Toward a Behavioral-Level End-to-End Framework for Silicon Photonics Accelerators

Emily Lattanzio*, Ranyang Zhou†, Arman Roohi‡, Abdallah Khreishah†, Durga Misra†, Shaahin Angizi†
*Webb School of Engineering, High Point University, High Point, NC, USA
†Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA
‡School of Computing, University of Nebraska–Lincoln, Lincoln, NE, USA
elattanz@highpoint.edu, shaahin.angizi@njit.edu

2022 IEEE 13th International Green and Sustainable Computing Conference (IGSC) | 978-1-6654-6550-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/IGSC55832.2022.9969371

***Abstract*—Convolutional Neural Networks (CNNs) are widely used due to their effectiveness in various AI applications such as object recognition, speech processing, etc., where the multiply-and-accumulate (MAC) operation contributes to $\sim$95% of the computation time. From the hardware implementation perspective, the performance of current CMOS-based MAC accelerators is limited mainly due to their von-Neumann architecture and corresponding limited memory bandwidth. In this way, silicon photonics has been recently explored as a promising solution for accelerator design to improve the speed and power-efficiency of the designs as opposed to electronic memristive crossbars. In this work, we briefly study recent silicon photonics accelerators and take initial steps to develop an open-source and adaptive crossbar architecture simulator for that. Keeping the original functionality of the MNSIM tool [1], we add a new photonic mode that utilizes the pre-existing algorithm to work with a photonic Phase Change Memory (pPCM) based crossbar structure. With inputs from the CNN's topology, the accelerator configuration, and experimentally-benchmarked data, the presented simulator can report the optimal crossbar size, the number of crossbars needed, and the estimation of total area, power, and latency.**

***Index Terms*—Silicon photonics, accelerator, convolutional neural network, crossbar**

I. Introduction

With the rise in computer vision and machine learning projects, CNNs have become essential in software algorithm developments. Especially, their strengths in feature extraction have made them applicable to a wide variety of applications from facial recognition to natural language processing [1], [2]. As CNNs have grown in usage, their corresponding hardware has had to develop to improve their performance and scalability. Current research focuses mainly on application-specific integrated circuits (ASIC) accelerators to reduce the power consumption and run time of these neural networks as opposed to general-purpose CPUs and GPUs. With multiply-and-accumulate (MAC) operation consisting of $\sim$95% of the CNN's computation time, selecting the best architecture is vital in optimizing the performance parameters [3]–[6].

Many CNN accelerators have been proposed in the electronic domain, which have limitations due to the von-Neumann structure. The separate memory and processing units lead to a high energy requirement in moving data and limited bandwidth in how fast data transfers can be made. Silicon Photonics is a promising alternative to CMOS-based electronics since its inherent parallel structure and light speed computations increase speed, bandwidth, and power-efficiency [7]–[9]. The optics domain reduces the complexity of matrix multiplication from $O(N^2)$ to $O(1)$ which leads to a reduction in computational time of MAC operations [2], [7], [8]. In addition, silicon photonics has allowed the creation of CMOS-compatible integrated photonic devices, which combine the speed of optical computation and the lower cost of CMOS manufacturing [7], [10]. As silicon photonics neural network accelerators have shown to be an efficient solution to the von-Neumann architecture bottleneck in conventional designs [11], [12], *developing a behavioral-level end-to-end framework for silicon photonics accelerators* is essentially and widely needed.

In this study, we take the initial steps to realize an open-source simulation framework for silicon photonics accelerators developed on top of MNSIM V1.1 [1]. We keep the existing ReRAM-crossbar functionality of MNSIM and add a new simulation mode for the photonics domain. This simulation mode is set to give a decent estimation of the total area, energy, latency, and power of the pPCM-crossbar-based CNN accelerators. The presented framework in this work is mainly developed based on the silicon photonics accelerator in [12] and tested for various crossbar sizes. This work can provide a proper guideline and flexibility in comparing electronics and photonics crossbar accelerators considering various network structures.

II. Background

A. Neuromorphic Computing with ReRAM Crossbars

Neural networks are computational models developed based on the functionality of neurons and synapses in the human brain. Inputs sent into a node are multiplied by weights, which are representative of neurotransmitters sent between cells, and added with the bias value of the node. Deep Neural Networks (DNNs) consist of multiple layers of these nodes to complete numerous MAC calculations. Through setting inputs/outputs and backpropagation, the values of the weights and biases are optimized to best model a set of data, which is later used to predict future results. Furthermore, applying non-linear activation functions to inner layers of neurons, such as ReLU, Sigmoid, and Tanh, allows networks to model the behavior of non-linear data-sets.

978-1-6654-6550-2/22/$31.00 ©2022 IEEE

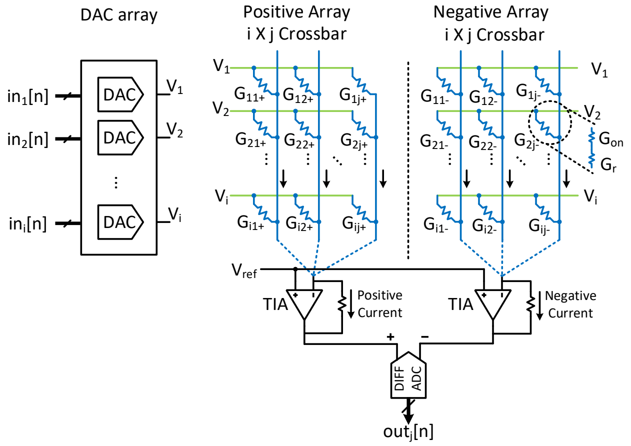


Fig. 1: Hardware implementation of a single $M \times M$ ReRAM crossbar array pair (positive and negative array) as an analog dot-product engine [15], [19].

In the realm of DNN acceleration, analog resistive crossbar memory is a popular memory array structure due to its high memory accessing bandwidth and *in-situ* computing capability. The current-mode weighted summation operations perform the MAC computations in the artificial neural network, making it one of the most promising candidates as the basic computing unit for neural network accelerator design [13]–[16]. Architecture ISAAC [17] uses this model to improve the throughput and energy by 14.8× and 5.5×, respectively, compared to a well-known ASIC architecture. PipeLayer [18] achieves the speedup and energy saving of 42.45× and 7.17×, respectively, compared with a GPU platform on average. However, many non-ideal effects, such as IR-drop (i.e., wire resistance), Stuck-At-Fault (SAF), thermal noise, shot and random telegraph noise [19], [20], are hampering the progress of hardware implementations of large-scale DNNs on ReRAM crossbar-based electronics accelerators.

The primary computation performed by analog ReRAM crossbars is the current-mode weighted summation operation (i.e., dot-product), where the architecture of the crossbar and its peripheral circuits are shown in Fig. 1. This array setup is widely used in crossbar-based dot-product engines [19], [21] for performing convolution computation with positive and negative kernel values. The inputs to the crossbar array are n-bit binary bit-strings, as shown in Fig. 1, which are first converted by the Digital-to-Analog Converter (DAC) array into voltages V_i. Since the reference voltage V_{ref} is set to $V_{\text{DD}}/2$, the current flowing into the differential ADC in the j-th column pair (i.e., two corresponding columns in the positive and the negative array) can be described as:

$$I_{\text{ADC},j} = \sum_{i=1}^{M} \Big((V_i - V_{\text{ref}}) \cdot (G^{+}_{i,j} - G^{-}_{i,j}) \Big) \tag{1}$$

where $G^{\pm}_{i,j}$ is the conductance of a ReRAM cell indexed by i and j in the positive and negative arrays, respectively. Equation (1) performs the dot-product computation between two vectors $\boldsymbol{V} - V_{\text{ref}}$ and $\boldsymbol{G}^{+}_{:,j} - \boldsymbol{G}^{-}_{:,j}$. However, a software-hardware co-design is essential when using the ReRAM crossbar array since mapping the DNN parameters into the crossbar-based accelerator requires a series of signal conversions, as introduced in [18], [19].

B. Behavioral-Level Simulators

While many crossbar-based neuromorphic computing simulators have been developed in the electronic domain such as MNSIM [1], Neurosim [22], etc., in this short study, we mainly focus on MNSIM. The ReRAM-Based Neuromorphic Computing System (MNSIM) simulates memristor-crossbar CNN accelerators to determine the most efficient crossbar size and crossbar number along with the corresponding area, energy, power, and latency of that design running neural networks. The weighting function is calculated through a matrix of conductance values saved in the memristor cells multiplied by a vector of input voltages. The activation function of the output voltages is performed through peripheral modules, which the user can design in the input configuration file to MNSIM. Along with accelerator configuration, the user inputs the topology of the CNN for the accelerator to use. Details such as the number of layers, types of layers, kernel size, stride, and input/output length are required. MNSIM optimizes the accelerator architecture by targeting one parameter specific, either total area, energy, power, or latency, which is selectable by the user. On the other hand, Neurosim [22] also estimates the area, latency, dynamic energy, and leakage power of accelerator architectures, but unlike MNSIM does not focus on CNNs. NeuroSim can simulate designs with SRAM, digital emerging nonvolatile memory (eNVM), and analog eNVM synaptic devices.

C. Silicon Photonics Accelerators

Current silicon photonics accelerators can be categorized into two groups: ones based on micro-resonators [7], [9], [23] and ones based on p-PCM [12]. An array of microring or microdisk resonators are commonly used in these accelerators to perform MAC computations. Alternatively, pPCM in a crossbar structure has also been used to perform the same calculations. A well-designed silicon photonics CNN accelerator has been presented based on a Microring Resonator (MRR) crossbar structure in [23]. In this design, the programmable nanophotonic processor removes inner loops, which have previously been shown to cause resonator-like feedback with MRRs, allowing it to be more applicable for large-scale integration. In addition to the crossbar structure, a frequency comb source is utilized along with an erbium-doped amplifier to create a multi-wavelength optical source. MAC operations are then completed through the multi-wavelength input being altered by the phase shifts of the MRR cells, which are set according to the weight matrix of the neural network.

Similar to this work, in [12], another silicon photonics crossbar-based CNN accelerator has been presented, that leverages the pPCM instead of MRRs. As shown in Fig. 2, each input is encoded into a wavelength on a photonic-chip-based microcomb and sent into the on-chip MAC unit of

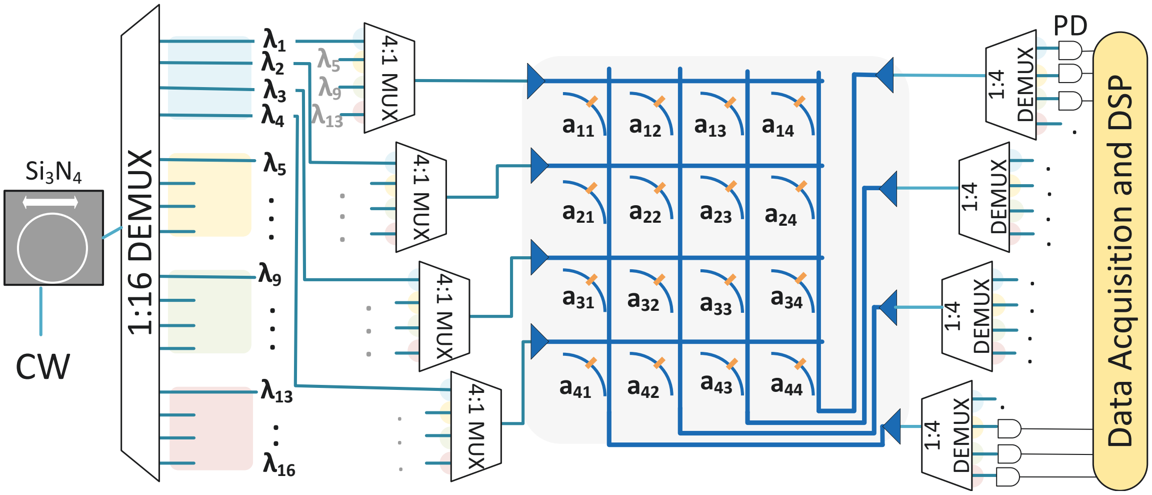


Fig. 2: Diagram of the all-optical dot-product engine. A photonic frequency comb generates the input vectors by using a continuous-wave (CW) laser device equipped with wavelength division multiplexers (MUXs). A wavelength multiplexer is used to group the entries of different input vectors sent to the on-chip MAC unit to perform the computation. Following the combination of the correct wavelengths using a wavelength division demultiplexer (DEMUX), the dot-product results are obtained from the photodetectors (PD) followed by digital signal processing (DSP) unit [12].

crossbars. The pPCM cells store the values of a kernel matrix, by absorbing a certain value of light determined by their phase configuration, and alter the wavelengths of the incoming signals to perform the weighing function and feature filtering. Our simulator is modeled around this accelerator design with the pPCM crossbar structure.

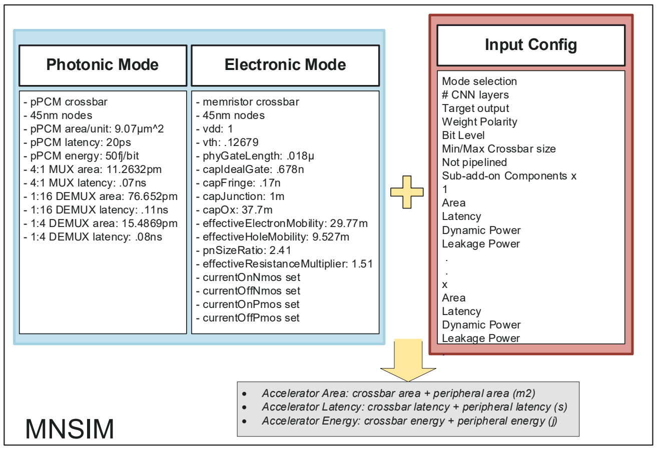
III. Proposed Flexible Framework

The proposed framework is developed on top of the MNSIM [1] and enhanced to support new features in a new photonics simulation mode with a simplified input file layout.

A. Input File Configuration

As shown in Fig. 3, the input file to the adapted MNSIM consists of two components: a network component and a configuration component. The network component remains identical to the original MNSIM [1], with each layer in the CNN topology being specified as either a convolutional or Fully-Connected (FC) layer with its corresponding input/kernel specifications. Within the configuration component, however, the user has the option to select between electronic and photonic modes. If neither is specified, the electronic mode is pre-assumed. As shown in the flowchart in Fig. 4, similar to the electronic mode, the user can opt for a performance optimization target (i.e., area, energy, power, or latency) in the photonic mode. Therefore, one of the optimization targets has to be selected for the program to determine a relevant and efficient accelerator architecture. The part is identical to the original input file configuration of MNSIM, where features such as bit level, minimum and maximum crossbar size, weight polarity, and a pipeline option are to be selected by the user.

The final part of the configuration component allows for flexibility in the accelerator design from the crossbar structure. Based on the user-input's performance data achieved from circuit simulators or experimental results, the tool offers the flexibility to add new extra components to the accelerator. The sub-array add-on components option in the input configuration file allows for peripheral component flexibility beyond the included components. Then, for each crossbar configuration, one set of added components is simulated to compute the accelerator's total area, energy, and latency.



Photonic Mode
- pPCM crossbar
- 45nm nodes
- pPCM area/unit: 9.07μm^2
- pPCM latency: 20ps
- pPCM energy: 50fj/bit
- 4:1 MUX area: 11.2632pm
- 4:1 MUX latency: .07ns
- 1:16 DEMUX area: 76.652pm
- 1:16 DEMUX latency: .11ns
- 1:4 DEMUX area: 15.4869pm
- 1:4 DEMUX latency: .08ns

Electronic Mode
- memristor crossbar
- 45nm nodes
- vdd: 1
- vth: .12679
- phyGateLength: .018μ
- capIdealGate: .678n
- capFringe: .17n
- capJunction: 1m
- capOx: 37.7m
- effectiveElectronMobility: 29.77m
- effectiveHoleMobility: 9.527m
- pnSizeRatio: 2.41
- effectiveResistanceMultiplier: 1.51
- currentOnNmos set
- currentOffNmos set
- currentOnPmos set
- currentOffPmos set

Input Config
Mode selection
# CNN layers
Target output
Weight Polarity
Bit Level
Min/Max Crossbar size
Not pipelined
Sub-add-on Components x
1
Area
Latency
Dynamic Power
Leakage Power
.
.
x
Area
Latency
Dynamic Power
Leakage Power

- *Accelerator Area: crossbar area + peripheral area (m2)*
- *Accelerator Latency: crossbar latency + peripheral latency (s)*
- *Accelerator Energy: crossbar energy + peripheral energy (j)*

Fig. 3: An overview of input/output files for photonic and electronic mode.

B. Adapting Algorithm to Photonic Domain

The presented simulator simulates multiple accelerator configurations to find which one produces the lowest target value (area, energy, or latency). As shown in Fig. 4, by altering the bit-level, crossbar size, and other input parameters, it simulates different accelerator options and calculates the number of crossbars required for each one along with the corresponding area, energy, and latency. Each of these performance results is

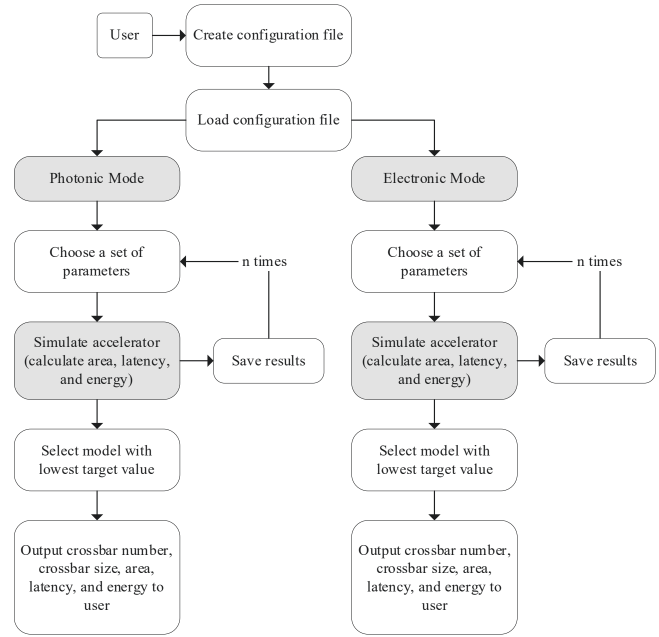

User
Create configuration file
Load configuration file
Photonic Mode
Electronic Mode
Choose a set of parameters
n times
Simulate accelerator (calculate area, latency, and energy)
Save results
Select model with lowest target value
Output crossbar number, crossbar size, area, latency, and energy to user
Choose a set of parameters
n times
Simulate accelerator (calculate area, latency, and energy)
Save results
Select model with lowest target value
Output crossbar number, crossbar size, area, latency, and energy to user

Fig. 4: Flowchart for the simulation modes in the updated MNSIM [1].

then saved, and the set of parameters that produces the lowest target value is outputted to the user.

We utilize MNSIM's original data-flow structure to adapt it to the photonic domain. We categorize the existing MNSIM algorithm as the "electronic mode" which is selectable by the user. The proposed photonic mode has a similar structure with fewer input parameters and different functions to compute area, energy, and latency. As mentioned, the essential structure for the photonic mode accelerator is modeled on top of [12], consisting of a pPCM matrix for MAC computation and multiplexers/demultiplexers for input/output generation. The performance statistics of pPCM used in the simulator are adopted from experimentally-benchmarked data and can be readily updated by the user as shown in Fig. 3.

IV. Validation Results

A. Performance Evaluation

To examine and compare the performance of a silicon photonics accelerator versus an electronic one, the enhanced MNSIM is used in this section to determine the area, energy, and latency at different crossbar sizes. For this experiment, we selected the well-known AlexNet architecture [24] and configured the input file accordingly. AlexNet was the first deep CNN successfully performing ImageNet classification task with 5 convolutional layers and 3 FC layers.

The achieved results from the simulator are plotted in log-scale in Fig. 5. Here we listed our main observations. As shown in Fig. 5(a), the photonic design shows significantly smaller energy consumption at crossbar size of 8×8 to 128×128 compared to the electronic design. Another observation is while the electronic design shows a remarkable reduction in energy consumption with an increase in crossbar size, the photonic one remains the lowest at these median sizes. We can see when the crossbar size goes beyond 256×256, the electronic accelerator consumes less energy. This stems from large size of combinational add-ons for photonic design as crossbar size grows.

TABLE I: Accuracy (%) of various acceleration techniques.

Implementation	MNIST	FashionMNIST	SVHN
Software Baseline	98.6	90.02	97.47
Electronic [17]	97.0	88.17	96.25
Photonic [11]	96.12	89.23	95.5

These trends swap in Fig. 5(b), with the silicon photonics accelerator's latency results decreasing significantly and the electronic latency results remaining stagnant. Besides the initial 8×8 crossbar size, all other models show a significant reduction in execution time with the photonic model. Therefore, our experiment confirms that a silicon photonics accelerator can be potential paradigm to outperform current electronic designs. Lastly, in Fig. 5(c), both the electronic and photonic accelerator designs decrease in the area with an increase in crossbar size, but at all crossbar sizes, the photonic area results remain smaller.

B. Inference Accuracy

We conducted experiments on several datasets, including MNIST [25], Fashion-MNIST [26], and SVHN [27] to analyze the accuracy of a photonic accelerator [11] vs. an electronic one [17]. MNIST is leveraged as a gray-scale dataset that contains 70,000 28×28 images of handwritten digits from 0 to 9, 60,000 images for training, and 10,000 images for testing sets. Similar to MNIST, Fashion-MNIST consists of 28×28 gray-scale images but includes 10,000 images for each training and testing set to form ten fashion categories. Finally, we also exploit SVHN with 73,257 training digits, 26,032 testing digits, and 531,131 additional digits for extra training data. The images are pre-processed to 20×20 from the original 32×32 cropped version and fed to the model.

We reported the inference accuracy of an in-house software baseline developed in python with the electronic and photonic designs in Table I. For each of the datasets, the inference accuracy of the electronic and photonic results remains within one percent of each other and two percent off the software baseline results. Since the accelerators are so comparable in accuracy, it leaves the performance results discussed in Section IV.A to be a tie-breaker between a photonic and electronic accelerator modeled with the enhanced MNSIM herein.

V. Conclusions and Future Works

As convolutional neural networks have grown in usage, their hardware has had to develop to improve their performance and scalability. Previous electronic-based CNN accelerators are limited due to their von-Neumann architecture and consequential high energy requirement in moving data. Silicon photonic's inherent parallel structure and light speed computations make it a promising alternative for the implementation of these accelerators. In this short study, we adapted MNSIM with a

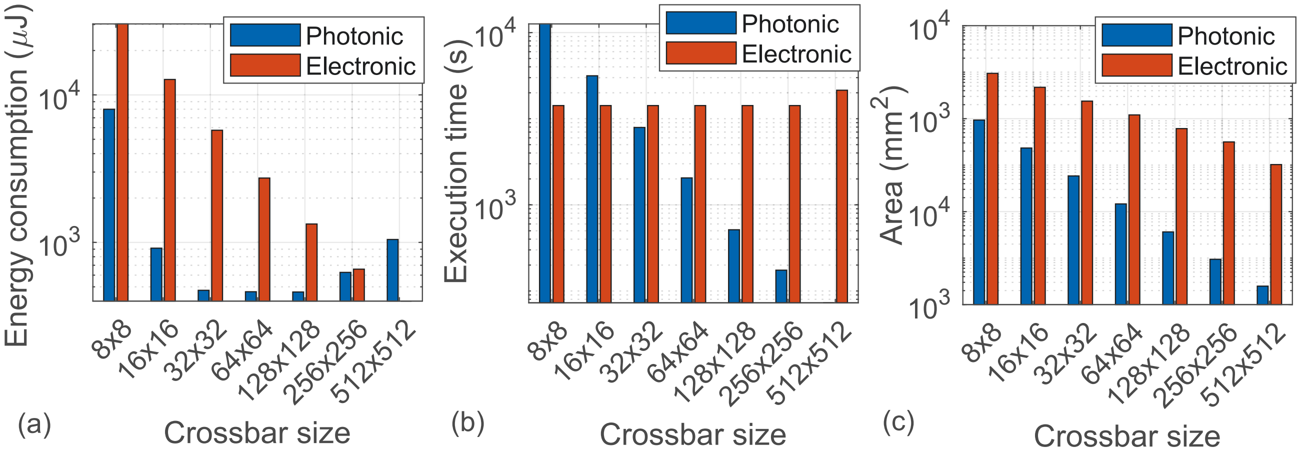


Fig. 5: Comparison of a silicon photonics accelerator design with an electronic ReRAM crossbar implemented in the enhanced MNSIM. To show the impact of the crossbar size on the performance parameters, we increase it from 8×8 to 512×512. Here, (a) Energy consumption, (b) Execution time, (c) Area.

new photonic mode that makes it to be the first freely available simulation platform for silicon-photonic crossbar accelerators. While the original electronic mode is based on a memristor-crossbar structure, its photonic mode is centered around a photonic Phase Change Memory crossbar structure. The user inputs into the network which mode they would like to use, the topology of the CNN they want to use the accelerator for, and their target output (options: area, energy, latency). Based on the selected target output, the enhanced MNSIM simulates accelerators of different crossbar sizes and outputs the model to the user that produces the lowest target result along with the model's total area, energy, and latency. Through comparing the simulated results at different crossbar sizes in the two modes, a silicon photonics design proved to be superior in reducing the physical size, execution time, and energy consumption of the accelerator model while maintaining within one percent of original performance accuracy.

In the future, we hope to add more adaptability to the tool. Currently, the sub-additional components options in the input configuration file add those components to each crossbar layer. We would like to add the option of user-entered components that will only be used in the first/last layer of the neural network. In addition, the photonic mode is currently solely based on a photonic Phase Change Memory crossbar structure. Our next step will be to allow for selecting components besides pPCM for the crossbar, such as Microring and Microdisk Resonators.

VI. Acknowledgment

This work is supported in part by the National Science Foundation under Grant No. 1852375, 2216772, and 2216773.

References

[1] L. Xia, B. Li, T. Tang, P. Gu, P.-Y. Chen, S. Yu, Y. Cao, Y. Wang, Y. Xie, and H. Yang, "Mnsim: Simulation platform for memristor-based neuromorphic computing system," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 5, pp. 1009–1022, 2017.

[2] 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.

[3] S. Angizi, Z. He, F. Parveen, and D. Fan, "Imce: Energy-efficient bit-wise in-memory convolution engine for deep neural network," in *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2018, pp. 111–116.

[4] A. Roohi, S. Angizi, D. Fan, and R. F. DeMara, "Processing-in-memory acceleration of convolutional neural networks for energy-effciency, and power-intermittency resilience," in *20th International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2019, pp. 8–13.

[5] S. Angizi, Z. He, A. S. Rakin, and D. Fan, "Cmp-pim: an energy-efficient comparator-based processing-in-memory neural network accelerator," in *Proceedings of the 55th Annual Design Automation Conference*, 2018, pp. 1–6.

[6] S. Angizi, S. Tabrizchi, and A. Roohi, "Pisa: A binary-weight processing-in-sensor accelerator for edge image processing," *arXiv preprint arXiv:2202.09035*, 2022.

[7] F. P. Sunny, E. Taheri, M. Nikdast, and S. Pasricha, "A survey on silicon photonics for deep learning," *ACM Journal of Emerging Technologies in Computing System*, vol. 17, no. 4, pp. 1–57, 2021.

[8] F. P. Sunny, A. Mirza, M. Nikdast, and S. Pasricha, "Robin: A robust optical binary neural network accelerator," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 20, no. 5s, pp. 1–24, 2021.

[9] F. Sunny, A. Mirza, M. Nikdast, and S. Pasricha, "Crosslight: A cross-layer optimized silicon photonic neural network accelerator," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 1069–1074.

[10] F. Sunny, M. Nikdast, and S. Pasricha, "Sonic: A sparse neural network inference accelerator with silicon photonics for energy-efficient deep learning," in *2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2022, pp. 214–219.

[11] W. Ma, Z. Liu, Z. A. Kudyshev, A. Boltasseva, W. Cai, and Y. Liu, "Deep learning for the design of photonic structures," *Nature Photonics*, vol. 15, no. 2, pp. 77–90, 2021.

[12] J. Feldmann, N. Youngblood, M. Karpov, H. Gehring, X. Li, M. Stappers, M. Le Gallo, X. Fu, A. Lukashchuk, A. S. Raja *et al.*, "Parallel convolutional processing using an integrated photonic tensor core," *Nature*, vol. 589, no. 7840, pp. 52–58, 2021.

[13] R. Zand, K. Y. Camsari, S. Datta, and R. F. DeMara, "Composable probabilistic inference networks using mram-based stochastic neurons," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 15, no. 2, pp. 1–22, 2019.

[14] 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," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 27–39, 2016.

[15] S. Angizi, Z. He, D. Reis, X. S. Hu, W. Tsai, S. J. Lin, and D. Fan,

"Accelerating deep neural networks in processing-in-memory platforms: Analog or digital approach?" in *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2019, pp. 197–202.

[16] M. Elbtity, A. Singh, B. Reidy, X. Guo, and R. Zand, "An in-memory analog computing co-processor for energy-efficient cnn inference on mobile devices," in *2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2021, pp. 188–193.

[17] 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," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 14–26, 2016.

[18] L. Song, X. Qian, H. Li, and Y. Chen, "Pipelayer: A pipelined reram-based accelerator for deep learning," in *2017 IEEE international symposium on high performance computer architecture (HPCA)*. IEEE, 2017, pp. 541–552.

[19] Z. He, J. Lin, R. Ewetz, J.-S. Yuan, and D. Fan, "Noise injection adaption: End-to-end reram crossbar non-ideal effect adaption for neural network mapping," in *Proceedings of the 56th Annual Design Automation Conference 2019*, 2019, pp. 1–6.

[20] R. Zand, K. Y. Camsari, S. D. Pyle, I. Ahmed, C. H. Kim, and R. F. DeMara, "Low-energy deep belief networks using intrinsic sigmoidal spintronic-based probabilistic neurons," in *Proceedings of the 2018 on Great Lakes Symposium on VLSI*, 2018, pp. 15–20.

[21] M. Hu, H. Li, Y. Chen, Q. Wu, G. S. Rose, and R. W. Linderman, "Memristor crossbar-based neuromorphic computing system: A case study," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 10, pp. 1864–1878, 2014.

[22] P.-Y. Chen, X. Peng, and S. Yu, "Neurosim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 12, pp. 3067–3080, 2018.

[23] S. Ohno, K. Toprasertpong, S. Takagi, and M. Takenaka, "Si microring resonator crossbar arrays for deep learning accelerator," *Japanese Journal of Applied Physics*, vol. 59, no. SG, p. SGGE04, 2020.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[25] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[26] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[27] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.