Towards ADC-Less Compute-In-Memory Accelerators for Energy Efficient Deep Learning

Utkarsh Saxena, Indranil Chakraborty, Kaushik Roy
School of Electrical and Computer Engineering, Purdue University {saxenau, ichakra, kaushik}@purdue.edu

***Abstract*—Compute-in-Memory (CiM) hardware has shown great potential in accelerating Deep Neural Networks (DNNs). However, most CiM accelerators for matrix vector multiplication rely on costly analog to digital converters (ADCs) which becomes a bottleneck in achieving high energy efficiency. In this work, we propose a hardware-software co-design approach to reduce the aforementioned ADC costs through partial-sum quantization. Specifically, we replace ADCs with 1-bit sense amplifiers and develop a quantization aware training methodology to compensate for the loss in representation ability. We show that the proposed ADC-less DNN model achieves 1.1x-9.6x reduction in energy consumption while maintaining accuracy within 1% of the DNN model without partial-sum quantization.**

***Index Terms*—Compute-in-memory, DNN acceleration, analog computing, quantization, hardware-software co-design**

I. Introduction

Compute-in-Memory (CiM) accelerators have shown great potential in efficient acceleration of DNNs [1], [2]. CiM accelerators achieve efficiency due to considerable reduction in data movements between the memory and the compute units [3], and performing computations in-memory in the analog domain. As analog signals are less tolerant of noise, analog to digital converters (ADCs) are needed to read out the results of in-memory operations and enable robust communication in the digital domain. Moreover, accurately capturing the dynamic range of analog partial-sums (PS) requires high precision ADCs which can consume as much as 60% of the total energy of CiM accelerators [4] (Fig.1).

It is necessary to mitigate the overhead posed by ADCs to improve the efficiency of CiM accelerators and make them more amenable for edge devices. Some CiM hardware demonstrations reduce ADC precision for sparse inputs [5], [6], while some research works use a reduced precision ADC with non-linear quantization [7]. However, the reduction in ADC precision in such hardware-only approaches is limited due to the need to maintain DNN accuracy. Alternatively, [8] proposed to eliminate ADC with modified sense amplifiers (SA) to perform in-situ Boolean operations. However, such an approach requires a large number of Boolean operations to perform complex MVM computations, resulting in limited efficiency improvements.

In this work, we propose hardware-software co-design to enable 1-bit (1b) PS quantization. Specifically, we develop a training methodology to achieve high accuracy with 1b ADC precision. Consequently, we are able to eliminate ADCs and use sense amplifiers (SA) for 1b analog to digital conversion of PS (Fig.2(c)). The proposed ADC-Less CiM accelerator for DNNs achieves high accuracy and high efficiency on moderately sized

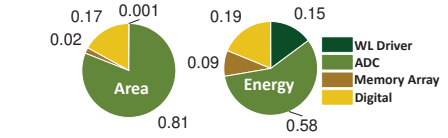


Fig. 1. Area and Energy braeakdown in CiM accelerators.

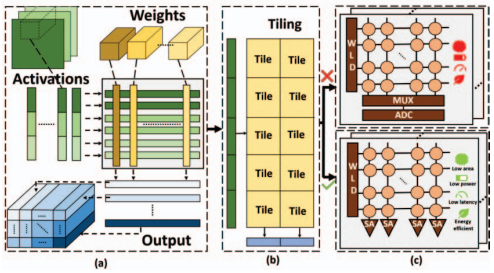


Fig. 2. Mapping DNN Layer on CiM accelerator array.

DNN workloads, making them a suitable computing substrate for resource constrained devices.

We make the following contributions:

- We use hardware/software co-design to develop ADC-Less CiM accelerators which perform in-memory MVM operation using 1b sense amplifiers.
- We develop a CiM hardware aware training methodology for ADC-Less DNNs and propose a 1b PS quantization function to achieve high accuracy.
- We provide an evaluation of the improvements obtained by our proposed ADC-Less CiM accelerators for DNN workloads.
- We provide a case study on the accuracy degradation of ADC-Less DNNs caused by device variations, a common issue in non-volatile memory devices.

II. Background

A. Executing DNN layers in CiM accelerators

When running a convolution (Conv) layer on CiM accelerators, weights are stored in memory array and activations are applied on the wordlines (WLs). The conv layer operation is converted to MVM operation between weights and activations as shown in Fig.2(a). Additionally, CiM accelerators use *tiling* and *bit-slicing* to scale to large DNN layers. *Tiling* refers to dividing a 2D weight matrix into multiple sub-matrices based on the size of the memory array as shown in Fig.2(b). *Bit-slicing* enables bit-serial computation in memory arrays to handle the gap in precision supported by the hardware (due to technological constraints) and the precision required by weights

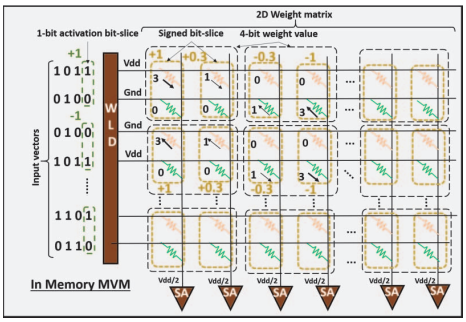


Fig. 3. Matrix Vector Multiplication in memory array.

Algorithm 1 Hardware aware implementation of Conv layer

Input: $\mathbf{A_l}, \mathbf{W_l}, \mathbf{n_w}, \mathbf{n_a}, \mathbf{s_w}, \mathbf{s_a}, \mathbf{N}$
Output: Layer Output $\mathbf{O_l}$

$G \leftarrow \text{ceil}(\frac{in\_channels * kH * kW}{N})$
while $N\%G \neq 0$ **do**
  $G \leftarrow G + 1$
end while
$W^q_{l,n_w} \leftarrow$ **Quantize**(W_l, n_w, s_w)
$A^q_{l,n_a} \leftarrow$ **Quantize**(A_l, n_a, s_a)
for $i \in [0, n_a)$ **do** ▷ iterative group Conv for W,A bit slice
  for $j \in [0, n_w)$ **do**
    $O_t \leftarrow$ **Conv**$(\mathbf{A_{q,i}}, \mathbf{W_{q,j}}, groups = \mathbf{G}, *kwargs)$
    $O^q_t \leftarrow$ **ADC-Less Quantize**(O_t)
    $O_{l,g} \leftarrow O_l + O^q_t$
  end for
end for
for $g \in [0, G)$ **do**
  $O_l \leftarrow O_l + O_{l,g}$ ▷ Accumulate across groups
end for

and activations. Weights are bit-sliced across multiple memory devices and activations bit-sliced and applied to the WLs across multiple timesteps. (Fig.3). Consequently, ADC outputs from different memory arrays and bit-slices are accumulated forming the conv output. Each memory array column computes a PS of the entire MVM operation between bit-sliced weights and the activations in the analog domain.

Fig. 3 shows an example of MVM operation between a 4-bit weight (2 slices of 2-bit) and 4-bit activation (4 slices of 1-bit). A group of two memory devices in a column store a signed bit-sliced weight and two WLs provide signed bit-sliced activation. Positive multiply output corresponds to current flowing into BL while negative multiply output corresponds to current flowing from BL as shown in 3. Total current flowing into/from BL corresponds to the MVM PS in the analog domain which is read by peripheral circuits Fig.2(c)). To accurately capture the dynamic range of the analog PS accumulated over BL, the minimum precision of ADCs required is given by,

$$ADC = log(N) + s_w + s_a - 2, \tag{1}$$

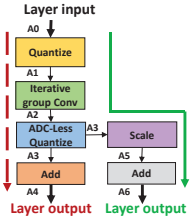
where N is the number of WLs turned ON, s_w and s_a are the precision of bit-slice of weights and activations, respectively. Reducing ADC precision quantizes PS and removing ADCs altogether while using SA for analog to digital conversion of PS quantizes it to 1b.

B. Related Works

Most works on CiM accelerators recognize the overhead posed by ADCs and attempt to mitigate it. Authors in [9] evaluate signal to noise ratio (SNR) metrics to mathematically obtain reduced ADC precision, achieving the best accuracy. The authors in [10] propose ADC quantization schemes that are resilient to non idealities presented by CiM hardware. However, their approach suffers from considerable degradation in DNN accuracy when using low precision ADCs.The authors in [11] proposed an input and weight splitting methodology to reduce ADC precision from 8-bit to 4-bit. Authors in [12] explored quantization aware training for binary Resnet on CiM accelerators with low ADC resolution.

III. ADC-Less DNNs

We perform quantization aware training to achieve high accuracy with ADC-Less DNNs. To emulate the DNN layer operation within CiM accelerator, we follow algorithm 1.Group parameter in conv layer is used to emulate *tiling* of Conv operation across multiple memory arrays. Each group corresponds to one memory array performing iterative Conv operations over bit-sliced activations and weights (algorithm 1). *Bit-slicing* of weights and activations is implemented by dividing them into n_w and n_a bit-slices of s_w and s_a bits, respectively, according to [13], [14]. As discussed later, we point out two issues specific to ADC-Less DNNs that limit the accuracy. One is considerable reduction in representation ability and the other is severe gradient clipping that occurs during training. With our proposed training methodology, ADC-Less DNNs achieve high accuracy on MNIST and CIFAR-10 datasets.



	Tensor Dim.	Tensor val.
A0	D	FP
A1	D·n_a	2^{s_a}
A2	D·n_a·n_w·G	$2^{\log(N)+s_w+s_a-2}$
A3	D·n_a·n_w·G	2
A4	D	(G·n_w·n_a+1)
A5	D·n_a·n_w·G	2
A6	D	$2^{(G \cdot n_w \cdot n_a)}$

(b)

Fig. 4. Representation ability of Convolution layer in ADC-Less DNNs.

A. Representation Ability

Fig.4(a) shows two alternatives to operations in Conv layer defined in algorithm 1. One is when quantized PS are added to get layer output (dotted arrow) and other is when quantized PS are scaled and added (solid arrow). Table in Fig.4(b) shows the dimensions of each intermediate tensor and different possible values for each element in the tensor. Higher the number of possible states for each element, higher would be the representation ability of that tensor. For the case of quantized networks, improved representation ability often translates to improved accuracy [15]. Binary PS quantization causes elements of tensor A3 in Fig.4 to have two possible values (± 1). When we compare possible values for tensor elements of A4 with A2, we observe that there is considerable amount of reduction in representation ability. Scaling the binary quantized PS values (± 1) by scale parameter a transforms them to $\pm a$. Adding multiple scaled 1b PS makes tensor A6 have higher representation ability than A4 leading to higher DNN accuracy.

B. Gradient Clipping and ADC-Less Quantization

Binary PS quantization function implemented by *sign* function during forward pass needs to be approximated during

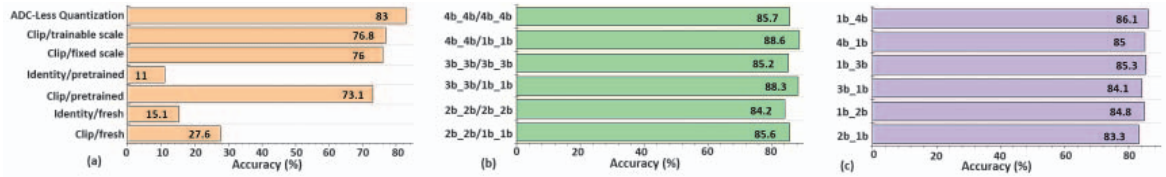


Fig. 5. Accuracy results on CIFAR-10 on ADC-Less Resnet-20. (a) Ablation study results on 1b quantized weights and activations. (b) Impact on accuracy under different bit-slice precision. Labels on Y-axis denote the precision: $W\_A/s_w\_s_a$ (c) Impact on accuracy with increasing weight and activation precision at 1b bit-slice. Labels on Y-axis denote the precision: W_A

backward pass while training. There are usually two flavors of approximation to sign function:

$$Clip: y = Hardtanh(-1, 1); Identity: y = x \quad (2)$$

Proper estimation of sign function is essential in achieving good accuracy while training [16]. $Identity$ approximation loses gradient information of quantization (gradient mismatch) while $Clip$ approximation loses gradient information outside clipping interval [16].

To better approximate the sign function and improve the representation ability of ADC-Less DNNs, we propose a modified ADC-Less quantization function. As shown in Fig.6, we use a scaled sign function for 1b PS quantization. The scale parameter $'a'$ improves the representation ability. During backward pass, the scale parameter is trained based on the amount of gradients lying in the clipping range. Compared to $Clip$ approximation, the clipping range is a trainable variable that is equal to $(-a, a)$. This improves the issue of gradient clipping during training since the training algorithm increases the scale parameter when considerable amount of gradients lie in the clipping range.

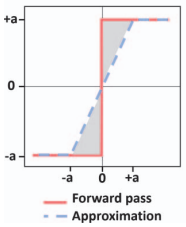


Fig. 6. ADC-Less Quantization

IV. Methodology and Experimental Results

In this section, we present the methodology adopted along with the accuracy results of ADC-Less DNNs and the hardware evaluation results for ADC-Less CiM accelerators.

A. Software and Hardware Evaluation Setup

For evaluating the impact on accuracy of the proposed training methodology for ADC-Less DNNs, we trained ADC-Less Resnet-20 and Resnet-18 model according to [16] on CIFAR 10 and ADC-Less LeNet-5 model on MNIST dataset.

We used PUMA [2] simulator for evaluating the performance gain obtained with ADC-Less CiM accelerators. Without loss of generality, we used PUMA to evaluate ADC-Less CiM accelerators on Resistive Random Access Memory (RRAM) based memory arrays.

B. Ablation Study

We performed an ablation study to evaluate the different design choices when training ADC-Less DNNs. Fig.5(a) shows the results on CIFAR-10 dataset with Resnet-20 model for 1b weights and activations. We observed that irrespective of initialization, the ADC-Less DNN does not train when $Identity$ approximation is used as shown in Fig.5(a). $Clip$ achieves higher accuracy when initialized with a pretrained Resnet-20 ($Clip/pretrained$). Further, we found that the accuracy was 0.5% better when the scale factor was trainable (Fig.5(a)). Finally, we evaluated the accuracy with our proposed ADC-Less quantization ($ADC\text{-}Less\ Quantization$) and it performed 6.2% better than $Clip/trainable\ scale$. This concurs with the identified limitations with training networks with 1b PS and demonstrates the effectiveness of the proposed quantization function.

TABLE I
Accuracy Results

W/A	s_w/s_a	***HP-ADC***	***ADC-Less***	***Gap***	***QAT***
MNIST					
1/1	1/1	99.0	98.9	0.1	98.8
2/2	2/2	99.2	99.0	0.2	98.9
3/3	3/3	99.2	99.1	0.1	99.1
4/4	4/4	99.3	99.2	0.1	99.1
CIFAR-10 (Resnet-20)					
1/1	1/1	85.5	83.0	2.5	73.1
2/2	1/1	88.0	85.6	2.4	79.8
3/3	1/1	89.3	88.2	1.1	85.5
4/4	1/1	89.5	88.6	0.9	87.2
CIFAR-10 (Resnet-18)					
1/1	1/1	91.3	87.4	3.9	82.5
2/2	2/2	92.3	91.2	1.1	86.0
3/3	3/3	92.7	91.9	0.8	87.5
4/4	4/4	93.0	92.4	0.6	89.8

C. Accuracy Results

The accuracy results are shown in Table 1. The accuracy is compared with two baselines: 1) DNNs with no PS quantization implemented using high precision ADCs (HP-ADC), and 2) Naive Quantization Aware Training (QAT) approach where DNNs are trained with 1b PS using $Clip$ approximation. For MNIST dataset, ADC-Less DNNs achieve accuracy within 0.2% of HP-ADC DNNs. For CIFAR-10, we observe a minor 0.9% degradation in accuracy with ADC-Less Resnet-20 and 0.6% degradation with ADC-Less Resnet-18 when compared with HP-ADC DNN. Also, ADC-Less DNN accuracy is always better than QAT.

Impact of precision. The accuracy with 1b bit-slice precision is better than the case with higher precision bit-slice ($n_w = 1, n_a = 1$) as shown in Fig.5(b). This is because binary bit-slice corresponds to higher representation ability. Fig.5(c) shows that increasing the activation precision improves accuracy, on an average 1.2% more than increasing weight precision by same amount.

Impact of memory array size. As the memory array size (N) increases, the accuracy decreases (Fig.7). Lower memory array size translates to higher accuracy due to higher representation ability. With larger memory array size, more number of accumulations within the memory array are quantized to 1b.

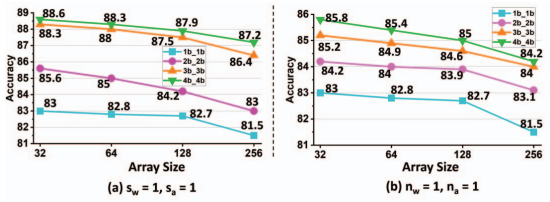


Fig. 7. Impact of memory array size on accuracy of ADC-Less DNN with (a) 1b bit-slice, (b) high precision bit-slice.

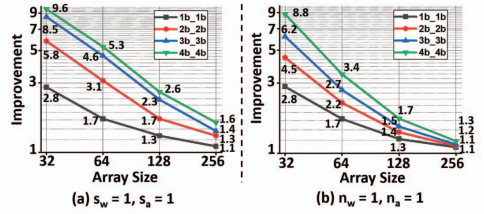


Fig. 8. Energy improvement with ADC-Less CiM accelerator running Resnet-20 for (a) 1b bit-slice, (b) high precision bit-slice. Labels represent weights and activation precision.

D. Energy Improvements

The energy improvements with ADC-Less CiM accelerators are shown over a baseline CiM accelerator with high precision ADC (eq. 1). Fig.8 shows the impact of precision on energy improvement on CIFAR-10 running on Resnet-20. We observe that the energy improvements are higher when weights and activation precisions are higher (8(a),(b)). We also observe that energy improvements are higher for the case of 1b bit-slice precision (8(a)) when compared with high precision bit-slice (Fig.8(b)). This is because more ADC conversions (albeit with lower precision) are eliminated in the case of 1b bit-slice.

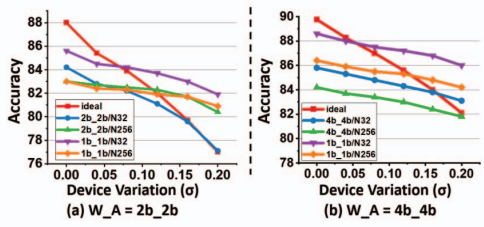


Fig. 9. Impact of device variation under different weight and activation precision: (a) 2-bit, (b) 4-bit. Labels represent the bit-slice precision for weights and activations and the memory array size.

E. Variation Analysis

Device variations is a pertinent issue in NVM based CiM accelerators. Device variations can be temporal, causing cycle to cycle variation in NVM device resistance or they can be spatial due to imperfections in the fabrication technology. As shown in [17], device variations can be modelled as log normal distribution with zero mean. We vary the standard deviation of device variation from 0 (no variation) to 0.2 (high device variation) and observe the impact on accuracy of HP-ADC DNNs and ADC-Less DNNs for a range of weight and activation precisions (Fig. 9(a),(b)). The accuracy numbers presented is the mean accuracy obtained over 50 trials. Fig.9 shows that the accuracy gradually degrades as the variation increases. However, the accuracy degradation for ADC-Less DNNs is much less severe when compared with HP-ADC DNNs. At moderate to high variation ($\sigma = 0.12 - 0.2$) the accuracy of ADC-Less DNNs ends up being higher than HP-ADC DNNs.

V. Conclusion

In this paper, we present a hardware/software co-design approach to improve energy efficiency of CiM accelerators by eliminating ADCs through 1-bit PS quantization. We train ADC-Less DNNs on CIFAR-10 and MNIST dataset and achieve accuracy close to DNNs without PS quantization (0.2% accuracy drop for MNIST dataset and 0.6% accuracy drop for CIFAR-10 dataset). The ADC-Less CiM accelerators achieve up to 9.6x energy improvement over baseline CiM accelerators. Additionally, we show that under realistic scenarios of NVM device variations, ADC-Less DNNs are less affected by perturbations and often have higher accuracy than DNNs without PS quantization.

Acknowledgment

This work was supported in part by the National Science Foundation, Vannevar Bush Faculty Fellowship, and by the Center for Brain-Inspired Computing (C-BRIC), one of six centers in JUMP, funded by Semiconductor Research Corporation (SRC) and DARPA.

References

[1] A. Biswas *et al.*, "Conv-sram: An energy-efficient sram with in-memory dot-product computation for low-power convolutional neural networks," *IEEE JSSC*, 2019.
[2] A. Ankit *et al.*, "Puma: A programmable ultra-efficient memristor-based accelerator for machine learning inference," in *ASPLOS*, 2019.
[3] N. Verma *et al.*, "In-memory computing: Advances and prospects," *IEEE Solid-State Circuits Magazine*, 2019.
[4] K. Roy *et al.*, "In-memory computing in emerging memory technologies for machine learning: An overview," in *DAC*, 2020.
[5] M. Ali *et al.*, "A 35.5-127.2 tops/w dynamic sparsity-aware reconfigurable-precision compute-in-memory sram macro for machine learning," *IEEE SSC-L*, 2021.
[6] J.-W. Su *et al.*, "16.3 a 28nm 384kb 6t-sram computation-in-memory macro with 8b precision for ai edge chips," in *ISSCC*, 2021.
[7] S. Yin *et al.*, "Xnor-sram: In-memory computing sram macro for binary/ternary deep neural networks," *IEEE JSSC*, 2020.
[8] J. Wang *et al.*, "A 28-nm compute sram with bit-serial logic/arithmetic operations for programmable in-memory vector computing," *IEEE JSSC*.
[9] S. K. Gonugondla *et al.*, "Fundamental limits on the precision of in-memory architectures," in *ICCAD*, 2020.
[10] W.-C. Wei *et al.*, "A relaxed quantization training method for hardware limitations of resistive random access memory (reram)-based computing-in-memory," *IEEE JXCDC*, 2020.
[11] Y. Kim *et al.*, "Input-splitting of large neural networks for power-efficient accelerator with resistive crossbar memory array," in *ISLPED*, 2018.
[12] ——, "Mapping binary resnets on computing-in-memory hardware with low-bit adcs," in *DATE*, 2021.
[13] X. Lin *et al.*, "Towards accurate binary convolutional neural network," ser. NeurIPS'17, 2017.
[14] S. Zhou *et al.*, "Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," 2018.
[15] Z. Liu *et al.*, "Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm," in *ECCV*, 2018.
[16] H. Qin *et al.*, "Forward and backward information retention for accurate binary neural networks," in *IEEE CVPR*, 2020.
[17] G. Charan *et al.*, "Accurate inference with inaccurate rram devices: Statistical data, model transfer, and on-line adaptation," in *DAC*, 2020.