PIXEL: Photonic Neural Network Accelerator

Kyle Shiflett*, Dylan Wright*, Avinash Karanth* and Ahmed Louri†
*School of Electrical Engineering and Computer Science, Ohio University, Athens, Ohio 45701
†Dept. of Electrical and Computer Engineering, George Washington University, Washington, DC 20052
Email: *{ks117713, dw437013, karanth}@ohio.edu, †louri@gwu.edu

***Abstract*—Machine learning (ML) architectures such as Deep Neural Networks (DNNs) have achieved unprecedented accuracy on modern applications such as image classification and speech recognition. With power dissipation becoming a major concern in ML architectures, computer architects have focused on designing both energy-efficient hardware platforms as well as optimizing ML algorithms. To dramatically reduce power consumption and increase parallelism in neural network accelerators, disruptive technology such as silicon photonics has been proposed which can improve the performance-per-Watt when compared to electrical implementation. In this paper, we propose PIXEL - Photonic Neural Network Accelerator that efficiently implements the fundamental operation in neural computation, namely the *multiply and accumulate* (MAC) functionality using photonic components such as microring resonators (MRRs) and Mach-Zehnder interferometer (MZI). We design two versions of PIXEL - a hybrid version that multiplies optically and accumulates electrically and a fully optical version that multiplies and accumulates optically. We perform a detailed power, area and timing analysis of the different versions of photonic and electronic accelerators for different convolution neural networks (AlexNet, VGG16, and others). Our results indicate a significant improvement in the energy-delay product for both PIXEL designs over traditional electrical designs (48.4% for OE and 73.9% for OO) while minimizing latency, at the cost of increased area over electrical designs.**

***Keywords*-deep neural network; machine learning; silicon photonics; accelerator; microring resonator; Mach-Zehnder interferometer;**

I. Introduction

Power dissipation has become a fundamental barrier to scaling computing system performance [1]. Modern computers based on Von Neumann architecture are power hungry and less effective for a wide range of tasks including perception, communication, learning and decision making than the human brain [2]. In fact, the human brain can compute 10^{18} multiply-and-accumulate (MAC/sec) using only 20 W of power [3]. Multicores have been proposed to alleviate the power constraints. However, the breakdown of Dennard's scaling has further exacerbated the problem by limiting the number of cores that can be simultaneously powered on with a fixed power budget and heat extraction rate [4]. Therefore, specialization and parallelization by designing application specific accelerators that exceed the efficiency and functionality of general purpose processors with the end goal of at least 10-100x improvement in power or performance appear to be one approach to overcome the power barrier [5], [6]. Examples of applications/functions in which accelerators are used include floating point coprocessors, graphical processing units (GPUs), network offloading functions, artificial neural networks (ANNs), Fast-Fourier Transforms (FFT), crypto processors, image co-processors, and many more.

In the accelerator domain, machine learning (ML) architectures such as Deep Neural Networks (DNNs) have achieved unprecedented accuracy on many modern applications such as image classification and speech recognition. ML algorithms take as input a set of training examples and discover patterns that enable them to make predictions on previously unseen test examples. Artificial neural networks (ANNs) have been designed to facilitate this learning process, gaining their inspiration from the neuron and synapse linked structure of biological brains. ANNs can be further expanded upon to form deep neural networks (DNNs) by placing numerous connected layers between the input and output of the network. These multiple hidden layers involve immense amounts of highly concurrent matrix-vector-multiplications (MVMs) between a network weight matrix and the input vector. In convolutional neural networks (CNNs) several of the layers perform multiply-and-accumulate (MAC) functions using the same kernel repeatedly on small windows in the input layer [7]. Since the MAC functionality is the fundamental and highly repeated operation in CNNs, large emphasis must be placed on this operation to exploit data parallelism in order to perform CNN hardware acceleration.

A large body of work has recently been introduced focusing on increasing ANN computing speed and power efficiency by developing electronic architectures (such as ASIC and FPGA chips) specifically tailored to improve the data computation or storage ability of the accelerator [8], [9], [7], [10], [11], [12], [13], [14]. Real life applications require CNNs with millions of MAC operations in each layer, composing several hidden layers, which poses a serious challenge for future scaling of ANNs in general. Moreover, electronic-based accelerators utilize broadcast and multicast buses for achieving parallelism and are still limited by electronic clock rates and ohmic losses [15], [7]. The challenge in implementing neural networks on hardware accelerators is two-fold: (i) accelerators cannot be easily scaled to maximally exploit the parallelism offered by neural nets, and (ii) data movement needs to be optimized to minimize energy consumption.

Emerging technology such as silicon photonics has the potential to provide high communication and processing bandwidths, minimal access latencies, and high power-efficiency [16], [17], [18], [19], [20], [21]. Photonics does not provide a convenient way to maintain or store logic levels, however, it can deliver an abundance of parallelism, high bandwidth, energy-efficiency and ease of implementing broadcast/multicast functionality (one-to-many and one-to-all), all of which can be harnessed to implement neural network functionality in ML accelerators. Linear transformations can be performed at the speed of light, detected at rates exceeding 100 GHz and some matrix operations can be performed without consuming significant power [2]. Recent Photonic Integrated Circuits (PIC) have altered the landscape of manufacturing photonic chips that integrate both active (lasers and detectors) and passive (waveguides, resonators and modulators) devices on the same platform with hybrid integration and paved way for developing photonic neural nets [22], [23]. Prior work on programmable photonics and neuromorphic photonic accelerators have focused on developing optical interconnect topologies (wavelength-division multiplexed banks of modulators, interconnected tunable couplers, etc) that show how to design optical compute engines. However, none of the prior work have shown the design of a photonic neural network accelerator with detailed architectural parameters such as area, power and timing.

In this paper, we leverage the unique advantages of photonic technology to design **PIXEL: Photonic Neural Network Accelerator** for inference by designing the basic Optical Multiply and Accumulate (OMAC) units and integrating the OMAC units with digital processing systems. The proposed work is based on the effective use of microring resonators (MRRs), Mach-Zehnder Interferometers (MZIs), optical waveguides and lasers for multiply and accumulate functionality and integrating the photonic components with electronic processing. Both MRRs and MZIs are mature technologies that have the required form factor (area-efficiency) as well as bandwidth-density for optical processing and integration. We design two versions of PIXEL - a hybrid version that multiples optically and accumulates electrically and a fully optical version that multiples and accumulates optically for various convolution neural networks (CNNs). The hybrid architecture relies on using only MRRs and is therefore area-efficient whereas all-optical uses MRRs and MZIs and consumes more area for implementing CNNs. We perform a detailed design space exploration that evaluates power, area and timing of different versions of photonic and electronic accelerators for different layers of convolution neural networks (AlexNet, VGG16, and others). Our results indicate a significant improvement in the energy-delay product for both PIXEL designs over traditional electrical designs (48.4% for OE and 73.9% for OO) while minimizing latency, at the cost of increased area over electrical designs. Our accelerator design improves parallelism, latency, energy efficiency, and scalability for various CNN applications using silicon photonics. The major contributions of the paper are as follows:

- **Optical MAC**: We propose two accelerators that implement efficient optical MAC functionality for DNNs using MRRs and MZIs. We show an electrical-optical hybrid version and an all-optical version of the proposed accelerator.
- **PIXEL**: We propose to integrate the two versions of MAC with electrical signal processing. We propose x- and y-dimension optical interconnects that allow energy-efficient data movement with proposed OMAC.
- **Scalability**: PIXEL provides a scalable platform to implement CNN architectures of various sizes while minimizing energy-delay product (EDP) for varying number of wavelengths and bits/wavelength.

II. Background: Photonics and Accelerator

In this section, we provide a brief background on photonic devices such as MRRs and MZIs that we use in our proposed PIXEL design. We also provide a brief introduction to accelerator functionality that we implement in our proposed Optical-MAC architecture.

A. Photonic Devices

1) Microring Resonators: Recent microring resonator (MRR) designs have been shown to be a very promising technology for optical interconnects, and have been widely used for modulation, demodulation, and switching functions. MRRs have a small footprint (7.5 μm radius) [18], low energy consumption (<100 fJ/bit) and have been demonstrated to modulate at 40Gb/s and beyond [19], [20]. Due to thermal sensitivity, ring heaters are used to ensure that the wavelength drift is avoided and signals can be accurately detected; however other solutions including athermal design [24], runtime thermal optimization [25] and backend switching [21] have been proposed.

Figure 1(a) show cascaded double MRR that couples light from Input Port, I_0 to Output Port, O_0 when no voltage is applied to MRRs (V_{off}). Similarly, light from Input Port, I_1 is coupled to Output Port, O_1 which forms the *bar* configuration as shown in Figure 1(d). Figure 1(b) shows that when a voltage (V_{on}) is applied to the MRRs, the resonant wavelength arriving from I_0 is in resonance with the MRR and the signal will couple through the cascaded MRRs to Output Port O_1. If the light is in resonance with MRRs from I_1, that signal will appear at O_0 forming the *cross* configuration. Consider that light is injected into input port I_0 only. This allows for the implementation of the logical bitwise AND operation or multiply. The implementation (Y=A AND B) is controlled by the incoming optical signal

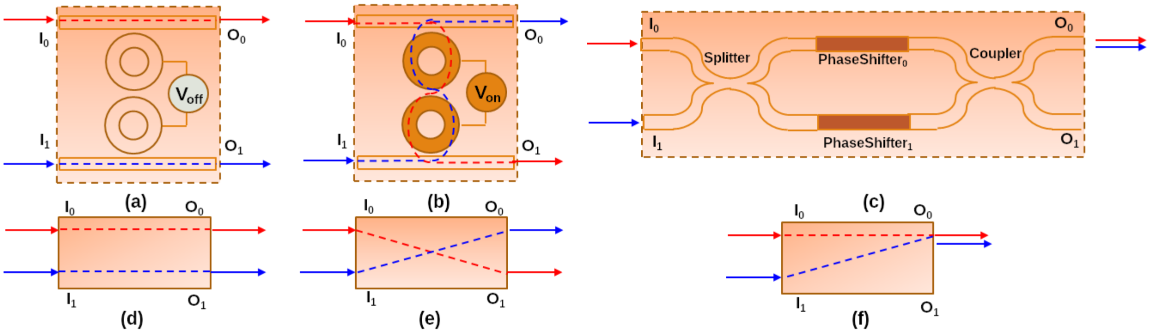


Figure 1: **(a) shows cascaded MRRs with V_{off} where light couples from Input Port, I_0 to Output Port, O_0. (b) shows cascaded MRRs with V_{on} where light couples from Input Port, I_0 to Output Port, O_1. (c) shows a MZI that combines signals from the two inputs ports I_0 and I_1 to output port O_0.**

(A) and the applied voltage to the MRRs (B). In order for the signal to appear at Output Port O_1 (Y=1), there must be an incoming optical signal (A=1) at Input Port I_0 and an applied high voltage to the MRRs (B=1). In all other input permutations (A=0,B=0; A=0,B=1; A=1,B=0), there will be no signal appearing at Output Port O_1 (Y=0). Therefore, we will use MRRs for multiply operation.

2) Mach-Zehnder Interferometer: The Mach-Zehnder Interferometer (MZI) allows for the splitting and coupling of two collimated beams of light. This device is highly configurable, and gives the ability to act as a tunable coupler. Figure 1(c) shows the layout of a MZI device. The device has two input ports connected to a splitter, which separates the two beams into the two phase-shifting arms. The two arms of the MZI have phase shifters 0 and 1 (ϕ_{upper} and ϕ_{lower}) that connect to a coupler. The coupler then diverts the beams to the two output ports. The MZI performs coupling with independent amplitude and phase shifting capabilities, and its ideal transfer matrix is given by:

$$h = je^{j\Delta}\begin{pmatrix} sin\theta & cos\theta \\ cos\theta & -sin\theta \end{pmatrix} \tag{1}$$

where:

$$\theta = \frac{\phi_{upper} - \phi_{lower}}{2} \tag{2}$$

$$\Delta = \frac{\phi_{upper} - \phi_{lower}}{2} \tag{3}$$

Each MZI can be configured to provide an independent power splitting ratio and overall phase shift by use of external electronic control signals applied to the two arms. This enables the MZI to operate as a directional coupler or more simply as an optical switch. When used as an optical switch, the MZI can operate in a bar state ($\phi_{upper} = [0, \pi]$; $\phi_{lower} = [\pi, 0]$) or in a cross state ($\phi_{upper} = \pi/2$; $\phi_{lower} = \pi/2$) as shown in Figures 1(d,e). This provides amplitude and phased controlled optical routing. Further, by adjusting ϕ_{upper} and ϕ_{lower} appropriately, and when $0 < \theta < \pi/2$, the MZI behaves as a tunable coupler combining the signals from both input ports to a single output port in an additive operation as shown in Figure 1(c,f). D.A.B. Miller recently showed on how to cascade and self-configure multiple MZIs to combine multiple input signals by adjusting the phase shifts on different arms of the individual MZI. Using this principle, MZI can be reconfigured to support different connection paths between its input and output ports and, hence, any kind of linear transformation (addition) can be implemented. Therefore, we will use MZIs for additive operation.

3) Photonic Link: In this work, we propose on-chip InP-based Fabry Perot lasers with short turn-on delay. On-chip lasers with dimensions 50 µm × 300 µm × 5 µm with each channel operating 128 wavelengths have been shown. Silicon waveguides, which have a smaller pitch of 5.5 µm, a lower propagation time of 10.45 ps/mm and a signal attenuation of 1.3 dB/cm are chosen due to ease of integration with other on-chip photonic components. Germanium-doped photodetectors along with back-end signal processing (transimpedance amplifiers (TIA), voltage amplifiers, clock and data recovery) to recover the transmitted bit. Two different optical-to-electrical (o/e) converters were used in our designs. The first o/e converter is a simple design utilizing a photodiode and shift registers to convert the serial optical pulses into parallel electrical signals. The second design must handle varying light-pulse amplitudes, and requires more complex logic for the o/e conversion than the first design. To determine the value of light pulses, a photodiode is used, which will output current proportional to the amount of light absorbed through electron-hole recombination. This current is then sent through an array of current comparators to determine the amplitude value of the signal, where back-end logic will then convert it into the bit-level data to be sent onward to the activation function circuitry.

B. Electrical MAC

MAC filters are designed to compute several inner products (IP) between input neuron lanes ($\mathcal{I}$) and synapse lanes ($\mathcal{S}$) before feeding it into a nonlinear function $f(x)$. Figure 2(a) displays the typical MAC implementation in the electrical domain. As shown, the number of input neuron lanes, filters, synapse lanes per filter, output neuron lanes ($\mathcal{O}$), and processing elements (PE) are all equal. The result for a single output neuron lane can be described as:

$$\mathcal{O}_k = f(\sum_j \sum_i \mathcal{I}_{i,j}\mathcal{S}_{i,j,k}) \tag{4}$$

where i represents the input neuron lane and synapse lane number, j represents the index of the input neuron lane and synapse lane, and k represents the output neuron lane index, which also correlates to the filter number in the PE. This process is repeated from ($k = 0$ to n) to get all indices for a single output neuron lane. This makes up a single PE and is done in all subsequent PEs in the MAC unit.

As an example, consider input neuron lane 0, INL$_0$ with 4 elements INL$_0$(e$_{inl0}$, e$_{inl1}$,e$_{inl2}$,e$_{inl3}$) each with 4 bits, <0010$_2$, 0100$_2$, 0110$_2$, 1001$_2$>represented as INL$_0$(2,4,6,9). Similarly assume that the other input neuron lanes are INL$_1$(0,1,3,4), INL$_2$(3,5,1,2) and INL$_3$ (8,2,8,6). For brevity sake, we consider only filter 0, with 4 synapse lanes, each with 4 elements of width 4 bits. Synapse Lane 0 in filter 0 is represented as F0[SL$_0$(e$_{sl0}$, e$_{sl1}$, e$_{sl2}$, e$_{sl3}$)]. Assume that the synapses in filter 0 have the following values: SL$_0$(6,9,13,11), SL$_1$(1,2,1,2), SL$_2$(2,3,4,5) and SL$_3$(3,1,3,1). In cycle 1, the following multiplications will occur in filter 0: Pm0 = INL$_0$(e$_{inl0}$) × SL$_0$(e$_{sl0}$, Pm1 = INL$_1$(e$_{inl0}$)) × SL$_1$(e$_{sl0}$), Pm2 = INL$_2$(e$_{inl0}$) × SL$_2$(e$_{sl0}$), and Pm3 = INL$_3$(e$_{inl0}$) × SL$_3$(e$_{sl0}$). All the partial multiplications will be summed with the output neuron lane ONL$_0$ (initialized to 0) to determine the first partial sum (Pm0 + Pm1 + Pm2 + Pm3 + ONL$_0$) which is written to ONL$_0$. In the above example, the first partial sum = (2×6 + 0×1 + 3×2 + 8×3 + 0) = 42. Once the entire window is computed, the final sum of 368 will be fed to the activation function (f) shown in Figure 3.

There are multiple ways of implementing an activation function in hardware. Approximation techniques are used to overcome the hard-to-realize implementations of these functions. The most common approaches include bit-level mapping schemes, lookup tables (LUT), piecewise linear (PL) approximation, piecewise nonlinear (PNL) approximation, and designs that are a hybrid of multiple approaches [26]. A modern hybrid hyperbolic tangent design based on PL approximation conjoined with bit-level mapping has been demonstrated to use minimal area with ultra-low gate-counts, while still rivaling common design latencies [27]. This hybrid approach has been used in the design of MAC units in this paper, and has allowed for high energy savings compared to traditional designs.

AND Shift-Accumulate: Through the modification of the MAC unit, Stripes (STR) takes advantage of parallelism that is innately present in DNNs by recognizing these properties on the bit level [28]. By breaking down each MAC operation into its bitwise elements, STR has accelerated the MAC functionality by using bitwise AND followed by a logical left-shift and accumulate. So, given a synapse $\mathcal{S}$ represented in p bits, and an input neuron $\mathcal{I}$, STR will process $\mathcal{S}$ bit-serially over p cycles. Each cycle, one bit of $\mathcal{S}$ and all of $\mathcal{I}$ go through an AND multiplication, accumulating the result into a running sum. The STR methodology is used in all accelerator designs represented in this paper.

III. OMAC: Optical MAC

A. Optical-Electrical MAC (OE)

The optical-electrical MAC hybrid accelerator shown in Figure 2(b) uses a combination of photonic devices and electrical circuitry to implement the AND shift-accumulate functionality. Our proposed hybrid design has optical AND with electrical shift-accumulate (OE). The optical AND utilizes an array of tuned MRRs to perform wavelength-division multiplexing (WDM), where each wavelength will either couple to the MRR when activated by the synapse lane and continue on into the multiplexed signal, or will not be allowed to pass through. The synapse lane controls the MRR, and represents an AND 1 when the MRR is activated, or an AND 0 when the MRR is deactivated. The number of wavelengths that each synapse lane filters is directly correlated to the number of input neuron lanes, as in the STR implementation value n. That is, for the optical designs in this paper, the number of wavelengths will be equated to the number of lanes, and will be referred to as lanes to provide consistency with the STR methodology across designs shown in Figure 2. Once the optical AND has completed, the signal will continue on to the electrical processing unit (EP). After undergoing an o/e conversion, the the AND values begin the shift-accumulation process with a CLA and left bit-shifter. Once all pulses in the neuron lanes have been transmitted against the synapse lanes, the resulted accumulation will go to the hyperbolic tangent activation function circuitry before entering the output neuron lane.

Consider the 4-OMAC configuration in Figure 2(b) such that OMAC 0 receives the multiplexed signal $\Lambda = (\lambda_0 + \lambda_1 + \lambda_2 + \lambda_3)$ with each tile tile transmitting the signal on a distinct wavelength to OMAC 0 in a multiple-write-single-read (MWSR) configuration. In this configuration, OMAC 0 will transmit ($\lambda_0 - \lambda_3$), OMAC 1 will transmit ($\lambda_4 - \lambda_7$), OMAC 2 will transmit ($\lambda_8 - \lambda_{11}$), and OMAC 3 will transmit ($\lambda_{12} - \lambda_{15}$). Therefore, the multiplexed signal received on each of the home channels will be ($\lambda_0 - \lambda_{15}$). Each OMAC will be outfitted with a single filter. OMAC 0 will implement filter 0, OMAC 1 will implement filter 1, and so on. The input neuron will also be distributed in a similar fashion. That is, OMAC 0 will fire (or transmit) input neuron lane 0 $\mathcal{I}_0$, OMAC 1 will fire $\mathcal{I}_1$, and so on. As shown in Figure 2(b), in cycle 1, OMAC 0 fires 4 bits per wavelength (0010$_2$, 0100$_2$, 0110$_2$, 1001$_2$) on 4 different home channels or waveguides such that the same information will be received by all of the tiles on the same wavelengths

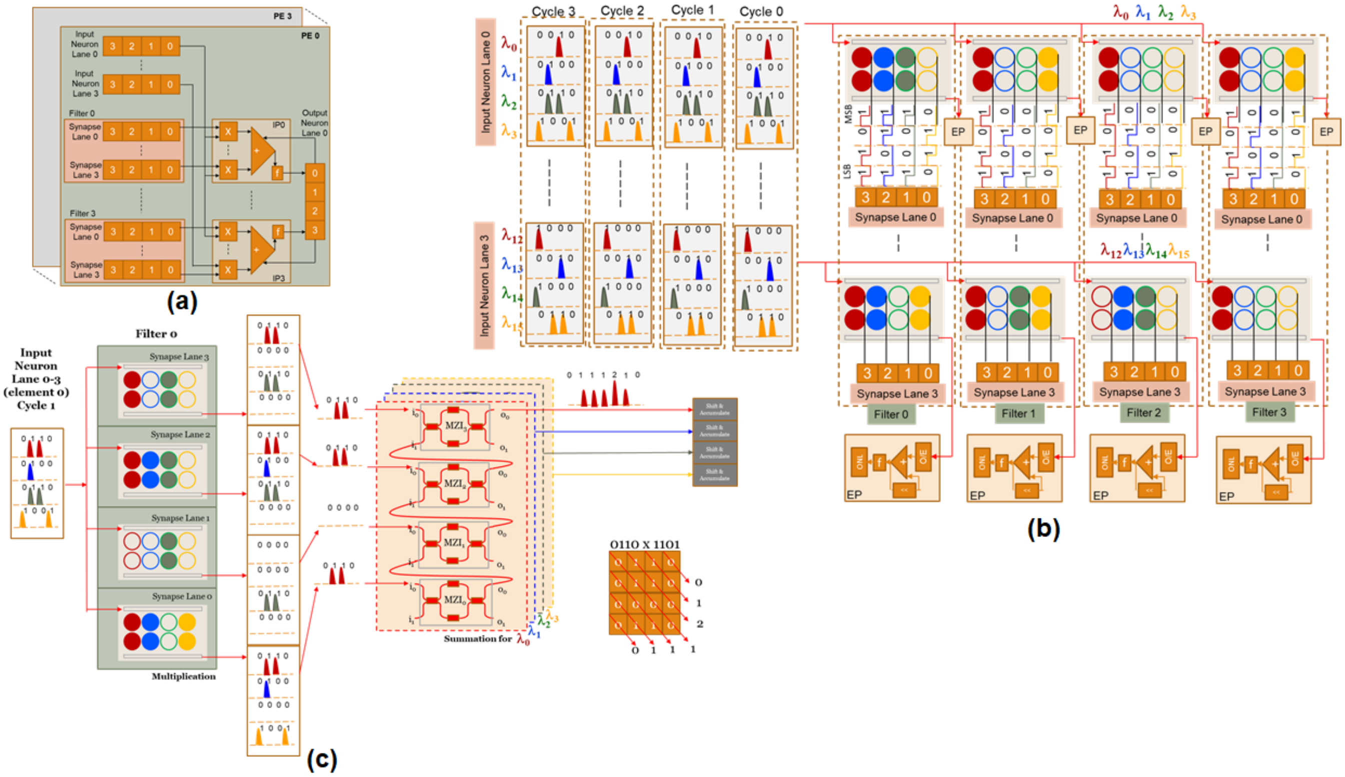


Figure 2: **(a) Basic STR configuration where bitwise multiplication and addition is performed. (b) Proposed OMAC unit that performs multiply optically whereas addition and shifting is performed electrically. (c) Extended OMAC with accumulation performed optically.**

($\lambda_0 - \lambda_3$). Similarly, OMAC 3 fires (1000_2, 0010_2, 1000_2, 0110_2) on wavelengths ($\lambda_{12} - \lambda_{15}$), which is recieved by all OMACs.

In each OMAC, different sunapse weights are associated with the MRRs to create synapse lanes $\mathcal{S}_0 - \mathcal{S}_3$. The bitwise AND operation between the incoming neuron and synapse will occur on the appropriate wavelength. On wavelength λ_0, 0010_2 is the incoming neuron data, and the MRR is turned off (0), therefore 0000_2 will appear on the lower waveguide and will be sent to the EP. This bitwise AND operation will occur such that the entire neuron datum is checked against a single synapse bit. As mentioned above, the next step for the signal is to enter the EP and undergo an o/e conversion and added to the partial sum. Once all 4 cycles of the running sum are computed across the 4 synapse lanes, the result is sent to the activation function where it will then appear at the output neuron lane. This technique reduces the OE MAC down to bitwise AND followed by shift-accumulate modeled after the STR design. So, for a synapse with p bits, the OE MAC requires p cycles to determine the partial sum for each synapse lane.

B. All-Optical MAC (OO)

The all-optical MAC utilizes optical devices for both the AND and shift-accumulate operations of the STR modified MAC methodology. The OO design uses WDM through the use of MRR for the AND operations as in the OE design. The main allure to this design is it's use of MZIs for low-latency, low-power shift-accumulate functionality. By cascading these MZIs together, the outputs of each sequential AND operation can undergo pure optical shift-accumulation. Synchronization of the signals output from the AND operation with the propagation delay of the MZI arms allows an optical pulse train to be delayed by one cycle in the MZI arms, which can then be added to the input of the next MZI.

Consider Figure 2(c), where element 0 in input neuron lanes $\mathcal{I}_0 - \mathcal{I}_3$ are fired by OMAC 0 in cycle 1. With the same four wavelengths (λ_0, λ_1, λ_2, λ_3) carrying the signals (0110_2, 0100_2, 0110_2, 1001_2) are simultaneously transmitted to Filter 0, which consists of the four synapse lanes ($\mathcal{S}_0 - \mathcal{S}_3$). After the bitwise AND operation in the MRRs, the output appears for all four synapses. By selective filtering through WDM using the MRRs, we can guide each wavelength ($\lambda_0 - \lambda_3$) emerging from the synapse lanes to a seperate MZI as shown in Figure 2(c).

For example, consider λ_0, where the output from synapse lanes ($\mathcal{S}_0$, $\mathcal{S}_2$, $\mathcal{S}_3$) is (0110_2). There is not output from synapse lane $\mathcal{S}_1$ since the MRRs are switched off. Each output from the synapse lanes are fed to a different MZI, take output λ_0 as an example. λ_0 (0110_2) feeds from synapse lane $\mathcal{S}_0$ to i_0 of MZI$_0$. Similarly, (0000_2, 0110_2, 0110_2) are fed to i_0 of MZI$_{1-3}$. By appropriately phase shifting ϕ_{lower} of MZI$_{0-3}$, it can be ensured that no optical signal emerges from the lower output o_1 of the MZI. Therefore, starting with the LSB (bit position 0) of (0110_2) as the input of MZI$_3$, bit 0 emerges from o_0 of MZI$_3$ at time t_0. For time t_1, consider bit position 1 of (0110_2) originating from MZI$_3$, and LSB of (0110_2) originating from MZI$_2$.

By ensuring that the path length connecting one MZI's o_0 to the next MZI's i_1 is equal to the bit transmission period, we can combine the two bits $(0 + 1)$ at the output, giving the resulting signal different amplitudes of light. The output signal will be then sent to the o/e converter where the final accumulated value will continue on to the activation function circuit.

C. PIXEL Architecture

Figure 3 shows the proposed PIXEL architecture where each OMAC consists of RF for filter weight storage, MAC unit that implements multiply and accumulate as described above. We consider neurons fired with photonic interconnects using both x- and y-dimensions. Front-end pre-processing of the data will fire the neurons repeatedly if needed and back-end processing of data will recover the information from the accumulation. The synapses are pre-loaded into the OMAC and the proposed design assumes timed firing of the neurons to implement the MAC functionality. The advantage of the proposed PNNA are as follows: (i) All neuron firing, and partial sums accumulation are in optical domain which significantly reduces energy consumption. While filter weights need to be pre-loaded to drive the MRRs, photonics could also be utilized to send the weight information on a specific channel to OMACs. (ii) PNNA architecture is scalable since the photonic drivers and receivers are located at x- and y-dimension E/O and O/E conversion. Except for active MRRs, all other components are passive and therefore, one can scale up by driving the optical signal with higher intensity. (iii) With two-dimensional connectivity, each row or column can be individually utilized/driven to solve a neural network problem. In what follows, we will evaluate the architectural parameters of implementing PIXEL architecture (power, delay, area).

IV. Performance Evaluation

A. Hardware Evaluation

To begin to evaluate the energy, area, latency of each design, an accurate understanding and description of every component is needed. Through simulation of each hardware component, the MAC can then be constructed to see how these parameters change with respect to the number of lanes in the design, as well as the number of bits per lane.

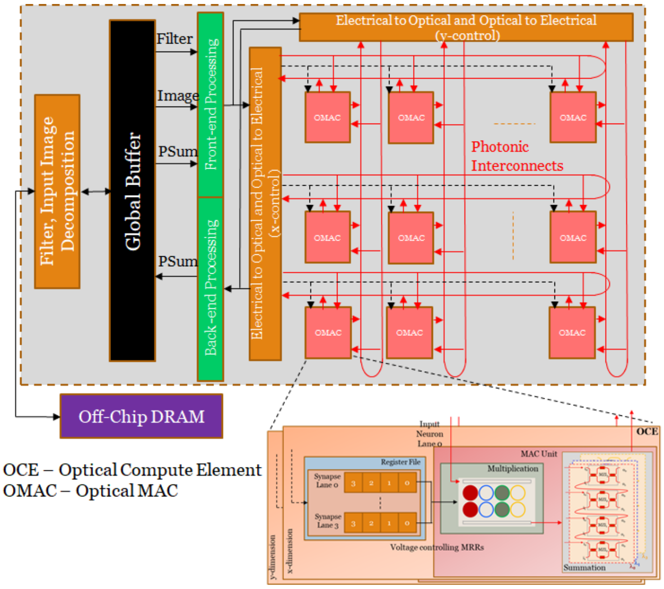


Figure 3: **PIXEL architecture consisting of OMAC where each OMAC contains photonic components to perform MAC operations. OCE are connected in x- and y-dimension with photonic interconnects.**

1) Electrical Devices: To evaluate the electrical device hardware parameters, it was necessary to obtain the gate-counts (GCs) of each component. Once the GCs for all of the devices is known, energy, area, and latency numbers can be calculate using technology parameters in the DSENT simulator [29]. Using the Bulk22LVT model in DSENT, single gates up to full devices and interconnects can be simulated. The Bulk22LVT model was used for electrical component simulation in the EE, OE, and OO desings.

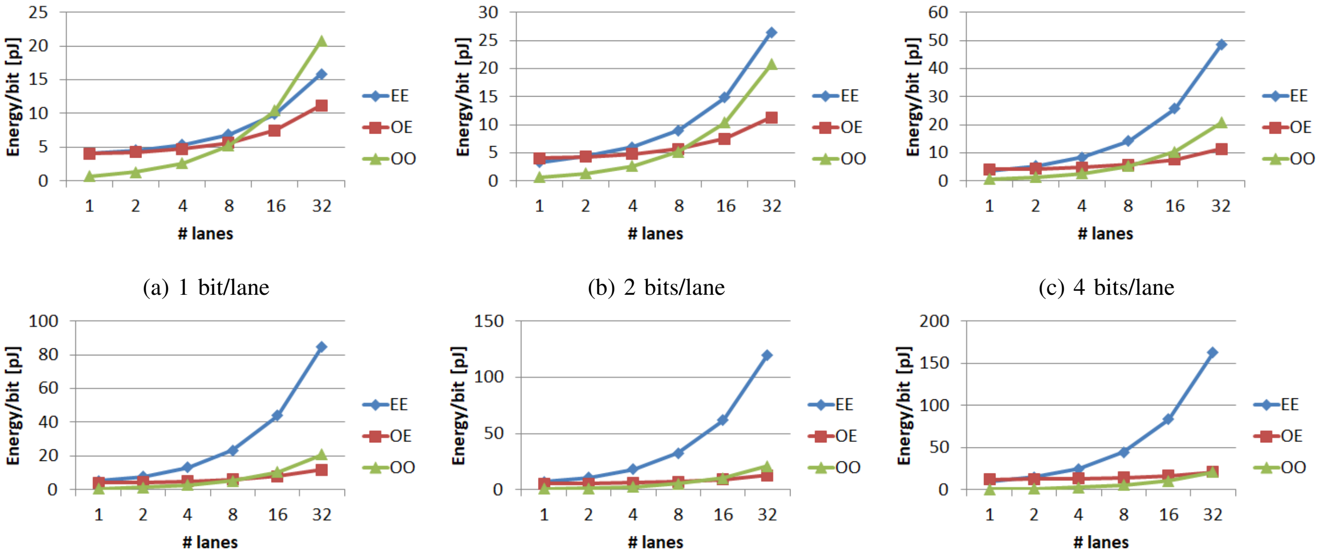
For example, a CLA's GC for a given number of bits, and the gate level depth (LD) is determined by equations below [30].

$$GC(n) = \frac{n^3 + 6n^2 + 47n}{6} \tag{5}$$

$$LD(n) = 4 + 2\lceil log_2(n-1) \rceil \tag{6}$$

Let us take $n = 8$ bits; this yields $GC(8) = 212$ and $LD(8) = 10$. Using DSENT's 22nm model, it can be calculated that 212 gates will occupy approximately $0.07\,\mathrm{nm}^2$, consuming $0.17\,\mu\mathrm{W}$ of power. The latency of the CLA can be estimated using the propagation delay of the Bulk22LVT model in conjunction with the LD of the design. With a $LD = 10$, it can be approximated that the 8-bit CLA will have a propagation delay of $2.95\,\mathrm{ns}$.

2) Photonic Devices: Recent works in silicon photonics have lead to ever-shrinking and increasingly-efficient devices. MRRs has been demonstrated with small footprints, (radius $r = 7.5\,\mu\mathrm{m}$) [18] and have been shown in WDM arrays to be highly efficient, consuming as little as $100\,\mathrm{fJ/bit}$ at $10\mathrm{Gb/s}$ [19]. Building off of these recent works, we have



(a) 1 bit/lane

(b) 2 bits/lane

(c) 4 bits/lane

(d) 8 bits/lane

(e) 16 bits/lane

(f) 32 bits/lane

Figure 4: **Energy/bit comparisons for a single MAC unit for a baseline electrical (EE), hybrid optical-electrical (OE) and all-optical (OO) designs for different number of lanes (wavelengths) and bits/lane.**

been able to use these results to give approximations for the WDM array used in the optical AND in the OE and OO implementations.

Given the radius of the MRR, we can determine the path length that the optical signal will travel. In the optical AND configuration, when a given wavelength λ is coupled to the appropriately tuned MRRs for that wavelength, the signal must pass around both MRRs to preserve the path direction shown in Figure 1(b). The path length travelled by a signal through both MRRs can be approximated using the S-shaped curved shown, which turns out to be two half-circumferences, or one circumference in length. Not including the path length to bring the signal in or out of the MRR array, the optical signal will need to travel $2\pi(7.5\,\mu\text{m}) \approx 47.1\,\mu\text{m}$.

With the path length now determined, it is quite easy to calculate the delay a signal will experience passing through the double MRR filter. Silicon has a refractive index $n_{Si} = 3.48$ at $1550\,\text{nm}$, and along with the path length $d = 47.1\,\mu\text{m}$, the delay is found through:

$$t_{MRR} = d\left(\frac{n_{Si}}{c}\right) = 0.547\,\text{ps} \tag{7}$$

MZIs have been shown to be highly energy efficient optical solutions to modulation, with some designs demonstrating as low as $32.4\,\text{fJ/bit}$ [31]. In this device, the phase-shifting arms of the MZI are $2\,\text{mm}$ in length, so in order to sync cascaded MZIs to the optical pulse frequency a precisely measured path must be placed from the output o_0 of the preceding MZI to the input port i_0 of the following MZI. This distance can be calculated in general through:

$$d_{path} = \frac{c(T_o - t_{MZI})}{n_{Si}} \tag{8}$$

or

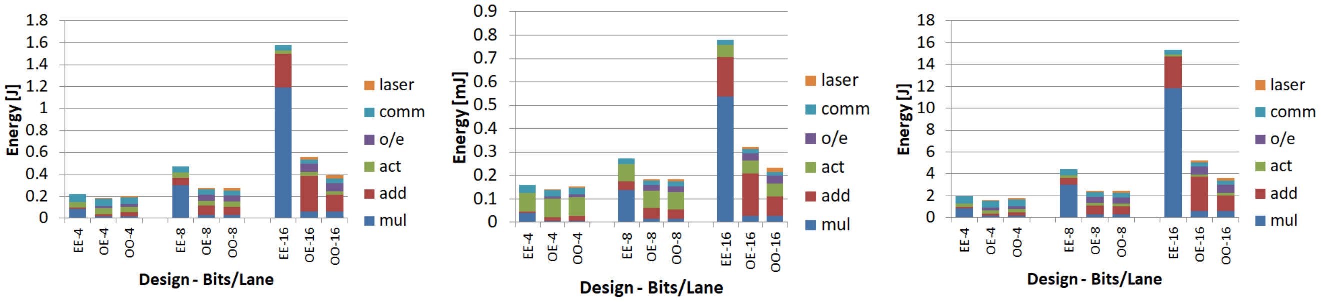
$$d_{path} = \frac{c}{n_{Si} f_o} - d_{MZI} \tag{9}$$

Where T_o is the optical period $(\frac{1}{f_o})$, t_{MZI} is the propagation delay of the MZI, and d_{MZI} is the path length of the MZI. Knowing the arm length of the MZI to be $2\,\text{mm}$, it can be calculated that $d = 6.77\,\text{mm}$ between MZIs at 10GHz. The number of MZIs for a given wavelength is the same as the number of bits that the wavelength carries. So, to accumulate n optical pulses on a single wavelength, the total accumulation length would be $d_{tot} = (n)d_{MZI} + (n-1)d_{path}$. The total propagation delay for the accumulation of 4 bit optical pulses would be:

$$t_{tot} = \Big(8(2\,\text{mm}) + 7(6.77\,\text{mm})\Big)\left(\frac{n_{Si}}{c}\right) = 0.736\,\text{ns} \tag{10}$$

B. CNN Evaluation

Several different CNN architectures (VGG16, AlexNet, ZFNet, ResNet-34, LeNet, GoogLeNet) were simulated in MATLAB to perform per-layer analysis of the number of computations (MVMs, multiplications, additions, activation functions) required for the inference phase of the network. It is necessary to compute the output shape (height, width,



(a) (b) (c)

Figure 5: **Energy consumption per component for (a) AlexNet, (b) LeNet, (c) VGG16 for all-electrical (EE), hybrid optical-electrical (OE) and all-optical (OO) architectures by considering the laser, communication, O/E, activation, addition and multiplication energy for 4, 8 and 16 bits/lane.**

Table I: **VGG16 computations [millions].**

Layer	MVM	Mul	Add	Act	Input Shape
Conv1	9.63	86.7	89.9	3.21	[224,224,3]
Conv2	206	1850	1853	3.21	[226,226,64]
Conv3	103	925	926	1.61	[114,114,64]
Conv4	206	1850	1850	1.61	[114,114,128]
Conv5	103	926	926	0.803	[58,58,128]
Conv6	206	1850	1850	0.803	[58,58,256]
Conv7	103	925	925	0.401	[30,30,256]
Conv8	206	1850	1850	0.401	[30,30,512]
Conv9	51.4	462	463	0.100	[16,16,512]
Conv10	51.4	462	463	0.100	[16,16,512]
FC1	10^{-6}	629	1259	629	[25088]
FC2	10^{-6}	16.8	33.6	16.8	[4096]
FC3	10^{-6}	16.8	33.6	16.8	[4096]

channels) of each convolutional layer. The output feature size can be calculated as

$$E = \frac{H - R + U}{U} \tag{11}$$

where H is the input feature size, R is the filter kernel size, and U is the stride size. Depending on the CNN architecture, padding may be added in accordance with the specifications of the architecture.

The number of matrix multiplications can now be determined using the output feature size. $N_{MVM} = E^2MC$ where M is the number of filters used in the convolutional layer, and C is the number of input channels. Next, the number of individual multiplications can be determined by $N_{mul} = R^2(N_{MVM})$, the number of additions can be determined by $N_{add} = N_{mul} + E^2M$, and the number of activation functions is $N_{act} = E^2M$.

Let us take the first convolutional layer (Conv1) of VGG16 for example. Conv1 has 64 filters with a kernel shape of (3,3), and the input shape fed to Conv1 is (224,224,3), so

$$N_{MVM} = 224^2(64)(3) = 9633792$$

$$N_{mul} = 3^2(N_{MVM}) = 86704128$$

and so on. Table I shows the per-layer analysis of VGG16 utilizing the calculations listed above.

C. Accelerator Evaluation

With all components of the MACs simulated to get their energy/bit, area, and propagation delays, plus the CNN architecture operations, an overall evaluation of the CNN accelerator designs can be performed.

Take the OE design in Figure 2(b) as an example for the AND shift-accumulate of two 4-bit words in all lanes as shown. The number of lanes is 4, and the number of bit per wavelength (bits/lane) is also 4. So, for a given wavelength, a synapse lane of $p = 4$ bits will require 4 cycles to compute the partial sum. This happens for all 16 wavelengths. The number of MRRs in the entire design is found to be 128, or 64 double MRR filters. It will take only 4 cycles to compute 16 4-bit ANDs, and (excluding laser power for now) the MRRs will consume only $128 \times 500\,\text{fJ} \times 4$ bits $\times 4$ cycles$=$ $1.024\,\text{nJ}$. After the o/e conversion in the EP, the bits must now be accumulated. A 4-bit CLA will have 58 gates, and at 1 GHz using the Bulk22LVT model it can be determined that all CLAs in the design will consume a total of $5.06\,\text{pJ}$. This type of analysis was done for every component in the design, including the laser sources, o/e converters, bit-shifters, and activation functions.

Once the energy consumption for each device is added up with interconnect overhead, the overall energy consumption for the OE design can be computed. Then pulling from the CNN computations performed earlier, the number of repetitions needed can be determined from the number of MVMs, as well as the energy consumption for every multiply, add, and activation function of the CNN architecure. Combining all of these gives the comprehensive performance of the accelerator for each CNN architecture on a per-layer basis.

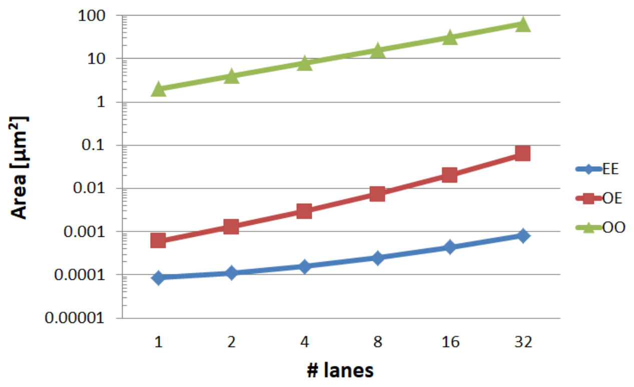


Figure 6: **Area comparison for 4 bits/lane for all-electrical (EE), hybrid (OE) and all-optical (OO) architectures.**

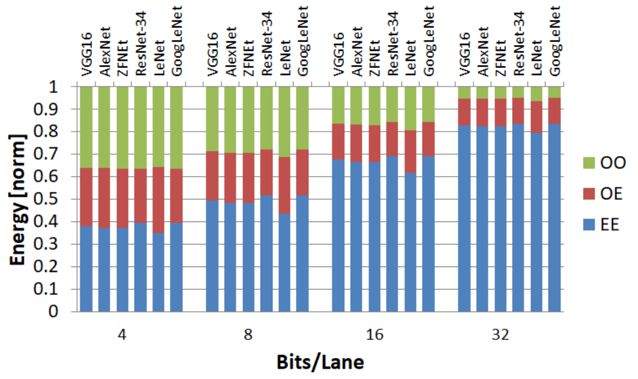


Figure 7: **Normalized energy for VGG16, AlexNet, ZFNet, ResNet-34, LeNet and GoogLeNet applications with 4, 8, 16 and 32 bits/wavelength for all-electrical (EE), hybrid optical-electrical (OE) and all-optical (OO) neural networks.**

V. Results

A. Single MAC Unit

The energy/bit for the three designs (EE, OE, OO) was computed from the device level up. It was necessary to vary both the number of lanes and the number of bits/lane to see how each design responded to the respective scaling. As it can be seen in Figure 4, the EE design grows quite large when scaling up both the number of lanes, and the number of bits per lane. As the number of bits/lane is increased, it can be seen how both optical designs' (OE, OO) energy/bit rises ever so slightly. This is because the number of optical devices like the MRR do not increase with the bits/lane, rather, they increase with respect to the number of lanes (wavelengths). The optical designs favor when the number of bits/lane is larger than the number of lanes, and the OO design drops drastically through the increase of the bits/lane due to the MZI's efficient accumulation ability.

An area analysis, shown in Figure 6, demonstrates how each design scales to changes in the number of lanes. It can be seen that the EE design occupies the least amount of area. This is expected as the 22nm model used allows complex logic to remain in a small amount of space. On the other hand, when compared to the logic gate size of 22nm technology, optical components are large. For logical AND, MRRs occupy a consideribly larger amount of area than the electrical implementation. MZIs are the largest device used in these designs, and as seen in the OO curve, their cascaded configuration contributes to a much larger area than both of the other designs. For 4 lanes at 4 bits/lane, it was found that the OE design occupied $2.78\,\mathrm{nm}^2$ more area than EE, and the OO design occupied $7.98\,\mu\mathrm{m}^2$ more area than EE.

B. Neural Network Inference Acceleration

1) Energy Efficiency: The simulation energy results across all 6 simulated CNN architectures for an inference show very promising numbers for both the OE and OO designs. Figure 7 shows the normalized energy consumption for the CNN architectures, and demonstrates each design's scaling response to changes in the number of bits/lane. Both OE and OO designs begin to outperform EE when the number of bits/lane is greater than the number of lanes. This offset allows the optical designs to utilize more optical pulses through their existing structures. This is opposed to increasing the number of wavelengths which would increase the number of optical devices. It can be seen that when the number of bits/lane is much greater than the number of lanes (32 bits/lane in 8 lanes), EE occupies a majority of the relative energy, while OO has a very small energy consumption.

Figure 5 shows the breakdown of each functional unit for 4 lanes in the AlexNet, LeNet, and VGG16 architectures. The analysis of each step in the acceleration design can be seen in these plots, and importantly, this shows how the laser source and o/e conversion in OE and OO contribute to the total energy consumption. Reference the 16 bit/lane group in the 5(a) AlexNet plot (EE-16, OE-16, OO-16). For multiplication, OE and OO's MRRs provide a high efficiency, consuming a mere 5.1% of the energy that EE does. For addition, both the EE and OE designs have similar of energies since they have electrical shift-accumulate, while OO's MRRs reduce the cost for addition by 53.8% over OE. The activation function circuitry remains the same across the designs, so there will be no variation there. There is a cost for the communication data as well; that is, the energy required to bring the data to the MACs and the energy required to send out the result. For the EE design, this is an electrical link to both bring the data in, and send it out. The OE and OO designs have a photonic link to bring the data in from laser sources, and an electrical link to carry the result out. The photonic link in the OE and OO designs consume slightly less energy than the electrical link over the

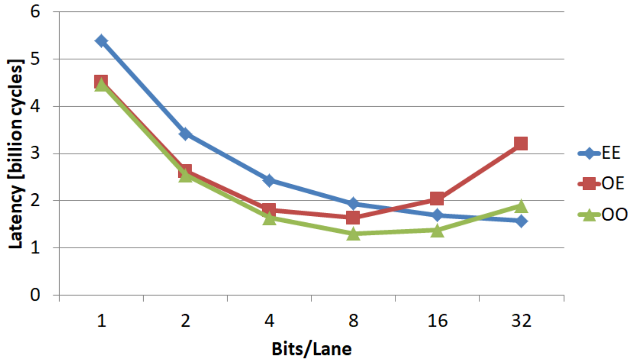


Figure 8: **Average latency for 8 lanes (wavelengths) for all-electrical (EE), hybrid (OE) and all-optical (OO) for 8 lanes (wavelengths) for different bits/lane (1-32).**

short distance.

The OE design does quite well when compared to EE, and the effect of its electrical processing (EP) unit for the shift-accumulate functionality becomes apparent for larger bits/lane. In the case of the OO design, the highly efficient MZIs respond quite well to changes in the bits/lane, keeping OO as the lowest energy consumption design for high numbers of bits/lane.

2) Latency: Latency is an important consideration for real-time CNN inferences, and our photonic designs keep latency to a minimum. Figure 8 shows the geometric mean for latency across the six CNN architectures for 8 Lanes with varying bits/lane. It can be seen that as the number of bits/lane is increased, the latency begins to fall. The EE design's latency consistently declines with an increasing bits/lane, but it can be seen that both optical designs have a U-shaped response. The latency for OE and OO designs begins to rise again since the larger bit count pushes the propagation delays over a cycle threshold. That is, only a certain amount of pulses can be clumped into a single cycle at the optical 10GHz before extra cycles are required to process this data.

It is also desirable to see the latency response on a per-layer basis for the CNN architectures. As an example, Figure 9 shows the latency at every layer for ZFNet operating with 8 lanes at 8 bits/lane. The latency relative difference between the three accelerator designs is consistent, and reinforces that the STR methodology scales very well to varing input sizes. The OE and OO designs do quite well in latency for this configuration, with the OO design having the least delay. For large convolutional layers like Conv 2, the absolute difference between OO and the other two designs is significant, while in less computationally demanding layers like the fully-connected layers, the absolute difference is not as great. In the Conv 2 later, OO is 31.9% faster than EE, and 18.6% faster than OE.

Table II: **Energy breakdown by component for 4 lanes, 16 bits/lane [mJ] for all-electrical (EE), hybrid (OE) and all-optical (OO) for various CNN applications.**

CNN	Des	Mul	Add	Act	o/e	Comm	Laser
ResNet -34	EE	3634	847	1.09	0	139	0
	OE	187	910	1.09	227	118	59.8
	OO	187	420	1.09	227	118	91.0
Goog-LeNet	EE	1578	368	1.22	0	60.4	0
	OE	81.0	396	1.22	98.8	51.4	26.0
	OO	81.0	183	1.22	98.8	51.4	35.1
ZFNet	EE	1225	313	34.2	0	46.9	0
	OE	62.9	336	34.2	76.6	39.9	20.1
	OO	62.9	155	34.2	76.6	39.9	30.4

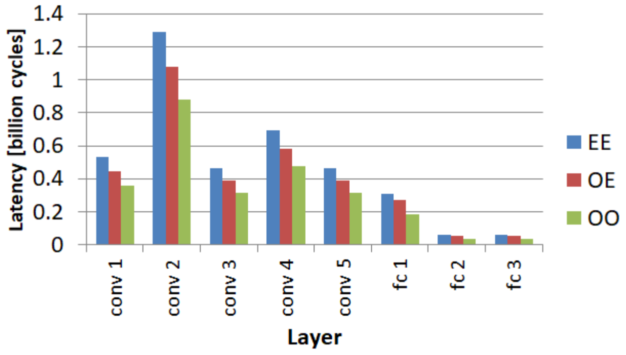


Figure 9: **ZFNet latency for 8 lanes with 8 bits/lane at different layers for all-electrical (EE), hybrid (OE) and all-optical (OO) architectures.**

3) Energy-Delay Product: The energy-delay product (EDP) will be a useful parameter in understanding the performance of the proposed designs across both the energy consumption and latency performance. Figure 10 shows the normalized EDP for the six CNN architectures. It can be seen that, again, the OO design offers the best performance when the number of bits/lane is high. This is promising considering the U-shape of the latency curve for OO, but it is still able to outperform the other two designs with it's high energy efficiency. The EDP for EE scales quickly with increasing bits/lane, as does the OE design. However, the OO design remains very low as the bits/lane is scaled up, staying resilient to changes in the input data size. For 4 lanes at 8 bits/lane, For 4 lanes at 16 bits/lane, OO's geometric mean of EDP is improved by 73.9% over EE and OE's by 48.4% over EE. This shows how the energy-efficient MRRs combined with the low-latency MZIs produce a worthy photonic acceleration platform for CNNs.

VI. Related Work

A. Photonic NoCs

Photonic NoC architectures have been proposed to overcome the bandwidth and throughput limitations of electrical

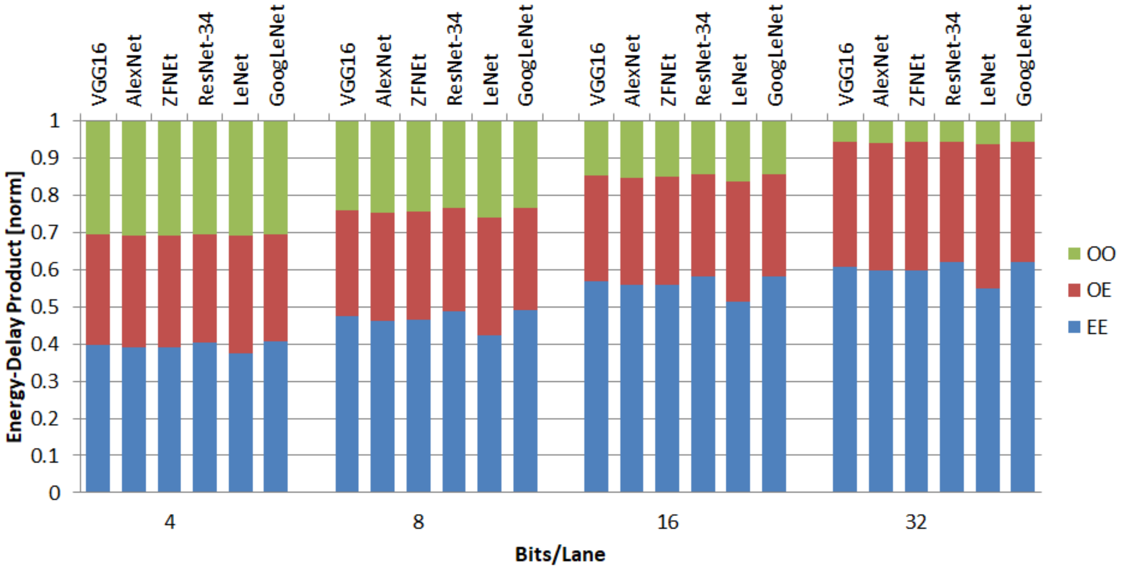


Figure 10: **Normalized EDP with 4 lanes for VGG16, AlexNet, ZFNet, ResNet-34, LeNet and GoogLeNet applications with 4, 8, 16 and 32 bits/wavelength for all-electrical (EE), hybrid optical-electrical (OE) and all-optical (OO) neural networks.**

interconnects for manycore architectures [32], [33], [34]. These include a wide variety of architectures ranging from rings, crossbars, decomposed crossbars and 3D stacked architectures [35], [36], [37], [38], [39]. Most photonic NoC architectures employ either Multiple-Write-Single-Read (MWSR) or Single-Write-Multiple- Read (SWMR) communication paradigms that trades off between energy consumption and performance. WDM has been used for increasing the bandwidth-density in manycore heterogeneous architectures and bandwidth and power scaling techniques have been proposed to further improve the on-chip communication [40], [41]. Photonic NoCs research has focused exclusively on improving the energy-efficiency of inter-core communication.

B. Programmable Photonics

Since photonics is advantageous for communication, optical devices and architectures have been investigated for optical computing. Optical logic gates have been proposed and implemented using myriad of techniques including self modulation of microring resonators, directed logic array and several recent programmable photonics initiatives. In [42], the authors demonstrate the design of optical AND and NAND logic using MRRs, whereas in [43], [44], [45], authors show the implementation of several logical operations using MRRs such as XNOR, XOR and design of priority encoders and basic adder units.

D.A.B. Miller recently showed on how to cascade multiple MZI to combine or add the amplitude of multiple input signals by adjusting the phase shifts on different arms of the individual MZI [46], [47], [48] and this was experimentally proven [49]. Using the principles of the universal beam coupler from Miller [46], a Field Programmable Photonic Array where a complete architectural solution of photonics device that could be programmed for the implementation of arbitrary simple, complex or even simultaneous circuits [50], [51]. This is analogous to Field Programmable Gate Array (FPGA) where arbitrary circuits can be designed for a programmable nanophotonic processor built with hundreds of MZIs. While prior work on programmable photonics is focused on designing flexible circuits with interconnected MZIs, in PIXEL we focus on combining MRRs and MZIs together to create optical MAC units to solve specific accelerator applications.

Interest in spiking neuromorphic networks (SNNs) using conventional CMOS circuits gained steam through several work such as True North from IBM [52], SpiNNaker [53], and others. However, limitation of neuromorphic processors that requires a large number of interconnects ($\sim$ 100s of many-to-one fan-in) and significant amount of multicasting has naturally created a tremendous interest in using photonics for such computation. Large-scale integrated photonic platforms have further helped in paving the way for developing neuromorphic photonic architectures.

Recent work by Pruncal et.al. have shown on the basic implementation of spiking photonic neural network (PNN) using the leaky integrate and fire (LIF) model [54], [55], [56]. PNN interacts with two tunable filter banks using MRRs - one filter bank represents excitatory connections whereas the other filter bank represents inhibitory connections and the two weighted subsets of the broadcast channels are dropped to a balanced photodiode where the output current represents the total power, thus computing the weighted sum of WDM inputs. This in turn will transduce an electronic signal which is capable of modulating a laser device, thereby achieving PNN functionality. In PIXEL, we incorporate a bank of MRRs for implementing bitwise AND operation, however backend processing either electrically or optically ensures the additive functionality.

C. Adaptive NoCs Using ML

Dynamic and adaptive NoCs have been proposed to increase network throughput, while reducing latency, optimizing power consumption, and increasing reliability for many-core systems. Dynamic-voltage frequency-scaling (DVFS) and power-gating (PG) have been used with deep reinforcement learning (Deep-RL) [57], [58] and ridge regression [59] techniques to increase energy efficiency of NoCs. ML has also been used to increase reliability in fault-tolerant NoC systems, through the utilization of Q-learning [60] and decision trees [61]. The work proposed in [62] uses a holistic approach with Q-learning that increases energy efficiency through multifunction adaptive channels, while increasing reliability with adaptive error detection and correction.

VII. Conclusion

In this paper, we have proposed two PIXEL photonic neural network accelerators based around MAC units: a hybrid optical and electrical design (OE) and an all optical design (OO). Our proposed designs have increased performance over traditional electrical accelerators through the minimization of energy consumption and latency. We have demonstrated the design-space exploration in determination of efficient lanes (wavelengths) and bits/lane values for our PIXEL accelerators, as well as an evaluation of the accelerators across several CNN architectures. We found that the optical bitwise multiplication utilizing MRRs gave a 94.9% increase in energy improvement for both OE and OO designs, while the OO design had a further 53.8% improvement for accumulation using MZIs over the electrical addition in the hybrid OE design. The all-optical OO design gave the best performance, having a minimal EDP for high bits/lane with an improvement of 73.9% over the all-electrical EE and 48.4% over the hybrid OE. Our OE and OO PIXEL designs exhibited efficient energies with minimal latency that leveraged the high parallelism of CNN architectures through the innate properties of optics, at the cost of increased areas for the designs.

Acknowledgment

This research was partially supported by NSF grants CCF-1513606, CCF-1703013, CCF-1901192, CCF-1513923, CCF-1547034, CCF-1547035, CCF-1547036, CCF-1702980, and CCF-1901165. We sincerely thank the anonymous reviewers for their excellent feedback.

References

[1] M. Horowitz, “Computing’s energy problem (and what we can do about it),” in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, Feb 2014, pp. 10–14.

[2] Y. Shen, N. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljacic, “Deep learning with coherent nanophotonic circuits,” *Nature Photonics*, vol. 11, 07 2017.

[3] B. J. Shastri, A. N. Tait, T. Ferreira de Lima, M. A. Nahmias, H.-T. Peng, and P. R. Prucnal, *Neuromorphic Photonics, Principles of.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2018, pp. 1–37. [Online]. Available: https://doi.org/10.1007/978-3-642-27737-5_702-1

[4] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, “Dark silicon and the end of multicore scaling,” in *Proceedings of the 38th Annual International Symposium on Computer Architecture*, ser. ISCA ’11. New York, NY, USA: ACM, 2011, pp. 365–376. [Online]. Available: http://doi.acm.org/10.1145/2000064.2000108

[5] W. W. Hwu and S. Patel, “Guest editors’ introduction: Accelerator architectures,” *IEEE Micro*, vol. 28, no. 04, pp. 4–12, jul 2008.

[6] R. Hameed, W. Qadeer, M. Wachs, O. Azizi, A. Solomatnikov, B. C. Lee, S. Richardson, C. Kozyrakis, and M. Horowitz, “Understanding sources of inefficiency in general-purpose chips,” in *Proceedings of the 37th Annual International Symposium on Computer Architecture*, ser. ISCA ’10. New York, NY, USA: ACM, 2010, pp. 37–47. [Online]. Available: http://doi.acm.org/10.1145/1815961.1815968

[7] 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, Jan 2017.

[8] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P.-l. Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, T. V. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C. R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, D. Killebrew, A. Koch, N. Kumar, S. Lacy, J. Laudon, J. Law, D. Le, C. Leary, Z. Liu, K. Lucke, A. Lundin, G. MacKean, A. Maggiore, M. Mahony, K. Miller, R. Nagarajan, R. Narayanaswami, R. Ni, K. Nix, T. Norrie, M. Omernick, N. Penukonda, A. Phelps, J. Ross, M. Ross, A. Salek, E. Samadiani, C. Severn, G. Sizikov, M. Snelham, J. Souter, D. Steinberg, A. Swing, M. Tan, G. Thorson, B. Tian, H. Toma, E. Tuttle, V. Vasudevan, R. Walter, W. Wang, E. Wilcox, and D. H. Yoon, “In-datacenter performance analysis of a tensor processing unit,” in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, ser. ISCA ’17. New York, NY, USA: ACM, 2017, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/3079856.3080246

[9] A. Putnam, A. M. Caulfield, E. S. Chung, D. Chiou, K. Constantinides, J. Demme, H. Esmaeilzadeh, J. Fowers, G. P. Gopal, J. Gray, M. Haselman, S. Hauck, S. Heil, A. Hormati, J.-Y. Kim, S. Lanka, J. Larus, E. Peterson, S. Pope, A. Smith, J. Thong, P. Y. Xiao, and D. Burger, “A reconfigurable fabric for accelerating large-scale datacenter services,” in *Proceeding of the 41st Annual International Symposium on Computer Architecuture*, ser. ISCA ’14. Piscataway, NJ, USA:

IEEE Press, 2014, pp. 13–24. [Online]. Available: http://dl.acm.org/citation.cfm?id=2665671.2665678

[10] A. Azizimazreah and L. Chen, "Shortcut mining: Exploiting cross-layer shortcut reuse in dcnn accelerators," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2019, pp. 94–105.

[11] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing fpga-based accelerator design for deep convolutional neural networks," in *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '15. New York, NY, USA: ACM, 2015, pp. 161–170. [Online]. Available: http://doi.acm.org/10.1145/2684746.2689060

[12] B. Reagen, P. Whatmough, R. Adolf, S. Rama, H. Lee, S. K. Lee, J. M. Hernández-Lobato, G. Wei, and D. Brooks, "Minerva: Enabling low-power, highly-accurate deep neural network accelerators," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, June 2016, pp. 267–278.

[13] H. Jang, J. Kim, J.-E. Jo, J. Lee, and J. Kim, "Mnnfast: A fast and scalable system architecture for memory-augmented neural networks," in *Proceedings of the 46th International Symposium on Computer Architecture*, ser. ISCA '19. New York, NY, USA: ACM, 2019, pp. 250–263. [Online]. Available: http://doi.acm.org/10.1145/3307650.3322214

[14] L. Song, J. Mao, Y. Zhuo, X. Qian, H. Li, and Y. Chen, "Hypar: Towards hybrid parallelism for deep learning accelerator array," 02 2019, pp. 56–68.

[15] H. Kwon, A. Samajdar, and T. Krishna, "Maeri: Enabling flexible dataflow mapping over dnn accelerators via reconfigurable interconnects," *SIGPLAN Not.*, vol. 53, no. 2, pp. 461–475, Mar. 2018. [Online]. Available: http://doi.acm.org/10.1145/3296957.3173176

[16] L. Xu, W. Zhang, Q. Li, J. Chan, H. L. R. Lira, M. Lipson, and K. Bergman, "40-gb/s dpsk data transmission through a silicon microring switch," *IEEE Photonics Technology Letters*, vol. 24, no. 6, pp. 473–475, March 2012.

[17] S. Manipatruni, K. Preston, L. Chen, and M. Lipson, "Ultra-low voltage, ultra-small mode volume silicon microring modulator." *Optics express*, vol. 18 17, pp. 18 235–42, 2010.

[18] 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," *Opt. Express*, vol. 19, no. 21, pp. 20 435–20 443, Oct 2011.

[19] X. Zheng, F. Liu, J. Lexau, D. Patil, G. Li, Y. Luo, H. D. Thacker, I. Shubin, J. Yao, K. Raj, R. Ho, J. E. Cunningham, and A. V. Krishnamoorthy, "Ultralow power 80 gb/s arrayed cmos silicon photonic transceivers for wdm optical links," *Journal of Lightwave Technology*, vol. 30, no. 4, pp. 641–650, Feb 2012.

[20] G. Li, A. V. Krishnamoorthy, I. Shubin, J. Yao, Y. Luo, H. Thacker, X. Zheng, K. Raj, and J. E. Cunningham, "Ring resonator modulators in silicon for interchip photonic links," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 19, no. 6, pp. 95–113, Nov 2013.

[21] M. Georgas, J. Leu, B. Moss, C. Sun, and V. Stojanovic, "Addressing link-level design tradeoffs for integrated photonic interconnects," in *CICC*, 2011, pp. 1–8.

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

[23] J. Ahn, M. Fiorentino, R. G. Beausoleil, N. Binkert, A. Davis, D. Fattal, N. P. Jouppi, M. McLaren, C. M. Santori, R. S. Schreiber, S. M. Spillane, D. Vantrease, and Q. Xu, "Devices and architectures for photonic chip-scale integration," *Applied Physics A*, vol. 95, no. 4, pp. 989–997, 2009. [Online]. Available: http://dx.doi.org/10.1007/s00339-009-5109-2

[24] L. Zhou, K. Kashiwagi, K. Okamoto, R. P. Scott, N. K. Fontaine, D. Ding, V. Akella, and S. J. B. Yoo, "Towards athermal optically-interconnected computing system using slotted silicon microring resonators and rf-photonic comb generation," *Applied Physics A*, October 2008.

[25] S. Manipatruni, R. Dokania, B. Schmidt, N. Droz, C. Poitras, A. Apsel, and M. Lipson, "Wide temperature range operation of micron-scale silicon electro-optic modulators," *Optics Letters*, vol. 33, no. 19, September-October 2008.

[26] A. H. Namin, K. Leboeuf, R. Muscedere, H. Wu, and M. Ahmadi, "Efficient hardware implementation of the hyperbolic tangent sigmoid function," in *2009 IEEE International Symposium on Circuits and Systems*, May 2009, pp. 2117–2120.

[27] B. Zamanlooy and M. Mirhassani, "Efficient vlsi implementation of neural networks with hyperbolic tangent activation function," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 1, pp. 39–48, Jan 2014.

[28] P. Judd, J. Albericio, T. Hetherington, T. M. Aamodt, and A. Moshovos, "Stripes: Bit-serial deep neural network computing," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Oct 2016, pp. 1–12.

[29] C. Sun, C. O. Chen, G. Kurian, L. Wei, J. Miller, A. Agarwal, L. Peh, and V. Stojanovic, "Dsent - a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling," in *2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip*, May 2012, pp. 201–210.

[30] O. A. L. A. Ridha, "Performance estimation of n-bit classified adders," *International Journal of Computer Applications*, vol. 80, no. 9, pp. 11–15, October 2013.

[31] J. Ding, R. Ji, L. Zhang, and L. Yang, "Electro-optical response analysis of a 40 gb/s silicon mach-zehnder optical modulator," *Journal of Lightwave Technology*, vol. 31, no. 14, pp. 2434–2440, July 2013.

[32] D. A. B. Miller, "Device requirements for optical interconnects to silicon chips," *Proceedings of the IEEE*, vol. 97, no. 7, pp. 1166–1185, July 2009.

[33] 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 39th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO 39. Washington, DC, USA: IEEE Computer Society, 2006, pp. 492–503. [Online]. Available: https://doi.org/10.1109/MICRO.2006.28

[34] A. Shacham, K. Bergman, and L. P. Carloni, "Photonic networks-on-chip for future generations of chip multiprocessors," *IEEE Transactions on Computers*, vol. 57, no. 9, pp. 1246–1260, Sept 2008.

[35] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. Jouppi, M. Fiorentino, A. Davis, N. Binkert, R. Beausoleil, and J. Ahn, "Corona: System implications of emerging nanophotonic technology," in *Computer Architecture, 2008. ISCA '08. 35th International Symposium on*, June 2008, pp. 153–164.

[36] A. K. 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 29th ACM on Interna-*

tional Conference on Supercomputing, ser. ICS '15, 2015, pp. 273–282.

[37] R. Morris, A. Kodi, and A. Louri, "Dynamic reconfiguration of 3d photonic networks-on-chip for maximizing performance and improving fault tolerance," in *Microarchitecture (MICRO), 2012 45th Annual IEEE/ACM International Symposium on*, 2012, pp. 282–293.

[38] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A. Choudhary, "Firefly: Illuminating future network-on-chip with nanophotonics," *SIGARCH Comput. Archit. News*, vol. 37, no. 3, pp. 429–440, Jun. 2009.

[39] N. Kirman and J. F. Martínez, "A power-efficient all-optical on-chip interconnect using wavelength-based oblivious routing," *SIGARCH Comput. Archit. News*, vol. 38, no. 1, pp. 15–28, Mar. 2010.

[40] 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)*, Feb 2018, pp. 480–491.

[41] Y. Demir and N. Hardavellas, "Slac: Stage laser control for a flattened butterfly network," in *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, March 2016, pp. 321–332.

[42] Q. Xu and M. Lipson, "All-optical logic based on silicon micro-ring resonators," *Opt. Express*, vol. 15, no. 3, pp. 924–929, Feb 2007. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-15-3-924

[43] Q. Xu and R. Soref, "Reconfigurable optical directed-logic circuits using microresonator-based optical switches," *Opt. Express*, vol. 19, no. 6, pp. 5244–5259, Mar 2011. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-19-6-5244

[44] C. Qiu, X. Ye, R. Soref, L. Yang, and Q. Xu, "Demonstration of reconfigurable electro-optical logic with silicon photonic integrated circuits," *Opt. Lett.*, vol. 37, no. 19, pp. 3942–3944, Oct 2012. [Online]. Available: http://ol.osa.org/abstract.cfm?URI=ol-37-19-3942

[45] L. Zhang, J. Ding, Y. Tian, R. Ji, L. Yang, H. Chen, P. Zhou, Y. Lu, W. Zhu, and R. Min, "Electro-optic directed logic circuit based on microring resonators for xor/xnor operations," *Opt. Express*, vol. 20, no. 11, pp. 11 605–11 614, May 2012. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-20-11-11605

[46] D. A. B. Miller, "Self-aligning universal beam coupler," *Opt. Express*, vol. 21, no. 5, pp. 6360–6370, Mar 2013. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-21-5-6360

[47] ——, "Perfect optics with imperfect components," *Optica*, vol. 2, no. 8, pp. 747–750, Aug 2015. [Online]. Available: http://www.osapublishing.org/optica/abstract.cfm?URI=optica-2-8-747

[48] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, "Experimental realization of any discrete unitary operator," *Phys. Rev. Lett.*, vol. 73, pp. 58–61, Jul 1994. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.73.58

[49] A. Ribeiro, A. Ruocco, L. Vanacker, and W. Bogaerts, "Demonstration of a 4&#x2009;&#x00d7;&#x2009;4-port universal linear circuit," *Optica*, vol. 3, no. 12, pp. 1348–1357, Dec 2016. [Online]. Available: http://www.osapublishing.org/optica/abstract.cfm?URI=optica-3-12-1348

[50] D. Pérez, I. Gasulla, and J. Capmany, "Field-programmable photonic arrays," *Opt. Express*, vol. 26, no. 21, pp. 27 265–27 278, Oct 2018. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-26-21-27265

[51] N. C. Harris, J. Carolan, D. Bunandar, M. Prabhu, M. Hochberg, T. Baehr-Jones, M. L. Fanto, A. M. Smith, C. C. Tison, P. M. Alsing, and D. Englund, "Linear programmable nanophotonic processors," *Optica*, vol. 5, no. 12, pp. 1623–1631, Dec 2018. [Online]. Available: http://www.osapublishing.org/optica/abstract.cfm?URI=optica-5-12-1623

[52] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014. [Online]. Available: https://science.sciencemag.org/content/345/6197/668

[53] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The spinnaker project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, May 2014.

[54] A. N. Tait, J. Chang, B. J. Shastri, M. A. Nahmias, and P. R. Prucnal, "Demonstration of wdm weighted addition for principal component analysis," *Opt. Express*, vol. 23, no. 10, pp. 12 758–12 765, May 2015. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-23-10-12758

[55] A. N. Tait, T. F. de Lima, E. Zhou, A. X. Wu, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal, "Neuromorphic photonic networks using silicon photonic weight banks," *Scientific Reports*, vol. 7, August 2017.

[56] 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, Nov 2014.

[57] Q. Fettes, M. Clark, R. Bunescu, A. Karanth, and A. Louri, "Dynamic voltage and frequency scaling in nocs with supervised and reinforcement learning techniques," *IEEE Transactions on Computers*, vol. 68, no. 3, pp. 375–389, March 2019.

[58] H. Zheng and A. Louri, "An energy-efficient network-on-chip design using reinforcement learning," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*, June 2019, pp. 1–6.

[59] M. Clark, R. Bunescu, A. Kodi, and A. Louri, "Lead: Learning-enabled energy-aware dynamic voltage/frequency scaling in nocs," in *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, June 2018, pp. 1–6.

[60] K. Wang, A. Louri, A. Karanth, and R. Bunescu, "High-performance, energy-efficient, fault-tolerant network-on-chip design using reinforcement learning," in *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2019, pp. 1166–1171.

[61] D. DiTomaso, T. Boraten, A. Kodi, and A. Louri, "Dynamic error mitigation in nocs using intelligent prediction techniques," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Oct 2016, pp. 1–12.

[62] K. Wang, A. Louri, A. Karanth, and R. Bunescu, "Intellinoc: A holistic design framework for energy-efficient and reliable on-chip communication for manycores," in *Proceedings of the 46th International Symposium on Computer Architecture*, ser. ISCA '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 589–600. [Online]. Available: https://doi.org/10.1145/3307650.3322274