AA-ResNet: Energy Efficient All-Analog ResNet Accelerator

Jongyup Lim, Myungjoon Choi, Bowen Liu, Taewook Kang, Ziyun Li, Zhehong Wang,
Yiqun Zhang, Kaiyuan Yang, David Blaauw, Hun-Seok Kim, and Dennis Sylvester
Department of Electrical Engineering and Computer Science,
University of Michigan, Ann Arbor, MI
Email: jongyup@umich.edu

Abstract—**High energy efficiency is a major concern for emerging machine learning accelerators designed for IoT edge computing. Recent studies propose in-memory and mixed-signal approaches to minimize energy overhead resulting from frequent memory accesses and extensive digital computation. However, their energy efficiency gain is often limited by the overhead of digital-to-analog and analog-to-digital conversions at the boundary of the compute-memory. In this paper, we propose a new in-memory accelerator that performs all computation in the analog domain for a large, multi-level neural network (NN) for the first time avoiding any digital-to-analog or analog-to-digital conversion overhead. We propose an all-analog ResNet (AA-ResNet) accelerator in 28-nm CMOS, achieving an energy efficiency of 1.2 µJ/inference and inference rate of 325K images/s for the CIFAR-10 and SVHN datasets in SPICE simulation.**

Keywords—**machine learning accelerator, in-memory computing, analog computing, deep residual learning**

I. Introduction

Today, the proliferation of IoT devices increases demand of machine learning accelerators designed for edge computing and reinforces the importance of energy efficiency of such accelerators. Especially, vast amount of energy consumption from frequent memory access to load the data during inference must be reduced to meet the limited energy budget of edge-computing.

Recently, various in-memory or mixed-signal approaches [1-4] address the issue and reduce energy consumption by replacing frequent memory read accesses and digital computations with in-memory and analog computations. In addition, recent studies propose modified training methods for mixed-signal based accelerators with low bit precision [3], and in-situ methods for minimizing accuracy degradation due to process variation [4]. However, all these approaches include digital-to-analog converters (DAC) and analog-to-digital converters (ADC) at the front and end of each hidden layer to store and broadcast features in digital representation [1-3]. Further, they implement the required non-linear (NL) functions in digital domain [2]. The DACs /ADCs are energy bottlenecks especially when high precision of weight or activation is required. The energy overhead gets even worse when implementing deep convolutional neural networks (CNNs). Hence prior mixed signal designs have been largely restricted to simple shallow networks. Other approaches implementing binarized CNN (BNN) in mixed-signal domain has been proposed using XNOR for multiplication and charge sharing techniques for addition [5,6]. BNN has the benefit of reducing computation complexity to a single bit, and as such, these mixed-signal accelerators reduce the DAC and ADC energy overhead, since they only have a single bit precision for both weights and activations. However, the BNN works well only for moderately sized networks (e.g., AlexNet and nine-layer networks with 328KB [5] / 295KB [6] of weights) and have a critical limit on the scalability to support very large networks that are difficult to train with binary weights.

To address the challenges, we propose the first multi-layer (total 18) all-analog ResNet (AA-ResNet) accelerator in 28nm CMOS with 32.2KB of weight storage, implementing not only convolution, but also NL function, storage of value for subsequence use, and routing between layers all in analog domain (Fig. 1). Weights and activations are in 4bit and 3~7bit precision, respectively, thereby offering significantly better precision compared to BNNs.

II. Overview

A. Convolution in pulse-to-charge domain

For every layer, the input activations are represented in the pulse-width domain as shown in Eq (1):

$$x_{i,j,d}^{(l)} = \Delta t_{i,j,d}^{(l)} \tag{1}$$

where $x_{i,j,d}^{(l)}$ is the input activation value of the i-th row, j-th column, d-th depth of l-th layer, and $\Delta t_{i,j,d}^{(l)}$ is the corresponding pulse width. In the proposed analog convolution, the weights, $w_{i,j,d,k}^{(l)}$ are the product of the 4-bit sign-and-magnitude digital values $W_{i,j,d,k}^{(l)}$ stored in the 6T SRAM arrays and the weight control DC current $I_{LSB}^{(l)}$ that charges capacitors in the analog integrators (Fig. 1). In Eq (2), k represents the kernel index.

$$w_{i,j,d,k}^{(l)} = W_{i,j,d,k}^{(l)} \cdot I_{LSB}^{(l)} \tag{2}$$

The input activation value determines the on-time of the DC current $I_{LSB}^{(l)}$ in Eq (2). The accumulated charge is proportional to $I_{LSB}^{(l)}$, the stored weight value, and the time period the current is turned on. Therefore, the accumulated charge represents the multiplication of the input and weight. Multiple wires are shorted together at the input of an analog integrator, which merges all of the charge flowing through the tied wires. Thus, the total integrated charge is equivalent to the convolution output, as shown in Eq (3a,b). Because the DC current has only a single polarity (pull down), a pair of the integrators integrate charge for the positive and negative convolution value separately.

$$Q_{i,j,k}^{+(l)} = \sum_d \sum_{j'=1,2,3} \sum_{\substack{i'=1,2,3 \\ sign\left(W_{i',j',d,k}^{(n)}\right) \geq 0}} \left|W_{i',j',d,k}^{(l)}\right| \cdot I_{LSB}^{(l)} \cdot \Delta t_{i+i'-2,j+j'-2,d}^{(l)} \tag{3a}$$

$$Q_{i,j,k}^{-(l)} = \sum_d \sum_{j'=1,2,3} \sum_{\substack{i'=1,2,3 \\ sign\left(W_{i',j',d,k}^{(n)}\right) < 0}} \left|W_{i',j',d,k}^{(l)}\right| \cdot I_{LSB}^{(l)} \cdot \Delta t_{i+i'-2,j+j'-2,d}^{(l)} \tag{3b}$$

$$y_{i,j,k}^{(l)} = Q_{i,j,k}^{+(l)} - Q_{i,j,k}^{-(l)} \tag{3c}$$

Subtraction for $y_{i,j,k}^{(l)}$ is performed in the pulse domain after voltage-to-pulse conversion as explained in section II.C.

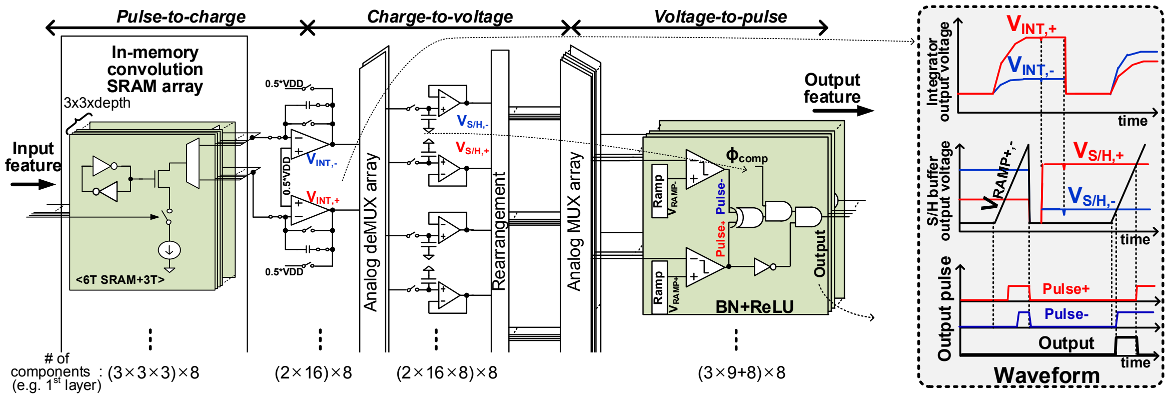


Fig. 1. A single layer(L1) structure of proposed AA-ResNet accelerator

In the proposed design, we also implement the feedforward shortcut, which is the key idea of ResNet [7,8] that improves accuracy of deep networks through residual learning. The shortcut connection, $y_{i,j,k}^{(l)} = \vec{w}^{(l)} \cdot \vec{x}^{(l)} + x_{i,j,k}^{(l-1)}$ is also calculated in the charge domain by tying the SRAM arrays and current path for $x_{i,j,k}^{(l-1)}$ to the input of the integrators. The implementation of residual learning is discussed in section III.A.

B. Sampling and holding in charge-to-pulse domain

The convolution results are stored in the analog (charge) domain and broadcasted at the proper timing to the NL function blocks and the next layers. The charge on the capacitors (Fig. 1) for the integrator pairs $Q\,_{i,j,k}^{+(l)}$ and $Q\,_{i,j,k}^{-(l)}$ can be directly converted into the voltage level $V_{INT}\,_{i,j,k}^{+(l)}$ and $V_{INT}\,_{i,j,k}^{-(l)}$, respectively, as in Eq (4a) and (4b), since the analog integrators hold the bottom plate of the integrator capacitors to a constant voltage of $0.5 \cdot V_{DD}$. The voltages $V_{INT}\,_{i,j,k}^{+(l)}$ and $V_{INT}\,_{i,j,k}^{-(l)}$ are sampled when the charge integration is complete. The buffers hold the sampled voltage to be fed into the NL block (Fig. 1).

$$V_{INT}\,_{i,j,k}^{+(l)} = \frac{1}{C} \cdot Q\,_{i,j,k}^{+(l)} + \frac{1}{2} \cdot V_{DD} \qquad (4a)$$

$$V_{INT}\,_{i,j,k}^{-(l)} = \frac{1}{C} \cdot Q\,_{i,j,k}^{-(l)} + \frac{1}{2} \cdot V_{DD} \qquad (4b)$$

C. NL function in voltage-to-pulse domain

NL function is performed in the analog domain, converting the convolution output from voltage to the pulse-width domain (Fig. 1). In the NL block, a ramp voltage, which monotonously rises in time, is compared with $V_{INT}\,_{i,j,k}^{+(l)}$ and $V_{INT}\,_{i,j,k}^{-(l)}$ using a comparator. The output of the comparator encodes the NL function value in the pulse-width domain (Fig. 2). Various non-linear functions such as ReLU can be realized by properly shaping the ramp voltage. In the proposed design, ReLU with batch normalization (BN) is implemented using the ramping voltages $V_{RAMP,k}^{+}$ and $V_{RAMP,k}^{-}$ generated by the ramp voltage generator structure discussed in section III.C.

$$V_{RAMP,k}^{+}\left(\Delta t_{i,j,k}^{+(l)}\right) = V_{INT}\,_{i,j,k}^{+(l)} \qquad (5a)$$

$$V_{RAMP,k}^{-}\left(\Delta t_{i,j,k}^{-(l)}\right) = V_{INT}\,_{i,j,k}^{-(l)} \qquad (5b)$$

We define $\Delta t_{i,j,k}^{+(l)}$ and $\Delta t_{i,j,k}^{-(l)}$ as the time between the start of the comparison and the point where $V_{RAMP,k}^{+}$ ($V_{RAMP,k}^{-}$) exceeds $V_{INT}\,_{i,j,k}^{+(l)}$ ($V_{INT}\,_{i,j,k}^{-(l)}$).

$$\Delta t_{i,j,k}^{(l+1)} = \begin{cases} \Delta t_{i,j,k}^{+(l)} - \Delta t_{i,j,k}^{-(l)}, & \Delta t_{i,j,k}^{+(l)} \geq \Delta t_{i,j,k}^{-(l)} \\ 0, & \Delta t_{i,j,k}^{+(l)} < \Delta t_{i,j,k}^{-(l)} \end{cases} \qquad (6)$$

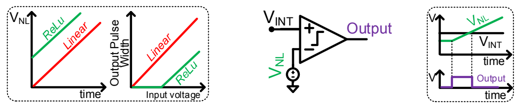
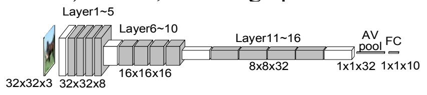


Fig. 2. Time domain mapping of non-linear voltage

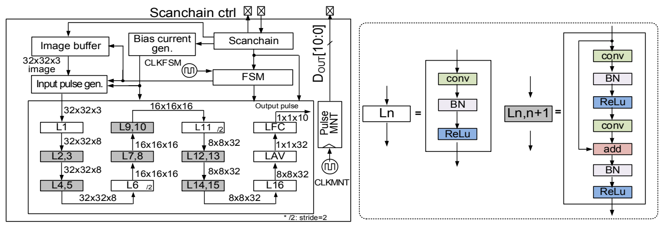
Using the logic gates in Fig. 1, the final output pulse width $\Delta t_{i,j,k}^{(l+1)}$ is obtained by the difference between $\Delta t_{i,j,k}^{+(l)}$ and $\Delta t_{i,j,k}^{-(l)}$ as in Eq (6), realizing ReLU.

D. Overall Structure

The proposed accelerator implements a modified ResNet [7,8] that consists of 16 convolution + BN + ReLU layers, an average pooling layer, and a fully connected (FC) layer as shown in Fig. 3. Layers colored in grey in Fig. 3 have feedforward shortcut connections that are unique to ResNets. The entire datapath of 19 layers is fully-pipelined to generate classification output at a very high throughput of one image per 64 cycles (64×48ns) or 325,520 image per second.



(a)



(b)

Fig. 3. (a) A diagram of modified ResNet and (b) overall hardware architecture

III. Implementation Details

A. In-memory convolution SRAM array cells

Fig. 4 shows in-memory convolution SRAM array cells with 3T-readout buffers. Stacked NMOS transistors offer tolerance to V_{DS} variation, generating linear currents. MSB of the weight (sign) selects one of the current conducting paths. The convolution layers with residual learning require an addition block, as shown at bottom of Fig. 4. In the addition block, instead of weight, a scaling factor $s[2:0]$ is stored in the SRAM cells. This is for aligning fixed point of convolution results and the input of the previous layer. Although activations are in the analog domain, we must consider their effective fixed-point representation. The scaling factors vary over different training dataset and different layers.

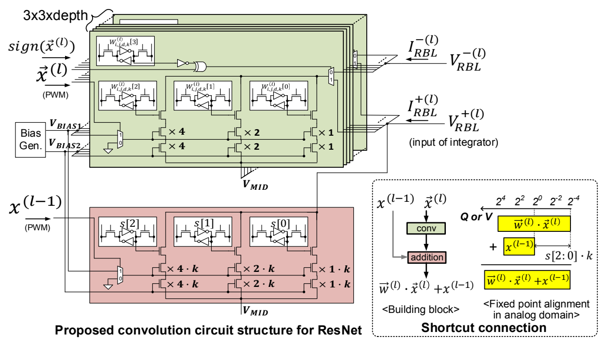


Fig. 4. Circuit structure of in-memory convolution SRAM array cells

B. *Analog integrators*

Analog integrators are composed of complementary folded cascode amplifier with an auto-zeroing scheme to cancel offset. The amplifier holds RBL voltage constant at 0.5V = $V_{DD}/2$, and this further improves the linearity of 3T-readout buffer (Figs. 1 & 4) by holding V_{DS} of the NMOS devices constant.

C. *S/H buffers and ramp voltage generators for BN and ReLU*

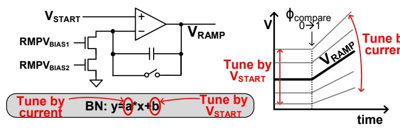
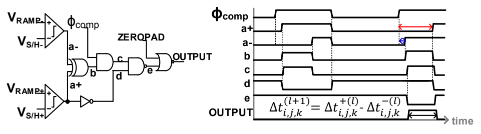


Fig. 5. Ramp voltage generator for BN and ReLU

S/H buffer samples the integrator output voltage on a capacitor after the integrated voltage is settled. The ramp voltage generator in Fig. 5 is used for non-linear voltage (BN+ReLU) generation. The slope of the ramp signal is determined by DC current level, which corresponds to the gain of the BN function. In addition, the bias of BN can be tuned with V_{START} which is the starting voltage level of V_{RAMP}. A pair of the ramp voltage generators share the DC current level (BN gain) but have separate bias voltage V^{+}_{START} and $V^{-}_{START} = 1\text{V} - V^{+}_{START}$, allowing for identical BN to be applied separately to positive and negative convolution results. Continuous comparators evaluate $V^{+}_{RAMP}/V^{-}_{RAMP}$ against buffered voltage pairs, generating a rising edge when V^{+}_{RAMP} and V^{-}_{RAMP} cross the buffered voltage pairs. Finally, ReLU is performed by passing these pulses through logic shown in Fig. 6(a). In the waveform of Fig. 6(b), the final output pulse is generated when $\Delta t^{+(l)}_{i,j,k} \geq \Delta t^{-(l)}_{i,j,k}$, with pulse width of $\Delta t^{(l+1)}_{i,j,k} = \Delta t^{+(l)}_{i,j,k} - \Delta t^{-(l)}_{i,j,k}$. If this inequality is not met, the output pulse width is zero (ReLU).



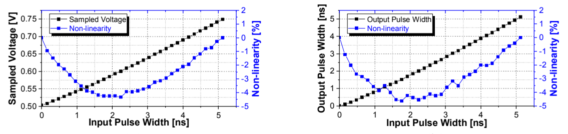
(a) (b)

Fig. 6. (a) ReLU structure and (b) output waveform

IV. Performance Evaluation

A. *Linearity of a single hidden layer*

Nonlinearity of a single layer including in-memory convolution SRAM arrays, integrators, S/H buffers, ramp generators, and BN+ReLU block is simulated in transistor-level SPICE simulation. Thanks to the constant RBL voltage and stacked readout buffers, nonlinearity incurred in convolution SRAM arrays and analog integrators is limited to 4.5% in the worst case (Fig. 7(a)). Nonlinearity in the voltage-to-output



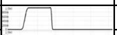
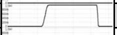
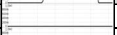
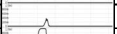
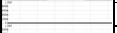
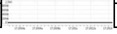
(a) (b)

Fig. 7. Linearity simulation result of (a) input pulse to voltage and (b) input pulse to output pulse

pulse domain is negligible compared to the nonlinearity from integrators and hence total nonlinearity in a single layer (input pulse to output pulse) is <4.8%.

B. *Multi-layer verification*

Multi-level operation is verified by co-simulation using transistor-level SPICE simulation of analog layers, and VCS for synthesized logics, simultaneously. A sample image is loaded to the input image buffer and the output pulse width of each layer is compared with output features from Matlab to ensure correct functionality. In Fig. 8, transistor-level SPICE simulation result of average pooling and FC layer match well with the output feature obtained from Matlab.

Class	SPICE sim. waveform	Pulse width [ps] (SPICE)	Normalized pulse width (SPICE)	Normalized output feature (Matlab)
0		302.81	0.512	0.494
9		0.00	0.000	0.000
8		**591.83**	**1.000**	**1.000**
7		0.00	0.000	0.000
6		0.00	0.000	0.000
5		109.10	0.184	0.162
4		373.00	0.630	0.635
3		0.00	0.000	0.000
2		0.00	0.000	0.000
1		0.00	0.000	0.000

(a) (b)

Fig. 8. (a) A sample SVHN image and (b) Transistor-level SPICE simulation result of average pooling + FC layers and comparison with

After validating individual components in SPICE simulations, we employ Verilog-A models of analog components to reduce simulation time and fully verify the wiring and timing of the full system with all layers.

C. *Analysis on noise and dynamic range*

An effective bit-precision of activations in the pulse-width domain can be calculated from the signal voltage range and noise level observed from analog blocks (or from pulse-width range and jitter). Noise statistics are estimated using transient noise simulations of a transistor-level SPICE netlist (Table 1).

Table 1: RMS noise from transient noise simulation

Noise source	RMS noise [mV]	RMS Jitter [ps]
SRAM array + Integrator	0.8838	18.1002
S/H buffer	0.7976	16.3345
Ramp generator	1.0787	22.0921
Comparator	0.4966	10.1700
Total	**1.6815**	**34.4373**

$\log_2\left(\frac{250mV}{1.6815mV}\right) \approx 7.22\,b$, which ignores dynamic range (DR) reduction after summation of positive and negative convolution values. In Fig. 9, an output pulse width of a layer $\Delta t^{(l+1)}_{i,j,k}$ (= $\Delta t^{+(l)}_{i,j,k}$ -$\Delta t^{-(l)}_{i,j,k}$) is represented in binary format to visualize effective bit precision of analog values. For the case of Fig. 9 (a), there is no additional DR loss except the loss from noise. On the other hand, the case shown in Fig. 9 (b) incurs additional DR loss after summation when $\Delta t^{+(l)}_{i,j,k} - \Delta t^{-(l)}_{i,j,k}$ are small. During training, the DR reduction after summation of positive and negative parts is modeled for each layer to minimize classification accuracy

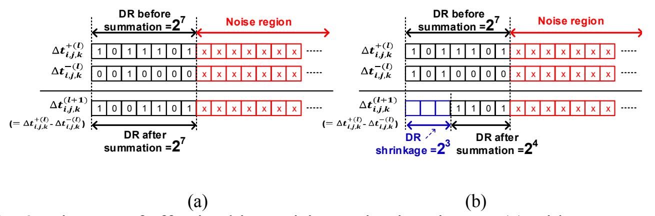


Fig. 9. Diagram of effective bit precision activations in case (a) without DR

loss. Overall, DR reduction varies from 2^0 to 2^4 among different layers, which corresponds to an effective bit loss from 0 to 4 bits. Including noise level and value reduction together, effective bit precisions of activations vary from 3 to 7 bits among different layers.

D. Accuracy Evaluation

To evaluate classification accuracy of the proposed accelerator, a Matlab model is designed based on the circuit structure including separate accumulation of positive/negative convolution results. Fixed-point activations are truncated after convolution based on dynamic range reduction of each layer. In addition, Gaussian noise in Section IV.C is added in each layer during training and testing and compared with truncation-only cases. SVHN and CIFAR-10 data sets are used for training and inference. From Fig. 10, with 3~7 effective-bit activation and 4b- weight, the proposed accelerator achieves 94.0% and 80.9% accuracy on SVHN and CIFAR-10 datasets, respectively. Accuracy degradation occurs due to both finite bit precision and circuit noise (Fig. 10). Comparing to identical noise conditions, but floating precision for both activations and weights, accuracy degradation is 1.6% and 3.9%. Compared to the case with identical bit precision without noise, the accuracy degradation is 3.2% and 5.0% on SVHN and CIFAR-10, respectively. Notice that using binary weights incur severe accuracy loss for the evaluated ResNet.

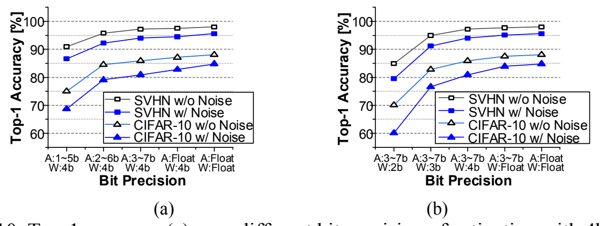


Fig. 10. Top-1 accuracy (a) over different bit precision of activation with 4b-weight and (b) over different bit precision of weight with 3~7b-activation

E. Energy breakdown

AA-ResNet energy consumption is simulated by SPICE for analog cores, and Prime Time PX with synthesized digital logics on actual image input vectors. Analog cores consume 94.8% of total energy (Fig. 11(a)), mostly by amplifier bias currents, while remaining energy is consumed by digital and analog peripherals. The energy consumption distribution among different layers mainly depends on layer dimension (Fig. 11 (b)). The 16th layer dominates (186nJ) due to many parallel

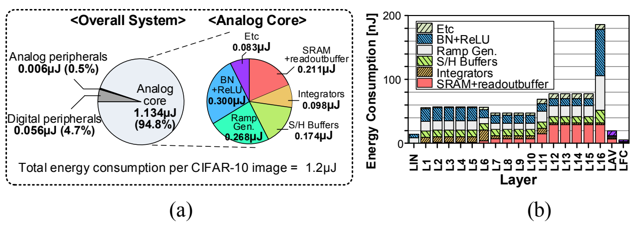


Fig. 11. (a) Energy breakdown of AA-ResNet accelerator and (b) energy distribution over different layers.

Table 2: Comparison with state-of-the-art

		This work[a]	[5]	[6]	[1]	[2]
Technology		**28nm**	28nm	65nm	65nm	55nm
Area		**11.9**[b]	5.95	17.6	1.44	3.4
Supply [V]		**1**	0.8 / 0.6	0.68 / 0.94 / 1.2	0.675 ~ 0.925	0.4 ~ 1
Power [mW]		**389**	0.899 / 0.094	21.8[c]	1.36[c]	Max 0.69
Core Circuit Type		**Analog**	Mixed-signal	Mixed-signal	Mixed-signal	Mixed-signal
Algorithm		**ResNet**	Binary CNN	Binary CNN	SVM	Stochastic RL
Dataset		**CIFAR-10 / SVHN**	CIFAR-10	CIFAR-10 /SVHN /MNIST	MIT-CBCL face detection data	Online learning from ultrasonic sensors
On-chip Memory [kB]		**Image: 6.75 Weight :32.2**	328	295	16	N/A
Accuracy [%]		**80.9 / 94.0**	86	84/ 94/ 98.6	96	N/A
MAC precision	Weight	**4b**	1b	1b	8b	6b
	Input	**Pulse width (eff.3~7b)**	1b	1b	8b	6b
1b-scaled[d] MAC performance [TOPS]		**33.1~77.2**	0.478 / 0.072	18.876	0.272[c]	0.078[c]
1b-scaled[d] MAC Efficiency [TOPS/W]		**85.1~198.5**	532 / 772	866	200[c]	112.32[c]
# of 1b-scaled[d] MAC Ops per CIFAR-10 image		**237.47M**	818.52M[c]	1441.47M[c]	N/A	N/A
CIFAR-10 image throughput rate [image/sec]		**325,520**	237 / 36	390	N/A	N/A
Energy per CIFAR-10 image [μJ/image]		**1.2**	3.79 / 2.61	3.55	N/A	N/A

a. Simulation based results b. Layout based c. Calculated based on other reported values
d. 1b-scaled: (weight precision) × (input precision) × original value

output channels that are connected to average pooling layer. Overall energy consumption is 1.2μJ per inference, which is 3× smaller compared to state-of-the art [5,6], achieved by avoiding ADCs and DACs.

V. Conclusions

In this paper, we proposed a design of the multi-bit precision AA-ResNet accelerator performing all operations in the analog domain to overcome DAC/ADC overhead. Evaluation showed 1.2 μJ energy consumption for an inference of the SVHN / CIFAR-10 data set, and 325,520 image/s of inference rate. We further analyzed nonlinearity in convolution, effective bit precision of activations from noise and DR shrinkage, and accuracy including the effects of noise and bit precision.

Acknowledgments

This work is supported in part by the Center for Applications Driving Architecture (ADA) of Joint University Microelectronics Program led by SRC and co-sponsored by DARPA.

References

[1] S. K. Gonugondla, et al., "A 42pJ/decision 3.12TOPS/W robust in-memory machine learning classifier with on-chip training," in *2018 IEEE ISSCC*, 2018, pp. 490–492.

[2] A. Amaravati, et al., "A 55-nm, 1.0–0.4V, 1.25-pJ/MAC Time-Domain Mixed-Signal Neuromorphic Accelerator With Stochastic Synapses for Reinforcement Learning in Autonomous Mobile Robots," *IEEE JSSC*, vol. 54, no. 1, pp. 75–87, Jan. 2019.

[3] R. S. Amant, et al., "General-purpose code acceleration with limited-precision analog computation," *in 2014 ACM/IEEE 41st ISCA* , 2014, pp. 505–516.

[4] K. Jia, et al, , "Calibrating Process Variation at System Level with In-Situ Low-Precision Transfer Learning for Analog Neural Network Processors," in *2018 55th ACM/ESDA/IEEE DAC*, 2018, pp. 1–6.

[5] D. Bankman, et al., "An Always-On 3.8μJ/86% CIFAR-10 Mixed-Signal Binary CNN Processor With All Memory on Chip in 28-nm CMOS," *IEEE JSSC*, vol. 54, no. 1, pp. 158-172, Jan. 2019.

[6] H. Valavi, et al., "A 64-Tile 2.4-Mb In-Memory-Computing CNN Accelerator Employing Charge-Domain Compute," *IEEE JSSC*, vol. 54, no. 6, pp. 1789–1799, Jun. 2019.

[7] K. He, et al., "Deep Residual Learning for Image Recognition," *arXiv:1512.03385 [cs]*, Dec. 2015.

[8] K. He, et al., "Identity Mappings in Deep Residual Networks," *arXiv:1603.05027 [cs]*, Mar. 2016.