Interconnect-Aware Area and Energy Optimization for In-MemoryAcceleration of DNNs

Gokul Krishnan, Sumit K. Mandal, Chaitali Chakrabarti, Jae-sun Seo, Umit Y. Ogras, and Yu Cao
Arizona State University

Editor's notes:
State-of-the-art in-memory computing (IMC) architectures employ an array of homogeneous tiles and severely underutilize processing elements (PEs). In this article, the authors propose an area and energy optimization methodology to generate a heterogeneous IMC architecture coupled with an optimized Network-on-Chip (NoC) for deep neural network (DNN) acceleration.

—Yiran Chen, Duke University

Deep neural networks (DNNs) achieve accuracy levels that exceed human-level perception for a variety of applications such as computer vision, natural language processing, and medical imaging. Higher accuracy comes with increased computational complexity and model size which in turn require more memory to store both the weights and activations. Due to limited on-chip memory capacity, this leads to a significant amount of communication with off-chip memory [1], whose energy is 1000× higher than the energy required to perform computations. Therefore, there is a strong need to minimize energy cost related to memory access.

Digital Object Identifier 10.1109/MDAT.2020.3001559
Date of publication: 11 June 2020; date of current version: 25 November 2020.

In-memory computing (IMC) has emerged as a promising method to address the memory access bottleneck. Both SRAM and nanoscale nonvolatile memory [e.g., resistive random access memory (ReRAM)]- based IMC hardware architectures provide a dense and parallel structure to achieve high performance and energy efficiency [2]–[4]. However, state-of-the-art IMC architectures that employ an array of homogeneous tiles (tiles having the same size) have severely underutilized crossbar arrays or processing element (PE). For example, Figure 1(a) shows that the most commonly used DNNs utilize less than 65% of PEs in an SRAM-based homogeneous IMC architecture with PE size of 256 × 256 [5]; the exceptions are the two VGG networks where input features in most layers are multiples of 256. The low utilization of PE arrays occurs due to the nonuniform distribution of weights across different layers [6]. Reduced utilization results in increased area, leading to additional on-chip interconnect and higher energy consumption.

Communication latency between different tiles also plays a crucial role in overall hardware performance. Conventional hardware architectures use

Copublished by the IEEE CEDA, IEEE CASS, IEEE SSCS, and TTTC
2168-2356/20©2020 IEEE

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI. Downloaded on May 31,2024 at 11:01:12 UTC from IEEE Xplore. Restrictions apply.

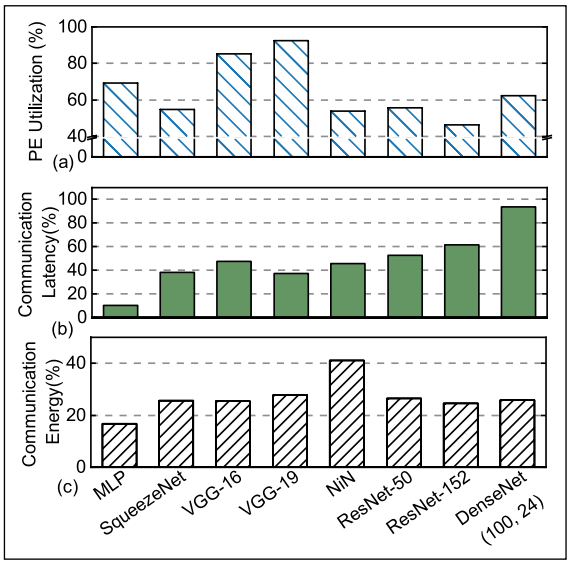

Figure 1. Experiments using NeuroSim [5] for different DNNs using homogeneous SRAM-based IMC architecture with P2P interconnect show that (a) less than 65% of the PEs are utilized, except for VGG network, (b) up to 90% of the end-to-end latency is spent on on-chip communication, and (c) communication energy constitutes 20%–40% of the total energy with NoC having one node per tile [1], [7].

H-Tree, point-to-point (P2P), and bus architectures [5] for on-chip communication, resulting in lower performance. Figure 1(b) shows that up to 90% of the total latency is spent on communication in a bus-based HTree interconnect [5]. In contrast, the network-on-chip (NoC)-based interconnect provides lower communication latency for DNN accelerators, as demonstrated in [1] and [2].

State-of-the-art hardware architectures typically implement multiple DNNs on the same NoC [1], [2]. Small DNNs, like NiN, underutilize the NoC, while large DNNs, such as DenseNet, lead to congestion. Moreover, if one NoC node is employed per tile as in [1] and [7], communication energy constitutes 20%–40% of the total energy, as shown in Figure 1(c). Here, the global interconnect energy from BookSim [8] is combined with the local interconnect energy (within tile) from NeuroSim [5]. As we show in the "Energy-aware NoC optimization" section, the initial layers account for a large portion of the total number of packets. Thus, if a tile is serviced by a dedicated router, the number of packets per router is very high for the initial layers. This increases the total communication latency and in-turn reduces energy efficiency.

In this work, we first propose an area-aware optimization technique that improves the PE array utilization. This is achieved by generating a heterogeneous tile-based IMC architecture that consists of tiles of different sizes, i.e., with different numbers of PEs where each PE is of the same size. Second, we minimize the communication energy across a large number of tiles using an NoC architecture with optimized tile-to-router mapping and scheduling. Overall, our proposed area and energy optimization methodology generates a heterogeneous IMC architecture coupled with an optimized NoC for DNN acceleration. Thorough experimental evaluations show up to 62% improvement in PE utilization, 78% reduction in area, and 78% lower energy-area product for a wide range of modern DNNs such as DenseNet (100, 24) and ResNet-152. The major contributions of this article are as follows.

- An area-aware optimization technique to maximize the PE array utilization to reduce area.
- An energy-aware NoC mapping and scheduling technique to minimize communication latency and energy.
- Experimental demonstration of the proposed methodology showing up to 78% reduction in energy-area product with respect to conventional IMC architectures.

Related work

Several SRAM [4] and ReRAM-based [2], [3], [9], [10] IMC DNN accelerators have been previously proposed, showing improvement in energy efficiency. Existing architectures assume a homogeneous tile structure, which leads to underutilization of the PE array [Figure 1(a)]. Underutilized PE arrays increase area and energy consumption due to both the crossbar and associated peripheral circuits. For example, the fifth layer of DenseNet (100, 24) of size $3 \times 3 \times 120 \times 24$ contains 25×10^3 weights to be mapped to five arrays of size 256×256. Using conventional mapping [2], only 22% of the PE array is utilized.

To address the utilization problem, a recent work proposed heterogeneous PEs (crossbars having different sizes) [10]. The PE array sizes are varied between 128×128 and 512×512 for the best utilization of crossbars across different DNNs. However, such an architecture is challenging to manufacture,

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI. Downloaded on May 31,2024 at 11:01:12 UTC from IEEE Xplore. Restrictions apply.

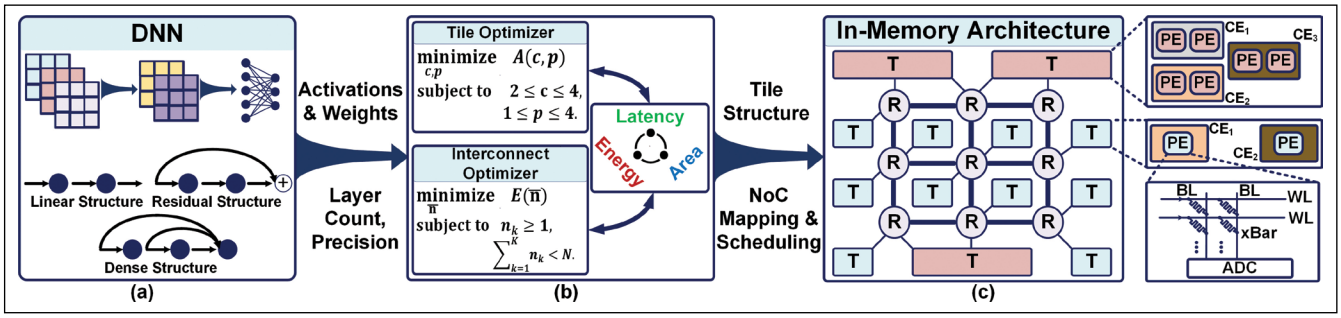

Figure 2. Overview of the proposed methodology to obtain an area- and energy-optimized IMC DNN accelerator. (a) DNNs with various connection structures, (b) the joint optimization technique, and (c) the generated heterogeneous IMC architecture with optimized interconnect. Each PE consists of the crossbar of the same size, with different number of PEs within each tile.

since different-sized PE arrays [11] require peripheral circuits to be custom designed. In this work, we propose an area optimization method to generate a heterogeneous tile architecture with different sizes of tiles, where each tile houses a different number of PEs, but each PE has the same crossbar size. This technique increases the utilization and reduces the chip area without incurring fabrication and design challenges associated with different sizes of PE arrays.

Both bus-based H-Tree interconnect [5] and NoC architectures [1], [2] have been employed for DNN accelerators. Since bus and H-Tree interconnect incur high area and latency costs, we limit the scope of this article to NoC-based DNN accelerators. The NoCs proposed in [1] and [2] assume a universal NoC architecture for all DNNs. This choice results in underutilization for small DNNs (e.g., NiN) and overutilization for large DNNs (e.g., DenseNet–100). In contrast, the proposed NoC optimization technique considers the nonuniform distribution of weights across different layers. Another traditional design choice is using one NoC router per tile which leads to higher energy consumption for DNNs with large tile counts. Therefore, we construct the NoC architecture using an energy-aware optimization technique.

Overview of the proposed methodology

Our methodology to generate an area and energy-optimized IMC architecture for a given DNN is described in Figure 2. It targets the following three objectives.

- Increased PE array utilization for overall area reduction.
- Optimized interconnect for energy-efficient on-chip communication.
- Integration of area-aware and energy-aware optimization to generate the heterogeneous IMC architecture.

The skeleton architecture consists of multiple tiles where each tile is built with several compute elements (CEs), and each CE consists of multiple PEs. Each PE is a crossbar array and all PEs are of the same size. The proposed hierarchical architecture enables efficient data transfer, reduced accumulator size, and distributed buffer structure. Figure 2(c) shows the skeleton architecture.

The methodology takes inputs such as the network structure and precision of the data for the target DNN, as illustrated in Figure 2(a). Additional inputs include a maximum number of routers and minimum/maximum number of PEs per CE, and CEs per tile. A joint optimization is performed on these inputs as shown in Figure 2(b). First, an area-aware optimization is done to improve the utilization of the PE arrays to produce the heterogeneous tile architecture. Second, an energy-aware NoC optimization is used to produce the optimal distribution of routers across different layers of DNNs on this architecture. Our optimization methodology also includes a scheduling technique to avoid congestion in the NoC, resulting in the reduction of NoC energy and end-to-end latency. The proposed method supports different IMC technologies such as SRAM, ReRAM, and phase change memory (PCM).

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI. Downloaded on May 31,2024 at 11:01:12 UTC from IEEE Xplore. Restrictions apply.

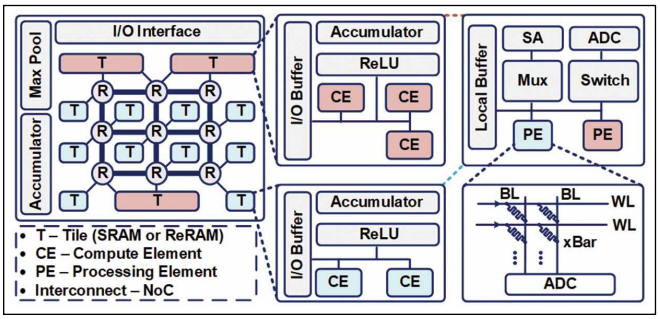

Figure 3. Hierarchical structure of the generated heterogeneous in-memory architecture with optimized NoC by utilizing the proposed area and energy optimization methodology.

Overview of generated architecture: The heterogeneous IMC architecture generated by the proposed methodology is shown in Figure 3. It consists of tiles along with nonlinear activation units, I/O buffer, and accumulators to manage data transfer efficiently. Each tile consists of a different number of CEs, buffers, and an accumulator. Each CE consists of several PE arrays, multiplexers, buffers, analog to digital converters (ADCs), and sample and hold circuits. In addition, the architecture does not use a digital to analog converter (DAC) and assumes all weights are stored on-chip similar to [2] and [3]. Such an assumption results in a large chip area for high-precision DNNs, but the chip area can be reduced by exploiting low-precision quantization and slim DNN models.

The proposed architecture employs NoC-based interconnect at the global tile level with an H-Tree interconnect at the CE level and bus interconnect at the PE level. Since the injection rate is much lower at the CE and PE levels than that at the tile level, H-Tree and bus interconnect provide ample performance. At the tile level, the proposed energy-optimized mesh NoC with X–Y routing algorithm is used. We consider mesh NoC as the interconnect topology since mesh NoC is the state-of-the-art interconnect topology both in the realm of computer architecture [12] and DNN accelerators [1], [2].

Area and energy optimization methodology

In this section, we describe our proposed methodology that performs area and energy optimizations to construct an optimized architecture for a given DNN.

Area-aware tile optimization: Mapping different DNNs to a homogeneous tile-based hardware architecture results in considerable underutilization of the PE arrays as shown in Figure 1(a). We propose an area-aware optimization to generate a heterogeneous tile-based (varying tile sizes) hardware architecture that increases PE array utilization and in-turn decreases the total area. Using the weight mapping scheme in [2], we calculate the number of rows N_k^r in

$$N_k^r = \left\lceil \frac{Kx_k \times Ky_k \times N_k^{if}}{(PE_x)_k} \right\rceil \quad (1)$$

and columns N_k^c in

$$N_k^c = \left\lceil \frac{N_k^{of} \times N_{\text{bits}}}{(PE_y)_k} \right\rceil \quad (2)$$

of PEs required for the kth layer of the DNN with K layers N_{bits}, N_k^{if}, N_k^{of}, $(PE_x)_k$, and $(PE_y)_k$ are defined in Table 1. The proposed area optimization applies to any other crossbar mapping method by changing the calculation of N_k^r and N_k^c. Next, we calculate the number of tiles T_k required for the kth layer of the DNN, as shown in the following equation:

$$T_k = \left\lceil \frac{N_k^r \times N_k^c}{c_k \times p_k} \right\rceil. \quad (3)$$

The parameters c_k and p_k are unknown and are found by the proposed optimization. To generate an optimal tile architecture, we propose an objective function that minimizes the residual area to increase PE utilization.

$$A_k(c_k, p_k) = (c_k \times p_k \times T_k - N_k^r \times N_k^c) \times T_k^2 \quad (4)$$

where the product of c_k, p_k, and T_k represents the number of PEs used to map the kth layer of the DNN. The product of N_k^r and N_k^c provides the actual number of PEs required to map the kth layer of the DNN. The difference between the two products signifies the residual area. Additionally, the total area and energy of the chip are directly proportional to the number of tiles required to realize the kth layer of

Table 1 Summary of notation.

Symbol	Definition	Symbol	Definition
T_k	Number of tiles in k^{th} layer	N_{bits}	Weight precision
Kx_k, Ky_k	Kernel size	c_k	Number of CEs in T_k
$(PE_x)_k$, $(PE_y)_k$	Size of the PE array	p_k	Number of PEs in c_k
I_k	Input activations in k^{th} layer	N_k^{if}, N_k^{of}	Number of i/p and o/p features

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI. Downloaded on May 31,2024 at 11:01:12 UTC from IEEE Xplore. Restrictions apply.

the DNN. Hence, we incorporate the area and energy cost of the hardware by multiplying the residual area by the square of T_k. Finally, we minimize the objective function with an upper and lower bound on c_k and p_k for all layers of the DNN as shown in the following equation:

$$\begin{aligned} &\underset{c_k,\ p_k}{\text{minimize}} && A_k(c_k, p_k), \qquad k = 1, \ldots, K \\ &\text{subject to} && c_{\min} \le c_k \le c_{\max} \\ & && p_{\min} \le p_k \le p_{\max}. \end{aligned} \tag{5}$$

where $c_{\min}$, $c_{\max}$, $p_{\min}$, and $p_{\max}$ are user-defined constraints that are an input to the optimization engine. By solving the problem in (5), we obtain the optimal number of CEs in a tile (c_k) and the number of PEs in each CE (p_k) for each layer of the DNN. This results in a heterogeneous-tiled IMC architecture with high PE array utilization and low area.

Energy-aware optimization for NoC: As a result of the proposed area-aware optimization, the total number of tiles in an IMC architecture can be very high. For example, DenseNet (100,24) requires 1088 tiles [5]. For such an architecture, one-to-one mapping of a router to tile [7] will require a large number of NoC routers and consume high power, as shown in Figure 4. Therefore, in our framework, we introduce an energy-aware optimization for the NoC.

Mapping tiles to routers: We first construct an objective function that represents the NoC energy consumption. Let n_k be the number of routers required for the kth layer of the DNN. The number of activations communicated between n_k and n_{k+1} routers is I_k. Hence, the number of activations between each source–destination pair is given by $I_k/(n_k \times n_{k+1})$. The total amount of communication volume can be found by adding this across all K layers and routers

$$E(\bar{\mathbf{n}}) = \left(\sum_{k=1}^{K-1} \frac{I_k}{n_k n_{k+1}}\right)\left(\sum_{k=1}^{K} n_k\right). \tag{6}$$

$E(\bar{\mathrm{n}})$ is proportional to the total communication energy of the DNN assuming that all transactions have a uniform size. We minimize this objective function with an upper bound on the total number of routers, N as

$$\begin{aligned} &\text{minimize} && E(\bar{\mathbf{n}}) \\ &\text{subject to} && n_k \ge 1, \qquad k = 1, \ldots, K \\ & && \sum_{k=1}^{K} n_k < N. \end{aligned} \tag{7}$$

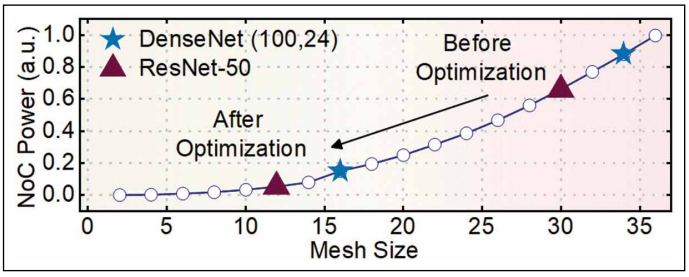

Figure 4. NoC optimization effectively reduces the power consumption because of its nonlinear dependence on the mesh size. We obtain NoC power through BookSim [8] simulations.

where the first constraint ensures that each layer of the DNN is associated with at least one router. N is a user-defined constraint (input to the optimization framework) that represents the maximum number of routers in the IMC architecture. At the end of this optimization, we obtain the number of routers needed for each layer (n_k) of the DNN.

Packet scheduling in NoC: If the activations of a layer are injected into the NoC in the order of computation, there is a high possibility of congestion resulting in high communication latency in the NoC. Therefore, we propose a scheduling technique for the NoC to schedule the activations between two layers of the DNN. The scheduling technique is applied on top of the optimal tile-to-router mapping for the NoC. This scheduling technique provides a starting time for activations from each source to destination pair in the NoC. Without loss of generality, we assume that all activations for a particular source–destination pair can be injected back-to-back.

Using the NoC topology and the routing algorithm, we first find the source–destination pairs which contend for the same link in the NoC. We model each source–destination pair (sd) as an individual task. The start time of the task corresponding to the pair sd is denoted by t_{sd} and the duration of the task equals to the number of packets for that pair (n_{sd}). Next, we put constraints on the start time of each task so that there is no contention between two transactions for the same link. The set of all tasks is denoted by T and the set of all nonoverlapping tasks is denoted by C. The following equation shows the formulation of the nonoverlap constraint, where the start time of two tasks is separated by the duration:

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI. Downloaded on May 31,2024 at 11:01:12 UTC from IEEE Xplore. Restrictions apply.

$$\begin{aligned} &\text{minimize} && t_{\text{terminal}} \\ &\text{subject to} && t_{mn} > t_{pq} + n_{pq} \vee t_{pq} > t_{mn} + n_{pq} \\ &&& \forall t_{mn,} \quad t_{pq} \in C \\ &&& t_{xy} > 0 \quad \forall t_{xy} \in T \\ &&& t_{\text{terminal}} > t_{xy} + n_{xy} \quad \forall t_{xy} \in T. \end{aligned} \tag{8}$$

Furthermore, the start time of all tasks are integers and greater than zero. We add one terminal task with the constraint that the start time of the terminal task (t_{terminal}) is greater than the start time of any of the source–destination pairs. We minimize t_{terminal} to obtain the optimal schedule for all source–destination pairs.

Experimental evaluation

Experimental setup

We evaluate the proposed methodology for a wide range of DNNs. An in-house simulator is developed to analyze the performance of the generated architecture for different DNNs. The inputs of the simulator primarily include the DNN structure, technology node, number of tiles, configuration of each tile, type of in-memory technology (ReRAM, SRAM, etc.), number of bits per cell, and frequency of operation. The circuit part and interconnect part of the simulator are calibrated with NeuroSim [5] and BookSim [8], respectively. The simulator performs the mapping of the entire DNN to a multitiled IMC architecture [2] based on the output from the area-aware optimization. The number of tiles and configuration of each tile is taken as input to perform the DNN mapping. The circuit simulator reports performance metrics, such as area, energy, and latency, of the computing logic. The interconnect performance is evaluated using the cycle-accurate NoC simulator. The circuit simulator provides the number of tiles per layer, activations, and the number of layers as output, which are taken as input by the NoC simulator. The NoC simulator computes the area, energy, and latency based on the number of routers and the computed schedules through our proposed approach. The overall performance of the architecture is calculated by combining the circuit-level and interconnect-level performance results. Finally, to evaluate the effectiveness of the proposed methodology, we compare the generated IMC architecture using 1T1R ReRAM bitcell/array [5] with state-of-the-art ReRAM-based IMC architectures.

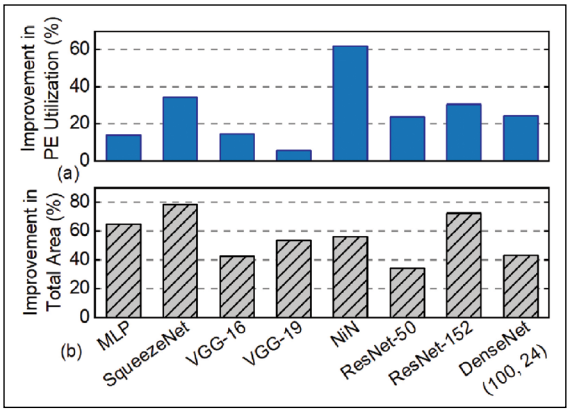

Figure 5. Improvement with respect to the baseline architecture (SRAM) in (a) PE utilization and (b) total area with the proposed area-aware optimization.

Baseline architecture: We incorporate IMC SRAM bitcell/array design [5] for the baseline architecture. We use 256 × 256 crossbar array with 32-nm technology [2]; all 256 rows are asserted together, analog multiply-and-accumulate (MAC) computation is performed along the bitline, and the analog voltage/current is digitized with a 4-bit flash ADC at the column periphery. The frequency of operation is 1 GHz. Parallel read-out is assumed for the crossbar with a 4-bit flash ADC at the column periphery [5]. We consider NoC bus width of 32. It should be noted that our proposed methodology applies to other values of these parameters.

Area-aware optimization for heterogeneous tiles

We compare both PE array utilization and the total chip area against the baseline architecture. The area-optimal tile architecture is obtained by following the methodology described in the "Area and energy optimization methodology" section. For our evaluation, we consider $c_{\min} = 2$, $c_{\max} = 4$, $p_{\min} = 1$, and $p_{\max} = 4$. Increasing the $p_{\max}$ beyond four results in very low PE array utilization. The optimal value of c_k and p_k is always less than 5, otherwise, the utilization starts reducing drastically. At the same time, the number of CEs is always more than 1 to keep the tile count reasonable, to limit the energy consumption.

Figure 5(a) shows the improvement in crossbar utilization with heterogeneous tile architecture for a range of DNNs. The improvement in utilization is the highest (62%) for NiN, and the least for VGG-19 (6%). The low improvement for VGG-19 is attributed to the baseline utilization being as high as 93%.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI. Downloaded on May 31,2024 at 11:01:12 UTC from IEEE Xplore. Restrictions apply.

This is because the number of input features in most layers are multiples of 256. The increase in PE utilization results in chip area reduction as shown in Figure 5(b). Compared to the homogeneous tile structure, we achieve a 79% reduction in area for SqueezeNet and 57% for NiN. For VGG-19, a higher area improvement is observed due to the reduction in both the number of PE arrays and associated peripheral circuits.

To better understand the efficacy of the proposed method, we analyze the layer-by-layer improvement in utilization for NiN in Figure 6(a). Two configurations of tile structures are obtained: 1) $c_k = 2$, $p_k = 1$ and 2) $c_k = 3$, $p_k = 2$. We note that, even with a high degree of freedom (15 possible combinations) for the optimization, only two configurations are chosen across the range of DNNs evaluated in this work.

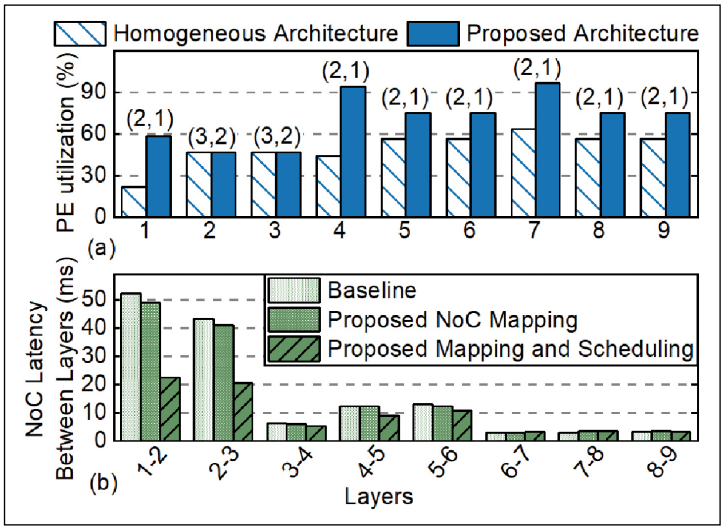

Figure 6. Layerwise improvement for NiN in (a) PE utilization for each layer with SRAM-based heterogeneous tile architecture. The tile structure for each layer (c_k, p_k) is shown on top of each bar and (b) communication latency for each layer with the proposed NoC optimization.

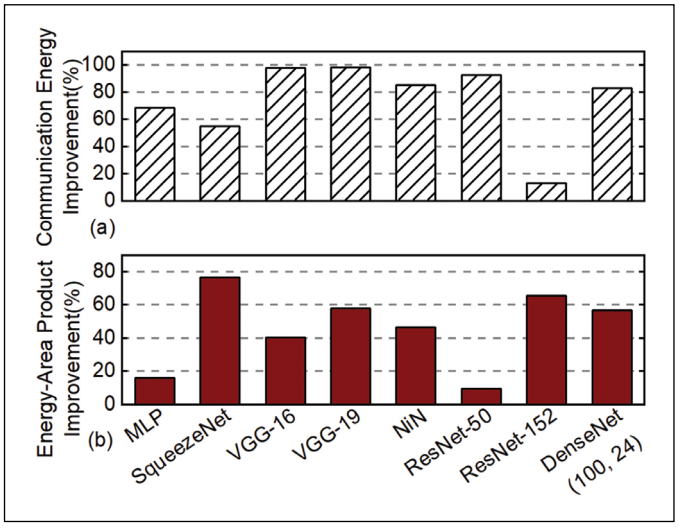

Figure 7. Improvement in (a) communication energy of the proposed energy-aware NoC optimization with respect to the baseline (SRAM) and (b) energy-area product of the generated SRAM-based architecture with respect to the baseline (SRAM).

Energy-aware NoC optimization

The proposed methodology includes an energy-aware tile-to-router mapping and scheduling technique for the NoC. The upper bound on the number of routers is set as three times the number of DNN layers to balance energy and performance. Figure 6(b) shows the improvement in latency for each layer of NiN due to the proposed NoC optimization. The proposed NoC mapping reduces the communication latency between layers 1 and 2 from 51 to 47 ms. As we integrate the NoC mapping with the scheduling technique, latency reduces further to 22 ms. The first three layers of NiN contain more than 50% of the total number of activations. Therefore, the proposed NoC mapping reserves more routers for the first three layers, resulting in a significant reduction in latency for those layers. Additionally, the total number of routers is reduced which reduces the NoC area. A direct consequence of both latency and area reduction is lower communication energy, as shown in Figure 7(a) with an average reduction of 74%. The energy reduction is the highest for the case of VGG networks—97%/98% for VGG-16/VGG-19. For ResNet-152, energy reduction is the lowest (15%), since the tiles are well distributed across layers for the baseline architecture, leaving less room for improvement.

Overall improvement

We compare the energy-area product of the generated architecture against the baseline to assess the overall improvement. The proposed approach achieves up to 78% improvement in energy-area product, as shown in Figure 7(b). This improvement is a direct consequence of the heterogeneous tile architecture (with different sizes of tiles) and optimized NoC. The improvement for ResNet-50 is 10% since the initial mapping result is already area- and energy-efficient.

Comparison with state-of-the-art architectures

Table 2 compares the ReRAM-based architecture generated by the proposed methodology with

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI. Downloaded on May 31,2024 at 11:01:12 UTC from IEEE Xplore. Restrictions apply.

Table 2. Inference performance results for VGG-19. *Reported in [9].

	Latency (ms)	Power/frame (W/frame)	FPS	EDAP (mm^2.ms.J)
Proposed Approach-ReRAM	2.69	4.2	372	0.208
AtomLayer [9]	6.92	4.8	145	1.58
PipeLayer [3]	2.6*	168.6	385	94.17
ISAAC [2]	8.0*	65.8	125	359.64

state-of-the-art works using the VGG-19 network as a representative example. The architectures have the following assumptions: crossbar of size 128 × 128, 2 bits/cell, 32-nm technology node, NoC flit width of 32 bits, and 16-bit precision for activations and weights. The generated ReRAM-based architecture achieves 2.56× improvement in frames per second (FPS) and 7.6× improvement in the energy-delay-area product (EDAP) than those of AtomLayer [9]. It consumes 40× lower power per frame along with 452× improvement in EDAP than PipeLayer [3] for comparable throughput. Moreover, there is a 3× improvement in inference latency compared to ISAAC [2]. The gain in performance is attributed to the high utilization of the crossbar arrays and the efficient tile-to-router mapping and scheduling for the NoC. Overall, the proposed area and energy optimization methodology generates a heterogeneous IMC architecture (ReRAM) with an optimized NoC that has lower EDAP and power-per-frame than prior works for better or comparable throughput.

THIS WORK PRESENTS an area and energy optimization methodology to generate a heterogeneous IMC architecture with optimized NoC for a given DNN. Unlike conventional DNN architectures that use homogeneous tiles, the architecture derived using the proposed area-aware optimization technique results in a heterogeneous architecture, where the tiles are of different sizes. The high energy efficiency is achieved through the proposed energy-aware optimization of the NoC architecture along with an associated scheduling technique. We show the efficacy of our proposed methodology for a wide range of DNN models from multilayer perceptron (MLP) to DenseNet and depths up to 152 layers (ResNet-152). We observe that the proposed methodology has an execution overhead of 12 s for small DNNs to 150 s for large DNNs. This shows that the proposed methodology is scalable with the size and depth of DNNs. The proposed methodology also supports emerging technologies (such as SRAM, ReRAM, and PCM) and applies to any mapping of weights on the crossbar. An evaluation of the architecture generated by the proposed methodology shows that our architecture achieves up to 7× improvement in the EDAP compared to state-of-the-art DNN accelerators. For future work, we will build upon the optimization techniques proposed in this work to develop a runtime reconfigurable architecture that utilizes the reuse of IMC crossbar arrays. ■

Acknowledgments

This work was supported in part by the Center for Brain-Inspired Computing (C-BRIC), one of the six centers in JUMP; in part by the Semiconductor Research Corporation program sponsored by the Defense Advanced Research Projects Agency (DARPA), National Science Foundation (NSF) CAREER under Award CNS-1651624; and in part by the Semiconductor Research Corporation under Grant 2938.001. Gokul Krishnan and Sumit K. Mandal contributed equally to this work.

References

[1] Y.-H. Chen et al., "Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 2, pp. 292–308, Jun. 2019.

[2] A. Shafiee et al., "ISAAC: A convolutional neural network accelerator with *in-situ* analog arithmetic in crossbars," in *Proc. ACM/IEEE ISCA*, 2016, pp. 14–26.

[3] L. Song et al., "PipeLayer: A pipelined ReRAM-based accelerator for deep learning," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2017, pp. 541–552.

[4] H. Valavi et al., "A 64-tile 2.4-Mb in-memory-computing CNN accelerator employing charge-domain compute," *IEEE J. Solid-State Circuits*, vol. 54, no. 6, pp. 1789–1799, Jun. 2019.

[5] P.-Y. Chen, X. Peng, and S. Yu, "NeuroSim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 12, pp. 3067–3080, Dec. 2018.

[6] Y. Ma et al., "Optimizing the convolution operation to accelerate deep neural networks on FPGA," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 7, pp. 1354–1367, Jul. 2018.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI. Downloaded on May 31,2024 at 11:01:12 UTC from IEEE Xplore. Restrictions apply.

[7] H. Kwon, A. Samajdar, and T. Krishna, "Rethinking NoCs for spatial neural network accelerators," in *Proc. 11th IEEE/ACM Int. Symp. Netw.-Chip*, Oct. 2017, pp. 1–8.

[8] N. Jiang et al., "A detailed and flexible cycle-accurate network-on-chip simulator," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw. (ISPASS)*, Apr. 2013, pp. 86–96.

[9] X. Qiao et al., "Atomlayer: A universal reRAM-based CNN accelerator with atomic layer computation," in *Proc. IEEE/ACM DAC*, Jun. 2018, pp. 1–6.

[10] Z. Zhu et al., "Mixed size crossbar based RRAM CNN accelerator with overlapped mapping method," in *Proc. IEEE/ACM ICCAD*, Nov. 2018, pp. 1–8.

[11] A. Lottarini et al., "Master of none acceleration: A comparison of accelerator architectures for analytical query processing," in *Proc. ACM/IEEE ISCA*, Jun. 2019, pp. 762–773.

[12] J. Jeffers, J. Reinders, and A. Sodani, *Intel Xeon Phi Processor High Performance Programming: Knights Landing Edition*. San Mateo, CA, USA: Morgan Kaufmann, 2016.

Gokul Krishnan is currently pursuing PhD in electrical engineering with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ. Krishnan has a BTech in electronics and communication engineering from Govt. Model Engineering College, Kochi, India (2016). He is a Student Member of the IEEE.

Sumit K. Mandal is currently pursuing PhD in electrical engineering with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ. Mandal has a BTech and an MTech in electronics and communication from the Kharagpur, Indian Institute of Technology, Kharagpur, Kharagpur, India (2015). He is a Student Member of the IEEE and ACM.

Chaitali Chakrabarti is a Professor with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ. Her research interests include VLSI algorithm-architecture co-design of signal processing and communication systems and all aspects of low-power embedded systems design. Chakrabarti has a PhD in electrical engineering from the University of Maryland, College Park, MD (1990). She is a Fellow of the IEEE.

Jae-sun Seo is an Assistant Professor with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ. His research interests include energyefficient hardware design for deep learning and neuromorphic computing. Seo has a PhD from the University of Michigan, Ann Arbor, MI (2010). He is a Senior Member of the IEEE.

Umit Y. Ogras is currently an Associate Professor with the School of Electrical, Computer and Energy Engineering, Arizona State University. He worked as a Research Scientist at the Strategic CAD Laboratories, Intel Corporation, Hillsboro, OR, from 2008 to 2013. Ogras has a PhD in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA (2007).

Yu Cao is a Professor of electrical engineering with Arizona State University, Tempe, AZ. His research interests include neural-inspired computing, hardware design for on-chip learning, and reliable integration of nanoelectronics. Cao has a PhD in electrical engineering from the University of California at Berkeley, Berkeley, CA (2002). He is a Fellow of the IEEE.

■ Direct questions and comments about this article to Gokul Krishnan and Sumit K. Mandal, School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA; gkrish19@asu.edu and skmandal @asu.edu.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI. Downloaded on May 31,2024 at 11:01:12 UTC from IEEE Xplore. Restrictions apply.