WiNN: Wireless Interconnect based Neural Network Accelerator

Siqin Liu*, Sushanth Karmunchi†, Soumyasanta Laha†, Savas Kaya*, Avinash Karanth*
*School of Electrical and Computer Engineering, Ohio University, Athens, Ohio 45701
†Department of Electrical and Computer Engineering, California State University, Fresno, Fresno, CA 93740
Email: *{ls847719, kaya, karanth}@ohio.edu, †{sushredredd058, laha}@mail.fresnostate.edu

***Abstract*—Deep Neural Networks (DNNs) have demonstrated promising performance in accuracy for several applications such as image processing, speech recognition, and autonomous systems and vehicles. Spatial accelerators have been proposed to achieve high parallelism with arrays of processing elements (PE) and energy efficient data movement using traditional Network-on-Chip (NoC) architectures. However, larger DNN models impose high bandwidth and low latency communication demands between PEs, which is a fundamental challenge for metallic NoC architectures. In this paper, we propose WiNN, a wireless and wired interconnected neural network accelerator that employs on-chip wireless links to provide high network bandwidth and single cycle multicast communication. We design separate wireless networks modulated with two different frequency bands one each for the weights and input. Highly *directional* antennas are implemented to avoid noise and interference. We propose *multicast-for-wireless* (MW) dataflow for our proposed accelerator that efficiently exploits the wireless channels' multicast capabilities to reduce the communication overheads. Our novel wireless transmitter integrates on-off keying (OOK) modulator with power amplifier that results in significant energy savings. Our simulation results show that WiNN achieves 74% latency reduction and 37.5% energy saving when compared to state-of-art metallic link-based accelerators, 38.1% latency reduction and 19.4% energy saving when compared to prior wireless accelerators for various neural networks (AlexNet, VGG16, and ResNet-50).**

***Index Terms*—Radio frequency, Wireless interconnect, Computing methodologies, Neural networks**

I. Introduction

Neural network algorithms, such as deep neural networks (DNNs), have demonstrated outstanding performance in accuracy surpassing humans over the past few years in performing artificial intelligence (AI) tasks, such as object detection, image recognition and classification [1], [2], [3]. However, the increase in prediction accuracy of DNNs comes at the cost of tremendous computation requirements with hundreds of layers and millions of parameters (60 million [3] to 10 billion [2]). This poses significant throughput and energy-efficiency challenges to efficiently compute and move data from memory to processing elements (PEs).

Spatial accelerators are the de facto solution to execute such highly parallel DNN workloads instead of using general purpose CPUs. As these accelerators are deployed at the edge, they are constrained by stringent power envelops and area budget. A large body of accelerators aiming at ML inference have been introduced recently to boost the computing speed and power efficiency [4]–[12]. Most of these accelerators are spatial in nature, i.e., an array of interconnected PEs are used to provide high throughput and parallelism. The on-chip dataflow between PEs and global buffers is optimized to maximize the data reuse and thereby, reduce the off-chip data movement. Reused data are either multicast or broadcast to PEs by the global buffer by customized dataflow patterns to improve energy-efficiency [13], [14].

As the number of PEs increases, the system performance may not scale accordingly due to the overhead of inter-PE and off-chip memory communication. In a spatial NN accelerator, the Network-on-Chip (NoC) plays a critical role in realizing high throughput and low latency. Most neural network accelerators operate in a pipelined fashion - a PE operation is triggered by data arrival, and the PE stalls if the next data to be processed is unavailable due to memory or network delay. Traditional interconnection system such as buses or crossbars are inefficient due to fundamental signaling or scaling limitations with increasing number of PEs [15]. Recent work has focused on energy-efficient and low latency NoC design specialized for DNN accelerators such as hierarchical mesh/buses, light weight micro switch and chubby-tree structures [10], [15], [16]. Nevertheless, the multicast energy consumption and high latency of long-distance communication of wired links limit the scalability of the accelerator.

Emerging wireless technology has the potential to provide high communication bandwidth, low access latencies, and high power efficiency [17] [18] [19]. Wireless technology offers several degrees of freedom including spatial, temporal, and frequency - all of which make it convenient to deliver high bandwidth, single-hop, distant independent on-chip communication to multiple receivers simultaneously. Few prior work have explored deploying the wireless communication for neural network accelerator. Most prior work have utilized wireless technology to broadcast or multicast weights or input activations on a single wireless channel to improve latency or energy performance [18], [19] [20]. However, none of the prior work have shown the comprehensive multi-bands wireless communication for neural network accelerators with customized dataflow tailored for wireless technology along with detailed transceiver technology design.

In this paper, we propose **WiNN**, a wireless and wired interconnected neural network accelerator that employs on-chip wireless links to provide high bandwidth and energy-efficient single cycle multicast communication of weights and

input activations. We propose *multicast-for-wireless* (MW) dataflow for WiNN that efficiently exploits the wireless channels' multicast capabilities to reduce the communication overheads. The proposed MW outperforms existing state-of-the art dataflows such as output stationary, weight stationary and row stationary when designed with wireless technology. Moreover, by exploring more than two frequency bands, we also provide the design space of partitioning and mapping MW dataflows to take advantage of additional frequency bands. The major contributions of this work are as follow:

- **Wireless Accelerator and Dataflow:** We propose a hybrid wireless and wired interconnected neural network accelerator. By employing wireless for multicasting weights and input activations, we reduce latency and improve energy-efficiency for data movement. Our customized dataflow, MW, exploits the unique wireless channels' capabilities of multicast and broadcast to improve execution latency.
- **Multi-band Wireless channels:** We use *directive* antenna for the x- and y-dimension wireless interconnects, which enables spatial division multiplexing to distribute weights and activation separately. We propose multi-band wireless channels using frequency division multiplexing that supports flexible partitioning and mapping.
- **Energy efficient transceiver:** Our novel wireless transmitter *integrates* on-off keying (OOK) modulator with power amplifier that results in significant energy saving for WiNN. A single transistor switch acts as the modulator of such an OOK transmitter. By switching the power amplifier only when '1' is observed, the average power dissipated is reduced by 50%.

II. Background

A. Deep Neural Networks (DNNs)

Deep Neural network (DNN) is an artificial neural network (ANN) with multiple middle layers between the input and output layers that can be trained to model the behavior of complex non-linear functions. Convolutional Neural Networks (CNNs) are a class of DNNs that are widely used for image processing. the computation of the convolutional layer dominates the complexity and energy consumption in the multiple layers of DNN. Convolutional layers convolve the input in the form of a raw image or an input activation map (the output of a previous convolution layer) with a filter to produce an output feature map as shown in Eq. 1:

$$\begin{aligned} O[m][x][y] =& B[m] + \sum_{k}^{C}\sum_{i}^{S}\sum_{j}^{R} I[k][Ux+i][Uy+j] \\ & \times W[m][k][i][j] \\ & 0 \leqslant m \leq M, 0 \leqslant x \leq F, 0 \leqslant y \leq E, \\ & E = (H-R+U)/U, F = (W-S+U)/U \end{aligned} \tag{1}$$

where S and R are the width and height of the filter volume; W and H are the width and height of the input map volume; F and E are the width and height of output map volume respectively.

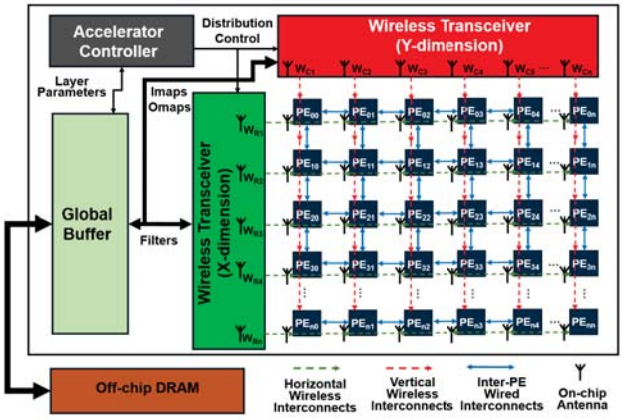

Fig. 1. An overview of proposed WiNN architecture. The two-dimensional PE array is connected by inter-PE electrical links and xy- dimension wireless interconnects.

C is the channel for both weight and input map, M is the number of filter volumes, and U is a the given stride size.

B. Dataflows and Communication Patterns

Several hardware accelerators have been proposed in the literature to efficiently implement neural network architectures over the past few years [5], [8], [11], [12]. The objective is to increase the throughput by taking advantage of the parallelism and improve the overall energy-efficiency when compared to general purpose CPUs. While MAC processing is confined to the PE array, data movement is dictated by the dataflow between the buffers and PE array. As dataflow determines the overall energy-efficiency, different dataflows such as Weight Stationary (WS), Output Stationary (OS), Row Stationary (RS) and No Local Reuse (NLR) have been proposed to minimize the data movement [21]. In WS for example, the weights remain fixed at the PE and the inputs change every cycle. This implies that accumulation of computed operations (reduction) needs inter-PE communication.

No matter which dataflow is deployed, the communication patterns cause three different traffic within the accelerator - scatter, gather and local [15]. Scatter is data distribution from the global buffer (GB) to the PE array. It involves either unicasting the weight and input map to specific PE, or multicasting to a row/column of PEs, depending on the dataflow strategy. Gather is the traffic flow by which multiple PEs send back data to the GB. It is either unicast or has many-to-one communication pattern, occurring at the end of the output computation. Local communication refers to the inter-PE communication. It could be the input map propagation or partial sum accumulation between neighbouring PEs.

III. WiNN Architecture

A. Accelerator Architecture

The proposed WiNN architecture is illustrated in Fig. 1. WiNN consists of a global buffer (GB), which connects the off-chip DRAM and the on-chip processing element (PE) array. Each PE consists of a local memory, computation

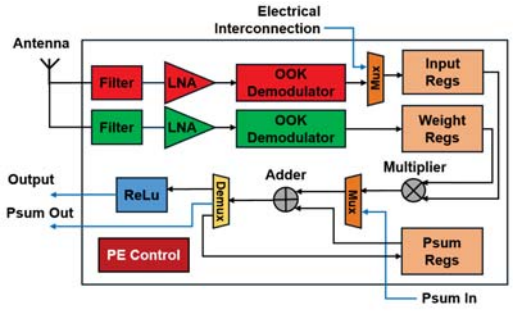

Fig. 2. Proposed PE microarchitecture. Wireless transceiver modules in red receive and demodulate input activations at 60GHz center frequency. Green transceiver modules correspond to weights that are demodulated with 70GHz carrier. The input activation register files (pink) obtain data either through wireless channels or through inter-PE electrical interconnects depending on the PE control.

unit, and wireless transceivers. GB stores weights and input feature maps that can be reused by the PE array. Wireless transceiver array (green and red boxes) assigns one transmitter and antenna for each row and column of the PE array, which comprises a X-Y dimension-order wireless network. We modulate the wireless channels of X-Y axis in different center frequencies to minimize the interference at the cross points. This ensures that weights and input maps are available to all the PEs at the same time regardless of the communication distance, thanks to frequency multiplexing and high-directivity of the antennas. Wired interconnects (blue line arrow) remain between the PEs for inter-PE data propagation to facilitate exchange of input features and output partial sums.

The fig. 2 shows the microarchitecture of the processing element (PE) where two wireless receivers obtain weights and input activations through the external antenna respectively. The two front-end filters work at different band-pass frequencies (for instance, green receives at 70GHz and red receives at 60GHz) to separately demodulate weights and input activations. Under the PE Control, the adder and multiplier fetch data from the register files and perform a multiplication and accumulation (MAC) operation per cycle. The partial sum (psum) to be accumulated is accessed either from the local psum register file or from the neighboring PE according to the specific dataflow.

B. Multicast for Wireless (MW) Dataflow

To best exploit the multicast capabilities of wireless channels, we propose a new dataflow MW, *multicast-for-wireless*. In essence, MW deploys the x- dimension wireless channels to multicast weights for each PE row and y- dimension wireless channels to multicast input activations for each PE column. Each PE is connected to adjacent PEs using wired interconnects to propagate the reused input activations. The psums are accumulated inside each PE.

To illustrate with a detailed walkthrough example, consider Fig. 3 which shows the proposed MW dataflow. Fig. 3(a) shows the convolution of a 3×3 filter on a 5×5 input map to obtain a 3×3 output map with a stride of one and no padding on a 3×3 PE array. Fig. 3(b) shows the data movement using both wireless and wired interconnects and the computation within each PE per cycle. At t_0, transmitters in the x-dimension transceiver array multicast W_{00} to all PEs (PE_{00}, PE_{01}, ... and PE_{22}) on frequency channels F_{R1}, F_{R2}, and F_{R3} respectively. Transmitters in the y-dimension multicast input maps in the convolution Sliding Window (SW) 1 to the PEs connected along each column on different frequency channels (PE_{00}, PE_{10} and PE_{20} on F_{C1}; PE_{01}, PE_{11} and PE_{21} on F_{C2} and PE_{02}, PE_{12} and PE_{22} on F_{C3} using directive antenna. As each PE only requires one input pixel from the multicast traffic, one extra cycle is spent by the PE to index the expected data according to the physical address of the PE. At cycle t_2, SW 1 shifts to SW 2. W_{01} is multicast to all PEs in all rows, similar to t_0. A new column of input maps (I_{03}, I_{13}, and I_{23}) are fetched and multicast by the wireless channel (F_{C3}) to the PEs in the column. F_{C1} and F_{C2} are set to the idle state. The rest of PEs retrieve the input map from neighboring PE through wired links (for instance, PE_{00} receives I_{01} from PE_{01}). The wireless input map distribution for each column also takes two cycles. When it reaches SW 3 and moves to SW 4 at t_6, W_{12} is multicast in one cycle. Input maps are unicast to the bottom row of PEs (PE_{20}, PE_{21}, PE_{22}), taking only one cycle because no input map index is required. Throughout the process, psums are always accumulated inside each PE until the convolution sliding window traverses the end of input maps. The pseudo code for the MW dataflow algorithm is presented in Algorithm 1.

Algorithm 1 MW algorithm on WiNN.

All weights are denoted as W[i, j], in which $0 \leq i < R$, $0 \leq j < S$
All input activations are denoted as I[p, q], in which $0 \leq p < H$, $0 \leq q < W$
All PEs have identifier PE[x, y], in which $0 \leq x < X$, $0 \leq y < Y$
for each input channel, $0 \leq c < C$ **do**
    Initialize all the PEs with W[0, 0] and I[p,q] ($0 \leq p < R$, $0 \leq q < S$) respectively
    for each weight, $0 \leq i < R$, $0 \leq j < S$ **do**
        // Weight distribution:
        for each PE, $0 \leq y < Y$, $0 \leq x < X$ **do**
            PE[x, y] = Global_Buffer[W[i,j]] // multicast through horizontal wireless channel F_x
        end for
        // Input activation distribution:
        for each PE, $0 \leq y < Y$, $0 \leq x < X$ **do**
            if $j == S - 1$ **then**
                if $x == X - 1$ **then**
                    PE[x, y] = Global_Buffer[I[$i + R - 1, j + S - 1$]] // unicast by vertical wireless channels F_y
                else
                    PE[x, y] = PE[$x + 1, y$] // vertical inter-PE propagation
                end if
            else
                if $y == Y - 1$ **then**
                    PE[x, y] = Global_Buffer[I[$i + R - 1, j + S - 1$]] // multicast by vertical wireless channels F_y
                else
                    PE[x, y] = PE[$x, y + 1$] // horizontal inter-PE propagation
                end if
            end if
            Psum[x, y] += W[i,j] × I[$i + R - 1, j + S - 1$]
        end for
    end for
end for
Clear the psum register file in PE and send the output back to GB through wired links

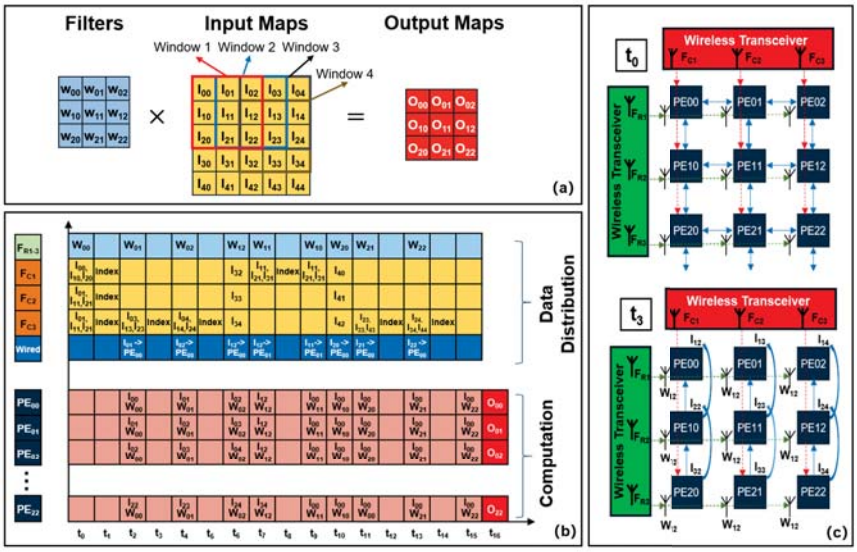

Fig. 3. A walkthrough example of WiNN architecture with multicast-for-wireless (MW) dataflow. (a) shows 3×3 filters (blue) and 5×5 input maps (yellow) are convolved to obtain 3×3 output maps (red). (b) shows the data movement of both wireless and wired interconnects and the computation cycle of PEs when (a) is implemented in WiNN. (c) shows the WiNN design with 3×3 PEs and inter-PE input map propagation at t_0 and t_3.

C. Scalability

WiNN architecture can be scaled to accommodate more PEs in different ways. One method to scale the architecture is to increase the concentration factor. By grouping 4 PEs together, we can multicast either the weight or input map to the PE cluster. These scaling approaches have been proposed for on-chip communication to reduce the router complexity [22].

WiNN architecture can be scaled also by expanding the PE array with more wireless frequency bands. Since transmitters work at the same power, integrating more PEs for x-dimension communication (weights) has no significant impact. However, for the y-dimension different input maps are sent to different columns. With multiple frequency bands, these input maps can be simultaneously sent on different frequencies. Multiple receiver circuits and antennas at different demodulation frequencies have to be integrated into the PE to achieve this multi-band design which can incur higher area cost. As shown in Fig. 3(c), three rows of PEs use 60GHz frequency and three columns of PEs use 70GHz. Since two frequency bands are used, we name the design WiNN-2. By applying multi-band channel for y-dimension, we develop WiNN-n architecture, where n represents the number of bands used by the wireless channels. In WiNN-4 for example, each y-dimension wireless channel carries 3 frequency bands, which provide dedicated channel for each PE in the column. Then, a column of input maps can be sent to the PEs in one cycle, instead of one additional cycle for indexing as depicted in Fig. 3(b). The total execution time for the example 3(a) on WiNN-4 is 10 cycles, 1.6 times faster than the baseline (WiNN-2).

D. Wireless Channel and Transceivers

In MW, the transmission along the rows and columns takes place simultaneously using two adjacent but different frequency channels. The use of the directional antennas alleviates the design of multiple transceiver in different frequency bands, thus avoiding design complexity and migration to power hungry BiCMOS or III-V technology if we were to scale up in frequency. Thanks to recent advances [23], [24], [25] especially in additive manufacturing and 3D chip integration, such highly directive antennas are easier to pursue either by in-plane dielectric engineering, structural guiding via etching, bonding or packaging elements. While certainly non trivial to build and test, use of directive antennas can ensure higher flexibility for dataflow in the proposed WiNN accelerator, as evident in the link budget analysis below (Fig. 4).

As an illustration of the concept and indicative of the potential of WiNN, the current design proposes to use 60 GHz and 70 GHz as the two frequency bands for transmission for the weights and inputs, respectively. A quarter wave monopole antenna with a very high directivity (around 5 dBi) has been considered for the MW. Such high-directivity requires either the use of metasurfaces or superstructures over the chip surface [24], [25] or loaded dielectrics [23] along with a quarter-wave monopole antenna.

Link Budget: A link budget is evaluated for the wireless communication of the transmitter data considering multiple design environments. As can be seen from Fig. 4(a) (b), the required transmit power decreases significantly with increasing antenna directivity for both distance and frequency. Fig. 4(c) shows the minimum dissipated DC power of the amplifier (Class-A) driving the antenna at the appropriate signal levels. Thus we can estimate how the overall TRx power changes with antenna directivity and frequency. The DC power is computed from the Power Added Efficiency (PAE) of 25% and the RF output power of the PA.

OOK Transceiver: The wireless communication in WiNN is

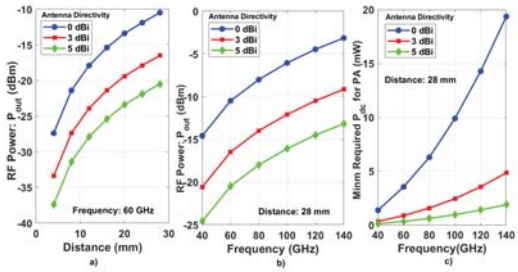

Fig. 4. The link budget analysis for 3 different antenna directivity including isotropic: (a) The RF power dissipation for different distance, (b) The RF power dissipation for different frequency, and (c) The estimated minimum DC power dissipation for different frequency.

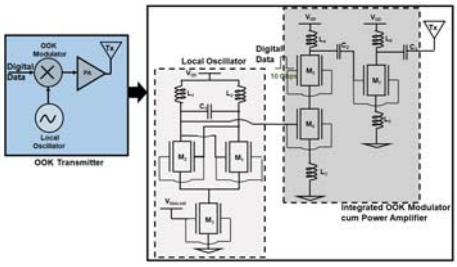

Fig. 5. The OOK transmitter circuit: Block diagram and the circuit implemented in 45nm FinFET technology. The single transistor switch, M_5, acts as the OOK modulator

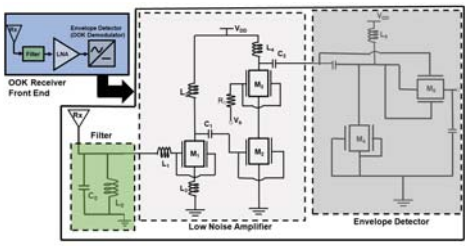

Fig. 6. The OOK receiver circuit: Block diagram and the circuit implemented in 45nm FinFET technology

achieved using the incoherent Amplitude Shift Keying (ASK) based On-Off Keying (OOK) modulation. The modulation scheme uses an integrated modulator and power amplifier in the transmitter and an envelope detector at the receiver for the demodulation as depicted in Fig. 5 and 6. The proposed CGM-FinFET transmitter consists of a differential LC oscillator and a two-stage cascade common-source power amplifier, stage one of which also acts as the OOK modulator. The data is fed into the driver transistor M4 when there is logic '1' that enables the switch M5. The single transistor switch thus acts as the modulator of such an OOK transmitter. The OOK receiver uses an energy efficient Low Noise Amplifier (LNA) and an envelope detector to demodulate the OOK modulated signal. An energy and area efficient active Dickson Rectifier [26] has been modified in the 45 nm CGM FinFET technology for the envelope detection of the OOK modulated signal.

Transceiver Performance: The OOK modulation ensures the

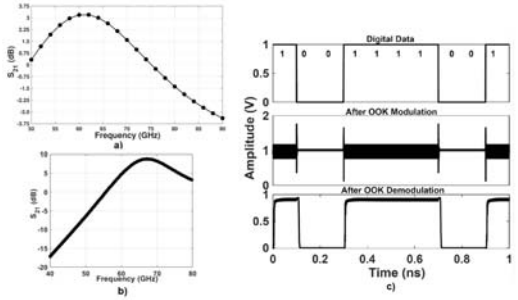

Fig. 7. (a) Gain of the PA (b) Gain of the LNA (c) OOK Modulation and Demodulation of the digital data.

power efficiency of the transceiver by avoiding the design of phase shift keying (PSK) modulators and phase locked loops (PLL). The output power of the power amplifier is -2 dBm to ensure sufficient energy to transmit to the maximum distance of 28 mm as envisaged for the WiNN architecture. Since the separation between the transmitter and the receiver are in the range of few millimeters, the gain of the amplifier is intentionally kept low at a peak gain of 3 dB to minimize the power dissipation. As depicted in Fig. 7(a), the 3-dB bandwidth is $\sim$12 GHz ranging from 55 GHz to 67 GHz at 2.25 dB satisfying the data rate requirement of the transmitter. The LNA in the receiver achieves a peak gain of 8 dB (Fig. 7(b)). The observed noise figure and the 1-dB compression point is 7 dB and -5 dBm respectively. The 3-dB bandwidth is $\sim$15 GHz ranging from 60 GHz to 75 GHz at a gain of 5 dB. The Dickson Rectifier demodulates the OOK modulated signal to retrieve the digital data at the receiver (Fig. 7(c)). Both the transmitter and receiver has been designed with the UC Berkeley CGM technology model at 45 nm [27].

IV. Performance Evaluation

A. Simulation Setup

We evaluate the proposed WiNN on three representative DNNs, i.e. AlexNet [1], VGG16 [2] and ResNet-50 [3] on CIFAR-10 dataset [28]. The propagation delay, power consumption and area overhead of each electrical component (Table I) are obtained through the RTL-level simulation by Design Compiler Ultra from Synopsys with FreePDK45 [29], a 45nm technology node design kit, released as an open-source model by NCSU. The transceiver circuit is implemented in the CGM 45 nm FinFET technology model from UC Berkeley [27]. We use a cycle accurate network simulator to obtain the latency and throughput of implementing the DNN benchmarks. A power model is further created to evaluate the energy consumption of WiNN when compared to other metallic NoC based accelerators. These include traditional mesh-based network, bus-based network, accelerators optimized NoC Microswitch [15] and hierarchical mesh as proposed in Eyeriss-v2 [10]. We also compare WiNN to two wireless interconnected DNN accelerators such as WiNoC [19] and WIENNA [20].

TABLE I
WINN AREA AND POWER BREAKDOWN WITH 256 PES AT 45 NM TECHNOLOGY NODE. WIRELESS RX AND TX ARE SIMULATED RESULTS FROM FIG 5 AND 6 WITH 60 GHZ CENTER FREQUENCY.

	Wireless Rx	Wireless Tx	PE (256x) +Mem	Global Buffer
Area (mm^2)	0.5	0.7	10.1	4.2
Power (mW)	8	25	307	147

B. Power and Area Model

Table I shows the simulation results of area overhead and power consumed by various components in WiNN architecture. The TRx area are largely contributed by inductors. We propose a power estimation model (equation 2) to fairly evaluate the energy performance of proposed WiNN architecture as well as other counterparts. In equation 2, l is index of a neural network layer, n is the index of an active PE, i is the index of a inter-PE electrical link, j is the index of a wireless channel, m is index of a wireless transceiver at global buffer, T denotes total number of cycles for one layer, P_{pe} denotes the power of PE, P_{el} denotes the power of electrical link, P_{wl} denotes the power of wireless link, and P_{wt} denotes the of power of wireless transceiver. We simulated a wired link with 8 Gbps and obtained 4.7pJ/bit energy consumption, which is used to compute the overall energy consumption of inter-PE wired interconnects in WiNN.

$$EnergyCost = \sum_{l}^{layers} (\sum_{n}^{N_{pe}} P_{pe} \times T_{pe_n} + \sum_{i}^{N_{el}} P_{el_i} \times T_{el_i} + \sum_{j}^{N_{wl}} P_{wl_l} T_{wl_j} + \sum_{t}^{T} \sum_{m}^{N_{wt}} P_{wt_t}) \quad (2)$$

C. Simulation Results

Execution Time: Fig. 8 shows the latency of WiNN on AlexNet, VGG16 and ResNet-50 with WS and MW dataflow for 256 PEs. Since RS and OS have identical number of multicast data that are transferred, we limit our comparison of MW to WS only. Micro-switch achieves up to 26% latency reduction on WS and 29% latency reduction on MW when compared to mesh network because of the compact switch architecture that achieves single-cycle unicast communication. Compared with traditional mesh and bus networks. H-Mesh reduces the latency by an average of 72% on WS and 85% on MW due to hierarchical design. WiNoC and WiNN both deploy wireless interconnects and achieve lower latency than other metallic counterparts due to the low latency, distance independent and one-cycle multicast communication. WiNN reduces the latency further by up to 14% on WS and 38% on MW compared to WiNoC as WiNoC only supports the broadcast of weights. The advantages of WiNN architecture are more pronounced on MW dataflow because WS dataflow emphasizes input features to be multicast/broadcast and inter-PE psum reduction, whereas MW dataflow emphasizes multicasting both weights and input activations.

Energy Consumption: The energy consumption of AlexNet, VGG16 and ResNet-50 with 256 PEs is shown in Fig. 9, which shows that WiNN has the highest energy efficiency among these networks. WiNN reduces the energy consumption by up to 37.5% compared to H-Mesh and 19.4% compared to WiNoC. The energy saving is essentially from the reduced latency of wireless multicast communication and the low power OOK transceiver. The weight multicast and input map propagation between neighboring PEs in MW is simpler than regular mesh topologies, which also contributes to the energy saving. The energy improvement through wireless communication can also be seen in WiNoC when compared to the metallic link based architectures. For instance, WiNoC reduces the energy consumption by 28.3% as compared to H-Mesh due to the efficient broadcasting of wireless interconnects. However, WiNoC is 19.4% less energy efficient than WiNN in ResNet-50 as WiNoC lacks support to multicast the input activations.

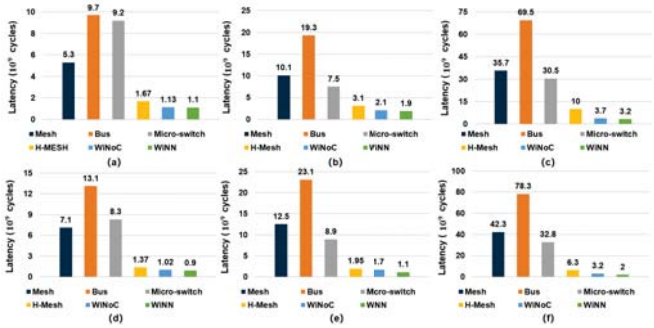

Fig. 8. Execution latency of (a) AlexNet (b) VGG16 and (c) ResNet-50 with weight stationary (WS) dataflow, (d) AlexNet (e) VGG16 and (f) ResNet-50 with multicast-for-wireless MW dataflow.

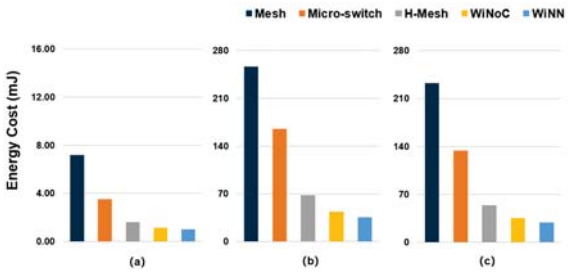

Fig. 9. Overall energy consumption of (a) AlexNet, (b) VGG16, and (c) ResNet-50 in WiNN with 256 PEs.

Scalability: We simulate multiple PE array sizes to evaluate the scalability of WiNN. Figure 10(a) shows the energy consumption of running AlexNet with 16×16, 32×32, and 64×64 PE arrays. As discussed in section 3.3, the PE array can be expanded to 32×32, as well as the number of wireless transceivers. However, we scale the PE array from 32×32 to 64×64 by increasing the concentration factor because of limited available wireless frequency bands. As shown in Figure 10, the mesh based accelerator scales the worst with 5.6 times more energy consumption when increasing from 256 PEs to 1024 PEs, while the increase is only 2.5 times for WiNN architecture. Figure 10(b)(c) show the energy breakdown of WiNN with 256 PEs and 4096 PEs, in which total energy consumption ratio of data movement (through both wired and

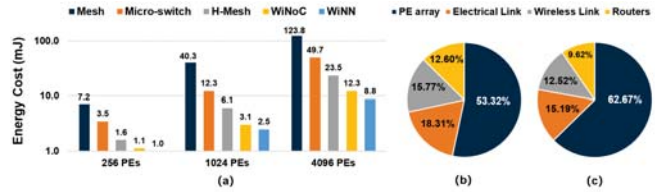

Fig. 10. Overall energy consumption of AlexNet on different PE array sizes of 256 PEs, 1024 PEs and 4096 PEs. Power breakdown for (b) 256 PEs and (c) 4096 PEs.

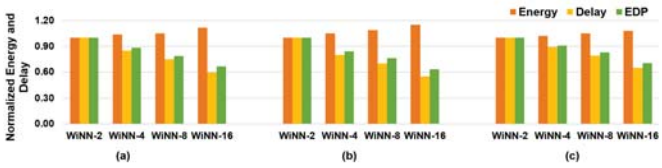

Fig. 11. Normalized energy, delay, and EDP for WiNN under different number of wireless bands configurations on (a) AlexNet, (b) VGG16 and (c) ResNet-50 network.

wireless) decreases from 34% to 28%. That is, the wired and wireless interconnection scheme in WiNN improves energy efficiency for large PE arrays.

Energy-Delay Product: Figure 11 shows the energy, delay, and energy delay product (EDP) of WiNN for AlexNet, VGG16, and ResNet-50 networks with multi-band wireless channel configurations. We normalized the results on the baseline version WiNN-2, in which two frequency bands are used for the wireless channels, one for the x- dimension, the other for the y- dimension. As shown in the figure, WiNN-4, WiNN-8, and WiNN-16 reduces the delay of running ResNet-50 by 11%, 21%, and 35% respectively when compared to WiNN-2. More frequency bands in y- dimension reduces the overall delay of the networks by alleviating input activation indexing, that is, PEs in a column receive the corresponding input activation in one cycle without requirement of discarding inputs from the multicast communication. Although incorporating multiple bands in one wireless channel increases the complexity of transceiver circuit and the overall energy consumption, multi-band channels achieves lower EDP values. For example, WiNN-4, WiNN-8, and WiNN-16 reduces the EDP of running ResNet-50 by 9%, 17%, and 30% respectively when compared to WiNN-2.

Accelerator Comparisons: A comprehensive comparison between the architectural parameters for different accelerators as well as their performance on AlexNet are given in Table II. Eyeriss V2 is a metallic interconnection based accelerator with a hierarchical mesh network optimized to configurably support multicast and unicast. When compared to Eyeriss V2, WiNN achieves 1.7× higher performance-per-watt and consumes 1.6× less energy. WiNoC and WIENNA are two accelerators with the wireless hybrid interconnection. WiNoC employs one wireless channel for low latency weight broadcast. When compared to WiNoC, WiNN achieves 1.9× higher performance-per-watt and consumes 1.1× less energy, even

TABLE II
HARDWARE PARAMETER SET UP AND PERFORMANCE COMPARISON OF WINN WITH EYERISS V2, WINOC, WIENNA.

	Eyeriss V2	WiNoC	WIENNA	WiNN
Technology	45nm	45nm	65nm	45nm
Area (mm^2)	9.5	14.2	1699	16.5
PE	256	256	16384	256
Core Frequency (MHz)	500	500	500	500
Peak Throughput GMACS	256	128	8192	128
Global Buffer (kB)	256	192	13312	192
Local SRAM (Byte)	410	128	/	128
Wireless Bandwidth (Gbps)	/	16	8-16	8
Wireless Frequency (GHz)	/	60	60	40-140
AlexNet Inference/J	625.4	884.9	/	1004.3
AlexNet TOPS/W	0.92	0.81	2.37-3.15	1.53
AlexNet Energy (mJ)	1.63	1.14	1.91-2.35	1.02

with a half of the wireless bandwidth. WIENNA relies on one wireless channel for high bandwidth interposer connection in 2.5D. When compared to WIENNA, WiNN achieves 2.3× less energy for AlexNet. Although WIENNA demonstrates at least 1.5× higher performance-per-watt, these numbers are simulated on their 64 PEs × 256 chiplets model, which cannot be directly compared to planar accelerators such as WiNN without 2.5D integration.

V. RELATED WORK

A significant amount of accelerators have been proposed recently to augment the parallelism and energy efficiency of DNN [4], [5], [6], [8], [10], [11], [12], [16]. Shidiannao [4] employed mesh-based interconnects for data distribution. Dadiannao [8] and Cambricon-X [6] relied on fat tree for balanced data transfer between the global buffer and PEs. Eyeriss [5] proposed separate buses to enhance the multicast communication but bandwidth was insufficient for DNN applications which hindered performance. Eyeriss V2 [10] addressed this challenge by proposing a hierarchical mesh network, which flexibly supported high bandwidth and data reuse. Kwon, et al. [15] analyzed traffic flows for DNN accelerators and proposed Microswitch network to achieve single-cycle communication for scatter and gather communication. Maeri [16] proposed a chubby-tree for efficient multicast and constructed the PE array with an adder tree to best exploit the interconnection for optimizing data movement.

Few prior work have applied wireless technology to DNN accelerators [19], [20]. WiNoC [19] proposed a hybrid wireless and wired interconnection for the accelerator to broadcast weights. WIENNA [20] is a wireless network of package (NoP) based 2.5D DNN accelerator, that employs wireless interconnects for high bandwidth and low latency interposer. Chiplets receive the inputs from the global buffer through wireless interconnects, while within the chiplet, PEs are interconnected by electrical links. These prior work have both relied on one wireless channel for communication at global buffer end to PE/chiplets. In WiNN, we incorporate multiple wireless bands and separate wireless channels in both x and y dimension of the accelerator to multicast both weights and input activations.

VI. Conclusions

In this paper, we proposed WiNN, a wireless and wired interconnected neural network accelerator that employs on-chip wireless links to provide high bandwidth and single cycle multicast communication. We further discussed the proposed *multicast-for-wireless* (MW) dataflow that efficiently exploits the wireless channels' multicast. We proposed a novel wireless transmitter with high energy efficiency. We ultimately evaluated the performance of WiNN as well as MW dataflow across several neural network accelerator architectures. Out simulation results show that WiNN achieves 74% latency reduction and 37.5% energy saving when compared to state-of-art metallic link-based accelerators, 38.1% latency reduction and 19.4% energy saving when compared to prior wireless accelerators for various neural networks (AlexNet, VGG16, and ResNet-50).

Acknowledgment

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

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[4] Z. Du, R. Fasthuber, T. Chen, P. Ienne, L. Li, T. Luo, X. Feng, Y. Chen, and O. Temam, "Shidiannao: Shifting vision processing closer to the sensor," in *2015 ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA)*, 2015, pp. 92–104.

[5] Y.-H. 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, 2016.

[6] S. Zhang, Z. Du, L. Zhang, H. Lan, S. Liu, L. Li, Q. Guo, T. Chen, and Y. Chen, "Cambricon-x: An accelerator for sparse neural networks," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2016, pp. 1–12.

[7] A. Parashar, M. Rhu, A. Mukkara, A. Puglielli, R. Venkatesan, B. Khailany, J. Emer, S. W. Keckler, and W. J. Dally, "Scnn: An accelerator for compressed-sparse convolutional neural networks," in *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, 2017, pp. 27–40.

[8] T. Luo, S. Liu, L. Li, Y. Wang, S. Zhang, T. Chen, Z. Xu, O. Temam, and Y. Chen, "Dadiannao: A neural network supercomputer," *IEEE Transactions on Computers*, vol. 66, no. 1, pp. 73–88, 2017.

[9] X. Zhou, Z. Du, Q. Guo, S. Liu, C. Liu, C. Wang, X. Zhou, L. Li, T. Chen, and Y. Chen, "Cambricon-s: Addressing irregularity in sparse neural networks through a cooperative software/hardware approach," in *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2018, pp. 15–28.

[10] Y.-H. Chen, T.-J. Yang, J. Emer, and V. Sze, "Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 2, pp. 292–308, 2019.

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

[12] B. Asgari, R. Hadidi, T. Krishna, H. Kim, and S. Yalamanchili, "Alrescha: A lightweight reconfigurable sparse-computation accelerator," in *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2020, pp. 249–260.

[13] Y. Chen, J. Emer, and V. Sze, "Using dataflow to optimize energy efficiency of deep neural network accelerators," *IEEE Micro*, vol. 37, no. 3, pp. 12–21, 2017.

[14] R. Guirado, H. Kwon, E. Alarcón, S. Abadal, and T. Krishna, "Understanding the impact of on-chip communication on dnn accelerator performance," in *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, 2019, pp. 85–88.

[15] H. Kwon, A. Samajdar, and T. Krishna, "Rethinking nocs for spatial neural network accelerators," in *2017 Eleventh IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*. IEEE, 2017, pp. 1–8.

[16] ——, "Maeri: Enabling flexible dataflow mapping over dnn accelerators via programmable interconnects," in *Proceedings of the International Conference on Architectural Support for Programming Languages and Operation Systems (ASPLOS)*, 2018.

[17] S. Deb, K. Chang, A. Ganguly, X. Yu, C. Teuscher, P. Pande, D. Heo, and B. Belzer, "Design of an efficient noc architecture using millimeter-wave wireless links," in *Thirteenth International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2012, pp. 165–172.

[18] B. Bahrami, M. A. J. Jamali, and S. Saeidi, "A novel hierarchical architecture for wireless network-on-chip," *Journal of Parallel and Distributed Computing*, vol. 120, pp. 307–321, 2018.

[19] M. Sinha, S. H. Gade, W. Singh, and S. Deb, "Data-flow aware cnn accelerator with hybrid wireless interconnection," in *2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*. IEEE, 2018, pp. 1–4.

[20] R. Guirado, H. Kwon, S. Abadal, E. Alarcón, and T. Krishna, "Dataflow-architecture co-design for 2.5 d dnn accelerators using wireless network-on-package," in *2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2021, pp. 806–812.

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

[22] J. Balfour and W. J. Dally, "Design tradeoffs for tiled cmp on-chip networks," in *ACM International conference on supercomputing 25th anniversary volume*, 2006, pp. 390–401.

[23] J. Wu, A. K. Kodi, S. Kaya, A. Louri, and H. Xin, "Monopoles loaded with 3-d-printed dielectrics for future wireless intrachip communications," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 12, pp. 6838–6846, 2017.

[24] Y. Song, Y. Wu, J. Yang, and K. Kang, "The design of a high gain on-chip antenna for soc application," in *2015 IEEE MTT-S International Microwave Workshop Series on Advanced Materials and Processes for RF and THz Applications (IMWS-AMP)*. IEEE, 2015, pp. 1–3.

[25] M. Alibakhshikenari, B. S. Virdee, C. H. See, R. A. Abd-Alhameed, F. Falcone, and E. Limiti, "High-gain metasurface in polyimide on-chip antenna based on crlh-tl for sub-terahertz integrated circuits," *Scientific reports*, vol. 10, no. 1, pp. 1–9, 2020.

[26] M. Awad, P. Benech, and J. Duchamp, "Design of dickson rectifier for rf energy harvesting in 28 nm fd-soi technology," in *2018 Joint International EUROSOI Workshop and International Conference on Ultimate Integration on Silicon (EUROSOI-ULIS)*. IEEE, 2018, pp. 1–4.

[27] Bsim.berkeley.edu. BSIM-CMG – BSIM Group. (2021, April 6). [Online]. Available: https://bsim.berkeley.edu/models/bsimcmg/

[28] R. C. Çalik and M. F. Demirci, "Cifar-10 image classification with convolutional neural networks for embedded systems," in *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, 2018, pp. 1–2.

[29] R. Thapa, S. Ataei, and J. E. Stine, "Wip. open-source standard cell characterization process flow on 45 nm (freepdk45), 0.18 µm, 0.25 µm, 0.35 µm and 0.5 µm," in *2017 IEEE International Conference on Microelectronic Systems Education (MSE)*, 2017, pp. 5–6.