2023 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED) | 979-8-3503-1175-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISLPED58423.2023.10244186

AR-PIM: An Adaptive-Range Processing-in-Memory Architecture

Teyuh Chou[1], Fernando Garcia-Redondo[2,3], Paul Whatmough[2,4], and Zhengya Zhang[1]

[1]University of Michigan, Ann Arbor, [2]Arm Research, [3]Now with IMEC, [4]Now with Qualcomm AI Research

***Abstract*—The crossbar-based processing-in-memory (PIM) architecture has garnered considerable attention for its potential in achieving high energy efficiency for deep neural networks (DNNs). The PIM hardware's accuracy depends heavily on the design and resolution of the analog-to-digital converters (ADCs). Regrettably, high-resolution ADCs tend to be costly and often dominate the overall energy and area of the PIM designs. We propose adaptive-range PIM (AR-PIM) architecture that enables the use of lower-resolution ADCs without sacrificing accuracy. This is achieved by leveraging sparsity in the weights and input activations and dynamically adjusting the number of input activations and distributing MAC operations across multiple cycles during runtime. We perform our evaluations using a commercial 7nm FinFET PDK and show that AR-PIM offers an appealing trade-off, delivering 1.7× higher energy efficiency and 4.3× better area benefits without losing accuracy. The latency overhead is modest, only 10% over a baseline PIM architecture.**

***Index Terms*—Processing in memory, deep neural network.**

I. Introduction

Deep neural network (DNN) has enjoyed exceptional successes in many applications and it has become one primary workload for modern computing hardware. Many works have demonstrated substantial acceleration for DNN workloads using a large amount of compute resources and power by GPUs, CPUs, FPGAs, and digital ASICs [1]–[4]. In mobile and IoT devices, the energy budget is severely limited, which requires hardware solutions of higher energy efficiency [5]–[7] to enable DNN processing on these devices.

Crossbar-based processing in memory (PIM) architecture, also known as compute in memory, is a promising candidate for DNN inference computation to improve performance and energy efficiency. In essence, PIM removes the memory wall by eliminating data movement between memory units and processing units and by performing multiply-accumulate (MAC) operations at the cross-point locations. However, PIM architecture requires complex analog circuits that must be carefully designed or else they can dominate the area and energy of the entire design [8]. These include digital-to-analog converters (DACs) for the digital inputs and analog-to-digital converters (ADCs) for digitizing the outputs between layers. The complexity of ADC and DAC grow exponentially with their resolution. Lowering the resolution of ADC and DAC is desirable for area- and energy-efficient solutions. However, an insufficient resolution generally results in accuracy degradation. Without further network engineering, the accuracy can drop dramatically for sub-8b ADC resolution.

979-8-3503-1175-4/23/$31.00 ©2023 IEEE

SRAM and NVM have been used as PIM memory devices. Nonvolatile memory (NVM) devices such as resistive RAM (RRAM), magnetoresistive RAM (MRAM), phase-change RAM (PCRAM), ferroelectric RAM (FeRAM), are suitable for PIM thanks to their nonvolatility, low standby power, and high density. However, the commercially available NVM devices are still at 22nm [9]–[12], lagging the scaling of logic devices. The process, voltage, temperature (PVT) variations of the NVM devices also require a diligent control. Although both SRAM and NVM suffer from variability, SRAM has no drift issues, infinite endurance, and always leads technology scaling. Comparing a 7nm SRAM to a 22nm RRAM or MRAM, a 7nm SRAM is a better candidate for PIM because of its energy efficiency if nonvolatility is not of concern.

In this work, we investigate PIM based on a 7nm SRAM and explore practical analog PIM design choices. The investigation points to a promising direction of utilizing data sparsity in an adaptive design to relax the high-resolution ADC requirements without accuracy loss. The contributions of this work are summarized below:

1) We provide an analysis of the practical design choices for 7nm SRAM PIM, accounting for the noise and non-idealities derived from the intrinsic nature of the SRAM cell and analog accumulation in the bitline.
2) We present runtime range detection and adaptive-range PIM (AR-PIM) to achieve high accuracy with minimal latency overhead even using a low-resolution ADC.
3) We benchmark AR-PIM against a baseline on multiple DNN workloads using MNIST dataset and ImageNet dataset, showing the energy efficiency and area improvement at minimal latency increment.

II. Preliminary and Related Work

PIM architectures can reduce data movement by adopting a weight-stationary approach. The weights are stored in a memory array and the input activations are passed to the array to perform computation. The weights are stored in a bit-parallel way across columns as in [13]–[15]. In computation, a vector of inputs is driven, one per wordline (WL). The cells along a column are turned on, and the currents are summed on the bitline (BL), accomplishing the dot product between the input vector and the weight vector stored on the column of the

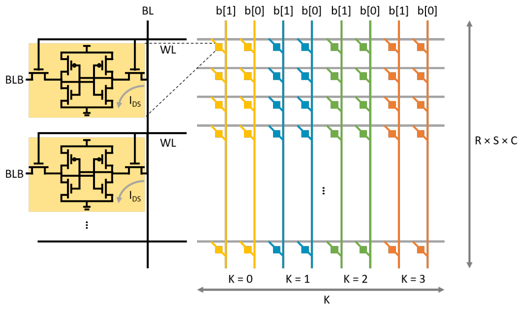


Fig. 1: The PIM mapping of the convolution operation where $R \times S$ is the kernel size, C is the input channel and K is the output channel. The 2b weight in the example is stored in two columns. The zoom-in view shows the 6T SRAM and current I_{DS} discharged by each bitcell. Both baseline PIM and AR-PIM adopt this mapping for convolution operation.

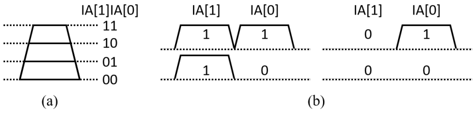


Fig. 2: The multi-bit (2b in the example) input representation of (a) WL pulse amplitude and (b) WL pulse train.

memory array. Across the columns, dot products are conducted in parallel, realizing vector-matrix multiplication (VMM).

An SRAM bitcell stores a value and its complement. When its WL turns on, the stored value drives BL and the complement drives BLB. In SRAM-based PIM, either BL or BLB can be taken as the output (Fig. 1). In this work, BL is taken as the output. The key design parameters are considered below.

Array Size. Using a larger array, more cells are activated in parallel, achieving higher performance; and the row and column peripheral circuits are amortized more effectively, leading to a higher compute density. A drawback of a larger array is the lower utilization in mapping smaller VMMs, leaving unused cells. A larger array also presents higher capacitive loading on WL and BL, resulting in a longer delay. Finally, a larger array implies the potential activation of more cells contributing current to the same BL, and thus the accumulation of more noise that may degrade the signal-to-noise ratio (SNR).

Input Encoding. The input activations can be encoded in two forms as shown in Fig. 2: (a) pulse amplitude or (b) pulse train, i.e., each bit of the multi-bit input is represented by a 1b pulse. The pulse train is more linear compared to pulse amplitude or width, and it can be better controlled [16], but the latency increases with the bitwidth. Past designs have combined pulse amplitude and pulse train [8], [17], [18].

BL Resolution. The BL resolution depends on the WL resolution (b_{WL}), the memory cell resolution, and the maximum allowable activated memory cells (N_{cells}) in a column. Since SRAM is a digital (1b) memory, the BL resolution is $b_{BL} = b_{WL} + \log_2 N_{cells}$. A higher BL resolution requires a higher-resolution ADC, which in turn significantly impacts power and area [8], [19]. A higher BL resolution also exacerbates PIM's variation and reduces the capability of error tolerance

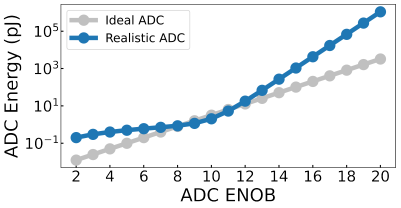


Fig. 3: ADC energy consumption with effective number of bits (ENOB) from [19].

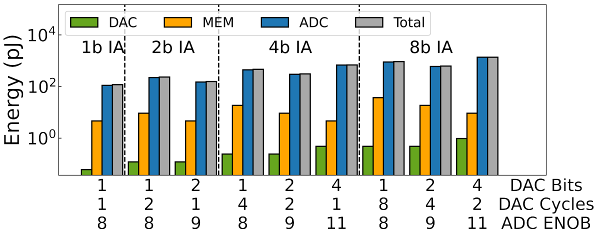


Fig. 4: Energy efficiency of 1b/2b/4b/8b input activations. The input encoding for each configuration is implemented by DAC bits and DAC cycles with the corresponding ADC resolution.

[20]. Recently, an all-digital PIM design [21] is proposed, the power and area of peripheral circuits increase with the BL resolution.

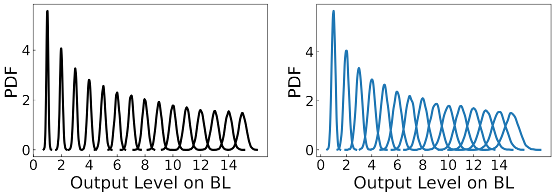
ADC Sharing. The ADC area can be significantly larger compared to the SRAM bitcell pitch. Placing one ADC per BL is difficult due to the physical layout constraint. Since the conversion time of ADC can be much shorter than the time it takes for the SRAM BL current to develop, sharing an ADC between BLs becomes a necessity, e.g., 1 ADC is shared by 4 BLs in [16]. ADC sharing requires extra circuits such as sample-and-hold circuit [8] or weighted capacitors [16] to store the BL value before the conversion starts.

III. PIM Design Considerations and Challenges

The following evaluations are based on the SPICE simulation of a 128×128 SRAM array in 7nm FinFET technology. The ADC energy is extracted from [19] and shown in Fig. 3. DACs are adopted from [22] and most circuit component models are adopted from [8]. The array size of 128 is chosen to obtain high utilization for DNN workloads as in [23].

A. Input Encoding

The energy with various input encoding choices is investigated. If the input activations are 1b, they can be encoded in 1b WL pulses. Since a BL is connected to 128 bitcells, the BL resolution is $b_{BL} = 8$. An 8b ADC consumes 96% of the total energy (Fig. 4), significantly higher than the energy of the memory access or the 1b DAC. For a 2b input activation, two input encoding options are available: 1b pulses over 2 cycles (with partial sums scaled and added digitally post-ADC) or a single-cycle 2b pulse, resulting in 8b and 9b BL resolution, respectively. A 9b ADC consumes 35% more energy than an 8b ADC (Fig. 3). Hence, two 8b analog-to-digital (A/D) conversions with DAC bits = 1 and DAC cycles = 2 result in 49% higher energy than one 9b A/D conversion with DAC



(a) (b)

Fig. 5: SRAM BL current levels from the SPICE simulation for WL voltage of (a) 0.8V and (b) 0.6V. Each Gaussian distribution represents one output level. The less overlap between two distributions means the better sensing margin for distinguishing two output levels.

bits = 2 and DAC cycles = 1, so the single-cycle 2b pulse encoding is preferable in terms of energy (Fig. 4).

Moving to 4b and 8b input activations, more input encoding options are available. The 2b-pulse encoding was found to be the sweet spot in energy consumption. However, regardless of the input encoding choice, the ADC dominates the energy consumption. A WL resolution of higher than 2b is not practical due to the significant escalation of ADC energy.

B. BL Current Levels under PVT Variations

If a bitcell storing a 1 is activated, the bitcell discharges one unit of current from BL. However, process variations complicate the picture. As more discharging bitcells on the same BL are activated, the distribution of current gets wider as in Fig. 5(a). The wide distribution makes it challenging to decode as few as 16 current levels. This insight suggests that the degradation of SNR due to process variations may make using a very high-resolution ADC inconsequential.

As the WL voltage level is reduced, e.g., in supporting WL pulse-amplitude input encoding, the current level boundaries are further obscured as seen in Fig. 5(b).

C. ADC Resolution and ADC Sharing

For this investigation, a reference SRAM-based PIM design is adopted: a 128×128 SRAM array; the input is provided bit serially; WL is encoded in 1b pulses. For example, an 8b input activation is passed to WL by a 1b DAC in 8 cycles. Therefore, the full resolution required at each BL is 8b. An 8b weight is stored in 8 SRAM bitcells in a row. A weight-stationary digital design was synthesized using an array of multiply-accumulates (MACs) with weight storage to mimic PIM. The partial sums are accumulated along a column of MACs. The digital design also follows the same bitwidth as the PIM design for comparison at the MAC level.

The energy efficiency of the PIM design is compared to the digital SIMD architecture based on [24] (with matched BL bitwidth to the PIM design for a fair comparison) in Fig. 6, assuming one ADC per BL. PIM achieves the best energy when the input activation and weight bitwidth are low. Also, note that PIM with an ADC that supports the full BL resolution ($R_{Full} = b_{BL}$) fares worse than the digital design. For the energy of PIM to be competitive, the ADC resolution needs to be reduced to 3b or 4b below the full BL resolution.

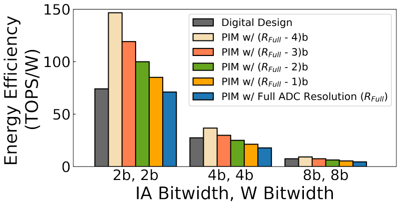
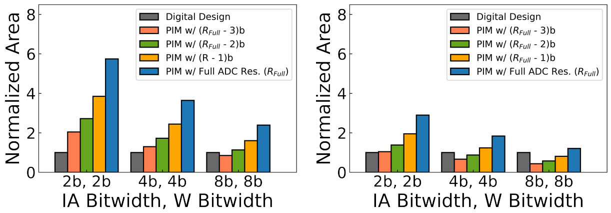


Fig. 6: The energy efficiency comparison of synthesized digital design and PIM designs with various ADC resolution settings and 3 Input Activation (IA) and Weight (W) bitwidth combinations.



(a) (b)

Fig. 7: The area comparison for synthesized digital design with the SIMD architecture and PIM designs with various ADC resolution settings for (a) 1 ADC per BL and (b) 1 ADC shared by 2 BLs.

The area of PIM is compared to the synthesized digital design in Fig. 7 for one ADC per BL. The ADC area is based on existing IPs and extrapolations. Due to the relatively large area of ADC, especially a high-resolution ADC, the area of PIM easily exceeds the digital design by up to 4.3×. Even with the short bitwidth of 2b input activation and 2b weight, the full BL resolution still requires the use of relatively high-resolution ADCs that consume a large area. When the ADC resolution is reduced to 3b below the full BL resolution ($R_{Full} - 3$), the area becomes comparable. Fig. 7(b) shows the configurations of one ADC shared by 2 BLs to reduce the PIM area. The results highlight the importance of reducing ADC resolution and increasing ADC sharing to keep the PIM area competitive.

D. Necessity to Control Resolution

The above sections highlight the challenges behind a high BL resolution and its feasibility due to the sensing margin. Reducing the ADC resolution is a must to make PIM more competitive in energy efficiency and area.

To control the BL resolution, WL resolution can be reduced by employing 1b-pulse encoding over multiple cycles and activating only a subset of rows at a time such as in [25]. However, these approaches requires more cycles to complete the computation. To address this operation limitation we proposed AR-PIM, described in Section IV.

IV. Adaptive Range-PIM (AR-PIM) Architecture

AR-PIM leverages data sparsity by controlling the BL range at runtime. This approach can prevent sacrificing the inference accuracy incurred by direct truncation on BL values with the reduced ADC resolution. For a bitcell to contribute to the BL current and increase the BL range, the bitcell needs to store a 1 and it needs to be activated. This implies that both the weight value and the input activation value are 1 (at the bit

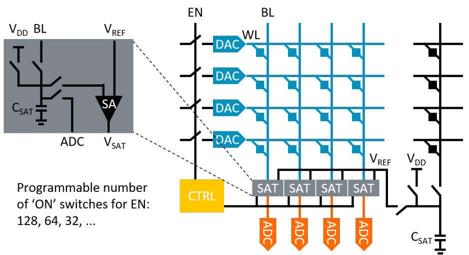


Fig. 8: AR-PIM architecture with BL saturation detection circuitry (SAT). The light-weight SAs and control with programmable activated number of WLs enable runtime range detection.

TABLE I: Array utilization, mean of BL value, and standard deviation of BL value. Note that the maximum of BL values is 128 for a 128×128 array.

Model	LeNet		AlexNet				
Dataset	MNIST		ImageNet				
Layer	CONV1	CONV2	CONV1	CONV2	CONV3	CONV4	CONV5
Array Utilization	2.64%	42.2%	94.5%	96.2%	96.4%	100%	100%
BL Value Mean	0.383	4.630	27.957	14.478	6.932	2.945	2.764
BL Value Std	1.038	4.277	17.429	12.804	6.123	2.977	2.999

Model	VGG11							
Dataset	ImageNet							
Layer	CONV1	CONV2	CONV3	CONV4	CONV5	CONV6	CONV7	CONV8
Array Utilization	21.1%	90%	100%	100%	100%	100%	100%	100%
BL Value Mean	6.420	9.030	9.906	5.907	6.798	3.905	3.463	2.925
BL Value Std	4.250	9.604	8.650	5.468	5.710	3.750	3.420	3.110

level). If either the weight value or the input activation value is 0, the bitcell does not contribute to the BL current or the BL range. Therefore, the ADC resolution quoted in the previous sections is the maximum resolution, while the effective BL range can be lower with the bit-level sparsity.

The presence of zeros in weights and input activations is referred to as sparsity. Sparsity exists even in unpruned models, especially with the rectified linear unit (ReLU) activation function that generates zero activations. Sparsity can be further increased using pruning algorithms to remove weights. In addition to word-level sparsity, plenty of bit-level sparsity exists in weights and input activations as identified by [26].

In DNN inference, a model is given and the weight sparsity is static. However, the activation sparsity is dynamic, namely input-dependent, and determined in runtime. As a result, when the computation of DNN inference is mapped to PIM, the BL range can vary due to the dynamic activation sparsity. We propose a technique to detect the runtime sparsity (or density) for the computation of DNN inference. If the density is low, an energy-inexpensive low-resolution ADC can be used; and if the density is high, the BL range can be adjusted by activating only a portion of the bitcells.

A. Runtime Range Detection

The runtime range detection can be implemented by reusing the SRAM sense amplifier (SA) in the readout circuitry and a reference column as shown in Fig. 8. The reference column stores a preset number of 1s to correspond to a given density level, e.g., 25% of the reference column storing 1 to represent a density of 25%. Prior to the detection, a readout from the reference column is performed by applying unit pulses on all WLs. The reference column's BL current is integrated on a sampling capacitor as the threshold voltage.

The range detection is done by an SRAM readout. The BL current is integrated on a sampling capacitor to be the BL voltage. The SA compares the BL voltage to the threshold voltage generated by the reference column. If the BL voltage is below the threshold voltage, the value of the column is higher than the reference value, and the SA sets $EN = 1$ to the controller.

The controller checks all BLs' SA outputs. If an SA signals $EN = 1$, the controller activates only 50% of the WLs and another round of SRAM readout follows for only a small number of saturated columns. In the next round, if one $EN = 1$, the controller activates just 25% of the WLs with the columns at which the BL value is above the reference value in subsequent SRAM readouts. The process continues until the BL value is reduced to the reference level or below, thereby controlling the BL range.

Fig. 9 illustrates BL range control. Assume that prior to the detection, the threshold voltage is set to represent a 25% density. In the example, in cycle 0, the first and the second BL density exceed the 25% threshold, and the controller only activates 50% of the rows in the subsequent cycle 1 and cycle 2. In cycle 1, the first BL density still exceeds the threshold, and the controller activates only 25% of the rows in the subsequent cycle 3 and cycle 4. By actively limiting the density below 25%, the effective BL resolution is reduced by 2b. The proposed range control adapts to the effective BL resolution by activating more or fewer bitcells, thus we call it adaptive-range PIM or AR-PIM.

B. Energy Minimization

To reduce the ADC resolution and improve the sensing margin, low-resolution ADCs are necessary. When adopting low-resolution ADCs, only a portion of all the rows in the array at a time can be activated and the BL values are read out sequentially. Therefore, it may result in more processing cycles, which in turn costs more energy and a longer latency. To amortize this overhead, AR-PIM exploits the lower effectual BL range in runtime originating from data density levels of input activations and weights.

The lowest energy is investigated by sweeping the input activation and weight density each from 5% to 75%. If the ADC resolution is set based on the effective BL range (using the IA/W density levels as the proxy indicator), the number of processing cycles and energy consumption can be minimized. Fig. 10 shows the result of choosing the appropriate ADC resolution represented as ENOB for optimal energy consumption.

V. Evaluation of Energy and Performance

The energy consumption of the AR-PIM architecture is evaluated using DNN workloads based on the bit-level sparsity of activations and weights. The DNN workloads include LeNet with MNIST dataset as well as AlexNet and VGG11 with ImageNet dataset. The energy of AR-PIM is highly dependent on two factors, the runtime data sparsity and the ADC resolution.

For LeNet with MNIST dataset, both activations and weights are quantized to 8b for evaluations. Table I shows

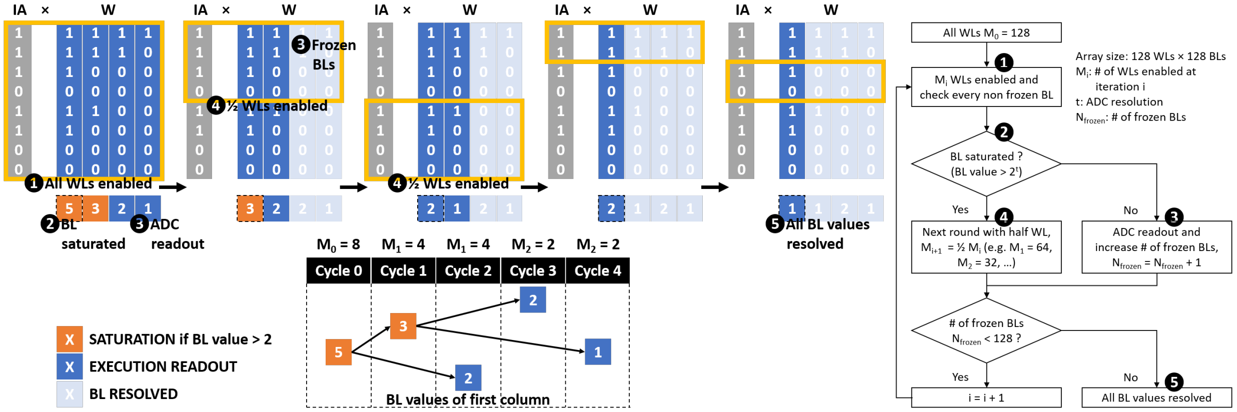

IA × W
1 All WLs enabled
2 BL saturated
3 ADC readout
3 Frozen BLs
4 ½ WLs enabled
5 All BL values resolved
M0 = 8
M1 = 4
M1 = 4
M2 = 2
M2 = 2
Cycle 0
Cycle 1
Cycle 2
Cycle 3
Cycle 4
BL values of first column
X SATURATION if BL value > 2
X EXECUTION READOUT
X BL RESOLVED
All WLs M0 = 128
1 Mi WLs enabled and check every non frozen BL
2 BL saturated ? (BL value > 2^t)
Yes
No
4 Next round with half WL, Mi+1 = ½ Mi (e.g. M1 = 64, M2 = 32, …)
3 ADC readout and increase # of frozen BLs, Nfrozen = Nfrozen + 1
# of frozen BLs Nfrozen < 128 ?
Yes
No
i = i + 1
5 All BL values resolved
Array size: 128 WLs × 128 BLs
Mi: # of WLs enabled at iteration i
t: ADC resolution
Nfrozen: # of frozen BLs

Fig. 9: Runtime range detection and adaptation of a simple 8×4 array example and the general flowchart for a 128×128 array.

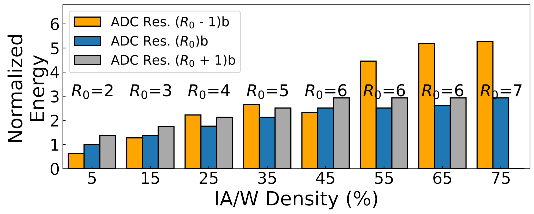


Fig. 10: Energy of AR-PIM for IA/W density each from 5% to 75% with conditions of ADC resolution R_0 - 1, R_0, R_0 + 1, where R_0 is the nominal value of ADC resolution settings depending on the product of array size, IA density, and W density.

the average BL values when running inference in the first and the second convolutional layers. Each layer is mapped to a 128×128 PIM module [8] as a baseline. In both layers, the average BL densities stay below 10%, making AR-PIM suitable as the effective BL range is low and consistent between columns.

Fig. 11(a) shows the normalized energy with different ADC resolution settings for the first and the second convolutional layers of LeNet. In the first convolutional layer, a lower ADC resolution reduces the energy consumption. The low utilization of cells along each column leads to a consistently low effective BL range. The low average BL value and the narrow distribution (Table I) allow the setting of a low ADC resolution to aggressively reduce the BL resolution to save the most energy.

In the second convolutional layer, a different trend is observed: when the ADC resolution is too low, the energy consumption increases. Different from the first layer, the utilization in the second layer of mapping is higher. The higher utilization results in a broader BL value distribution (Table I). The lowest ADC resolution could result in a large number of extra cycles and more energy. The energy-optimal ADC resolution can be set to capture most of the BLs, leaving only a small number of extra cycles to capture the remaining BLs.

Fig. 11(b) shows the latency implications of different ADC resolution settings. Generally speaking, the lower the ADC resolution, the higher the latency. The energy-optimal points tend to be low-resolution points where the latency does not increase excessively.

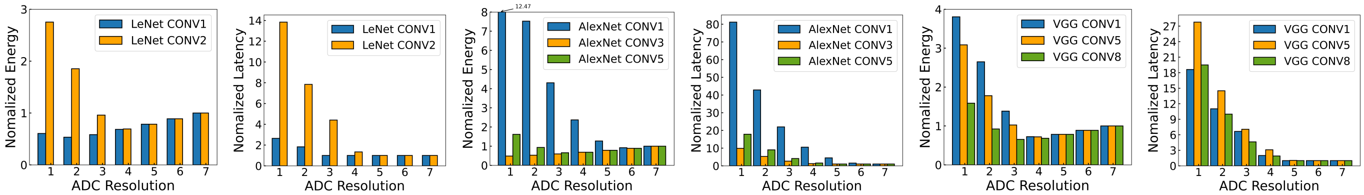
Fig. 11(c) and Fig. 11(d) show the normalized energy and latency of AR-PIM with different ADC resolution settings for the first, middle, and last convolutional layers in AlexNet with ImageNet dataset. The activations and weights are quantized to 16b and 12b in evaluations while maintaining the inference accuracy. The mean of BL values decreases and the distribution gets narrower in deep layers as in Table I.

Similar behavior can be observed in VGG11 as in Fig. 11(e) and Fig. 11(f). Fig. 12 shows the accuracy and energy tradeoff between with and without AR-PIM. The accuracy can be recovered in lower-resolution ADCs with minimal energy increase. Across different DNN workloads, AR-PIM can minimize the energy consumption while maintaining the inference accuracy over the baseline PIM with 7b ADC resolution. As a result, AR-PIM improves the energy efficiency over the baseline PIM. By using AR-PIM, latency-constrained applications can achieve up to 1.7× higher energy efficiency.

VI. Conclusion

This work explores the design boundary for analog PIM using SRAM in a 7nm process. The input encoding, BL range, ADC resolution, and ADC sharing are studied to analyze their impacts on the energy efficiency and the area cost of analog PIM. From the analyses, we conclude that low-bitwidth quantized NNs are more suitable to be deployed on analog PIM to save energy on power-constrained mobile devices. Addressing the challenges behind the deployment of multi-bit matrices in analog accelerators, AR-PIM is presented with a runtime BL range detection mechanism to adapt to a lower effective BL range.

AR-PIM eliminates the need for high-resolution ADCs and reduces the energy consumption of the ADCs. By adapting to the lower effectual BL range, AR-PIM also enhances the variation tolerance and the sensing margin. Considering the energy gain and latency overhead together, our evaluations show that AR-PIM provides 1.7× higher energy efficiency over the baseline PIM with 4.3× area reduction while maintaining the inference accuracy.



(a) (b) (c) (d) (e) (f)

Fig. 11: Energy consumption and latency of AR-PIM compared and normalized to the baseline PIM with 7b ADC resolution (the rightmost bar in each figure) for each layer running (a)(b) LeNet using MNIST dataset, (c)(d) AlexNet using ImageNet dataset, and (e)(f) VGG11 using ImageNet dataset.

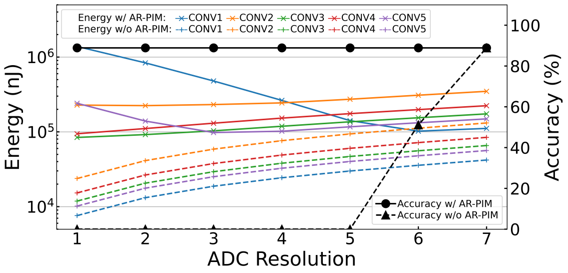


Fig. 12: Accuracy and energy trade-off with ADC resolution settings.

Acknowledgment

We thank Mudit Bhargava for advice and discussion. The work at the University of Michigan was supported in part by NSF CCF-1900675.

References

[1] D. Ciregan, U. Meier *et al.*, "Multi-column deep neural networks for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3642–3649.

[2] "NVIDIA Grace CPU." [Online]. Available: https://www.nvidia.com/en-us/data-center/grace-cpu/, 2021.

[3] J. Fowers, K. Ovtcharov *et al.*, "A configurable cloud-scale DNN processor for real-time AI," in *Proceedings of the International Symposium on Computer Architecture*, 2018, pp. 1–14.

[4] N. P. Jouppi, C. Young *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the International Symposium on Computer Architecture*, 2017, pp. 1–12.

[5] T. Chen, Z. Du *et al.*, "DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning," in *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, 2014, pp. 269–284.

[6] Y.-H. Chen, J. Emer *et al.*, "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," in *Proceedings of the International Symposium on Computer Architecture*, 2016, pp. 367–379.

[7] Y. Chen, T. Luo *et al.*, "DaDianNao: A Machine-Learning Supercomputer," in *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, 2014, pp. 609–622.

[8] A. Shafiee, A. Nag *et al.*, "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in *Proceedings of the International Symposium on Computer Architecture*, 2016, pp. 14–26.

[9] P. Jain, U. Arslan *et al.*, "A 3.6 Mb 10.1 Mb/mm^2 embedded non-volatile ReRAM macro in 22nm FinFET technology with adaptive forming/set/reset schemes yielding down to 0.5 V with sensing time of 5ns at 0.7 V," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2019, pp. 212–214.

[10] L. Wei, J. G. Alzate *et al.*, "A 7Mb STT-MRAM in 22FFL FinFET technology with 4ns read sensing time at 0.9 V using write-verify-write scheme and offset-cancellation sensing technique," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2019, pp. 214–216.

[11] Y.-D. Chih, Y.-C. Shih *et al.*, "A 22nm 32Mb embedded STT-MRAM with 10ns read speed, 1M cycle write endurance, 10 years retention at 150°C and high immunity to magnetic field interference," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2020, pp. 222–224.

[12] C.-C. Chou, Z.-J. Lin *et al.*, "A 22nm 96KX144 RRAM macro with a self-tracking reference and a low ripple charge pump to achieve a configurable read window and a wide operating voltage range," in *IEEE Symposium on VLSI Circuits*, 2020, pp. 1–2.

[13] J. Zhang, Z. Wang *et al.*, "A machine-learning classifier implemented in a standard 6T SRAM array," in *IEEE Symposium on VLSI Circuits*, 2016, pp. 1–2.

[14] S. K. Gonugondla, M. Kang *et al.*, "A 42pJ/decision 3.12TOPS/W robust in-memory machine learning classifier with on-chip training," in *IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2018, pp. 490–492.

[15] A. Biswas and A. P. Chandrakasan, "Conv-RAM: An energy-efficient SRAM with embedded convolution computation for low-power CNN-based machine learning applications," in *IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2018, pp. 488–490.

[16] Q. Dong, M. E. Sinangil *et al.*, "A 351TOPS/W and 372.4 GOPS compute-in-memory SRAM macro in 7nm FinFET CMOS for machine-learning applications," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2020, pp. 242–244.

[17] P. Chi, S. Li *et al.*, "PRIME: A novel processing-in-memory architecture for neural network computation in ReRAM-based main memory," in *Proceedings of the International Symposium on Computer Architecture*, 2016, pp. 27–39.

[18] L. Song, X. Qian *et al.*, "PipeLayer: A pipelined ReRAM-based accelerator for deep learning," in *IEEE International Symposium on High Performance Computer Architecture*, 2017, pp. 541–552.

[19] B. Murmann, "ADC Performance Survey 1997-2021 (ISSCC & VLSI Symposium)." [Online]. Available: http://web.stanford.edu/ murmann/adcsurvey.html, 2021.

[20] S. K. Gonugondla, M. Kang *et al.*, "A variation-tolerant in-memory machine learning classifier via on-chip training," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 11, pp. 3163–3173, Nov 2018.

[21] Y.-D. Chih, P.-H. Lee *et al.*, "An 89TOPS/W and 16.3 TOPS/mm^2 all-digital sram-based full-precision compute-in memory macro in 22nm for machine-learning edge applications," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2021, pp. 252–254.

[22] S. Cosemans, B. Verhoef *et al.*, "Towards 10000TOPS/W DNN inference with analog in-memory computing–A circuit blueprint, device options and requirements," in *IEEE International Electron Devices Meeting (IEDM)*, 2019, pp. 22–2.

[23] T.-J. Yang and V. Sze, "Design considerations for efficient deep neural networks on processing-in-memory accelerators," in *IEEE International Electron Devices Meeting (IEDM)*, 2019, pp. 22–1.

[24] B. Moons, R. Uytterhoeven *et al.*, "Envision: A 0.26-to-10TOPS/W subword-parallel dynamic-voltage-accuracy-frequency-scalable Convolutional Neural Network processor in 28nm FDSOI," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2017, pp. 246–247.

[25] W.-H. Chen, K.-X. Li *et al.*, "A 65nm 1Mb nonvolatile computing-in-memory ReRAM macro with sub-16ns multiply-and-accumulate for binary DNN AI edge processors," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2018, pp. 494–496.

[26] J. Albericio, A. Delmás *et al.*, "Bit-pragmatic deep neural network computing," in *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, 2017, pp. 382–394.