2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA) | 978-1-6654-2027-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/HPCA53966.2022.00066

SPACX: Silicon Photonics-based Scalable Chiplet Accelerator for DNN Inference

Yuan Li, Ahmed Louri
Department of Electrical and Computer Engineering
George Washington University, Washington, DC 20052
Email: liyuan5859, louri@gwu.edu

Avinash Karanth
School of Electrical Engineering and Computer Science
Ohio University, Athens, Ohio 45701
Email: karanth@ohio.edu

***Abstract*—In pursuit of higher inference accuracy, deep neural network (DNN) models have significantly increased in complexity and size. To overcome the consequent computational challenges, scalable chiplet-based accelerators have been proposed. However, data communication using metallic-based interconnects in these chiplet-based DNN accelerators is becoming a primary obstacle to performance, energy efficiency, and scalability. The photonic interconnects can provide adequate data communication support due to some superior properties like low latency, high bandwidth and energy efficiency, and ease of broadcast communication. In this paper, we propose SPACX: a Silicon Photonics-based Chiplet ACcelerator for DNN inference applications. Specifically, SPACX includes a photonic network design that enables seamless single-chiplet and cross-chiplet broadcast communications, and a tailored dataflow that promotes data broadcast and maximizes parallelism. Furthermore, we explore the broadcast granularities of the photonic network and implications on system performance and energy efficiency. A flexible bandwidth allocation scheme is also proposed to dynamically adjust communication bandwidths for different types of data. Simulation results using several DNN models show that SPACX can achieve 78% and 75% reduction in execution time and energy, respectively, as compared to other state-of-the-art chiplet-based DNN accelerators.**

***Keywords*-DNN, Chiplet, Accelerator, Silicon Photonics**

I. INTRODUCTION

Emerging deep neural network (DNN) models often exhibit significant increase in model complexity and size for higher inference accuracy [1]–[8]. Consequently, computing systems must scale up in processing power, on-chip memory capacity, and data communication to efficiently process the large-scale DNN models [9], [10]. As the scaling of a monolithic chip slows down due to the stringent constraints of power density and fabrication cost [11], [12], chiplet-based architectures [11], [13] have been recently proposed for scalable DNN inference applications [13]–[16]. However, in chiplet-based DNN accelerators, data communication using metallic-based interconnects is posing a major obstacle to the performance, energy efficiency, and scalability. The fundamental limitations of the metallic-based interconnects, especially those spanning across chiplets, are (1) high latency and low bandwidth which inevitably lead to system performance degradation [13], [17], (2) prominent latency discrepancy between single-chiplet and cross-chiplet communications which makes the orchestration of data communication imposed by parallel computing in DNN models challenging [13], and (3) excess energy of long-distance communication frequently observed in chiplet-based architectures [18], [19].

Photonic interconnects can potentially overcome the fundamental limitations of the metallic-based interconnects [18], [20]. Low-loss waveguide can distribute data to processing elements (PEs) in a single chiplet or across several chiplets without requiring multiple hops [18], maintaining low and uniform communication latency. Communication bandwidth can be increased by techniques such as wavelength-division multiplexing (WDM) and space-division multiplexing (SDM) [21]. Photonic interconnects have been shown to achieve high energy efficiency as the communication distance increases [17]. More importantly, the salient ease of broadcast property of photonic interconnects [18], [22] makes them especially suitable to support the prevalent broadcast communication observed in DNN inference applications.

Prior photonic network designs [23]–[34] for either on-chip or chiplet-based data communication often target communication in CPUs or GPUs, and exhibit equal bandwidth between arbitrary nodes. Several prior designs intentionally disable the broadcast capability [25], [26], [30]. However, the highly regular and non-uniform communication in DNN inference applications [1], [15], [16] makes the costly equal bandwidth allocation unnecessary. Prevalent broadcast communication in DNN inference applications [1], [16] cannot be adequately supported as well. In addition, previous DNN dataflows [13], [35]–[40] are designed for accelerators with only the metallic-based interconnects, as a result, optimized to consume more data in memory hierarchies closer to computation units. These dataflows do not promote data broadcast because broadcast communication cannot be efficiently supported by underlying accelerators. A dataflow tailored to photonic interconnects is necessary because (1) data communication distance is not a primary concern as in the previous dataflows, and (2) features uniquely related to photonic interconnects such as ease of broadcast communication shall be fully considered.

In this paper, we propose **SPACX: Silicon Photonics-based Chiplet ACcelerator**. SPACX includes a new photonic network and corresponding dataflow co-optimized for DNN inference applications. We also explore multiple broadcast granularities enabled by different configurations of the photonic network and their implications on system performance and energy efficiency. Besides, a flexible bandwidth allocation

978-1-6654-2027-3/22/$31.00 ©2022 IEEE
DOI 10.1109/HPCA53966.2022.00066

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on October 06,2024 at 09:05:01 UTC from IEEE Xplore. Restrictions apply.

scheme is developed to dynamically adjust communication bandwidths for different types of data based on DNN layer parameter information. Simulation studies with several DNN models [2]–[5] show that SPACX can achieve 78% and 75% reduction in execution time and energy, respectively, as compared to other state-of-the-art chiplet-based DNN accelerators with either metallic-based [13] or photonic [30] interconnects. The major contributions of this paper include:

- **A Photonic Network Design**. We develop a hierarchical photonic network that seamlessly extends the connection between the global buffer (GB) and chiplets to PEs, enabling one-hop data communication from the GB to arbitrary PEs. The photonic network adequately supports orthogonal single-chiplet (from the GB to all PEs on a single chiplet) and cross-chiplet (from the GB to specific PEs on all chiplets) broadcast communications.
- **A Broadcast-Enabled Output-Stationary Dataflow**. The proposed dataflow enforces single-chiplet and cross-chiplet broadcast of input features and weights by spatially allocating computations with shared input features and weights to PEs on a chiplet and corresponding PEs on all chiplets, respectively. Such allocation exploits the orthogonal broadcast capability of the proposed photonic network to obtain high data parallelism and high energy efficiency of data communication. Furthermore, output-stationary nature of the developed dataflow significantly reduces intermediate data exchange between PEs which incurs excessive electrical-to-optical (E/O) and optical-to-electrical (O/E) signal conversions.
- **Broadcast Granularity Exploration**. We explore using SDM technique to support multi-granularity broadcast communication. By adding extra waveguides, proposed photonic network can be tuned to several configurations to support both single-chiplet and cross-chiplet broadcast communications at multiple granularities (the number of broadcast destination PEs). We explore the selection of network configurations based on DNN layer parameters and implications on performance and energy efficiency.
- **Flexible Bandwidth Allocation**. We enhance the proposed photonic network with a flexible bandwidth allocation scheme. This scheme adjusts the communication bandwidth by tuning the numbers of wavelengths for different types of data, based on DNN layer and system parameters obtained offline. This scheme helps improve network utilization and reduce PE stalls.

II. BACKGROUND AND MOTIVATION

A. Photonic Interconnects

1) Basic Photonic Components: Figure 1 demonstrates a single photonic link with WDM. An off-chip laser source emits light with different wavelengths λ0 and λ1. Light is then coupled into a waveguide using an optical coupler [41]. At the transmission side, two micro-ring resonators

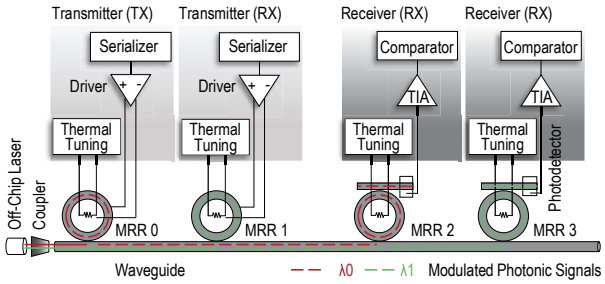


Figure 1. A photonic link including two sets of transmitter and receiver and two multiplexed wavelengths λ0 and λ1.

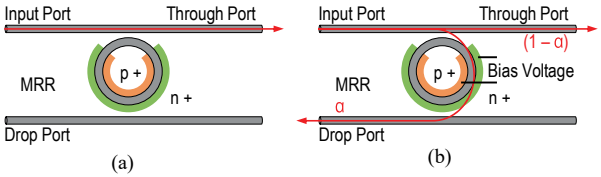


Figure 2. An optical tunable splitter working at (a) off-resonance and (b) a transient state between on-resonance and off-resonance with a split ratio of $\alpha/(1-\alpha)$ when a bias voltage is applied.

(MRRs) [42], [43] labeled `MRR0` and `MRR1` work as optical modulators to modulate wavelengths λ0 and λ1 using input electrical signals, respectively. At the receiving side, another two MRRs labeled `MRR2` and `MRR3` work as optical filters, each of which selects a specific modulated wavelength and forwards it to a corresponding photodetector [18]. The electrical signals generated by the photodetectors are amplified by the transimpedance amplifiers (TIAs) and forwarded to comparators to retrieve the original data. An MRR, works as either an optical modulator or filter, is tuned by a resistive heater controlled by a thermal tuning unit to mitigate thermal and process variations [18], [26]. As a signal can propagate through a waveguide at a speed close to the speed of light [21], one-hop data communication is viable in typical chiplet-based architectures regardless of the relative position of source and destination nodes. Prior work [24], [44]–[46] has also shown that as many as 64 wavelengths can be multiplexed within a single waveguide with each wavelength operating at 10 `Gbps` data rate. Since the transmitters and receivers account for a significant fraction of the overall energy consumption in a photonic link, high energy efficiency is achieved as communication distance increases.

2) Optical Tunable Splitter: Different from MRRs working at either on-resonance or off-resonance as optical modulators and filters, an optical tunable splitter works in the transient state between on-resonance and off-resonance [47]. As shown in Figure 2, the regions inside and outside an optical tunable splitters are doped to form a PIN diode structure. The optical tunable splitter is disabled (at off-resonance) if no bias voltage is applied, as shown in Figure 2 (a). Light traverses from the input port to the through port in this case. When applying an appropriate bias voltage to the PIN diode, light from the input port is split to two parts

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on October 06,2024 at 09:05:01 UTC from IEEE Xplore. Restrictions apply.

and forwarded to the drop port (α fraction) and the through port ($1-\alpha$ fraction), respectively. The split ratio is $\alpha/(1-\alpha)$ in this case. By tuning the bias voltage, the split ratio in the range of 0.4 to 1.8 can be obtained [47]. The bias voltage is tuned by a digital-to-analog converter (DAC) with a delay less than 500 ps [47]. Multiple optical tunable splitters can be cascaded [48] in the case when a split ratio outside the range of 0.4 to 1.8 is required.

We can construct a broadcast channel using a transmitter and multiple receivers with the corresponding optical filters replaced by optical tunable splitters, each of which tuned to an appropriate split ratio. Implementing broadcast channels instead of unicast channels shown in Figure 1 can significantly reduce the number of transmitters at the cost of moderate increase in laser power, leading to high energy efficiency of data communication. Single-chiplet or cross-chiplet broadcast communication is enabled when the receivers are attached to PEs in a single chiplet or specific PEs in all chiplets, respectively. The orthogonal single-chiplet and cross-chiplet communications are achieved when multiple broadcast channels with respective wavelengths are multiplexed in the same waveguide.

3) Existing Photonic Network Designs: Several prior photonic network designs [17], [23]–[25], [30], [31] have been developed for chiplet-based architectures. These network designs, similar to the ones developed for monolithic-chip architectures [26]–[29], [32]–[34], only target generic communication in CPUs or GPUs. The resulting equal bandwidth setup between arbitrary nodes may lead to unnecessarily high implementation cost and energy overhead, given the highly regular and non-uniform communication in DNN inference applications [1], [16]. Furthermore, the prevalent broadcast communication in DNN inference applications [1], [16] cannot be adequately supported by prior network designs [25], [26], [30] in which broadcast capability is intentionally disabled. Unlike prior photonic network designs, SPACX photonic network is tailored to support the predetermined regular data communication, especially broadcast communication, in DNN inference applications with minimal implementation cost and energy overhead.

B. Communication in DNN Inference

1) DNN Computation: The computation in a convolution layer can be formulated as a multi-dimensional nested loop over weight kernels, input feature maps (ifmaps), and output feature maps (ofmaps). The dimensions include the height (`r`) and width (`s`) of weight kernels, the height (`h`) and width (`w`) of ifmaps, the number of input channels (`c`), and the number of output channels (`k`). The height (`e`) and width (`f`) of ofmaps are not independent and can be derived from the previous dimensions. Figure 3 shows the computation operations and data communication involved in a convolution layer. Weights and input features are read-only input data while partial sums (psums) are read-and-write intermediate

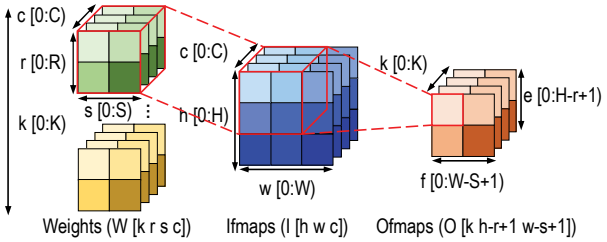


Figure 3. Computation and data communication in a convolution layer.

```
1  for c = [ 0 : C )
2      for k = [ 0 : K )
3          for h = [ 0 : H )
4              for w = [ 0 : W )
5                  for r = [ 0 : R )
6                      for s = [ 0 : S )
7                          O [k h - r + 1 w - s + 1] += W [k r s c] * I [h w c]
```

Figure 4. The nested loop representation of a convolution layer.

data used to generate the output features, which are output data. Figure 4 is a nested loop representation of a convolution layer assuming that both batch size and stride equal one.

2) Prevalent Broadcast Communication: Data communication involved in a convolution layer includes transmitting weights and input features to the PEs and gathering generated psums or output features from the PEs. Data communication in DNN inference applications is regular and predetermined by the nested loop and computing system parameters. As we can observe in Figure 4, the same weight `W[k r s c]` is broadcast to different destinations if the `h` (`line3`) and `w` (`line4`) loops are processed in parallel. Similarly, the same input feature `H[h w c]` is broadcast to different destinations if the `k` (`line2`), `r` (`line5`), and `s` (`line6`) loops are processed in parallel. However, leveraging broadcast opportunities of one input data type (weight or input feature) could lead to losing the broadcast opportunities of the other. For example, the broadcast opportunities of weights are fully leveraged by processing the `h` and `w` loops in parallel, however, at the cost of completely losing the broadcast opportunities of input features. With the goal of maximizing the overall broadcast opportunities, SPACX dataflow simultaneously broadcasts weights and input features by processing the `k`, `h`, and `w` loops in parallel with an output-stationary dataflow. Specifically, weight `W[k r s c]` is broadcast to a set of PEs that hold different `I[h w c]` (`h∈[0:H)`, `w∈[0:W)`, and `c` with the same value as in `W[k r s c]`). Input feature `I[h w c]` is broadcast to another set of PEs that hold different `W[k r s c]` (`k∈[0:K)`, `r` with the value of `h-e+1`, `s` with the value of `w-f+1`, `c` with the value as in `I[h w c]`). The simultaneous broadcast of input features and weights can be perfectly supported by the orthogonal single-chiplet and cross-chiplet broadcast capability of the proposed SPACX photonic network, respectively.

3) Emerging Dataflow Optimization Goals: Optimizing data locality is the major optimization goal of several prior dataflows [13], [35]–[40]. For example, [13], [49] improve

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on October 06,2024 at 09:05:01 UTC from IEEE Xplore. Restrictions apply.

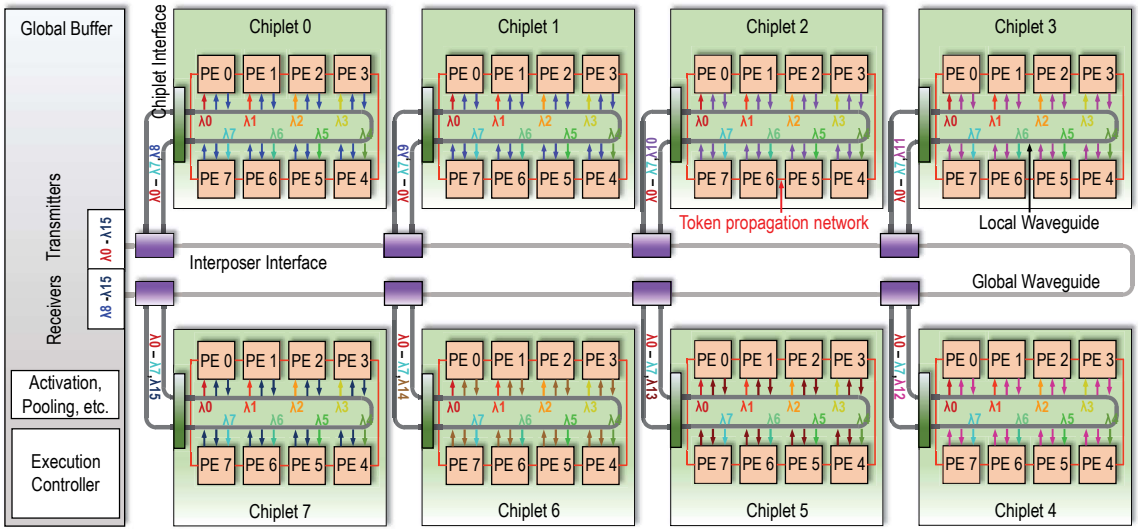


Figure 5. SPACX architecture with 8 accelerator chiplets and 8 processing elements (PEs) per accelerator chiplet. Each local waveguide is connected to the global waveguide through a chiplet interface and an interposer interface. A single-bit token propagation network is implemented on each accelerator chiplet to determine the PE with active PE-to-GB communication. 16 wavelengths $\lambda 0$, $\lambda 1$, ..., $\lambda 15$ are utilized for either single-chiplet or cross-chiplet broadcast communication.

weight locality while [36] improves output feature locality, incurring respective broadcast communication of input features and weights. This is because of the fundamental latency and energy limitations of the metallic-based interconnects. The resulting broadcast communication is not the primary optimization goal but created by the nature of convolution computation. In fact, the broadcast communication is not well-supported but emulated by several unicast communications in many metallic-based networks like [13]. Since photonic interconnects largely overcome the limitations of metallic-based interconnects, the original dataflow optimization goals may become less relevant while emerging optimization goals increasingly gain importance. Locality or data communication distance is no longer a primary concern as in the previous dataflows due to the distance-independent feature of photonic interconnects, while broadcast opportunities which used to be not well leveraged can significantly improve energy efficiency of data communication. We consider optimization goals uniquely related to the photonic interconnects such as maximizing broadcast communication and reducing E/O and O/E signal conversions in the SPACX dataflow.

III. SPACX ARCHITECTURE

A. Photonic Network Architecture

Figure 5 shows the SPACX architecture with `M=8` chiplets and `N=8` PEs on each chiplet. In this example, 16 wavelengths are implemented for either single-chiplet or cross-chiplet broadcast communication. All the photonic components (e.g., waveguides, MRRs, photodetectors, optical tunable splitters) are implemented on a silicon interposer [12] while all the electrical components are implemented on the GB die or accelerator chiplets. We can observe two levels of waveguides: a global waveguide connecting all accelerator chiplets, and a local waveguide on each accelerator chiplet. Please note that identical network architecture is used in both levels to provide seamless data communication between the GB and arbitrary PEs. The physical placement of the GB die and accelerator chiplets, as well as the placement of the global and local waveguides, is not necessarily the same as in Figure 5, which only highlights the hierarchical architecture of the SPACX photonic network.

B. Wavelength Allocation

We categorize all available wavelengths into two groups: `X` wavelengths for cross-chiplet broadcast communication ($\lambda 0$ to $\lambda 7$) and `Y` wavelengths for single-chiplet broadcast communication and PE-to-GB unicast communication ($\lambda 8$ to $\lambda 15$). The corresponding PEs on all chiplets are designated the same wavelength in `X` ($\lambda 0$ for `PE0`). All PEs on a chiplet are designated the same wavelength in `Y` ($\lambda 8$ for `Chiplet0`). Although wavelengths are equally divided into two groups in Figure 5 (`X=Y=8`), this is not always true in other configurations of the SPACX architecture. In Section V, we will discuss how SDM technique can be utilized to decouple `X` and `Y` from `N` and `M`, and enable flexible multi-granularity broadcast communication.

C. Interposer Interface and Chiplet Interface

We take interposer and chiplet interfaces located between the global waveguide and local waveguide on `Chiplet0` shown in Figure 6 as an example. As wavelengths $\lambda 0$-$\lambda 7$ are utilized for cross-chiplet broadcast communication, 8 optical tunable splitters are tuned to a split ratio of `1/7` to forward one eighth of power in these wavelengths from the global waveguide to the local waveguide in the

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on October 06,2024 at 09:05:01 UTC from IEEE Xplore. Restrictions apply.

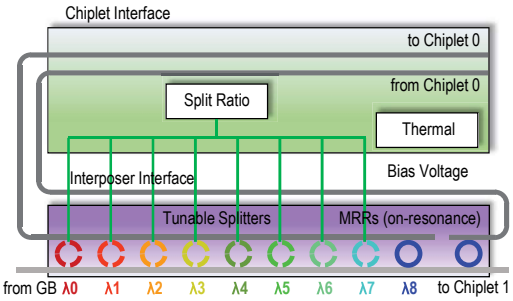


Figure 6. Interposer interface and chiplet interface attached to Chiplet0.

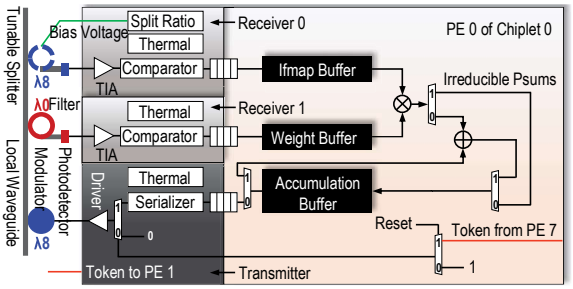


Figure 7. PE architecture and the associated transmitter and receivers, taking PE0 of Chiplet0 as an example.

interposer interface. Meanwhile, since wavelength λ8 is utilized for single-chiplet broadcast communication and PE-to-GB unicast communication on Chiplet0, all power of this wavelength is forwarded from the global waveguide to the local waveguide through an optical filter working at on-resonance. Another optical filter working at on-resonance is utilized to forward the modulated wavelength λ8 from the local waveguide to the global waveguide. The DAC to control the split ratio of all optical tunable splitters and the thermal tuning units are implemented on the chiplet interface.

D. Support for Different Communications

1) Cross-Chiplet Broadcast Communication: In the SPACX photonic network, cross-chiplet broadcast communication is utilized to send the same data from the GB to PEs that are located on different chiplets but share the same position. For example, wavelength λ0 is allocated to the broadcast communication from the GB to PE0 on all chiplets. Data shared by PE0 on all chiplets is used to modulate wavelength λ0 at the GB side. The optical tunable splitters associated with wavelength λ0 in all interposer interfaces are tuned to appropriate split ratios (1/7 split ratio for Chiplet0, 1/6 split ratio for Chiplet1, ..., 1/0 split ratio for Chiplet7), to forward an equal fraction of power of wavelength λ0 to each chiplet, and in the end to each PE0 through a specific local waveguide. Similarly, wavelengths λ1 - λ7 are allocated to broadcast communications from the GB to PE1 - PE7 on all chiplets.

2) Single-Chiplet Broadcast Communication: In the SPACX photonic network, single-chiplet broadcast communication is utilized to send the same data from the GB to all PEs on a chiplet. For example, wavelength λ8 is allocated to the broadcast communication from the GB to all PEs on Chiplet0. Data shared by all PEs on Chiplet0 is used to modulate wavelength λ8 at the GB side. An optical filter on the interposer interface attached to Chiplet0 works at on-resonance to completely forward the power of wavelength λ8 to the local waveguide on Chiplet0. Along the local waveguide, optical tunable splitters in all PEs are tuned to appropriate split ratios (1/7 split ratio for PE0, 1/6 split ratio for PE1, ..., 1/0 split ratio for PE7), to forward an equal fraction of power of wavelength λ8 to each PE. Similarly, wavelengths λ9 - λ15 are allocated to broadcast communications from the GB to Chiplet1 - Chiplet7.

E. PE-to-GB Unicast Communication

The aforementioned cross-chiplet and single-chiplet broadcast communications in the SPACX photonic network only address input data transmission from the GB to PEs. Output data transmission from PEs to the GB is also implemented using global and local waveguides. To reduce implementation complexity and cost, all local PEs on a chiplet share a single wavelength (the same wavelength for single-chiplet broadcast communication, wavelength λ8 for Chiplet0). Output data generated in different local PEs is sequentially transmitted to the GB. A token propagation mechanism is therefore needed to determine which local PE currently has access to the PE-to-GB unicast communication channel. The local PE holding the token can modulate the specific wavelength using its output data to be transmitted (wavelength λ8 for all PEs in Chiplet0). The modulated wavelength is forwarded from the local waveguide to the global waveguide through another optical filter on the interposer interface, and in the end, to the GB. Upon completion of transmission, the token is released and forwarded to the next local PE through the token propagation network. The token is originally held by PE0 on each chiplet after reset. Because of the uniform computation operations across all PEs, the proposed token propagation mechanism exhibits two features. First, conventional token arbitration waveguide in [34] is replaced by a single-bit electrical ring, as the adjacent downstream PE is guaranteed to hold output data ready for transmission upon the completion of output data transmission of current PE. Second, each local PE is allocated an equal-duration time slot for output data transmission. As the local PEs compete for a single wavelength, the communication bandwidth for PE-to-GB unicast communication is relatively low. We address this issue through proper dataflow optimization in Section IV.

F. Other Modules in SPACX Architecture

Figure 7 shows the architecture of PE0 in Chiplet0 as in the SPACX architecture shown in Figure 5. PE0 includes two receivers and one transmitter. Receiver0 is equipped with an optical tunable splitter on wavelength λ8 to receive

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on October 06,2024 at 09:05:01 UTC from IEEE Xplore. Restrictions apply.

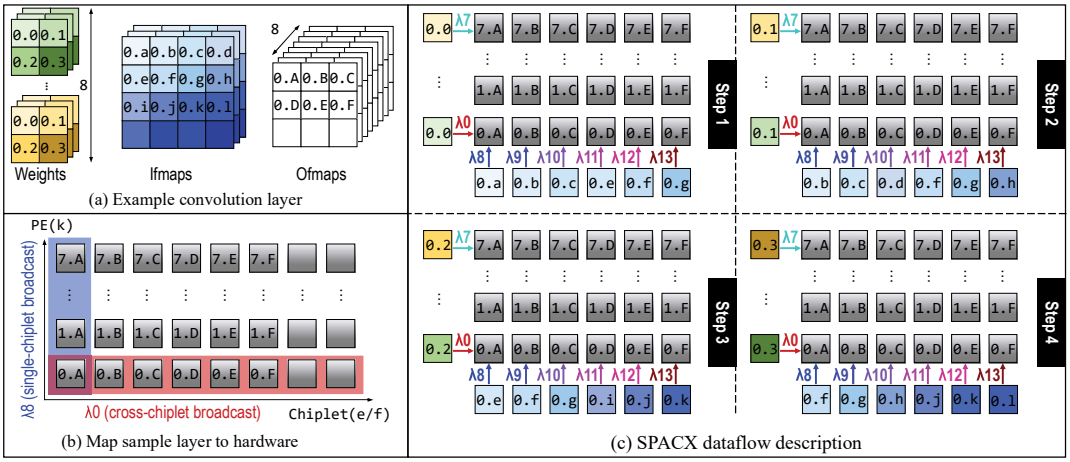


Figure 8. Processing a convolution layer `[r s e f c k]=[2 2 4 4 3 8]` on the SPACX architecture. SPACX dataflow processes output features on the same `e/f` plane on different chiplets, and output features with different `k` values on different PEs on a single chiplet.

data from single-chiplet broadcast communication while `Receiver1` is equipped with an optical filter on wavelength `λ0` to receive data from cross-chiplet broadcast communication. The transmitter is equipped with an optical modulator on wavelength `λ8` for PE-to-GB unicast communication. The SPACX PE architecture includes separate buffers for input features, weights, and psums or output features. A multiply-and-accumulate (MAC) module is responsible for computation operations. `PE0` holds the token upon reset and passes it to `PE1` after the output features stored in the accumulation buffer have been transmitted.

We implement computation units for auxiliary operations such as pooling, activation, and normalization on the GB die as shown in Figure 5, as these operations often demand high communication bandwidth and are fast to process [6], [50]. In this paper, we focus on accelerating convolution and fully-connected layers. An execution controller (similar to the RISC-V processor in [13]) is responsible for offloading computation operations to PEs and orchestrating data communication by tuning the optical tunable splitters on the interposer interfaces and PEs. As PEs in the SPACX architecture process aligned computation operations and incur predetermined data communication, the tuning process is performed offline prior to the execution of a DNN layer. For simplicity, we do not show the control network between the execution controller and the interposer interfaces and PEs in Figure 5. The execution controller also determines the bandwidth allocation scheme discussed in Section VI as it is also achieved by tuning the optical tunable splitters. The tuning latency is set to be 500 `ps` [47].

IV. BROADCAST-ENABLED SPACX DATAFLOW

SPACX dataflow is developed according to three features specifically related to the proposed photonic network. First, the proposed photonic network can support highly energy-efficient orthogonal single-chiplet and cross-chiplet broadcast communications. Second, data exchange between PEs could lead to frequent and costly E/O and O/E signal conversions. Third, the PE-to-GB communication bandwidth in the proposed photonic network is relatively low.

Consider the convolution layer shown in Figure 8 (a) as an example. We present different weight kernels (output channel `k` dimension) with different colors, and label a weight in a specific weight kernel using `A.B` terminology where `A` and `B` represent the input channel in the `c` dimension and the position of this weight in `A` input channel, respectively. Input features are labeled using the same `A.B` terminology as for weights. For output features, `A` in the `A.B` terminology represents the output channel in the `k` dimension while `B` represents the position of an output feature in `A` output channel. Figure 8 (b) describes how the example convolution layer in Figure 8 (a) is mapped to the SPACX architecture shown in Figure 5 to fully exploit the orthogonal single-chiplet and cross-chiplet broadcast capability of the proposed photonic network. We map two rows of output features in an ofmap to different chiplets (`E2=2` and `F2=3` in the dataflow shown in Figure 9) while filling the PEs on each chiplet with corresponding output features in other ofmaps (`K3=8`). As we allocate output features on the same ofmap to different chiplets, cross-chiplet broadcast capability of the proposed photonic network is exploited for weight broadcast. Meanwhile, as we allocate corresponding output features on different ofmaps to PEs on a chiplet, single-chiplet broadcast capability of the proposed photonic network is exploited for input feature broadcast. From the PE point of view, both types of input data required for convolution operations, weights and input features, are transmitted through broadcast communications.

Figure 8 (c) describes the detailed computation operations and data communications involved in one iteration of the `c` loop (`line13`) in Figure 9. Since `R=S=2` according to Figure 8 (a), the computation operations are performed in

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on October 06,2024 at 09:05:01 UTC from IEEE Xplore. Restrictions apply.

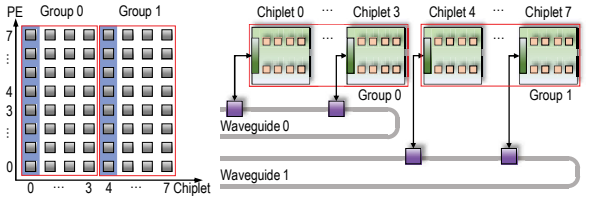
```
1  // package level
2  for e1 = [ 0 : E1 )
3      for f1 = [ 0 : F1 )
4          parallel_for k1 = [ 0 : K1 )
5              parallel_for e2 = [ 0 : E2 )
6                  parallel_for f2 = [ 0 : F2 )
7  // chiplet level
8  for k2 = [ 0 : K2 )
9      parallel_for e3 = [ 0 : E3 )
10         parallel_for f3 = [ 0 : F3 )
11             parallel_for k3 = [ 0 : K3 )
12 // PE level
13 for c = [ 0 : C )
14     for r = [ 0 : R )
15         for s = [ 0 : S )
16             k = k3 + K3 * (k2 + K2 * k1)
17             e = e3 + E3 * (e2 + E2 * e1)
18             f = f3 + F3 * (f2 + F2 * f1)
19             O [k e f] += W [k r s c] * I [r + e - 1 s + f - 1 c]
```

Figure 9. SPACX dataflow nested loop representation.

four steps. We focus on computation operations and data communications related to two PEs labeled `0.A` and `0.F`. Computation operations and data communications related to other PEs can be easily inferred from Figure 8 (c). In `Step1`, weight labeled `0.0` and in the green weight kernel is simultaneously transmitted to PEs labeled `0.A` and `0.F` using cross-chiplet broadcast communication on wavelength $\lambda 0$. Meanwhile, input features labeled `0.a` and `0.g` are transmitted to PEs labeled `0.A` and `0.F` using single-chiplet broadcast communications on wavelengths $\lambda 8$ and $\lambda 15$, respectively. PEs labeled `0.A` and `0.F` perform computation operations and generate psums `0.0`×`0.a` and `0.0`×`0.g`, respectively. Similar data communications and computation operations are performed in the following three steps and the generated intermediate psums are locally accumulated. This concludes the completion of one iteration of the `c` loop. The psums generated in all iterations of the `c` loop are locally accumulated to obtain the output features.

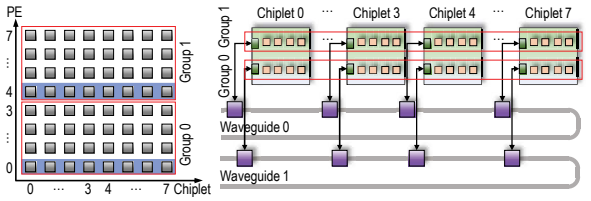
V. BROADCAST GRANULARITY EXPLORATION

Maintaining the mapping algorithm shown in Figure 8 (a) and (b) could lead to low PE utilization for particular convolution layers. For example, consider a convolution layer with parameters `[r s e f c k]=[2 2 2 2 3 16]`, in which the number of output features on an ofmap is `e`×`f=4` while the number of output channels is `k=16`. When mapping this convolution layer to the SPACX architecture shown in Figure 5, we observe that only 4 chiplets are utilized (`e`×`f`<`M`). Meanwhile, the computation operations along the `k` dimension have to be iteratively performed (`k`>`N`) even though there are idle chiplets in the system. To resolve this issue, we divide the chiplets into two groups and perform cross-chiplet broadcast communication within each group instead of across all chiplets in the system. This approach enables `line4` of the SPACX dataflow shown in Figure 9. Figure 10 shows how chiplets are divided into two groups: `Chiplet0` - `Chiplet3` in `Group0` and `Chiplet4` - `Chiplet7` in `Group1`. To realize independent cross-chiplet broadcast communication in each group, one additional global waveguide is implemented as shown in Figure 10



(a) Broadcast groups (b) Network configuration

Figure 10. Limit cross-chiplet broadcast communication in separate groups and the corresponding network configuration (configuration `B` in Table I).



(a) Broadcast groups (b) Network configuration

Figure 11. Limit single-chiplet broadcast communication in separate groups and the corresponding network configuration (configuration `C` in Table I).

(b). In addition to potentially improving PE utilization, this approach also has implications on the number of required wavelengths. The two shaded chiplets (`Chiplet0` and `Chiplet4` in Figure 10 (a)) can share one wavelength for single-chiplet broadcast communication, as they are physically separated and connected to `Waveguide0` and `Waveguide1`, respectively.

Consider another convolution layer with parameters `[r s e f c k]=[2 2 4 4 3 4]`, in which the number of output features on an ofmap is `e`×`f=16` while the number of output channels is `k=4`. This represents an opposite situation as compared to the previous example. When mapping this convolution layer to the SPACX architecture shown in Figure 5, we observe that only 4 PEs on each chiplet are utilized (`k`<`N`). Meanwhile, the computation operations along the `e`/`f` dimensions have to be iteratively performed (`e`×`f`>`M`) even though there are idle PEs on each chiplet. To resolve this issue, we divide the PEs on each chiplet into two groups and perform single-chiplet broadcast communication within each group instead of across all PEs on the chiplet. This approach enables `line9` and `line10` of the SPACX dataflow shown in Figure 9. Figure 11 shows how PEs on each chiplet are divided into two groups: `PE0` - `PE3` in `Group0` and `PE4` - `PE7` in `Group1`. To realize independent single-chiplet broadcast communication in each group, one additional global waveguide and one additional local waveguide on each chiplet are implemented as shown in Figure 11 (b).

Implementing multiple cross-chiplet and single-chiplet broadcast groups enables fine-grained mapping of DNN layers with diverse layer parameters on the SPACX architecture, potentially increasing PE utilization. Furthermore, this approach has implications on system scalability and energy efficiency of data communication. Table I lists four SPACX photonic network configurations corresponding to different broadcast granularities. Configuration `A` is the original

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on October 06,2024 at 09:05:01 UTC from IEEE Xplore. Restrictions apply.

Table I
FOUR DIFFERENT CONFIGURATIONS OF THE SPACX ARCHITECTURE.

Configuration	A	B	C	D
No. of global waveguide	1	2	2	4
No. of local waveguide per chiplet	1	1	2	2
No. of wavelengths	16	12	12	8
No. of PEs per waveguide	64	32	32	16
No. of MRRs in interfaces	80	80	96	96

SPACX photonic network architecture shown in Figure 5. Configurations `B` and `C` are architectures with finer cross-chiplet and single-chiplet broadcast granularities, respectively. Configuration `D` simultaneously achieves fine cross-chiplet and single-chiplet broadcast granularities. Configuration `D`, which can be considered as the combination of configurations `B` and `C`, only exhibits moderate increase in implementation cost (the number of required MRRs). This is because the number of interposer interfaces per chiplet increases to 2 while the number of MRRs on each interposer interface decreases to 6 (4 optical tunable splitters and 2 optical filters). Enabling fine broadcast granularity as in configuration `D` incurs significant decrease in required laser power at the cost of moderate increase in the overall MRR power. This is how the energy efficiency of data communication in the SPACX architecture is improved.

VI. FLEXIBLE BANDWIDTH ALLOCATION

The PE architecture shown in Figure 7 has equal communication bandwidth for weights and input features. However, in many DNN layers, the communication bandwidth demands for weights and input features are different due to several factors such as the layer parameters and data reuse status in PE local buffers. We develop a flexible bandwidth allocation scheme discussed in this section to address this issue.

The proposed scheme achieves bandwidth allocation by tuning the numbers of wavelengths for transmission of weights and input features. In particular, the scheme supports cross-chiplet input feature multicast using wavelengths originally allocated for cross-chiplet weight broadcast. This feature leverages the convolution reuse [1] of input features. As shown in Figure 12, the same input feature `0.f` is utilized to generate output features `0.A`, `0.B`, `0.D`, and `0.E`. Since these output features are generated in `PE0` on `Chiplet0`, `Chiplet1`, `Chiplet3`, and `Chiplet4`, the original wavelength $\lambda 0$ for cross-chiplet broadcast communication can be utilized for cross-chiplet multicast of input feature `0.f` if it is available. To support the cross-chiplet multicast communication, it is necessary to identify the subset of chiplets sharing a particular input feature. From Figure 12 we observe that `min(S,F2)`×`min(R,E2)`×`K1` chiplets share an input feature. We construct the subset of chiplets sharing a particular input feature by tuning the optical tunable splitters on interposer interfaces attached to chiplets outside the subset at off-resonance while tuning the optical tunable splitters on interposer interfaces attached to chiplets in the subset to appropriate split ratios. Along the `e/f` dimensions, cross-chiplet input feature multicast can only be performed sequentially on each specific wavelength in wavelength group `X`. Along the `k` dimension, cross-chiplet input feature multicast can be done in parallel using different wavelengths in wavelength group `X`.

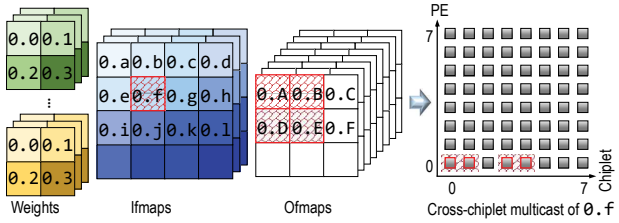


Figure 12. Cross-chiplet input feature multicast using the wavelength originally designated to cross-chiplet weight broadcast.

Similarly, the proposed scheme also supports single-chiplet weight multicast using wavelengths originally allocated for single-chiplet input feature broadcast. This feature leverages the convolution reuse [1] of weights. A particular weight is shared by `E3`×`F3` local PEs in each chiplet and multicast to these PEs using the wavelength originally designated to single-chiplet input feature broadcast. To support the single-chiplet multicast communication, it is necessary to identify the subset of PEs sharing a particular weight. This is done by tuning the optical tunable splitters attached to PEs outside the subset at off-resonance while tuning the optical tunable splitters attached to PEs in the subset to appropriate split ratios. Along the `k` dimension, single-chiplet weight multicast can only be performed sequentially on each specific wavelength in wavelength group `Y`. Along the `e/f` dimensions, single-chiplet weight multicast can be done in parallel using different wavelengths in wavelength group `Y`.

VII. EVALUATION METHODOLOGY

A. Simulation Platform

In order to evaluate the SPACX architecture and other chiplet-based DNN accelerators [13], [30], we extend the open-source MAESTRO simulator [51] to support the non-uniform latency and bandwidth distribution between local PEs and PEs on separate chiplets. The execution time includes computation time and communication time. The extended simulator tracks the number of arithmetic operations and the number of accesses to each memory hierarchy (e.g., local buffer, GB, off-chip DRAM) and derives the computation time and communication time, respectively. The derivation takes the hierarchical network architecture into account and enforces the communication bandwidth limit of each link. The delay for tuning optical tunable splitters is set to be 500 `ps` [47] and included in the communication time. We also assume that the communication time is maximally overlapped by the computation time.

B. Power Model

We evaluate the power consumption of arithmetic operations using `Synopsys Design Compiler`. The power

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on October 06,2024 at 09:05:01 UTC from IEEE Xplore. Restrictions apply.

Table II
NETWORK PARAMETERS

Simba	Chiplet level	Electrical mesh 20 Gbps / PE read / write bandwidth
	Package level	Electrical mesh 320 Gbps / chiplet read / write bandwidth
POPSTAR	Chiplet level	Electrical mesh 20 Gbps / PE read / write bandwidth
	Package level	Photonic crossbar 310 Gbps / chiplet read bandwidth 100 Gbps / chiplet write bandwidth 10 wavelengths, 10 Gbps / wavelength
SPACX	Chiplet level	20 Gbps / PE read bandwidth 10 Gbps / PE write bandwidth (shared)
	Package level	340 Gbps / chiplet read bandwidth 20 Gbps / chiplet write bandwidth 24 wavelengths, 10 Gbps / wavelength

Table III
MODERATE PHOTONIC PARAMETERS

Component	Value	Component	Value
Laser source	5 dB [45]	Ring drop	1 dB [62]
Coupler	1 dB [45]	Ring through	0.02 dB [54]
Splitter	0.2 dB [46]	Photodetector	0.1 dB [45]
Waveguide	1 dB/cm [45]	Waveguide-to-receiver	0.5 dB [63]
Waveguide bend	1 dB [63]	Receiver sensitivity	-20 dBm [45]
Waveguide crossover	0.05 dB [63]	Ring heating	2 mW [59]

Table IV
AGGRESSIVE PHOTONIC PARAMETERS

Component	Value	Component	Value
Laser source	5 dB [45]	Ring drop	0.7 dB [54]
Coupler	1 dB [45]	Ring through	0.01 dB [64]
Splitter	0.2 dB [46]	Photodetector	0.1 dB [45]
Waveguide	1 dB/cm [45]	Waveguide-to-receiver	0.5 dB [63]
Waveguide bend	0.01 dB [65]	Receiver sensitivity	-26 dBm [66]
Waveguide crossover	0.05 dB [63]	Ring heating	320 μW [57]

consumption values of accessing in-package memory hierarchies (local buffer and GB) and off-chip DRAM are obtained using CACTI 6.0 [52] and DRAMSim2 [53], respectively. The power consumption values of metallic-based interconnects are obtained using DSENT [54] and parameters in [55], while the power consumption values of photonic interconnects are derived from Equation (1):

$$P_{total} = P_{TX} + P_{RX} + P_{laser} \quad (1)$$

The overall power consumption $\texttt{P}_{\texttt{total}}$ includes power consumption of transmitter circuitry $\texttt{P}_{\texttt{TX}}$, power consumption of receiver circuitry $\texttt{P}_{\texttt{RX}}$, and laser power $\texttt{P}_{\texttt{laser}}$. We calculate $\texttt{P}_{\texttt{TX}}$ and $\texttt{P}_{\texttt{RX}}$ using the same parameters as in [56], [57] and scale to 28 nm technology [58]. Please note that the power consumption of MRR thermal heating has already been included in $\texttt{P}_{\texttt{TX}}$ and $\texttt{P}_{\texttt{RX}}$. The values of $\texttt{P}_{\texttt{TX}}$ and $\texttt{P}_{\texttt{RX}}$ are 2.9 mW and 2.6 mW, respectively, when a moderate 2 mW [59] MRR thermal heating power consumption is assumed. The laser power $\texttt{P}_{\texttt{laser}}$ includes four parts: photodetector sensitivity $\texttt{P}_{\texttt{rs}}$, overall insertion loss $\texttt{C}_{\texttt{loss}}$, extinction ratio power penalty $\texttt{P}_{\texttt{extinction}}$, and system margin $\texttt{M}_{\texttt{system}}$ as shown in Equation (2):

$$P_{laser} = P_{rs} + C_{loss} + P_{extinction} + M_{system} \quad (2)$$

Table III and Table IV list the moderate and aggressive photonic parameters, respectively, from which the laser power $\texttt{P}_{\texttt{laser}}$ can be derived. We adopt the moderate photonic parameters listed in Table III for power and energy estimation, unless otherwise stated. $\texttt{P}_{\texttt{extinction}}$ represents the power penalty cased by extinction ratio which is assumed to be 2 dB [60]. System margin $\texttt{M}_{\texttt{system}}$ is assumed to be 4 dB [61]. The purpose of the system margin is to allocate a certain amount of power to additional sources of power penalty that may develop during the system lifetime.

C. Chiplet-based DNN Accelerators for Comparison

The proposed SPACX architecture is compared against two other chiplet-based DNN accelerators named Simba [13] and POPSTAR [30]. Simba includes metallic-based interconnects implemented at both package and chiplet levels. To the best of our knowledge, there are no chiplet-based DNN accelerators with pure photonic interconnects. Hence, we select a chiplet-based architecture POPSTAR originally designed for generic applications and replace the CPU chiplets with accelerator chiplets in Simba to construct another chiplet-based DNN accelerator for comparison purpose. We assume that SPACX architecture and the two other chiplet-based DNN accelerators all include M=32 chiplets and N=32 PEs per chiplet. SPACX architecture adopts broadcast granularities of e/f=8 and k=16 unless otherwise stated. To maintain the same PE computation capability, MAC vector width in SPACX is set to be 32. SPACX PE buffer size is 4 kB while the PE buffer size of Simba and POPSTAR is 43 kB [13]. The difference in PE buffer size reflects the SPACX design philosophy of trading data locality for massive broadcast communications. Simulation results shown in Section VIII prove its effectiveness when metallic-based interconnects are completely replaced by photonic interconnects. SPACX GB size is set to be 2 MB, which is the same as in Simba and POPSTAR [13]. Weights and input features are assumed to be 8-bit wide while psums are assumed to be 24-bit wide [13]. The network parameters of the SPACX architecture and two other chiplet-based DNN accelerators are listed in Table II. We attempt to keep the bandwidth values at both package and chiplet levels comparable. However, some bandwidth values cannot be tuned to be exactly the same due to the specific features of different network architectures. The purpose of including Simba and POPSTAR in evaluation is to evaluate the benefits from technology and architecture design, respectively.

D. Benchmarks

We utilize four DNN models in our evaluation, ResNet-50 [3], VGG-16 [2], DenseNet-201 [4], and EfficientNet-B7 [5]. ResNet-50 includes more variations of weight kernel size and computation intensity, while VGG-16 includes more communication-intensive fully connected layers that can test network performance in extreme scenarios. There are 21

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on October 06,2024 at 09:05:01 UTC from IEEE Xplore. Restrictions apply.

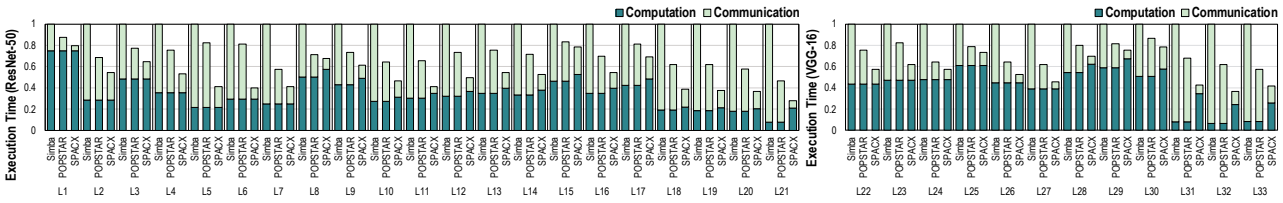


Figure 13. Per-layer execution time breakdown of chiplet-based DNN accelerators for ResNet-50 and VGG-16 DNN models (normalized to Simba).

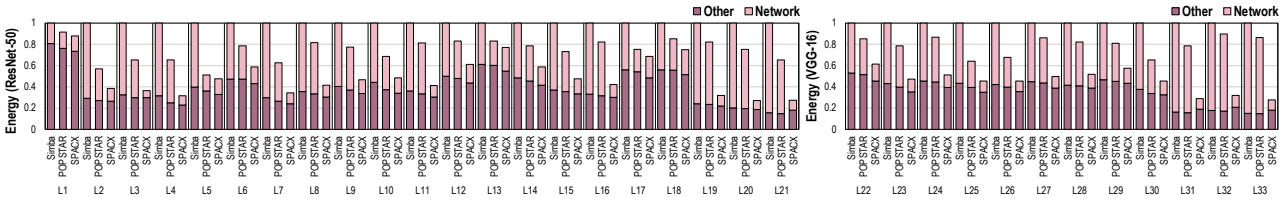


Figure 14. Per-layer energy breakdown of chiplet-based DNN accelerators for ResNet-50 and VGG-16 DNN models (normalized to Simba).

and 12 different convolution or fully connected layers in ResNet-50 and VGG-16. We test all 33 layers in layer-by-layer manner, as each layer has different layer parameters which have implications on performance and energy of the SPACX architecture and other two chiplet-based DNN accelerators. Please note that we have removed redundant layers with the same layer parameters in ResNet-50 and VGG-16. For example, `res2a_branch1` in ResNet-50 has been removed because it has the same layer parameters as `res2[a-c]_branch2c`. In addition, we accumulate the execution time and energy values of all layers to obtain an implication of the overall execution time and energy in a complete inference pass. Please note that only the convolution and fully connected layers are taken into account during the accumulation process. We do not report the layer-by-layer execution time and energy of DenseNet-201 and EfficientNet-B7 due to the large layer counts in these two DNN models.

VIII. EXPERIMENT RESULTS

A. Execution Time and Energy

1) Layer-by-Layer Execution Time and Energy: Figure 13 shows the per-layer execution time comparison of chiplet-based DNN accelerators for ResNet-50 and VGG-16. Execution time is broken down into two parts: computation time including accessing local buffers and performing arithmetic operations, and communication time between PEs and the GB or off-chip DRAM. SPACX achieves 46% and 24% reduction in execution time on average, respectively, when compared to Simba and POPSTAR. The execution time reduction mainly comes from decrease in communication time part. POPSTAR has lower communication time than Simba due to low-latency inter-chiplet communication enabled by the photonic crossbar. SPACX achieves lower communication time than POPSTAR because the proposed photonic network provides low-latency intra-chiplet communication, and also adequate broadcast communication support that alleviates the contention at the GB side. Note that execution time reduction is significant in layers with intensive data communication like the fully connected layers `L21` and `L31`-`L33`. In these layers, the computation time in SPACX is higher than that in Simba and POPSTAR because the small `e/f` values have led to low chiplet utilization.

Figure 14 shows the per-layer energy comparison of chiplet-based DNN accelerators for ResNet-50 and VGG-16. Overall energy can be broken down into two parts: network energy, and energy of the MAC units and memory hierarchy labeled as `other`. SPACX achieves on average 52% and 37% reduction in energy, respectively, as compared to Simba and POPSTAR. The energy reduction is mainly from decrease in execution time and network energy. POPSTAR achieves lower network energy than Simba because of the implementation of energy-efficient photonic interconnects. SPACX achieves lower network energy than POPSTAR due to fewer MRRs and associated heaters required and less frequent E/O and O/E signal conversions. Note that energy reduction is significant in layers with intensive data communication.

2) Overall Execution Time and Energy: In addition to the layer-by-layer estimation of execution time and energy, we present the overall execution time and energy of chiplet-based DNN accelerators for four different DNN models in Figure 15. Unlike in layer-by-layer experiment where each layer is separately executed and data is always initially stored in off-chip DRAM, we exploit the data reuse in the GB between successive layers here. The execution time and energy only account for the convolution layers and fully connected layers in the four DNN models. POPSTAR achieves on average 39% and 28% reduction in execution time and energy, respectively, as compared with Simba. We consider this as benefits from adopting photonics technology. SPACX achieves on average 64% and 65% reduction in execution time and energy when compared with POPSTAR. We consider this as benefits from the proposed architectural design as both architectures adopt photonics technology. SPACX achieves on average 78% and 75% reduction in execution time and energy, respectively, when compared with Simba, indicating the combined effects of technology and architecture innovations of SPACX.

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on October 06,2024 at 09:05:01 UTC from IEEE Xplore. Restrictions apply.

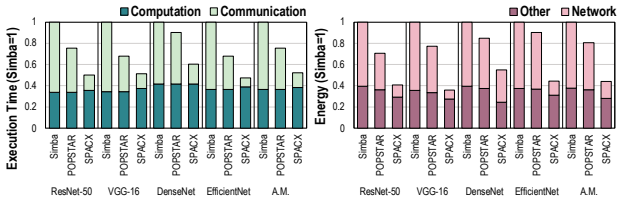


Figure 15. Execution time (left) and energy (right) breakdown of chiplet-based DNN accelerators for a complete inference pass (normalized to Simba).

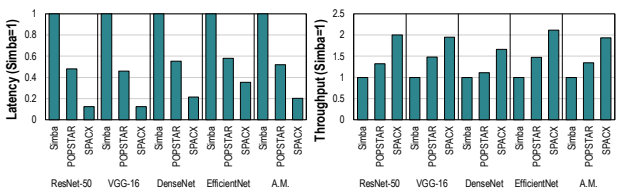


Figure 16. Communication latency (left) and throughput (right) of chiplet-based DNN accelerators for a complete inference pass (normalized to Simba).

B. Network Latency and Throughput

Figure 16 presents the communication latency and throughput comparison of chiplet-based DNN accelerators for four different DNN models. We monitor these two network-related metrics to illustrate the benefits from the proposed photonic network over electrical mesh and photonic crossbar. Latency is the time elapsed between generating and receiving of a data packet, while throughput is defined as the average number of data packets received in a unit time period. As compared to Simba, POPSTAR and SPACX achieve 48% and 80% latency reduction, respectively, which indicates the low communication latency enabled by photonic interconnects. Meanwhile, POPSTAR and SPACX achieve 35% and 93% throughput increase, respectively, as compared to Simba. Low latency and high throughput of SPACX photonic network comply with the significant decrease in communication time of SPACX, compared to POPSTAR and Simba in Figure 15.

C. Impact of SPACX Dataflow

Figure 17 shows the execution time and energy comparison when assuming different dataflows in the SPACX architecture. `WS` is a weight-stationary dataflow [13] with parallel mapping along `k` and `c` dimensions. The orthogonal broadcast feature of the SPACX photonic network is not fully exploited as only parallel mapping along `k` dimension leverages broadcast of input features. Parallel mapping along `c` dimension incurs spatial psum reduction, and leads to frequent and costly E/O and O/E signal conversions. `OS(e/f)` is an output-stationary dataflow [36] with parallel mapping along `e/f` dimensions. The orthogonal broadcast feature is still not fully exploited as parallel mapping along `e/f` dimensions only leverages broadcast of weights. By contrast, SPACX dataflow perfectly matches the underlying photonic network by enabling parallel mapping along `k` and `e/f` dimensions. SPACX dataflow achieves on average 68% and 21% reduction in execution time, respectively, as compared to `WS` and `OS(e/f)`. Energy reduction achieved by SPACX dataflow is 75% and 27%.

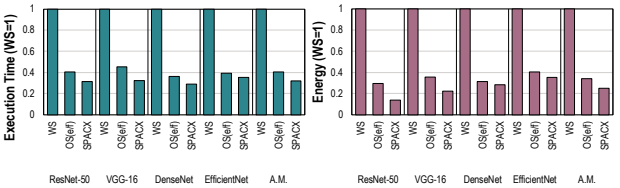


Figure 17. Execution time (left) and energy (right) when applying different dataflows to SPACX architecture (normalized to weight-stationary dataflow).

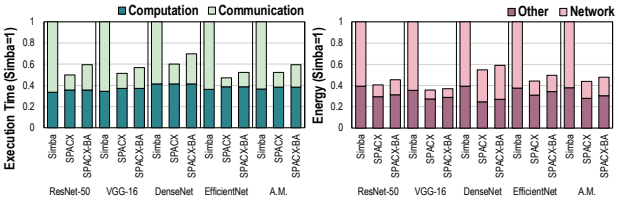


Figure 18. Execution time (left) and energy (right) comparison of SPACX with or without bandwidth allocation (BA) (normalized to Simba).

D. Impact of Bandwidth Allocation Scheme

We evaluate the effectiveness of the bandwidth allocation scheme discussed in Section VI by comparing the execution time and energy of the SPACX architecture with SPACX without bandwidth allocation (labeled `SPACX-BA`) for four different DNN models. From Figure 18 we can observe that, disabling the bandwidth allocation scheme leads to on average 14% increase in execution time due to additional stalls caused by network under-utilization. The impact of the bandwidth allocation scheme on energy is more complicated. On the one hand, it can potentially lead to decrease in energy due to lower execution time. On the other hand, it generates more multicast or even unicast communications, which incurs higher E/O and O/E signal conversion energy.

E. SPACX Network Power and Energy Analysis

1) SPACX Photonic Network Power: Figure 19 and 20 show the overall, laser, and transceiver (MRRs and associated heaters) power consumption of the SPACX photonic network when respective moderate parameters shown in Table III and aggressive parameters shown in Table IV are assumed. We observe significant decrease in overall power, laser power, and transceiver power when aggressive parameters are assumed, indicating the potential power efficiency improvement of the proposed photonic network from advances of related photonic components. By examining the laser power shown in Figure 19 (b) and Figure 20 (b), we find that the minimal laser power is achieved when both single-chiplet broadcast granularity (`k` granularity) and cross-chiplet broadcast granularity (`e/f` granularity) are at 4. Increasing either `k` or `e/f` granularity incurs linear increase in insertion loss, hence exponential increase in laser power. Meanwhile, decreasing either `k` or `e/f` granularity after a certain point also leads to higher laser power because the laser power decrease due to insertion loss decrease is offset by laser power increase due to waveguide duplication. By examining the transceiver power shown in Figure 19 (c) and Figure 20 (c), we find that the minimal

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on October 06,2024 at 09:05:01 UTC from IEEE Xplore. Restrictions apply.

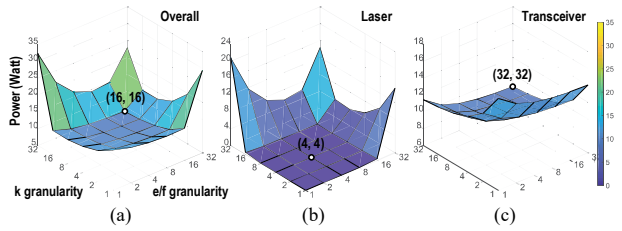


Figure 19. The overall (a), laser (b), and transceiver (c) power consumption of SPACX photonic network when moderate parameters are assumed.

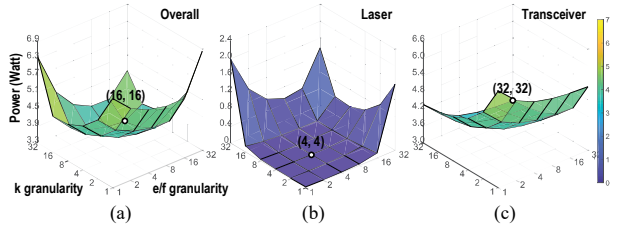


Figure 20. The overall (a), laser (b), and transceiver (c) power consumption of SPACX photonic network when aggressive parameters are assumed.

transceiver power is achieved when both k granularity and e/f granularity are at 32. This is because coarser granularity leads to fewer MRRs required in interposer interfaces, and in the end, lower transceiver power. Since the minimal laser power and transceiver power are achieved at different k and e/f granularities, the aggregated overall power reaches minimal value at another point where k granularity and e/f granularity are both at 16. However, we choose broadcast granularities of e/f=8 and k=16 to achieve balanced improvement on both energy efficiency and execution time.

2) SPACX Photonic Network Energy: Figure 21 (a) shows the energy breakdown of a ResNet-50 inference pass when moderate and aggressive parameters are applied to POPSTAR and SPACX. Using aggressive parameters brings additional 19% and 8% energy reduction for POPSTAR and SPACX, respectively, as compared to Simba. The SPACX photonic network consumes 23.9 mJ and 8.4 mJ for a ResNet-50 inference pass when moderate and aggressive parameters are assumed, respectively, as shown in Figure 21 (b).

F. Scalability Exploration

We examine the the execution time and energy of the SPACX architecture for a ResNet-50 inference pass when varying the number of chiplets (M) and number of PEs per chiplet (N). All values are normalized to the M=32 N=32 SPACX architecture. We make the following observations. First, electrical interconnects can completely offset the benefit from system scaling as we observe increase in execution time when increasing M in Simba. Second, POPSTAR and SPACX both scale better than Simba in terms of execution time and energy due to the photonics technology. Third, POPSTAR consumes higher energy than Simba when the system scale is small because of the costly E/O and O/E signal conversions. Fourth, the energy gap between POPSTAR and SPACX increases as system scales up due to the much more

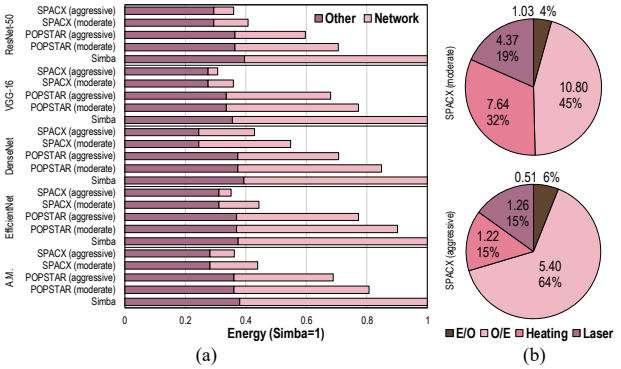


Figure 21. Energy breakdown of chiplet-based DNN accelerators (a) and SPACX photonic network (b) for a ResNet-50 inference pass.

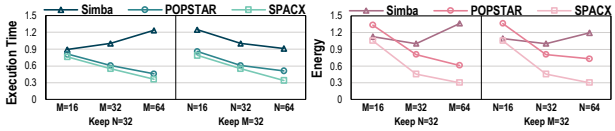


Figure 22. Execution time (left) and energy (right) of SPACX architecture by varying the number of chiplets (M) and number of PEs per chiplet (N).

MRRs and associated heaters required in POPSTAR.

G. Area Estimation

We calculate the SPACX PE area (excluding the transmitter and two receivers) to be 0.72 mm^2 using Synopsys Design Compiler and a 28 nm technology library. We assume that the area for a transmitter or a receiver is 0.0096 mm^2 per wavelength [67]. Hence, the area overhead of the peripheral circuity of transmitter and receivers in a PE is around 4%. There are 132 MRRs underneath a chiplet which has an area of 4.07 mm^2. The overall MRR area is 0.01 mm^2 when assuming 5 μm MRR radius [68]. The overall micro-bump area is 0.68 mm^2 when assuming 4 wires per MRR and 36 μm micro-bump pitch size [69]. As most MRRs and micro-bumps can be implemented under a chiplet, we assume that they do not incur additional area overhead.

IX. CONCLUSION

In this paper, we propose a chiplet-based DNN accelerator with photonic interconnects named SPACX. Salient features of the SPACX architecture include a photonic network that supports seamless orthogonal single-chiplet and cross-chiplet broadcast communications, a tailored dataflow that exploits high-performance and ease of broadcast communication of photonic interconnects to maximize parallelism in DNN inference applications, and a flexible bandwidth allocation scheme that dynamically adjusts communication bandwidths for different types of data based on layer and system parameters obtained prior to the execution of a DNN layer. The combined benefits of these features provide adequate, flexible, and energy-efficient communication support for scalable chiplet-based DNN accelerators. Simulation studies using multiple DNN models show that the SPACX architecture

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on October 06,2024 at 09:05:01 UTC from IEEE Xplore. Restrictions apply.

achieves significant reduction in execution time and energy, and exhibits better scalability, as compared to other state-of-the-art chiplet-based DNN accelerators.

ACKNOWLEDGMENT

This research was partially supported by the National Science Foundation grants CCF-1702980, CCF-1812495, CCF-1901165, CCF-1953980, CCF-1513606, CCF-1703013, and CCF-1901192. We sincerely thank anonymous reviewers for their excellent feedback.

REFERENCES

[1] V. Sze, Y. Chen, T. Yang, and J. S. Emer. Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017.

[2] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, pages 1–14, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017.

[5] M. Tan and Q. V. Le. Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 6105–6114, 2019.

[6] X. Yang and *et. al*. Interstellar: Using Halide's Scheduling Language to Analyze DNN Accelerators. In *Proceedings of the ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 369–383, 2020.

[7] J. Li, A. Louri, A. Karanth, and R. Bunescu. CSCNN: Algorithm-Hardware Co-Design for CNN Accelerators Using Centrosymmetric Filters. In *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 612–625, 2021.

[8] J. Li, A. Louri, A. Karanth, and R. Bunescu. GCNAX: A Flexible and Energy-Efficient Accelerator for Graph Convolutional Neural Networks. In *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 775–788, 2021.

[9] R. Mayer and H. A. Jacobsen. Scalable Deep Learning on Distributed Infrastructures: Challenges, Techniques, and Tools. *ACM Computing Survey*, 53(1):1–37, 2020.

[10] N. P. Jouppi and *et. al*. In-Datacenter Performance Analysis of a Tensor Processing Unit. In *Proceedings of the ACM/IEEE International Symposium on Computer Architecture (ISCA)*, pages 1–12, 2017.

[11] A. Kannan, N. E. Jerger, and G. H. Loh. Enabling Interposer-based Disintegration of Multi-Core Processors. In *Proceedings of the IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 546–558, 2015.

[12] X. Hu, D. Stow, and Y. Xie. Die Stacking Is Happening. *IEEE Micro*, 38(1):22–28, 2018.

[13] Y. S. Shao and *et. al*. Simba: Scaling Deep-Learning Inference with Multi-Chip-Module-Based Architecture. In *Proceedings of the IEEE/ACM International Symposium on Microarchitecture (MICRO)*, page 14–27, 2019.

[14] G. Ascia, V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti. Improving Inference Latency and Energy of DNNs through Wireless Enabled Multi-Chip-Module-based Architectures and Model Parameters Compression. In *Proceedings of the IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, pages 1–6, 2020.

[15] Y. Li, A. Louri, and A. Karanth. Scaling Deep-Learning Inference with Chiplet-based Architecture and Photonic Interconnects. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, pages 931–936, 2021.

[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 (TPDS)*, pages 1–14, 2021.

[17] P. Fotouhi, S. Werner, J. Lowe-Power, and S. J. Ben Yoo. Enabling Scalable Chiplet-based Uniform Memory Architectures with Silicon Photonics. In *Proceedings of the International Symposium on Memory Systems (MEMSYS)*, page 222–334, 2019.

[18] D. A. B. Miller. Device Requirements for Optical Interconnects to Silicon Chips. *Proceedings of the IEEE*, 97(7):1166–1185, 2009.

[19] H. Zheng, K. Wang, and A. Louri. Adapt-NoC: A Flexible Network-on-Chip Design for Heterogeneous Manycore Architectures. In *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 723–735, 2021.

[20] R. Soref. The Past, Present, and Future of Silicon Photonics. *IEEE Journal of Selected Topics in Quantum Electronics*, 12(6):1678–1687, 2006.

[21] K. Bergman, L. P. Carloni, A. Biberman, J. Chan, and G. Hendry. *Photonic Network-on-Chip Design*. Springer, 2014.

[22] K. Shiflett, A. Karanth, R. Bunescu, and A. Louri. Albireo: Energy-Efficient Acceleration of Convolutional Neural Networks via Silicon Photonics. In *Proceedings of the ACM/IEEE International Symposium on Computer Architecture (ISCA)*, pages 860–873, 2021.

[23] Y. Demir, Y. Pan, S. Song, N. Hardavellas, J. Kim, and G. Memik. Galaxy: A High-Performance Energy-Efficient Multi-Chip Architecture Using Photonic Interconnects. In *Proceedings of the ACM International Conference on Supercomputing (ICS)*, page 303–312, 2014.

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on October 06,2024 at 09:05:01 UTC from IEEE Xplore. Restrictions apply.

[24] P. Grani, R. Proietti, V. Akella, and S. J. Ben Yoo. Design and Evaluation of AWGR-based Photonic NoC Architectures for 2.5D Integrated High Performance Computing Systems. In *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 289–300, 2017.

[25] A. Narayan, Y. Thonnart, P. Vivet, C. F. Tortolero, and A. K. Coskun. WAVES: Wavelength Selection for Power-Efficient 2.5D-Integrated Photonic NoCs. In *Proceedings of the Design, Automation and Test in Europe Conference (DATE)*, pages 516–521, 2019.

[26] A. Narayan, Y. Thonnart, P. Vivet, and A. K. Coskun. PROWAVES: Proactive Runtime Wavelength Selection for Energy-Efficient Photonic NoCs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, pages 1–14, 2020.

[27] Y. Kao and H. J. Chao. BLOCON: A Bufferless Photonic Clos Network-on-Chip Architecture. In *Proceedings of the IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, pages 81–88, 2011.

[28] N. Kirman, M. Kirman, R. K. Dokania, J. F. Martinez, A. B. Apsel, M. A. Watkins, and D. H. Albonesi. Leveraging Optical Technology in Future Bus-based Chip Multiprocessors. In *Proceedings of the IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 492–503, 2006.

[29] C. Li, M. Browning, P. V. Gratz, and S. Palermo. LumiNOC: A Power-Efficient, High-Performance, Photonic Network-on-Chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 33(6):826–838, 2014.

[30] Y. Thonnart, S. Bernabé, J. Charbonnier, C. Bernard, D. Coriat, C. Fuguet, P. Tissier, B. Charbonnier, S. Malhouitre, D. Saint-Patrice, M. Assous, A. Narayan, A. Coskun, D. Dutoit, and P. Vivet. POPSTAR: A Robust Modular Optical NoC Architecture for Chiplet-based 3D Integrated Systems. In *Porceedings of the Design, Automation and Test in Europe Conference (DATE)*, pages 1456–1461, 2020.

[31] A. Narayan, Y. Thonnart, P. Vivet, A. Joshi, and A. K. Coskun. System-Level Evaluation of Chip-Scale Silicon Photonic Networks for Emerging Data-Intensive Applications. In *Proceedings of the Design, Automation and Test in Europe Conference (DATE)*, pages 1444–1449, 2020.

[32] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A. Choudhary. Firefly: Illuminating Future Network-on-Chip with Nanophotonics. In *Proceedings of the ACM/IEEE International Symposium on Computer Architecture (ISCA)*, 2009.

[33] A. K. Ziabari, J. L. Abellán, R. Ubal, C. Chen, A. Joshi, and D. Kaeli. Leveraging Silicon-Photonic NoC for Designing Scalable GPUs. In *Proceedings of the ACM International Conference on Supercomputing (ICS)*, pages 273–282, 2015.

[34] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. P. Jouppi, M. Fiorentino, A. Davis, N. Binkert, R. G. Beausoleil, and J. H. Ahn. Corona: System Implications of Emerging Nanophotonic Technology. In *Proceedings of the ACM/IEEE International Symposium on Computer Architecture (ISCA)*, pages 153–164, 2008.

[35] Y. Chen, J. Emer, and V. Sze. Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks. In *Proceedings of the ACM/IEEE International Symposium on Computer Architecture (ISCA)*, page 367–379, 2016.

[36] Z. Du, R. Fasthuber, T. Chen, P. Ienne, L. Li, T. Luo, X. Feng, Y. Chen, and O. Temam. ShiDianNao: Shifting Vision Processing Closer to the Sensor. In *Proceedings of the ACM/IEEE International Symposium on Computer Architecture (ISCA)*, page 92–104, 2015.

[37] S. Chakradhar, M. Sankaradas, V. Jakkula, and S. Cadambi. A Dynamically Configurable Coprocessor for Convolutional Neural Networks. In *Proceedings of the ACM/IEEE International Symposium on Computer Architecture (ISCA)*, page 247–257, 2010.

[38] L. Cavigelli, D. Gschwend, C. Mayer, S. Willi, B. Muheim, and L. Benini. Origami: A Convolutional Network Accelerator. In *Proceedings of the ACM Great Lakes Symposium on VLSI (GLVLSI)*, page 199–204, 2015.

[39] H. J. Yoo, S. Park, K. Bong, D. Shin, J. Lee, and S. Choi. A 1.93 TOPS/W Scalable Deep Learning/Inference Processor with Tetra-Parallel MIMD Architecture for Big Data Applications. In *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)*, pages 80–81, 2015.

[40] T. Chen, Z. Du, N. Sun, J. Wang, C. Wu, Y. Chen, and O. Temam. DianNao: A Small-Footprint High-Throughput Accelerator for Ubiquitous Machine-Learning. In *Proceedings of the ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, page 269–284, 2014.

[41] R. Marchetti, C. Lacava, L. Carroll, K. Gradkowski, and P. Minzioni. Coupling Strategies for Silicon Photonics Integrated Chips. *Photonics Research*, 7(2):201–239, 2019.

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

[43] K. Shiflett, D. Wright, A. Karanth, and A. Louri. PIXEL: Photonic Neural Network Accelerator. In *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 474–487, 2020.

[44] S. Van Winkle, A. Karanth, R. Bunescu, and A. Louri. Extending the Power-Efficiency and Performance of Photonic Interconnects for Heterogeneous Multicores with Machine Learning. In *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 480–491, 2018.

[45] R. Morris, A. Karanth, and A. Louri. Dynamic Reconfiguration of 3D Photonic Networks-on-Chip for Maximizing Performance and Improving Fault Tolerance. In *Proceedings of the IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 282–293, 2012.

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on October 06,2024 at 09:05:01 UTC from IEEE Xplore. Restrictions apply.

[46] S. Werner, J. Navaridas, and M. Luján. Designing Low-Power, Low-Latency Networks-on-Chip by Optimally Combining Electrical and Optical Links. In *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 265–276, 2017.

[47] E. Peter, A. Thomas, A. Dhawan, and S. R. Sarangi. Active Microring based Tunable Optical Power Splitters. *Optics Communications*, 359:311–315, 2016.

[48] J. Bashir, E. Peter, and S. R. Sarangi. A Survey of On-Chip Optical Interconnects. *ACM Computing Survey*, 51(6):1–34, 2019.

[49] M. Sankaradas, V. Jakkula, S. Cadambi, S. Chakradhar, I. Durdanovic, E. Cosatto, and H. P. Graf. A Massively Parallel Coprocessor for Convolutional Neural Networks. In *Proceedings of the IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, pages 53–60, 2009.

[50] M. Gao, J. Pu, X. Yang, M. Horowitz, and C. Kozyrakis. Tetris: Scalable and Efficient Neural Network Acceleration with 3D Memory. In *Proceedings of the ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 751–764, 2017.

[51] H. Kwon, P. Chatarasi, V. Sarkar, T. Krishna, M. Pellauer, and A. Parashar. MAESTRO: A Data-Centric Approach to Understand Reuse, Performance, and Hardware Cost of DNN Mappings. *IEEE Micro*, 40(3):20–29, 2020.

[52] N. Muralimanohar, R. Balasubramonian, and N. P. Jouppi. CACTI 6.0: A Tool to Model Large Caches. *HP laboratories*, 27:1–24, 2009.

[53] P. Rosenfeld, E. Cooper-Balis, and B. Jacob. DRAMSim2: A Cycle Accurate Memory System Simulator. *IEEE Computer Architecture Letters (CAL)*, 10(1):16–19, 2011.

[54] C. Sun, C. O. Chen, G. Kurian, L. Wei, J. Miller, A. Agarwal, L. S. Peh, and V. Stojanovic. DSENT - A Tool Connecting Emerging Photonics with Electronics for Opto-Electronic Networks-on-Chip Modeling. In *Proceedings of the IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, pages 201–210, 2012.

[55] J. M. Wilson and *et. al*. A 1.17 pJ/b 25Gb/s/pin Ground-Referenced Single-Ended Serial Link for Off-and On-package Communication in 16nm CMOS using a Process- and Temperature-Adaptive Voltage Regulator. In *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)*, pages 276–278, 2018.

[56] R. Polster, Y. Thonnart, G. Waltener, J. Gonzalez, and E. Cassan. Efficiency Optimization of Silicon Photonic Links in 65-nm CMOS and 28-nm FDSOI Technology Nodes. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 24(12):3450–3459, 2016.

[57] A. Joshi, C. Batten, Y. Kwon, S. Beamer, I. Shamim, K. Asanovic, and V. Stojanovic. Silicon-photonic clos networks for global on-chip communication. In *Proceedings of the IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, pages 124–133, 2009.

[58] A. Stillmaker and B. Baas. Scaling Equations for the Accurate Prediction of CMOS Device Performance from 180 nm to 7 nm. *Integration*, 58:74–81, 2017.

[59] G. Li, X. Zheng, J. Yao, H. Thacker, I. Shubin, Y. Luo, K. Raj, J. E. Cunningham, and A. V. Krishnamoorthy. 25Gb/s 1V-Driving CMOS Ring Modulator with Integrated Thermal Tuning. *Optics Express*, 19(21):20435–20443, 2011.

[60] C. DeCusatis. *Handbook of Fiber Optic Data Communication: A Practical Guide to Optical Networking*. Academic Press, 2013.

[61] A. V. Krishnamoorthy, R. Ho, X. Zheng, H. Schwetman, J. Lexau, P. Koka, G. Li, I. Shubin, and J. E. Cunningham. Computer Systems based on Silicon Photonic Interconnects. *Proceedings of the IEEE*, 97(7):1337–1361, 2009.

[62] H. Jayatilleka, M. Caverley, N. A. F. Jaeger, S. Shekhar, and L. Chrostowski. Crosstalk Limitations of Microring-Resonator based WDM Demultiplexers on SOI. In *IEEE Optical Interconnects Conference*, pages 48–49, 2015.

[63] R. Morris and A. Karanth. Power-Efficient and High-Performance Multi-level Hybrid Nanophotonic Interconnect for Multicores. In *Proceedings of the IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, pages 207–214, 2010.

[64] S. Pasricha and S. Bahirat. OPAL: A Multi-Layer Hybrid Photonic NoC for 3D ICs. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 345–350, 2011.

[65] 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.

[66] A. Biberman, K. Preston, G. Hendry, N. Sherwood-Droz, J. Chan, J. S. Levy, M. Lipson, and K. Bergman. Photonic network-on-chip architectures using multilayer deposited silicon materials for high-performance chip multiprocessors. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 7(2):1–25, 2011.

[67] Y. Thonnart and *et al.* A 10 Gb/s Si-Photonic Transceiver with 150μW 120μs-Lock-Time Digitally Supervised Analog Microring Wavelength Stabilization for 1Tb/s/mm^2 Die-to-Die Optical Networks. In *Proceeding of the IEEE International Solid-State Circuits Conference (ISSCC)*, pages 350–352, 2018.

[68] G. Li, X. Zheng, H. Thacker, J. Yao, Y. Luo, I. Shubin, K. Raj, J. E. Cunningham, and A. V. Krishnamoorthy. 40 Gb/s Thermally Tunable CMOS Ring Modulator. In *International Conference on Group IV Photonics*, pages 1–3, 2012.

[69] H. Zheng, K. Wang, and A. Louri. A Versatile and Flexible Chiplet-based System Design for Heterogeneous Manycore Architectures. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, pages 1–6, 2020.

Authorized licensed use limited to: Chinese University of Hong Kong. Downloaded on October 06,2024 at 09:05:01 UTC from IEEE Xplore. Restrictions apply.