2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA) | 978-1-6654-3333-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISCA52012.2021.00072

Albireo: Energy-Efficient Acceleration of Convolutional Neural Networks via Silicon Photonics

Kyle Shiflett*, Avinash Karanth*, Razvan Bunescu†, and Ahmed Louri‡
*Ohio University, †The University of North Carolina at Charlotte, ‡George Washington University
Email: *{ks117713, karanth}@ohio.edu, †rbunescu@uncc.edu, ‡louri@gwu.edu

***Abstract*—With the end of Dennard scaling, highly-parallel and specialized hardware accelerators have been proposed to improve the throughput and energy-efficiency of deep neural network (DNN) models for various applications. However, collective data movement primitives such as multicast and broadcast that are required for multiply-and-accumulate (MAC) computation in DNN models are expensive, and require excessive energy and latency when implemented with electrical networks. This consequently limits the scalability and performance of electronic hardware accelerators. Emerging technology such as silicon photonics can inherently provide efficient implementation of multicast and broadcast operations, making photonics more amenable to exploit parallelism within DNN models. Moreover, when coupled with other unique features such as low energy consumption, high channel capacity with wavelength-division multiplexing (WDM), and high speed, silicon photonics could potentially provide a viable technology for scaling DNN acceleration.**

In this paper, we propose Albireo, an analog photonic architecture for scaling DNN acceleration. By characterizing photonic devices such as microring resonators (MRRs) and Mach-Zehnder modulators (MZM) using photonic simulators, we develop realistic device models and outline their capability for system level acceleration. Using the device models, we develop an efficient broadcast combined with multicast data distribution by leveraging parameter sharing through unique WDM dot product processing. We evaluate the energy and throughput performance of Albireo on DNN models such as ResNet18, MobileNet and VGG16. When compared to current state-of-the-art electronic accelerators, Albireo increases throughput by 110 X, and improves energy-delay product (EDP) by an average of 74 X with current photonic devices. Furthermore, by considering moderate and aggressive photonic scaling, the proposed Albireo design shows that EDP can be reduced by at least 229 X.

***Keywords*-optical computing; silicon photonics; hardware acceleration; deep neural networks**

I. Introduction

The breakdown of Dennard scaling [19] coupled with the computation intensity of deep neural networks (DNN) [14] has prompted the development of specialized hardware accelerators for improved performance on DNN inference. These domain-specific accelerators have been shown to greatly outperform general purpose processors that rely on single instruction multiple data (SIMD) and single instruction multiple thread (SIMT) parallelism, which are typically found in central processing units (CPUs) and graphics processing units (GPUs) [54].

As the size and complexity of DNN models continues to increase, hardware accelerators with various optimizations have been proposed to improve throughput and reduce energy consumption. Current accelerator designs employ spatial architectures [7], exploit structured data patterns (e.g. sparsity and quantization) [22], utilize approximate computing schemes [40], and prioritize data movement in near-data processing [11]. This recent shift in the computing paradigm for DNNs has led architects to explore the benefits of emerging technologies such as superconducting logic [27], memristor arrays [49], logic folding [16], and silicon photonics [5] to further improve throughput and energy efficiency of DNN acceleration.

The use of silicon photonics has traditionally been in communication systems, and more recently as an energy-efficient alternative for on-chip interconnects [2]. Photonic interconnects are extremely well suited for collective operations that involve data to be broadcast or multicast [12], since signals can be easily split without the need for replication. Such data distribution is prevalent in convolutional neural networks (CNN), where locally-connected receptive fields of the input volume have shared parameters (kernels) [17]. This is more difficult for electronic-based data distribution where broadcasts incur high energy costs [30], which is often circumvented through spatial or local reuse of data [54].

With the introduction of energy-efficient microring resonators (MRR) and tunable Mach-Zehnder modulators (MZM), photonic interconnects can deliver high modulation speeds (>50 GHz) [43], consume low energies (<70 fJ/bit) [18], [59], and provide high channel capacities via wavelength-division multiplexing (WDM). As silicon photonics continues to mature, it is becoming increasingly promising as an alternative to digital electronics for high-speed and energy-efficient computation. The inherent parallelism of optics provides the high bandwidth density necessary to efficiently scale computation for DNNs [9]. Silicon photonics has the potential for $O(N)$ energy scaling for $O(N^2)$ fixed-point operations, whereas energy tends to scale with $O(N^2)$ for digital systems [41]. The combination of high-speed low-energy devices and fundamental parallelism of optics potentially makes silicon photonics the next

978-1-6654-3333-4/21/$31.00 ©2021 IEEE
DOI 10.1109/ISCA52012.2021.00072

scalable solution that efficiently unites both computation and data movement [37]. It must be noted that photonic devices do not provide a drop-in replacement for digital components. Rather, they have their own set of properties that yield a completely different, yet potentially beneficial, computation structure that can be exploited for DNNs.

There has been significant research in the past few years where several groups have proposed photonic technology for implementing both spiking neural networks (SNN) and DNNs [5], [36], [45], [52], [55]. Prior works have insufficiently characterized device precision limitations due to crosstalk and noise thus far, underscoring the need for a detailed exploration before designing system-level DNN accelerators with photonics. These recently proposed photonic architectures accelerate generic dot products, but have not fully exploited the potential speedup by leveraging the shared parameters found in CNNs.

In this paper, we propose **Albireo**, an analog photonic architecture for scaling DNN acceleration. By characterizing photonic devices such as MRRs and MZMs using photonic simulators, we develop a realistic expectation of device capability for system level acceleration. The precision limitations imposed by MRRs and MZMs are quantified, which is the driving factor for the Albiero architecture. We present a set of photonic computation schemes that naturally exploit the shared parameters and multicast data distribution found in CNNs, which reduces energy consumption and significantly increases throughput for CNN applications. Albireo implements an efficient broadcast combined with multicast data distribution, and leverages parameter sharing through unique WDM dot product processing in the proposed photonic locally-connected units (PLCU). Albireo improves upon prior work by not only leveraging shared parameters, but by doing so through passively overlapping receptive fields, significantly increasing computation parallelism.

We evaluate the energy and throughput performance of the proposed Albireo architecture on CNN models such as ResNet18 [24], MobileNet [26], and VGG16 [53], and compare the results with both photonic and electronic accelerators. We evaluate three Albireo designs: a conservative (C) estimate using current photonic devices that have been demonstrated, a moderate (M) estimate using future photonic device parameters that yields comparable energy efficiency to current state-of-the-art electronic accelerators, and an aggressive (A) estimate that makes Albireo a high performance successor to current electronic accelerators. When compared to low-power state-of-the-art electronic accelerators like Eyeriss [7], ENVISION [39], and UNPU [32], Albireo-C improves throughput by at least 20 X (110 X on average), while improving EDP performance by 74 X on average. Albireo-M estimates further improve performance by reducing EDP by an average of 275 X. With Albireo-A estimates, throughput is improved by an average of 177 X, and EDP is improved by at least 229 X (690 X on average).

The major contributions of this paper are as follows:

- **Characterization of Photonic Device Models:** We develop detailed MRR and MZM device configurations using a combination of extensive modeling and simulation in Lumerical INTERCONNECT. These models are used to evaluate the crosstalk and noise margins for the proposed optical subsystems of the hardware accelerator, which determines the precision levels that can be achieved for computation.
- **Photonic Hardware Accelerator:** We propose an analog photonic accelerator that implements an efficient broadcast combined with multicast data distribution, and leverages parameter sharing through unique WDM dot product processing in photonic locally-connected units (PLCU). The use of MZMs for multi-wavelength multiplication and star couplers for multicasting makes Albireo distinct from other photonic accelerators, and improves convolution energy efficiency and latency.
- **Performance Evaluation:** We evaluate the energy and throughput performance of Albireo on CNN models like ResNet18 [24], MobileNet [26], and VGG16 [53], and compare the results with both photonic and electronic accelerators. As silicon photonics is still maturing, we provide estimates that take into account current as well as future device projections.

This paper is organized as follows: In Section II-A, we provide background on the convolution operation; in Section II-B, we explain optical arithmetic operations; in Section III, we describe the proposed Albireo architecture; in Section IV, we analyze the performance of Albireo; in Section V, we discuss the related work; and in Section VI, we conclude the paper.

II. Preliminaries and Background

A. Convolution Operation

In this section, we give a brief description of the convolution operation. Each convolution layer in a CNN performs the convolution operation, which is a series of dot products between kernels W and receptive fields in the input volume A. An input volume is a three-dimensional data structure with width A_x, height A_y, and depth A_z. A receptive field is a region of the input volume where the kernel is applied. Each two-dimensional width-height slice is a channel in the input volume, and the number of channels is equivalent to A_z. Each dot product between a kernel and a receptive field in the input volume produces an activation in the output volume B, and the application of a kernel over the entire input volume yields a two-dimension feature map. Each element of a kernel is referred to individually as a weight. The depth B_z of the output volume is equal to the number of feature maps, which is equal to the number of kernels W_m. The kernels are applied with stride S, which is the number of elements that the kernel is moved in a single dimension

from one dot product to the next. Assuming square inputs and receptive fields (dimension $x = y$), the dimensions of a feature map are:

$$B_x = B_y = \left\lceil \frac{A_x - W_x + 2P_x}{S_x} \right\rceil + 1 \qquad (1)$$

where P is the zero padding of the input volume. The shape of the output volume is $B_x \times B_y \times W_m$. Figure 1 shows the convolution between two kernels and an input volume.

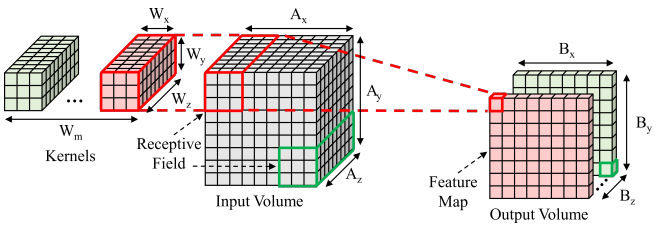

Figure 1. **Convolution operation between kernels and an input volume.**

Algorithm 1 describes the convolution operation that occurs in a single layer of a CNN. The square brackets in Algorithm 1 index elements along a dimension. The dimensionality is as follows: $A[z][y][x]$, $W[m][z][y][x]$, and $B[z][y][x]$. We use the indexing operator ":", where [:] means all indices along that dimension, and [x:y] means indices x to y-1. The function f is a nonlinear activation function, commonly the rectified linear unit (ReLU).

Algorithm 1 Convolution operation for a single input volume

```
1: function CONV(A, W)
2:     for m ← 0; step 1; while m < W_m do
3:         y_B ← 0
4:         for y_A ← 0; step S; while y_A < A_y do
5:             x_B ← 0
6:             for x_A ← 0; step S; while x_A < A_x do
7:                 a ← A[:][y_A : y_A+W_y][x_A : x_A+W_x]
8:                 w ← W[m][:][:][:]
9:                 B[m][y_B][x_B] ← f(a · w)
10:                x_B ← x_B+1
11:            end for
12:            y_B ← y_B+1
13:        end for
14:    end for
15: end function
```

B. Optical Arithmetic

In this section, we explain how analog multiplication and addition are performed with photonic devices, and how these devices are utilized to compute optical dot products. We also quantify the precision limitations that ring-based photonic processors encounter due to optical crosstalk and noise, which is the driving factor for the Albireo architecture.

The basic implementation of optical computation in Albireo is as follows. Data is represented in the optical domain with the optical power amplitudes of signals, and these optical signals are routed through the chip using silicon waveguides. The optical signals are modulated to carry certain input operands to the photonic devices that perform computation. The resulting outputs of these computations are generated through attenuation and combination of various optical signals. Output signal data is captured by photodiodes (PD), which convert optical signals into an electrical current that is directly proportional to the incident optical power.

1) Optical Multiplication: Multiplication is achieved by scaling the optical power of a signal, which is easily done by attenuating the signal if the multiplier is less than one. Scaling a signal by a multiplier greater than one would require supplementary optical power to be introduced from additional laser sources. To minimize laser power consumption, optical signals should be multiplied by values (kernel weights) in the [0,1] interval, and the output optical power of a photonic multiplier will be $0 \leq P_{\text{out}} \leq P_{\text{in}}$.

To perform optical multiplication, we use a Mach-Zehnder modulator (MZM) with the layout shown in Figure 2(a). The MZM is able to multiply a signal through destructive interference, which occurs by the upper arm of the device applying a differential phase shift $\Delta\phi$ to half of the input signal. The phase shifter is a doped junction, and an applied voltage causes a phase shift through a refractive index change (plasma dispersion effect). The output power of the MZM is:

$$P_{\text{out}} = \frac{P_{\text{in}}}{2} + \frac{P_{\text{in}}}{2}\angle\Delta\phi \qquad (2)$$

where $0 \leq \Delta\phi \leq \pi$. For example a $\Delta\phi = \pi$ phase shift is a multiply by 0, and a $\Delta\phi = 0$ is a multiply by 1.

MZMs have the advantage of being wavelength independent as long as the path lengths of both arms are equal, and as long as the Y-branches have a broadband response for the range of input wavelengths. When utilizing WDM as shown in Figure 2(b), a MZM is capable of multiplying several input signals by the same kernel weight in parallel. WDM is possible since different wavelengths do not interfere.

2) Optical Addition: Addition is achieved by combining multiple optical signals into a single waveguide, where each optical signal is carried on a different wavelength. The combined optical power of each signal is represented as the addition of their individual optical powers. By having each signal carried on a separate wavelength, destructive interference is avoided in the combination waveguide. The combination waveguide is then fed to a PD. The PDs used in Albireo detect the total optical power across all input wavelengths, so the addition of the optical signal powers is converted to an electrical current.

We utilize microring resonators (MRR) for optical addition, which are closed-loop waveguides shown in Figure 2(c). MRRs are wavelength filters that perform addition by demultiplexing/multiplexing selective wavelengths into a single waveguide. The resonant wavelength is coupled

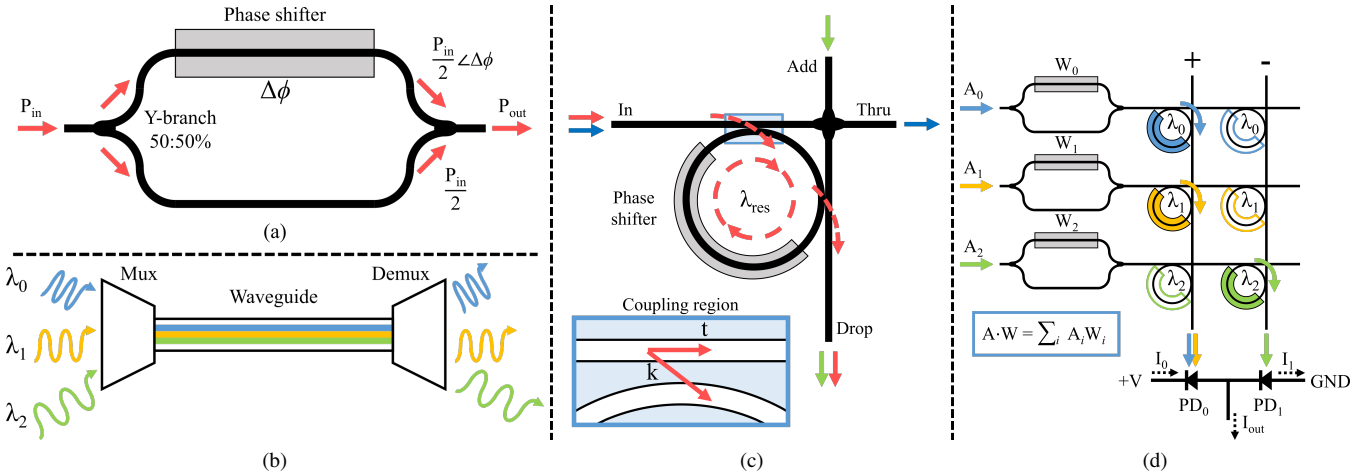

Figure 2. **Photonic devices for optical arithmetic: (a) Mach-Zehnder modulator. (b) Waveguide showing wavelength-division multiplexing. (c) Double-bus microring resonator. (d) Optical dot product operation, showing accumulation for positive and negative weights.**

into the MRR from the *In* port and is then coupled to the *Drop* port waveguide. Signals on other wavelengths continue propagating unimpeded by the MRR to the *Thru* port.

The resonant wavelength is a function of the effective refractive index of the waveguide n_{eff}, the ring's circumference L, and m whole number of wavelengths that fit within the ring [6]:

$$\lambda_{\text{res}} = \frac{n_{\text{eff}} L}{m}, \quad m \in \mathbb{Z}^+ \tag{3}$$

MRRs can also modulate signals through the plasma dispersion effect, since $\Delta\lambda_{\text{res}} \propto \Delta n_{\text{eff}}$. They may be "turned off" by applying a voltage and causing λ_{res} to shift out of resonance with a signal, allowing that signal to pass the ring without getting coupled.

3) Optical Dot Products: Dot products and the fundamental MAC operation constitute the convolution operation, which is achievable with photonics by using MZMs for multiplicaton and MRRs for accumulation. The architecture shown in Figure 2(d) computes the optical dot product between the optical input vector A and the MZM weight vector W, where each input signal is carried on a different wavelength.

Each MZM multiplies one input signal A_i by a weight W_i, and the resulting optical powers are combined on one of the two accumulation waveguides, which are needed to perform the addition of positive and negative signals. MRRs switch the multiplied signals depending on the applied weight, whether it was positive or negative. Input signals multiplied by a positive weight are accumulated on the positive waveguide, and signals multiplied by a negative weight are accumulated on the negative waveguide. The PDs at the end of each waveguide convert the incident optical power into an electric current. The balanced PD arrangement shown in Figure 2(d) will produce a positive current for the positive waveguide's signals, and a negative current for the negative waveguide. Subtraction is required to complete the dot product, and is achieved with the difference of these two currents, shown as I_{out}. The output current is:

$$I_{\text{out}} = R_0 \sum_i P_i^+ - R_1 \sum_j P_j^- \tag{4}$$

where R_0 and R_1 are the responsivity (units of A/W) of the photodiodes PD_0 and PD_1, respectively. P_i^+ is the optical power of a positively-weighted signal, and P_j^- is the power of a negatively-weighted signal. $R_0 = R_1$ for all designs in this paper.

C. Limited Precision of Photonic Dot Products

1) Noise Limitations: Noise can be introduced into the photonic computation from many sources, however we outline just the main sources in this section. The first source of noise is relative intensity noise (RIN), which is the normalized optical power fluctuations from the laser sources. RIN is described by a power spectral density (PSD) in units of decibels relative to the carrier per hertz (dBc/Hz). RIN will introduce noise in the current output at the PDs. The second noise source is shot current. Shot noise is a discrete event and follows a Poisson probability distribution, but for high event rates it is well approximated by a normal distribution [10]. The shot current is:

$$I_{\text{shot}} = \mathcal{N}(0,\, 2q_e I_{\text{PD}} \Delta f) \tag{5}$$

where q_e is the elementary charge, I_{PD} is the current of the photodiode, and Δf is the bandwidth. The third noise source is Johnson-Nyquist (thermal) noise:

$$I_{\text{therm}} = \mathcal{N}(0,\, \frac{4k_B T}{R_{\text{f}}} \Delta f) \tag{6}$$

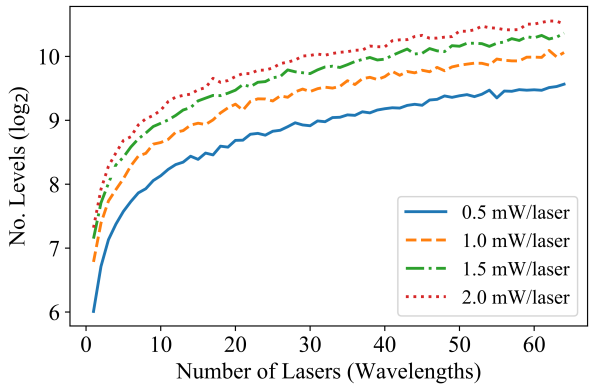

Figure 3. **Noise analysis of for photonic dot products independent of crosstalk. Noise has a much lower influence on precision than crosstalk.**

where k_B is the Boltzmann constant, T is the temperature, and R_f is the feedback resistance of the transimpedance amplifier (TIA) that converts the PD current into a voltage.

Noise causes variations in the accumulated signals, which decreases the number of discernible optical power amplitudes (levels). The number of levels indicates the MAC precision that the system can support. With parameters $\Delta f = 5\,\mathrm{GHz}$, $T = 300\,\mathrm{K}$, and $\mathrm{RIN} = -140\,\mathrm{dBc/Hz}$, we found that RIN contributes the least to the total noise with typical photonic circuit laser powers. This means that increasing the input optical power from the lasers can increase the precision of the system. Precision is gained by increasing laser power until RIN surpasses shot and thermal noise.

We evaluated system noise independent of crosstalk, and plotted precision versus the number of wavelengths for increasing laser power in Figure 3. The diminishing returns for increasing laser power are clearly seen, and we found that 10 bits of precision is achievable with a 2 mW optical laser source with as few as 20 wavelengths. While we use the terminology "bits of precision" for analog photonic computation, what we are actually describing is the $\log_2$ of the number of separable optical power amplitudes at the output. For example, if there are 450 separable optical power amplitudes at the output, $\log_2(450) \approx 8.81$, which implies that the system fully supports 8 bits of precision without error. Computation would become approximate at 9 bits of precision due to some output value distributions overlapping a decision threshold. While these output value distributions may overlap a decision threshold with a small probability, it cannot be guaranteed that an error will not occur, and therefore the computation must be considered approximate. We must also take crosstalk between MRRs into account when determining the precision, which we discuss in the next subsection.

2) MRR Crosstalk Limitations: The amount of crosstalk between MRRs will dictate the precision that Albireo can support. A MRR's transmission repeats at wavelengths that fit a whole number of times in the ring, and the spacing of resonances is the free spectral range (FSR):

$$\mathrm{FSR} = \frac{\lambda_\mathrm{res}^2}{n_\mathrm{g} L} \tag{7}$$

where n_g is the group refractive index of the ring [6]. WDM systems that use MRRs must operate within this FSR, which imposes a limit on the number of wavelengths that can be accumulated by a series of MRRs.

A wider FSR would reduce crosstalk between MRRs, but decreasing the ring's circumference to increase the FSR also increases the resonance full width at half max (FWHM). The density of signals must be considered, which is indicated by finesse:

$$\mathrm{Finesse} = \frac{\mathrm{FSR}}{\mathrm{FWHM}} \tag{8}$$

Finesse is constant regardless of L in an ideal (lossless) MRR. Finesse can be increased independent of L by tuning the power coupling coefficients, which will decrease the FWHM without affecting the FSR. The FWHM of a double-bus MRR is:

$$\mathrm{FWHM} = \frac{(1 - t_1 t_2 a)\lambda_\mathrm{res}^2}{\pi n_\mathrm{g} L \sqrt{t_1 t_2 a}} \tag{9}$$

where t^2 is the power transmission coefficient of the coupling region, a^2 is the single-pass amplitude transmission in the ring, and $a^2 = e^{-\alpha L}$, where α is the loss per unit length [6]. In an ideal MRR ($a = 1$), t^2 is related to the power cross-coupling coefficient k^2 by $k^2 + t^2 = 1$. The power cross-coupling coefficient represents the fraction of optical power coupled into the ring resonator from the *In* port, or the fraction of optical power coupled to the *Drop* port from the ring. These coupling coefficients are determined by the gap between the coupled waveguides and length of the coupling region. Note that there are two coupling regions for the double bus ring used in this design, each with its own transmission coefficient, as indicated by t_1 and t_2 in Equation 9. We use $k_1 = k_2$ in the rings, since this symmetric coupling criterion yields critical coupling [6].

Lowering k^2 decreases the FWHM of the MRR (Figure 4(a)), which reduces the amount of crosstalk from adjacent resonant wavelengths. Lowering crosstalk amplitude increases the number of distinguishable optical amplitudes in the system. Reducing k^2 also increases the MRR's finesse, which allows for more signals to fit within the FSR. There are temporal consequences for decreasing k^2 (Figure 4(b)), and a signal will undergo considerable loss if the MRR modulation frequency is too high. The number of discernible optical levels is the precision that the system can support, which is represented in base 2 to get an indication of bit precision in Figure 4(c).

Reduced model precision like 8-bit integer quantization is common among energy-efficient architectures, which has been shown to yield competitive accuracy for computer

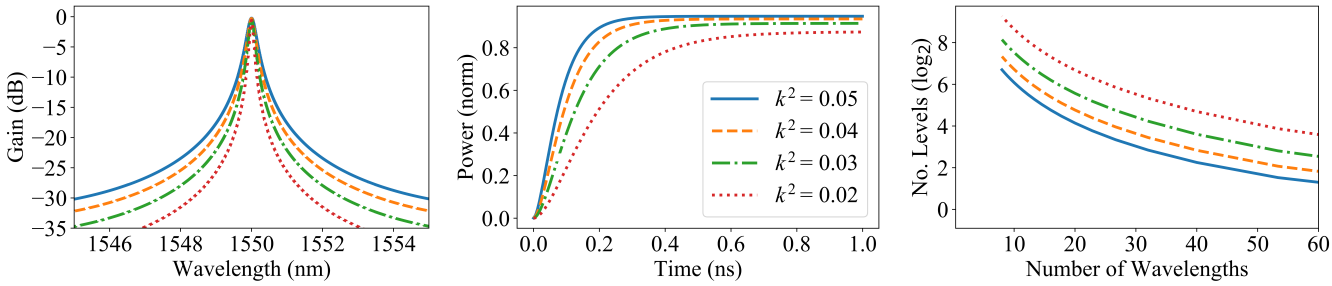

Figure 4. **MRR k^2 design space exploration. (a) The optical power spectrum at the *Drop* port of an MRR. (b) Temporal response at the *Drop* port of an MRR. (c) Precision versus number of wavelengths for an MRR accumulator.**

vision tasks while improving inference time and energy consumption [28]. From Figure 4(c), both $k^2 = 0.02$ and $k^2 = 0.03$ can support 8 bits of precision for a small number of wavelengths, but $k^2 = 0.02$ has poor temporal response in Figure 4(b). For around 20 wavelengths, $k^2 = 0.03$ can support 6 bits of precision, but this is only for positive accumulation. With the inclusion of the negative waveguide, a photonic dot product is able to increase its bit precision by about 1 bit because it is doubling the number of values represented without increasing the number of wavelengths in the FSR (given some additional crosstalk). This means that 7 bits is the worst case precision for $k^2 = 0.03$ with 20 wavelengths, which guides decisions for the proposed Albireo subsystems.

The kernel weights in a neural network layer follow a bell-shaped distribution [23], so there will be more crosstalk around the mean of the distribution, and less crosstalk for the tails of the distribution. An MRR accumulator could possibly support more optical power levels, since more "important" or more "influential" features are weighted higher (in the tails of the distribution) than others.

III. Albireo Architecture Design

This section provides the details of the subsystems in the proposed Albireo architecture. We discuss how the processing units are grouped with supplementary electronics, and detail the broadcast combined with multicast dataflow for efficient convolution partitioning across the processing groups.

A. Photonic Locally-Connected Unit

The MAC architecture introduced in Figure 2(d) does not utilize the MZM's ability to multiply several wavelengths at once. This returns to the concept of parameter sharing in CNNs, where kernels are applied across the entire input volume, and several inputs are multiplied by same kernel weight. Parameter sharing is implemented with MZMs, which will compute on several receptive fields concurrently. There will be an associated output element for each receptive field concurrently processed, which means multiple dot products will be computed in parallel. This increases the number of wavelengths multiplied by each MZM, and increases the number of MRRs needed to accumulate each output. Introducing more wavelengths and receptive fields into a photonic dot product processor will expand the ring resonators into a crossbar-like grid. There will be a balanced PD output for each receptive field simultaneously processed. This is best explained through an illustration in Figure 5(a), which shows the photonic locally-connected unit (PLCU), the basic building block in the Albireo architecture.

A PLCU has shape $N_m \times N_d$, where N_m is the number input waveguides, and N_d is the number of balanced PD outputs. A PLCU has N_m MZMs and $2N_mN_d$ switching MRRs arranged in a grid. We design the PLCU with $N_m = 9$ input waveguides since a common shape for CNN kernels is 3×3, and allows the PLCU to hold an entire channel of the kernel's weights in the MZMs. Kernel shapes other than $W_x \times W_y = N_m$ are still compatible with this architecture. For example, a kernel with $W_x \times W_y > N_m$ will not completely fit in the PLCU's MZMs, and will therefore require additional cycles to complete the dot product.

A larger number of wavelengths increases the amount of parallel computation that can take place in each PLCU, but also increases crosstalk and causes a reduction in precision. The number of wavelengths in a PLCU is $W_y(N_d+W_x-1)$, assuming a square kernel and $W_x \times W_y = N_m$. Our goal is at least 7 bits of precision with $k^2 = 0.03$ for temporal performance, which is achievable at around 20 wavelengths as discussed in Section II-C. The PLCU is designed with $N_d = 5$, which yields 21 total wavelengths with $N_m = 9$. The proposed $N_m \times N_d$ PLCU is shown in Figure 5(a).

Each PLCU processes a single channel of the convolution, and computes N_d concurrent receptive fields. The inputs for a single cycle computation with a stride of 1 are shown in Figure 5(a). Each color represents a different row in the input volume, and overlapping receptive fields in each row produce a multicast pattern since multiple input elements are

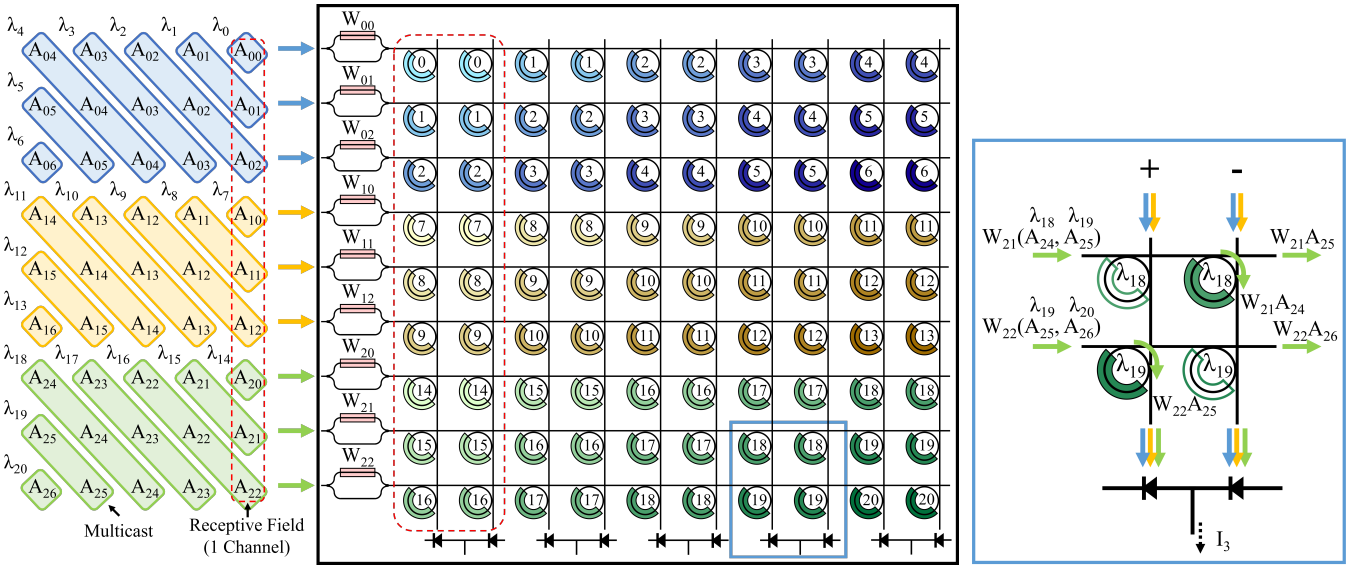

Figure 5. **(a) The proposed PLCU with $N_m = 9$ and $N_d = 5$. The overlapping receptive fields produce a multicast pattern. (b) Select signals being switched by MRRs and accumulated at the balanced PDs.**

subject to the same kernel weights.

These inputs correspond to an input field with shape $W_y(N_d + W_x - 1)$, where A_{ij} indicates the input element at row i and column j, W_{ij} is the kernel weight at row i and column j for the same channel. A single 3×3 receptive field and the corresponding accumulation waveguides are shown in the dashed red border in Figure 5(a). Figure 5(b) details the filtering and switching of the MRRs for select signals, where W_{21} is negative and W_{22} is positive.

B. Photonic Locally-Connected Group

Even though a PLCU is constrained to 21 wavelengths, a larger number of wavelengths ($\geq$64) can be supported by on-chip networks for data distribution [58]. This allows the clustering of multiple PLCUs into a photonic locally-connected group (PLCG) to process multiple channels of the input volume in parallel. Each PLCU in the PLCG operates on a set of inputs that fall into a separate FSR. We assume data distribution can support 64 wavelengths, and with 21 wavelengths per PLCU, that gives $N_u = 3$ PLCUs that can be clustered into a group with a total of 63 wavelengths.

Given that a PLCG contains N_u PLCUs and processes N_u channels in parallel, each cycle will produce N_d partial outputs that need to be aggregated over $\lceil \frac{W_x}{N_u} \rceil$ cycles to complete the dot product. This depth-priority processing is shown in Figure 7, and is further discussed in Section III-C. This creates no partial sum writes back to memory since the entire dot product is aggregated before the kernel is moved and applied to another set of receptive fields. This is beneficial since data movement can consume magnitudes more energy than computation [25]. The PLCG's stationary accumulation of partials causes writes to memory only when the entire activation is complete. The partial sums that are created are repetitively added and registered in the PLCG's aggregation unit, and the layout of a PLCG is shown in Figure 6(b).

One of the obstacles that silicon photonics currently faces is data storage. Although photonic memories have made significant progress over the past decade [3], there is no robust replacement for digital buffering. This means aggregation of partials must be done in electronics and requires optical-to-electrical (O/E) conversion. O/E conversion is implemented first at the PDs, which as discussed in Section II-B, converts the accumulated optical powers into a directly-proportional electrical current. This current is then fed to a transimpedance amplifier (TIA), which acts as a current-to-voltage converter and amplifies the induced current signal to a suitable voltage level for further processing. This analog voltage signal is then converted into a digital value with an analog-to-digital converter (ADC), and aggregated over a number of cycles in the digital domain. When the dot products are complete, i.e. all partials have been aggregated for the output element, the digital values have the ReLU activation function applied and are output to memory. The PDs, TIAs, ADC, adder, and activation subcomponents of the aggregation unit are shown in Figure 6(b). A PLCG's aggregation unit has N_d TIAs and adders.

C. Data Distribution and Albireo Chip

The input signal multicast in each PLCU is implemented using star couplers, which is free propagation region that mixes several inputs [44]. Star couplers physically broadcast all inputs to all output ports, however Albireo utilizes these signals in a multicast manner. For example, in Figure 5(a)

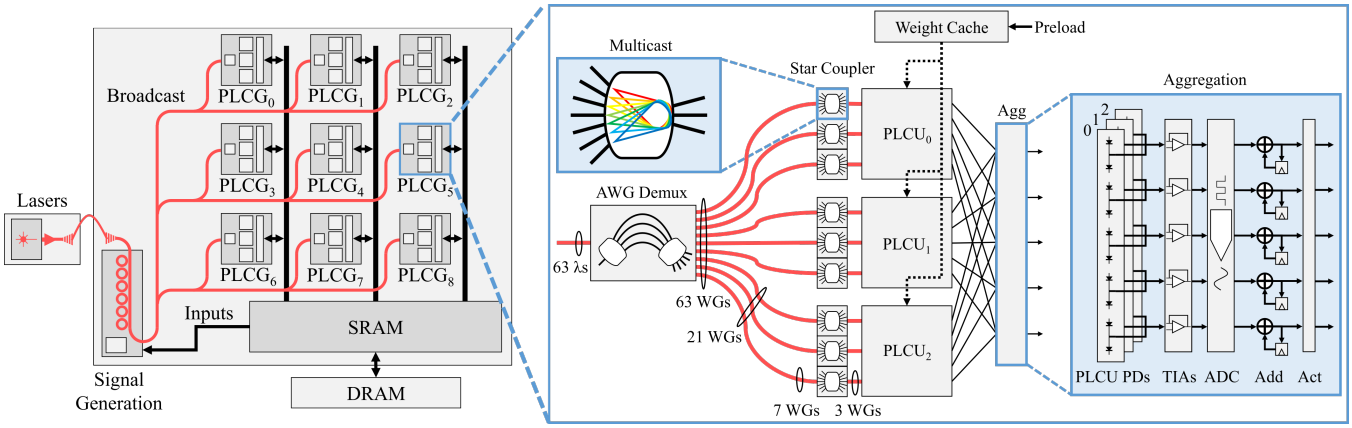

Figure 6. **(a) Albireo chip architecture with $N_g = 9$, showing input signal broadcasting. (b) PLCG architecture with $N_u = 3$, $N_m = 9$, and $N_d = 5$, showing star coupler multicast and aggregation unit.**

the signal A_{00} that appears on λ_0 is only used in the top row (row 0), but that signal also appears on rows 1 and 2 where it is unused. The star coupler takes $N_d + W_x - 1$ waveguides, each with a demultiplexed wavelength, and multiplexes the signals into W_x output waveguides that are fed to a set of MZMs in the PLCU. All input wavelengths are delivered to a PLCG through a single waveguide, which are then demultiplexed into their own waveguides by an arrayed waveguide grating (AWG) [57]. The demultiplexing using AWG and then multicasting via star couplers is shown in Figure 6(b). AWGs and star couplers are passive devices, and consume no power.

Each PLCG operates on a single kernel, and several kernels are applied in a CNN layer. These kernels all operate on the same input volume, so it is practical to compute on multiple kernels in parallel. This can be achieved by introducing multiple PLCGs and broadcasting the same inputs to each of them. Broadcasting with photonics is straightforward, signals are easily split using a series of Y-branches. This broadcasting is shown in Figure 6(a).

The Albireo architecture incorporates multiple PLCGs into a single chip, and we design Albireo with $N_g = 9$ PLCGs. More PLCGs could be implemented in the chip, which would increase the the amount of parallel processing, but also increase area and power. We chose N_g based on the area constraints since photonic devices are large compared to digital logic. Off-chip lasers are responsible for delivering the optical power onto the chip, which is then modulated by a bank of MRRs to generate the input signals. These input signals are then broadcast to each PLCG to compute partial dot products. A global SRAM buffer is responsible for holding inputs, kernel weights, and activations. Each PLCG also has a smaller cache for holding the kernel weights, which are initially preloaded. Input values and kernel weights undergo an electrical-to-optical conversion (E/O) with digital-to-analog converters (DAC) when applied at the modulating bank of MRRs and MZMs. The proposed Albireo chip is shown in Figure 6(a).

Algorithm 2 Convolution partitioning on Albireo

```
1:  function AlbireoConv(A, W)
2:      parallel for m ← 0; step 1; while m < W_m do          ▷ N_g instances
3:          y_B ← 0
4:          for y_A ← 0; step S; while y_A < A_y do
5:              x_B ← 0
6:              for x_A ← 0; step S; while x_A < A_x do
7:                  for c ← 0; step N_u; while c < W_z do
8:                      B[m][y_B][x_B : x_B+N_d] ← B[m][y_B][x_B : x_B+N_d]
                                                  + PlcgDot(A, W, m, y_A, x_A, c)
9:                  end for
10:                 B[m][y_B][x_B:x_B+N_d] ← f(B[m][y_B][x_B:x_B+N_d])
11:                 x_B ← x_B+N_d
12:             end for
13:             y_B ← y_B+1
14:         end for
15:     end parallel for
16: end function

17: function PlcgDot(A, W, m, y_A, x_A, c)
18:     parallel for i ← 0; step 1; while i < N_d do          ▷ N_d instances
19:         a[i] ← A[c : c+N_u][y_A : y_A+W_y][x_A+i : x_A+i+W_x]
20:         w[i] ← W[m][c : c+N_u][:][:]
21:         z[i] ← a[i] · w[i]
22:     end parallel for
23:     return z
24: end function
```

The partitioning of convolution on Albireo is shown in Algorithm 2. Line 2 computes on N_g kernels in parallel (one kernel per PLCG), and this parallel computation is the result of photonic broadcasting of the input volume. Line 8 is the aggregation of partials over N_u consecutive channels, and line 10 applies the activation function f once all partials are aggregated. Line 17 is the function that computes the N_d concurrent dot products in the PLCG, which is possible due to parameter sharing and the photonic multicasts in the star couplers.

Figure 7 shows the dataflow for a single kernel and single PLCG. In cycle 1, The first N_u channels of the kernel are applied at the MZMs in the PLCUs, where channel 0 is ap-

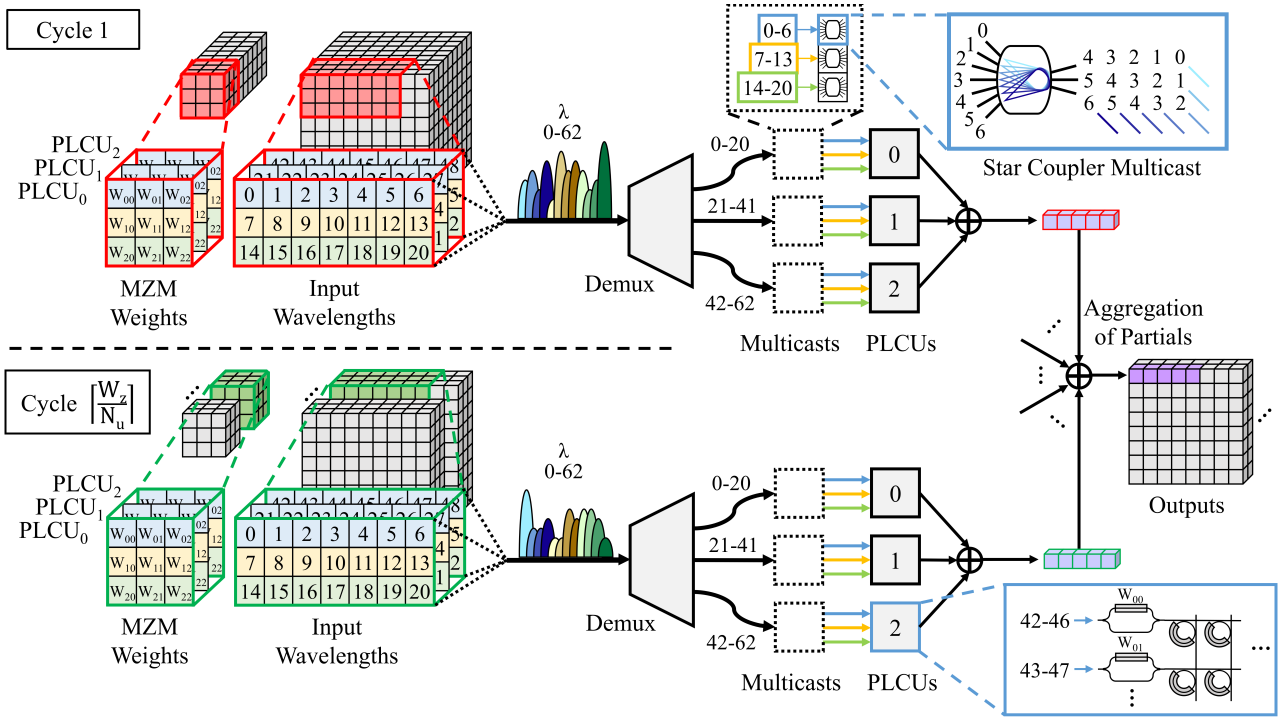

Figure 7. **Dataflow in a PLCG with $N_u = 3$, $N_m = 9$, $N_d = 5$, and $W_x = W_y = 3$. The depth-first aggregation of partials is shown from cycle 1 to cycle $\lceil \frac{W_z}{N_u} \rceil$, which completes the dot product and produces the output elements shown in purple.**

plied in PLCU$_0$ and so on. The $N_u \times W_y \times (N_d + W_x - 1)$ field of the input volume is modulated by the signal generation MRR bank, where each element is on a separate wavelength and transmitted over a single waveguide to the PLCG. Each of these wavelengths are then demultiplexed into their own waveguide via the AWG. Each set of $W_y \times (N_d + W_x - 1)$ waveguides undergoes multicasting, where each $(N_d + W_x - 1)$ sized row from the input volume undergoes a separate multicast at independent star couplers. Once multicasting is complete, the signals then continue on to the PLCU to compute the N_d concurrent dot products. The $N_u \times N_d$ partials created in the group are reduced to N_d partials by adding the currents from corresponding PDs across each PLCU. The N_d partials then enter the aggregation unit of the PLCG, where they undergo O/E conversion and are registered to be added across the remainder of the $\lceil \frac{W_z}{N_u} \rceil$ cycles.

While Albireo's architecture is targeted at efficient computation of convolutional layers, the architecture also supports fully-connected (FC) layers. It is easiest to think of FC layer implementation in Albireo in terms of convolution. That is, each element in the output is computed by a kernel that has a receptive field that is the size of the entire input volume. The convolution of this kernel with the input volume, which is now a single dot product, is equivalent to an FC computation for a single output element. There is one kernel applied for each output element. When computing an FC layer, only one PD accumulation column in a PLCU is utilized since no parameter sharing occurs. Aggregation across PLCUs within a PLCG still occurs in this mapping.

Albireo can also implement depthwise separable convolutions, like those found in MobileNet [26]. The depthwise kernels are applied in each PLCU as in the regular convolution case, however aggregation is not performed across channels for depthwise kernels. The pointwise kernels used in depthwise separable convolution require a different input mapping, and each MZM applies a weight from each channel of the 1×1 kernel. The optical inputs to each PLCU is a 2-dimensional slice of shape $N_m \times N_d$, where N_m is now the number of input channels, and N_d is the number of receptive fields. Each balanced PD pair still handles a single receptive field, and these are aggregated over the channels of the pointwise kernel between the PLCUs as in the regular convolution mapping.

IV. Performance Evaluation

A. System Setup

We perform 3 estimates of the proposed Albireo architecture: a conservative (C), moderate (M) and aggressive (A) estimate. Albireo-C uses photonic devices that have been demonstrated to date. This gives an indication of what Albireo is capable of using current technology. The Albireo-M estimates are the device performance needed to

Table I
DEVICE POWER ESTIMATES FOR CONSERVATIVE, MODERATE, AND AGGRESSIVE CONFIGURATIONS.

Device	Conservative	Moderate	Aggressive
MRR	3.1 mW [38]	388 µW	155 µW
MZM	11.3 mW [1]	1.41 mW	565 µW
Laser	37.5 mW @ 20 °C [15]	1.38 mW	1.38 mW
TIA	3 mW [46]	1.5 mW	300 µW
ADC	29 mW @ 5 GS/s [21]	14.5 mW @ 5 GS/s	2.9 mW @ 8 GS/s
DAC	26 mW @ 5 GS/s [47]	13 mW @ 5 GS/s	2.6 mW @ 8 GS/s

have similar energy consumption as current state-of-the-art electronic accelerators. Since silicon photonics is an emerging technology, the moderate estimate sets a target performance for photonic device engineers to pursue. The aggressive devices are future estimates that would make Albireo-A a high performance successor to current electronic accelerators, reducing metrics like energy-delay product by at least 100 X. The aggressive device assumptions made here are still well above the low-energy predictions for photonic devices made in [37], which defines a set of approaches for scaling laser and modulator energies into the attojoule range. The device power parameters used for each of these estimates is shown in Table I.

The proposed photonic processors were designed and verified in Lumerical INTERCONNECT: Photonic Integrated Circuit Simulator [35]. We simulated the performance of Albireo using a combination of Python and the crosstalk, noise, scattering, and temporal analysis from Lumerical INTERCONNECT. Memory subsystems were simulated using the PCACTI tool [48], which is an extension of the CACTI cache modeling tool [4] for FinFET and recent CMOS devices.

Table II shows the list of optical parameters used for the photonic devices. These optical parameters are from simulation and demonstrated (referenced) devices, and are used for all 3 (C, M, A) estimates of the Albireo architecture. The memory subsystem estimates are for 7 nm FinFET technology. The global SRAM buffer has 256 kB of storage and a footprint of 0.59×0.34 mm^2. The PLCG kernel caches have 16 kB of storage and a footprint of 0.092×0.085 mm^2.

Photonic processing requires high amounts of E/O and O/E conversions, which can easily become a bottleneck by the DACs and ADCs. The DACs [47] and ADCs [21] we utilize support 8-bit precision and operate at 5 GS/s, which limits the modulation rate to 5 GHz for Albireo-C and Albireo-M. We optimistically raise the sampling rate to 8 GS/s for aggressive estimates in Albireo-A. Higher sampling rates are achievable at this precision, but at the cost of much higher power consumption [60].

Albireo's performance is evaluated on CNN models including VGG16 [53], ResNet18 [24], MobileNet [26], and AlexNet [31]. We perform a per-layer analysis to yield latency, energy, and EDP for an inference on these CNN

Table II
OPTICAL DEVICE PARAMETERS USED IN THE ALBIREO ARCHITECTURE.

Device	Parameter	Value
Waveguide	$w \times h$	500×220 nm
	$n_{\text{eff}}, n_{\text{g}}$	(2.33, 4.68) @ λ=1550 nm
	loss	1.5 dB/cm (straight) [13]
		3.8 dB/cm (bent) [13]
Y-branch	loss	0.3 dB [61]
	area	1.2×2.2 µm^2 [61]
Microring resonator	radius	5 µm
	loss	0.39 dB
	k^2	0.03
	FSR	16.1 nm
	area	20×20 µm^2
Mach-Zehnder modulator	loss	1.2 dB [1]
	area	300×50 µm^2 [1]
Star coupler	loss	1.3 dB [34]
	area	750×350 µm^2 [34]
Arrayed waveguide grating	channels	64
	loss	2.0 dB
	crosstalk	-34 dB [29]
	FSR	70 nm [57]
	area	5×2 mm^2 [57]
Laser	RIN	-140 dBc/Hz
	area	400×300 µm^2 [15]
PIN photodiode	responsivity	1.1 A/W [51]
	dark current	25 pA @ 1 V [51]
	area	40×40 µm^2 [51]

models. The image input to each of these CNN models is assumed to have dimensions $224 \times 244 \times 3$.

We compare Albireo with two recent photonic neural network accelerators PIXEL [52] and DEAP-CNN [5]. PIXEL is a mixed-signal photonic accelerator built using MRRs for bitwise logical operations and MZMs for analog accumulation. DEAP-CNN utilizes the MRR weight banks proposed in [56] for dot products, and uses voltage addition for accumulation of partial sums across filter channels. We apply the same conservative device parameters (Table I) to PIXEL and DEAP-CNN, and scale their architectures to meet a 60 W power consumption threshold. We also assume 7 nm digital electronics for these photonic architectures so all device powers are consistent when comparing with Albireo. We obtain a fair comparison between these architectures by using the same device assumptions and holding the designs to the same power constraints. We compare with the 9-PLCG Albireo design, which consumes only 22.7 W of power, so we also provide comparison with a 60 W version of Albireo, which is scaled up to 27 PLCGs. Both DEAP-CNN and Albireo operate at 5 GHz, while PIXEL operates at 10 GHz. DEAP-CNN is unable to support 3×3 shaped kernels with more than 113 channels and has no infrastructure in place to handle partial sums of kernels larger than this. When comparing with Albireo, we have made the optimistic assumption in favor of DEAP-CNN that their architecture can support these larger kernels, which appear in the CNN benchmarks used for evaluation. The PIXEL architecture that we compare against is their 8-bit "OO" optical MAC unit. We scale the number of PIXEL

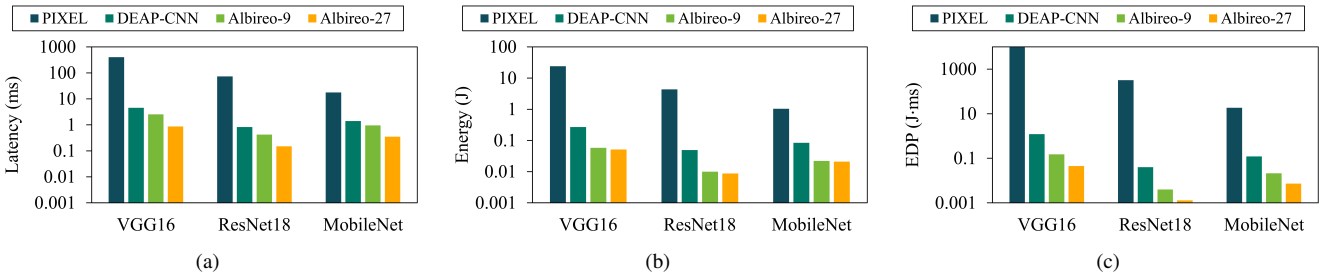

Figure 8. **CNN inference benchmark comparison with photonic accelerators PIXEL [52] and DEAP-CNN [5] using conservative photonic device parameters. Designs are held to 60 W power budget. Performance shown for: (a) latency, (b) energy, (c) energy-delay product.**

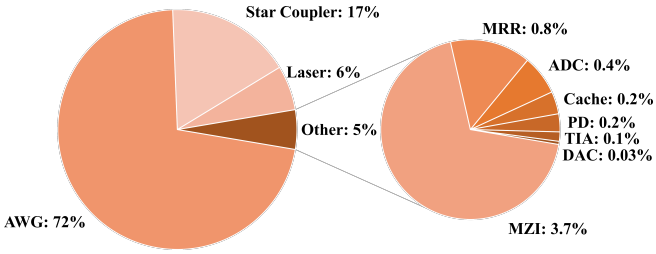

Figure 9. **Albireo chip area breakdown by component.**

8-bit optical MAC units to meet the 60 W power constraint.

Albireo is compared against three energy-efficient state-of-the-art electronic accelerators: Eyeriss [7], [8], ENVISION [39], and UNPU [32], and each accelerator represents a different energy-efficient computation technique. Eyeriss is a spatial architecture that takes advantage of row-stationary dataflow to reduce energy consumption. ENVISION uses subword parallel MACs with dynamic voltage, frequency, and bit precision scaling. UNPU is lookup table-based bit-serial processor with variable bit precision. The latency and energy efficiency of these electronic architectures are the reported performance taken directly from their publications. It is important to note that since these are the reported performances from each electronic accelerator's respective publication, these performances are only valid for the technology generation used in each accelerator's evaluation. Both Eyeriss and UNPU performance results are for 65 nm technology, and ENVISION results are for 28 nm technology, while Albireo's digital electronics are estimated assuming 7 nm technology.

B. Results

The Albireo architecture occupies an estimated 124.6 mm^2, most of which is for photonic data distribution with the AWGs (72%) and star couplers (17%). Although a single AWG uses 8% of the total area, these are passive diffractive devices and do not consume energy. The MZMs are the largest computation device, occupying 3.7% of the total area. MZMs are competitive for fast multiplication despite their large footprint. An MZM achieves 333 GOPS/mm^2 when multiplying just a single optical input at 5 GHz modulation. For comparison, a recent approximate 8-bit multiplier achieves just 7.3 GOPS/mm^2 [20], 46 X lower than the MZM. This performance gap is further widened when the MZM multiplies several input wavelengths at once in a WDM system. The area breakdown for all components in the Albireo chip architecture is shown in Figure 9. Photonic devices have relatively large footprints when compared to digital electronics, so hybrid photonic-electronic circuits like Albireo should expect a majority of the area to be occupied by photonic devices. Table III provides a breakdown of the total device powers, which illustrates how the device estimates affect power scaling in Albireo.

Table III
DEVICE POWER BREAKDOWN FOR EACH ALBIREO ESTIMATE.

	Albireo-C		**Albireo-M**		**Albireo-A**	
	Power (W)	Portion	Power (W)	Portion	Power (W)	Portion
MRR	7.52	33.1%	0.94	15.2%	0.38	23.2%
MZI	3.45	15.2%	0.43	6.9%	0.17	10.4%
Laser	2.36	10.4%	0.09	1.5%	0.12	7.3%
TIA	0.14	0.62%	0.07	1.1%	0.01	0.61%
DAC	7.93	34.9%	3.98	64.3%	0.80	48.8%
ADC	1.31	5.8%	0.65	10.5%	0.13	7.9%
Cache	0.03	0.13%	0.03	0.48%	0.03	1.8%
Total	22.7	100%	6.19	100%	1.64	100%

When compared to the recent photonic accelerators PIXEL and DEAP-CNN, Albireo outperforms in all metrics. The comparison between these accelerators is shown in Figure 8. On average, the regular 9-PLCG (Albireo-9, 22.7 W) architecture improves latency by 79.5 X and 1.7 X when compared to PIXEL and DEAP-CNN, respectively. Latency is further improved when scaling to the same power constraints with a 27-PLCG (Albireo-27, 58.8 W) architecture, giving average reductions of 225 X and 4.8 X when compared to PIXEL and DEAP-CNN, respectively. The Albireo-27 design reduces average energy consumption by 226 X and 4.9 X for PIXEL and DEAP-CNN, respectively, and reduces EDP by 50,957 X and 23.9 X for PIXEL and DEAP-CNN, respectively. We compare on a combination metric that indicates how efficiently the architectures utilize WDM for computation in units of energy per wavelengths used.

Table IV
CNN INFERENCE BENCHMARK COMPARISON WITH STATE-OF-THE-ART DIGITAL PROCESSORS EYERISS [7], [8], ENVISION [39], AND UNPU [32].

	AlexNet						VGG16					
	Eyeriss[a]	Envision[b]	UNPU[a]	**Albireo-C**	**Albireo-M**	**Albireo-A**	Eyeriss	Envision	UNPU	**Albireo-C**	**Albireo-M**	**Albireo-A**
Latency (ms)	25.9	21.3	2.89	0.13	0.13	0.080	1252	598.8	54.6	2.55	2.55	1.60
Energy (mJ)	7.19	0.94	0.84	2.90	0.80	0.13	295.4	15.6	16.2	58.1	15.7	2.56
EDP (mJ·ms)	186.1	20.0	2.42	0.37	0.10	0.010	370 k	9341	886.9	148.2	40.1	4.09
GOPS/mm^2	1.75	18.2	15.7	44.7 395.0[c]	44.7 395.0[c]	72.6 641.8[c]	0.77	13.8	17.7	48.8 431.1[c]	48.8 431.1[c]	77.7 687.1[c]
GOPS/W/mm^2	6.29	411.9	53.9	2.00 17.7[c]	7.26 64.2[c]	44.7 395.0[c]	3.3	531.3	59.1	2.14 18.9[c]	7.92 70.0[c]	48.6 429.4[c]

[a] 65 nm technology
[b] 28 nm technology
[c] active area only

Albireo has 30.9 X better WDM efficiency than DEAP-CNN on average, and 1680 X better WDM efficiency compared to PIXEL.

The performance of Albireo compared with state-of-the-art digital accelerators is shown in Table IV. When averaged across all 3 accelerators, Albireo-C improves latency by 110 X and EDP by 74.2 X. Alberio-M consumes roughly equal energy to both ENVISION and UNPU, and reduces EDP by an average of 275 X. Eyeriss is an outlier for EDP, so we directly compare Albireo-M and Albireo-A with ENVISION and UNPU for this metric. Albireo-M reduces EDP by 23.1 X and 216 X for UNPU and ENVISION, respectively. Albireo-A further improves performance by giving an average of 177 X lower latency, and improving EDP by 229 X and 2137 X for UNPU and ENVISION, respectively. Table IV also includes performance with respect to area since there are significant area differences between photonic and electronic circuits.

V. RELATED WORK

Silicon photonic accelerators are a new research effort to design the next generation of scalable and energy-efficient processors for DNN inference. HolyLight [33] is a multi-tile architecture that utilizes silicon microdisk resonators for pipelined matrix-vector multiplication. DNNARA [45] accelerates DNNs using the residue number system for computation, which is implemented through 2×2 optical switches for modulo arithmetic. We forego comparison with HolyLight and DNNARA because holding them to a 60 W power budget using realistic photonic device parameters renders them impractical for competitive CNN inference.

PIXEL [52] is a mixed-signal photonic accelerator built using MRRs for bitwise logical operations and cascaded MZMs for analog accumulation. PIXEL's OMAC processor is not as area efficient as Albireo's PLCU because the PLCU uses MRRs for accumulation, while PIXEL uses large MZMs for accumulation. PIXEL also does not efficiently utilize WDM in its MZMs. Each MZM accumulates a single wavelength, which increases the number of MZMs in their design. DEAP-CNN [5] utilizes the MRR weight banks proposed in [56] to compute dot products, and uses voltage addition for accumulation of partial sums across filter channels. Albireo is more energy efficient than DEAP-CNN because it uses fewer devices while reducing latency with parameter sharing. DEAP-CNN requires 2034 DACs for signal generation and MRR weight banks, while Albireo uses only 306 DACs (6.6 X fewer). Also, DEAP-CNN uses 113 TIAs, while Albireo uses only 45 TIAs.

Programmable photonics is a related, yet different concept to this work. Programmable photonic MZM meshes like those in [42] are restricted to performing unitary matrix operations, a constraint that is not applicable to Albireo. Cascading two MZM unitary meshes with an attenuation layer in between creates an architecture like in [50], which implements the singular value decomposition of a matrix to perform matrix-vector multiplication. These meshes have the benefit of being trainable and can implement universal unitary operations, whereas Albireo is less flexible and designed as a computation accelerator for convolutions.

VI. CONCLUSIONS

With the end of Dennard scaling, highly-parallel hardware accelerators have been proposed to improve the throughput and energy-efficiency of DNN models. Emerging technology like silicon photonics could provide the efficiency necessary to further scale DNN acceleration. In this paper, we presented Albireo, a photonic neural network accelerator that exploits multicast data patterns found in DNNs. We developed new models based on realistic photonic device limitations that prior works have not addressed in sufficient detail. Albireo increases parallel computation through novel dot product processing in PLCUs, and leverages broadcasts to compute on several kernels concurrently. Albireo reduces EDP by at least 24 X on CNN benchmarks when compared to recent photonic accelerators. With conservative estimates, Albireo improves latency by 110 X and EDP by 74 X on average when compared to state-of-the-art electronic accelerators. With aggressive estimates, Albireo improves latency by 177 X on average and EDP by at least 229 X.

ACKNOWLEDGEMENTS

This research was partially supported by NSF grants CCF-1513606, CCF-1702980, CCF-1703013, CCF-1812495,

CCF-1901165, CCF-1901192, and CCF-1953980. We thank the anonymous reviewers for their excellent feedback.

REFERENCES

[1] S. Akiyama, T. Baba, M. Imai, T. Akagawa, M. Takahashi, N. Hirayama, H. Takahashi, Y. Noguchi, H. Okayama, T. Horikawa, and T. Usuki, "12.5-gb/s operation with 0.29-v·cm vπl using silicon mach-zehnder modulator based-on forward-biased pin diode," *Opt. Express*, vol. 20, no. 3, pp. 2911–2923, Jan 2012. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-20-3-2911

[2] T. Alexoudi, N. Terzenidis, S. Pitris, M. Moralis-Pegios, P. Maniotis, C. Vagionas, C. Mitsolidou, G. Mourgias-Alexandris, G. T. Kanellos, A. Miliou, K. Vyrsokinos, and N. Pleros, "Optics in computing: From photonic network-on-chip to chip-to-chip interconnects and disintegrated architectures," *Journal of Lightwave Technology*, vol. 37, no. 2, pp. 363–379, 2019.

[3] T. Alexoudi, G. T. Kanellos, and N. Pleros, "Optical ram and integrated optical memories: a survey," *Light: Science & Applications*, vol. 9, no. 1, p. 91, 2020.

[4] R. Balasubramonian, A. B. Kahng, N. Muralimanohar, A. Shafiee, and V. Srinivas, "Cacti 7: New tools for interconnect exploration in innovative off-chip memories," *ACM Trans. Archit. Code Optim.*, vol. 14, no. 2, Jun. 2017. [Online]. Available: https://doi.org/10.1145/3085572

[5] V. Bangari, B. A. Marquez, H. Miller, A. N. Tait, M. A. Nahmias, T. F. de Lima, H. Peng, P. R. Prucnal, and B. J. Shastri, "Digital electronics and analog photonics for convolutional neural networks (deap-cnns)," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1–13, 2020.

[6] 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*, vol. 6, no. 1, pp. 47–73, 2012. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/lpor.201100017

[7] Y. Chen, J. Emer, and V. Sze, "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 2016, pp. 367–379.

[8] Y. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, 2017.

[9] Q. Cheng, J. Kwon, M. Glick, M. Bahadori, L. P. Carloni, and K. Bergman, "Silicon photonics codesign for deep learning," *Proceedings of the IEEE*, vol. 108, no. 8, pp. 1261–1282, 2020.

[10] T. T. Cheng, "The normal approximation to the poisson distribution and a proof of a conjecture of ramanujan," *Bulletin of the American Mathematical Society*, vol. 55, pp. 396–401, 1949.

[11] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie, "Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 2016, pp. 27–39.

[12] S. V. R. Chittamuru, S. Desai, and S. Pasricha, "Swiftnoc: A reconfigurable silicon-photonic network with multicast-enabled channel sharing for multicore architectures," *J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 4, Jun. 2017. [Online]. Available: https://doi.org/10.1145/3060517

[13] L. Chrostowski, Z. Lu, J. Flueckiger, X. Wang, J. Klein, A. Liu, J. Jhoja, and J. Pond, "Design and simulation of silicon photonic schematics and layouts," in *Silicon Photonics and Photonic Integrated Circuits V*, L. Vivien, L. Pavesi, and S. Pelli, Eds., vol. 9891, International Society for Optics and Photonics. SPIE, 2016, pp. 185 – 195. [Online]. Available: https://doi.org/10.1117/12.2230376

[14] J. Cong and B. Xiao, "Minimizing computation in convolutional neural networks," in *2014 International Conference on Artificial Neural Networks (ICANN)*, 2014, pp. 281–290.

[15] A. Descos, C. Jany, D. Bordel, H. Duprez, G. Beninca de Farias, P. Brianceau, S. Menezo, and B. Ben Bakir, "Heterogeneously integrated iii-v/si distributed bragg reflector laser with adiabatic coupling," in *39th European Conference and Exhibition on Optical Communication (ECOC 2013)*, 2013, pp. 1–3.

[16] A. Dhar, X. Wang, H. Franke, J. Xiong, J. Huang, W. mei Hwu, N. S. Kim, and D. Chen, "Freac cache: Folded-logic reconfigurable computing in the last level cache," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2020, pp. 102–117.

[17] S. Dieleman, J. De Fauw, and K. Kavukcuoglu, "Exploiting cyclic symmetry in convolutional neural networks," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, p. 1889–1898.

[18] P. Dong, S. Liao, D. Feng, H. Liang, D. Zheng, R. Shafiiha, C.-C. Kung, W. Qian, G. Li, X. Zheng, A. V. Krishnamoorthy, and M. Asghari, "Low vpp, ultralow-energy, compact, high-speed silicon electro-optic modulator," *Opt. Express*, vol. 17, no. 25, pp. 22 484–22 490, Dec 2009. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-17-25-22484

[19] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *2011 38th Annual International Symposium on Computer Architecture (ISCA)*, 2011, pp. 365–376.

[20] D. Esposito, A. G. M. Strollo, E. Napoli, D. De Caro, and N. Petra, "Approximate multipliers based on new approximate compressors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 12, pp. 4169–4182, 2018.

[21] M. Guo, J. Mao, S. W. Sin, H. Wei, and R. P. Martins, "A 5 gs/s 29 mw interleaved sar adc with 48.5 db sndr using digital-mixing background timing-skew calibration for direct sampling applications," *IEEE Access*, vol. 8, pp. 138 944–138 954, 2020.

[22] 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 *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 2016, pp. 243–254.

[23] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, p. 1135–1143.

[24] 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)*, 2016, pp. 770–778.

[25] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014, pp. 10–14.

[26] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017.

[27] K. Ishida, I. Byun, I. Nagaoka, K. Fukumitsu, M. Tanaka, S. Kawakami, T. Tanimoto, T. Ono, J. Kim, and K. Inoue, "Supernpu: An extremely fast neural processing unit using superconducting logic devices," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2020, pp. 58–72.

[28] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2704–2713.

[29] S. Kamei, A. Kaneko, M. Ishii, T. Shibata, Y. Inoue, and Y. Hibino, "Crosstalk reduction in arrayed-waveguide grating multiplexer/demultiplexer using cascade connection," *Journal of Lightwave Technology*, vol. 23, no. 5, pp. 1929–1938, 2005.

[30] A. Karkar, T. Mak, K. Tong, and A. Yakovlev, "A survey of emerging interconnects for on-chip efficient multicast and broadcast in many-cores," *IEEE Circuits and Systems Magazine*, vol. 16, no. 1, pp. 58–72, 2016.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th*

International Conference on Neural Information Processing Systems - Volume 1, ser. NIPS'12. Red Hook, NY, USA: Curran Associates Inc., 2012, p. 1097–1105.

[32] J. Lee, C. Kim, S. Kang, D. Shin, S. Kim, and H. Yoo, "Unpu: An energy-efficient deep neural network accelerator with fully variable weight bit precision," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 173–185, 2019.

[33] W. Liu, W. Liu, Y. Ye, Q. Lou, Y. Xie, and L. Jiang, "Holylight: A nanophotonic accelerator for deep learning in data centers," in *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2019, pp. 1483–1488.

[34] S. Lu, C. Yang, Y. Yan, G. Jin, Z. Zhou, W. H. Wong, and E. Y. B. Pun, "Design and fabrication of a polymeric flat focal field arrayed waveguide grating," *Opt. Express*, vol. 13, no. 25, pp. 9982–9994, Dec 2005. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-13-25-9982

[35] Lumerical Inc. [Online]. Available: https://www.lumerical.com/products/

[36] A. Mehrabian, Y. Al-Kabani, V. J. Sorger, and T. El-Ghazawi, "Pcnna: A photonic convolutional neural network accelerator," *2018 31st IEEE International System-on-Chip Conference (SOCC)*, Sep 2018. [Online]. Available: http://dx.doi.org/10.1109/SOCC.2018.8618542

[37] D. A. B. Miller, "Attojoule optoelectronics for low-energy information processing and communications," *Journal of Lightwave Technology*, vol. 35, no. 3, pp. 346–396, 2017.

[38] S. Moazeni, S. Lin, M. Wade, L. Alloatti, R. J. Ram, M. Popović, and V. Stojanović, "A 40-gb/s pam-4 transmitter based on a ring-resonator optical dac in 45-nm soi cmos," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 12, pp. 3503–3516, 2017.

[39] B. Moons, R. Uytterhoeven, W. Dehaene, and M. Verhelst, "14.5 envision: A 0.26-to-10tops/w subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28nm fdsoi," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, 2017, pp. 246–247.

[40] T. Moreau, M. Wyse, J. Nelson, A. Sampson, H. Esmaeilzadeh, L. Ceze, and M. Oskin, "Snnap: Approximate computing on programmable socs via neural acceleration," in *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, 2015, pp. 603–614.

[41] M. A. Nahmias, T. F. de Lima, A. N. Tait, H. Peng, B. J. Shastri, and P. R. Prucnal, "Photonic multiply-accumulate operations for neural networks," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1–18, 2020.

[42] S. Pai, B. Bartlett, O. Solgaard, and D. A. B. Miller, "Matrix optimization on universal unitary photonic devices," *Phys. Rev. Applied*, vol. 11, p. 064044, Jun 2019. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevApplied.11.064044

[43] D. Patel, A. Samani, V. Veerasubramanian, S. Ghosh, and D. V. Plant, "Silicon photonic segmented modulator-based electro-optic dac for 100 gb/s pam-4 generation," *IEEE Photonics Technology Letters*, vol. 27, no. 23, pp. 2433–2436, 2015.

[44] S. Pathak, D. V. Thourhout, and W. Bogaerts, "Design trade-offs for silicon-on-insulator-based awgs for (de)multiplexer applications," *Opt. Lett.*, vol. 38, no. 16, pp. 2961–2964, Aug 2013. [Online]. Available: http://ol.osa.org/abstract.cfm?URI=ol-38-16-2961

[45] J. Peng, Y. Alkabani, S. Sun, V. J. Sorger, and T. El-Ghazawi, "Dnnara: A deep neural network accelerator using residue arithmetic and integrated photonics," in *49th International Conference on Parallel Processing - ICPP*, ser. ICPP '20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3404397.3404467

[46] M. Rakowski, Y. Ban, P. De Heyn, N. Pantano, B. Snyder, S. Balakrishnan, S. Van Huylenbroeck, L. Bogaerts, C. Demeurisse, F. Inoue, K. J. Rebibis, P. Nolmans, X. Sun, P. Bex, A. Srinivasan, J. De Coster, S. Lardenois, A. Miller, P. Absil, P. Verheyen, D. Velenis, M. Pantouvaki, and J. Van Campenhout, "Hybrid 14nm finfet - silicon photonics technology for low-power tb/s/mm2 optical i/o," in *2018 IEEE Symposium on VLSI Technology*, 2018, pp. 221–222.

[47] B. Sedighi, M. Khafaji, and J. C. Scheytt, "8-bit 5gs/s d/a converter for multi-gigabit wireless transceivers," in *2011 6th European Microwave Integrated Circuit Conference*, 2011, pp. 192–195.

[48] A. Shafaei, Y. Wang, X. Lin, and M. Pedram, "Fincacti: Architectural analysis and modeling of caches with deeply-scaled finfet devices," in *2014 IEEE Computer Society Annual Symposium on VLSI*, 2014, pp. 290–295.

[49] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, "Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 2016, pp. 14–26.

[50] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljačić, "Deep learning with coherent nanophotonic circuits," *Nature Photonics*, vol. 11, no. 7, pp. 441–446, 2017.

[51] Z. Sheng, L. Liu, J. Brouckaert, S. He, and D. V. Thourhout, "Ingaas pin photodetectors integrated on silicon-on-insulator waveguides," *Opt. Express*, vol. 18, no. 2, pp. 1756–1761, Jan 2010. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-18-2-1756

[52] K. Shiflett, D. Wright, A. Karanth, and A. Louri, "Pixel: Photonic neural network accelerator," in *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2020, pp. 474–487.

[53] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[54] V. Sze, Y. Chen, T. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[55] A. N. Tait, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal, "Broadcast and weight: An integrated network for scalable photonic spike processing," *Journal of Lightwave Technology*, vol. 32, no. 21, pp. 4029–4041, 2014.

[56] A. N. Tait, A. X. Wu, T. F. de Lima, E. Zhou, B. J. Shastri, M. A. Nahmias, and P. R. Prucnal, "Microring weight banks," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 22, no. 6, pp. 312–325, 2016.

[57] A. van Wijk, C. R. Doerr, Z. Ali, M. Karabiyik, and B. I. Akca, "Compact ultrabroad-bandwidth cascaded arrayed waveguide gratings," *Opt. Express*, vol. 28, no. 10, pp. 14 618–14 626, May 2020. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-28-10-14618

[58] S. Van Winkle, A. K. Kodi, R. Bunescu, and A. Louri, "Extending the power-efficiency and performance of photonic interconnects for heterogeneous multicores with machine learning," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2018, pp. 480–491.

[59] M. R. Watts, W. A. Zortman, D. C. Trotter, R. W. Young, and A. L. Lentine, "Vertical junction silicon microdisk modulators and switches," *Opt. Express*, vol. 19, no. 22, pp. 21 989–22 003, Oct 2011. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-19-22-21989

[60] A. Zandieh, N. Weiss, T. Nguyen, D. Haranne, and S. P. Voinigescu, "128-gs/s adc front-end with over 60-ghz input bandwidth in 22-nm si/sige fdsoi cmos," in *2018 IEEE BiCMOS and Compound Semiconductor Integrated Circuits and Technology Symposium (BCICTS)*, 2018, pp. 271–274.

[61] Y. Zhang, S. Yang, A. E.-J. Lim, G.-Q. Lo, C. Galland, T. Baehr-Jones, and M. Hochberg, "A compact and low loss y-junction for submicron silicon waveguide," *Opt. Express*, vol. 21, no. 1, pp. 1310–1316, Jan 2013. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-21-1-1310