. L. Abellán, M. E. Acacio, and T. Krishna. A novel network fabric for efficient spatio-temporal reduction in flexible dnn accelerators. In *2021 15th IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, pages 1–8, 2021.

[21] D.-R. Oh, K.-J. Moon, W.-M. Lim, Y.-D. Kim, E.-J. An, and S.-T. Ryu. An 8b 1gs/s 2.55mw sar-flash adc with complementary dynamic amplifiers. In *2020 IEEE Symposium on VLSI Circuits*, pages 1–2, 2020.

[22] Y. Okawachi, B. Y. Kim, M. Lipson, and A. L. Gaeta. Chip-scale frequency combs for data communications in computing systems. *Optica*, 10(8):977–995, 08 2023.

[23] E. Qin, A. Samajdar, H. Kwon, V. Nadella, S. Srinivasan, D. Das, B. Kaul, and T. Krishna. Sigma: A sparse and irregular gemm accelerator with flexible interconnects for dnn training. In *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 58–70, 2020.

[24] Y. S. Shao, J. Clemons, R. Venkatesan, B. Zimmer, M. Fojtik, N. Jiang, B. Keller, A. Klinefelter, N. Pinckney, P. Raina, S. G. Tell, Y. Zhang, W. J. Dally, J. Emer, C. T. Gray, B. Khailany, and S. W. Keckler. Simba: Scaling deep-learning inference with multi-chip-module-based architecture. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '52, page 14–27, New York, NY, USA, 2019. Association for Computing Machinery.

[25] K. Shiflett, A. Karanth, R. Bunescu, and A. Louri. Albireo: Energy-efficient acceleration of convolutional neural networks via silicon photonics. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, pages 860–873, 2021.

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[27] A. Sludds, S. Bandyopadhyay, Z. Chen, Z. Zhong, J. Cochrane, L. Bernstein, D. Bunandar, P. B. Dixon, S. A. Hamilton, M. Streshinsky, A. Novack, T. Baehr-Jones, M. Hochberg, M. Ghobadi, R. Hamerly, and D. Englund. Delocalized photonic deep learning on the internet's edge. *Science*, 378(6617):270–276, 2022.

[28] C. Sun, M. Wade, Y. Lee, J. Orcutt, L. Alloatti, M. Georgas, A. Waterman, J. Shainline, R. Avizienis, S. Lin, B. Moss, R. Kumar, F. Pavanello, A. Atabaki, H. Cook, A. Ou, J. Leu, Y.-H. Chen, K. Asanović, and V. Stojanovic. Single-chip microprocessor that communicates directly using light. *Nature*, 528:534–538, 12 2015.

[29] F. P. Sunny, A. Mirza, M. Nikdast, and S. Pasricha. Robin: A robust optical binary neural network accelerator. *ACM Trans. Embed. Comput. Syst.*, 20(5s), sep 2021.

[30] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017.

[31] S. Vatsavai, V. Karempudi, I. Thakkar, A. Salehi, and T. Hastings. Sconna: A stochastic computing based optical accelerator for ultra-fast, energy-efficient inference of integer-quantized cnns, may 2023.

[32] S. S. Vatsavai, V. S. P. Karempudi, and I. Thakkar. An optical xnor-bitcount based accelerator for efficient inference of binary neural networks, 2023.

[33] L. Virot, P. Crozat, J.-M. Fedeli, J.-M. Hartmann, D. Marris-Morini, E. Cassan, F. Boeuf, and L. Vivien. Germanium avalanche receiver for low power interconnects. *Nature communications*, 5:4957, 09 2014.

[34] B. Wang, Z. Huang, W. V. Sorin, X. Zeng, D. Liang, M. Fiorentino, and R. G. Beausoleil. A low-voltage si-ge avalanche photodiode for high-speed and energy

efficient silicon photonic links. *Journal of Lightwave Technology*, 38(12):3156–3163, 2020.

[35] Q. Xu, D. Fattal, and R. G. Beausoleil. Silicon microring resonators with 1.5-μm radius. *Opt. Express*, 16(6):4309–4315, 03 2008.

[36] E. Yang and T. Lehmann. High gain operational amplifiers in 22 nm cmos. In *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2019.

[37] J. Yang and J. Li. Application of deep convolution neural network. In *2017 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 229–232, 2017.

[38] C. Ye, S. Khan, Z. R. Li, E. Simsek, and V. J. Sorger. λ-size ito and graphene-based electro-optic modulators on soi. *IEEE Journal of Selected Topics in Quantum Electronics*, 20(4):40–49, 2014.

[39] F. Zokaee, Q. Lou, N. Youngblood, W. Liu, Y. Xie, and L. Jiang. Lightbulb: A photonic-nonvolatile-memory-based accelerator for binarized convolutional neural networks. In *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1438–1443, 2020.

Vita

Bobby Bose

Place of Birth:

- Lexington, KY, USA

Education:

- University of Kentucky, Lexington, Kentucky
  M.S. in Computer Engineering, Expected Dec. 2023
- University of Kentucky, Lexington, Kentucky
  B.S. in Computer Engineering, Dec. 2022

Positions:

- Graduate Researcher
  University of Kentucky
  Aug. 2022 - Dec. 2023
- CAD Intern
  Infineon Technologies
  Aug. 2021 - Dec. 2021
- Undergraduate Researcher
  University of Kentucky
  Mar. 2020 - July 2022

Honors:

- Best Master Student Research Symposium Award 2023
- GLSVLSI 2021 Best Paper Award

Publications:

- Bobby Bose and Ishan Thakkar. 2021. Characterization and Mitigation of Electromigration Effects in TSV-Based Power Delivery Network Enabled 3D-Stacked DRAMs. In Proceedings of the 2021 on Great Lakes Symposium on VLSI (GLSVLSI '21). Association for Computing Machinery, New York, NY, USA, 101–107.