IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 June 2023, accepted 5 August 2023, date of publication 15 August 2023, date of current version 18 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3305432

RESEARCH ARTICLE

AIMC Modeling and Parameter Tuning for Layer-Wise Optimal Operating Point in DNN Inference

IMAN DADRAS[1,3], GIUSEPPE M. SARDA[2,3], NATHAN LAUBEUF[2,3], DEBJYOTI BHATTACHARJEE[3], AND ARINDAM MALLIK[3]

[1]Intelligent Materials and Systems Laboratory (IMS Laboratory), Institute of Technology, University of Tartu, 50411 Tartu, Estonia
[2]ESAT-MICAS, KU Leuven, 3000 Leuven, Belgium
[3]imec, 3001 Leuven, Belgium

Corresponding author: Iman Dadras (iman.dadras@ut.ee)

This work was supported in part by the European Research Council (ERC) under Grant 101088865; in part by the European Union's Horizon 2020 Research and Innovation Program under Grant 857263 and Grant 101070374; in part by the Flanders AI Research Program, Katholieke Universiteit (KU) Leuven, Estonian Research Council, under Grant 1084; and in part by the Estonian Centre of Excellence in ICT Research and Doctoral School of the European Institute of Innovation and Technology (EIT) Manufacturing, funded by the European Union (EU).

ABSTRACT Analog in-memory computing (AIMC) has been utilized in convolutional neural networks (CNNs) edge inference engines to solve the memory bottleneck problem and increase efficiency. However, AIMC analog-to-digital converters (ADCs) restricted resolution imposes quantization of output activations that can reduce the accuracy without meticulous optimization. A study conducted output quantization calibration and obtained configurations with which low-resolution ADCs did not affect the accuracy. The configurations were layer-specific. Therefore, a real-time quantization adjustment was required. AIMC output quantization is adjusted by controlling analog gain entangling it with analog parameters and nonlinear functions. AIMC dynamic output quantization control without interrupting its operation has been an unsettled problem until now. This paper introduces a technique for imposing output quantization configurations obtained from calibration processes on AIMC through circuit parameters setup. The technique permits on-the-fly quantization adjustments enabling layer-wise calibration that increases achievable network accuracies on AIMC platforms. As a case study, we deployed the method on the AIMC macro of an artificial intelligence (AI) inference engine SoC platform with a RISC-V processor and hybrid DIgital-ANAlog accelerators (DIANA). We related its controllable circuit parameters with the quantization configuration in a look-up table. This case study has noteworthy side benefits in identifying platform limitations due to nonlinearities and design imperfections. These limitations are investigated, and design advice that is transferable to future AIMC designs is provided to avoid imperfections such as mismatch, bias voltage drop, and interconnect delay. In addition, the study of output quantization from different levels of abstraction leads to design guidelines to facilitate dynamic quantization control during the application phase.

INDEX TERMS Analog in-memory computing (AIMC), deep neural network (DNN), convolutional neural network (CNN), application-specific integrated circuit (ASIC), artificial intelligence hardware acceleration, modeling, characterization, quantization.

I. INTRODUCTION

Artificial intelligence (AI) has been recognized as "the new electricity" for its potential to revolutionize the industry [1]. It demonstrated vast applicability in various domains, from natural language processing (NLP) [2], [3], image classification and object recognition [4], [5] to stock market trading [6], [7]. In computer vision applications, convolutional neural networks (CNNs) showed outstanding ability due to their spatial kernels [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Vivek Kumar Sehgal.

CNNs require a high computational load indicating parallelization possibility. A vast effort exists to fully harness parallel computing architectures for higher efficiency [9]. GPUs [10], FPGAs [11], and application-specific integrated circuits (ASICs) [12] can leverage a higher level of parallelism than conventional processors. Nevertheless, the development cost and time, along with the performance and efficiency, increase from GPUs to ASICs. In specific applications where energy and speed constraints are limited, ASICs are the only viable solution. Research is moving towards more efficient and accurate systems to accelerate CNN at the edge. A very promising acceleration method consists of computing MAC operations in the analog domain directly in memory cells.

The unmet need for efficiency by digital computing in edge devices caused a resurgence in analog computing. Analog computing can be exploited to increase efficiency at the cost of accuracy reduction [13]. The analog domain represents a number by a single signal without resolution restriction and performs MAC operations with one device per input [14]. This characteristic makes analog accelerators strong candidates for applications with limited energy budgets. Analog in-memory computing (AIMC) combines analog efficiency with in-memory computing (IMC) to overcome the memory wall bottleneck by merging processor and memory units, pushing energy efficiency by orders of magnitude [15]. However, device nonlinearities, mismatches, and noise impact the analog computation's accuracy. On top of that, analog circuits lack flexibility as their behavior, as well as the data flow, are fixed at design time. Thus, an analog macro engineered for a particular workload or required precision may not be efficient when the requirements change. CNNs show high resilience to errors and reduced parameter precision but with limitations and different output sensitivity to different layers in the network [16]. These observations promise high accuracy and efficiency with a hybrid digital-AIMC accelerator that can split the workload in agreement with accuracy and efficiency requirements.

Along this line, the DIANA SoC [17] integrates three cores in a complete system: a RISC-V CPU, a digital accelerator, and an AIMC-based accelerator [18]. The RISC-V processor controls the system and allocates the workload among the two accelerators. The AIMC accelerator is designed to achieve high utilization and efficiency with moderate accuracy for layers with a high number of channels. Layers with fewer parallelization possibilities and more severe sensitivity to accuracy are assigned to the digital accelerator. This structure allows DIANA to achieve high efficiency without a decrease in network accuracy by allocating the execution of different layers in the digital or analog core. DIANA's AIMC macro is used as a case study in this paper while analyzing the applicability of the technique presented here for other AIMC platforms.

The AIMC paradigm can be implemented with various cell technologies. Non-volatile memories (NVMs) form dense crossbar arrays to perform parallel MAC operations [19]. NVMs are enabled with emerging technologies such as resistive random access memory (RRAM), [20] phase-change memory (PCM) [21], and spintronics [22]. However, NVM's technological drawbacks, like read and write non-idealities [23], low reliability, and temperature dependency [19], make the design of AIMC macro challenging and motivate designers towards more standard technologies such as CMOS-based SRAMs [24], [25], [26], [27]. This last type of cell is the one used in DIANA.

ADCs are essential parts of AIMCs. They convert voltages proportional to the MAC operation results to digital data. Consequently, they quantize the output activations to fewer levels than the MAC operation result requires. At this stage, careful calibration is required to reach an accuracy comparable with the baseline [28]. The quantization configurations are set by selecting circuit analog parameters. A methodology is missing to link the quantization configurations obtained from thorough optimizations [28] to physical circuits. This method should determine the mechanism by which the quantization parameters, obtained at the software level, are imposed on AIMC. It can be a modeling technique that connects the circuit-level parameters to quantization configurations.

There are some efforts on AIMC modeling. Spetalnick et al. [29] combine system and circuit models and simulations to analyze the SRAM AIMC design space and spot efficiency gains and losses. Kein et al. [30] integrate an AIMC cell model to gem5-x simulator for full-system simulation in the design phase. However, to the authors' knowledge, there is no model that selects the circuit parameters according to output quantization calibration.

The lack of a quantization imposition technique has hindered the optimal use, in terms of computation accuracy, of the AIMC. Moreover, non-idealities of the analog circuit should be known for a correct accuracy evaluation and compensation strategy. The characterized non-idealities can also be mitigated in future AIMC designs.

In this paper, we contribute to the AIMC paradigm with the following developments:

- A technique is developed to link the AIMC circuit analog parameters to its output quantization. It allows on-the-fly implementations of layer-wise quantization calibrations like [28] on AIMCs, significantly increasing the achievable classification accuracy on the platform. The study of the output quantization mechanism also gives guidelines for AIMC designs to better exploit the ADC output range.
- The method is applied to DIANA's AIMC macro as a case study. As a result, a linear model of the DIANA's AIMC is developed. The model can translate the quantization parameters obtained from calibration or training to controllable circuit parameters. The Controllable parameters are an external bias voltage and a programmable PWM unit time. Thus, the quantization configuration is set by adjusting these parameters in

a real-time manner. A look-up table summarizes the model and eases its application.
- Important non-idealities for accuracy are characterized and modeled. Methods are proposed to avoid, compensate, and in future designs, improve non-idealities.

Section II discusses the technique required to impose the quantization parameters on AIMC output. Section III briefly introduces the DIANA's AIMC macro. This SoC is used in the rest of the paper as an example to apply the suggested method. Section IV presents the experimental results. Section V implements the method on DIANA as a model, and section VI concludes the paper.

II. AIMC OUTPUT QUANTIZATION CONTROL THROUGH CIRCUIT PARAMETERS

Quantization is used to reduce the computational cost of CNN and match the edge applications restrictions [31]. The quantization is applied to input activations and weights to reduce their bit precision to b_a and b_w, respectively. The output of a convolution should ideally be coded with l_o levels:

$$l_o = (2^{b_a} - 1) \times (2^{b_w} - 1) \times n_a + 1 \tag{1}$$

where n_a is the number of accumulations in the MAC operation. Some works reported re-quantization at the activation layer to directly produce quantized input activations for the next layer [32], [33]. This output quantization is necessary for AIMCs due to their restricted ADC resolutions. As an example, a layer with 16 7-bit input channels and 3 by 3 kernels with 2-bit weights would need 11-bit ADCs according to (1) that are prohibitively power-hungry [34].

The output quantization should be calibrated according to the CNN model and activation distribution in each layer or even channel in order to achieve high network accuracy [31]. Calibration is the process of determining the clipping range $[\alpha, \beta]$, a range that data out of it will be mapped to its limits before quantization.

There are different calibration approaches. A straightforward way is to set the clipping range to the maximum and minimum values of the to-be-quantized data. This method increases the dynamic range and reduces the resolution. So, other approaches, like using percentile [35] and optimizing the data loss [36], take a smaller range to increase resolution mitigating the effect of outliers. Another method is to learn the quantization parameters during the training [28], [33].

Laubeuf et al. [28] conducted research on output quantizations with AIMCs. They showed improvement in accuracy with a layer-wise output quantization calibration over the network-wide counterpart. Their work used a DIANA-like AIMC macro and did a Pytorch simulation to show dynamic output quantization control with adjusting pulse width modulation (PWM) unit time. Accuracies on par with the baseline are achieved for Resnet-20 on CIFAR-10 and Resnet-18 on ImageNet after output quantization optimization. The paper showed the importance of AIMC output quantization calibration and its enforcement possibility via circuit parameters selection.

However, results from [28] are not directly applicable to AIMCs, because first, their assumptions in the simulation are different from the actual chip structure. Second, they only analyzed the unit time adjustment for a fixed bias voltage value. The bias voltage can be used for fine-tuning the quantization parameters in DIANA, as its values are continuous, unlike discrete unit time values. Also, using the combination of bias voltage and unit time increases the designers' degree of freedom, so they can optimize the chip also for power and performance vs. accuracy [17], for example. And third, the nonlinear behavior and second-order effects of AIMC are neglected in their linear Pytorch model. There is a gap between their high-level study and the low-level AIMC circuits. To fill the gap, it is required to investigate the quantization from both network and circuit-level perspectives to unveil the output quantization mechanism in AIMCs.

The ADC thresholds in AIMCs are usually uniform and symmetric. Therefore, the output quantization is also uniform and symmetric from the high-level network perspective. Under this assumption, the scale factor is defined as a floating point number with which the data multiplies before discretization. As it should convert a data from $[-\beta, \beta]$ to $2^b - 1$ levels, scale factor can be calculated with:

$$S = \frac{2^b - 1}{2\beta} \tag{2}$$

in which b is the quantization (ADC) bitwidth. The quantization output (O) with this scale factor is then obtained as:

$$O = int(S \times O_{mac}) \tag{3}$$

where O_{mac} is the MAC operation result.

From the circuit perspective, there is a gain (A_v) that determines the voltages at the ADC input (V_{adc}) proportional to the MAC operation result.

$$V_{adc} = A_v \times O_{mac} \tag{4}$$

This voltage is then converted to digital at the ADCs according to the ADC quantization steps (δ_{adc}).

$$O = int(\frac{V_{adc}}{\delta_{adc}}) \tag{5}$$

Comparing (4) and (5) with (3) the scale factor from the circuit perspective is

$$S = \frac{A_v}{\delta_{adc}} \tag{6}$$

Therefore, there are two ways to control the AIMC output quantizer, adjusting ADC quantization steps or AIMC analog gain. ADC quantization steps are usually optimized and fixed according to the voltage dynamic range and ADC bit precision. On the other hand, it is preferred to support quantization dynamic control via analog gain.

To control the quantization via analog gain, one or more parameters to change the analog gain have to be devised during the design. These parameters should be able to change easily to set different quantization set-ups

while executing different layers. In addition, the controllable analog gain range should be adequately wide to support different possible quantization configurations. The effect of the gain-controlling parameters on other performance figures should also be taken into account. Because the gain-controlling parameters have effects on other performance figures, it is beneficial to have more of these parameters. This gives more flexibility to optimize the affected performance by tuning the correct parameter for specific applications. Moreover, the relationship between the parameters and quantization should be defined clearly. Due to analog devices' higher-order effects and non-idealities, it usually cannot be done analytically. Thus, a measurement and characterization campaign may be needed.

The parameters that can be used for analog gain control are as diverse as AIMC structures. For example, in the memristor-based [37], [38] architectures, the memristor value and the DAC gain are related to the analog gain. Memristor values are programmable, and there is a possibility to consider a scale on them. In designs with PWM DAC [39], [40], the PWM unit time is a potential parameter to be easily programmed to control the gain and modulate the quantization. In DIANA, there are PWM unit time and current limiting transistor bias voltage for this purpose. The fact that DIANA SoC uses a combination of two parameters that one, e.g. bias voltage, has a non-linear relationship with the analog gain, and the other is a commonly used parameter in time-domain AIMC makes DIANA a good example to be a case study in this paper.

This section developed a technique for controlling the AIMC output quantization. The relationship between gain and quantization was defined, and it was shown that the analog gain is preferred to set the quantization parameters. Hence, gain-controlling parameters should be utilized in AIMC to dynamically adjust quantization for each layer. In the next section, DIANA's AIMC will be introduced to be used as an example for this technique. The parameters involved in its analog gain will be identified, and their relationship with quantization parameters will be studied.

III. DIANA'S AIMC MACRO

This section first briefly describes the AIMC macro implementation integrated into DIANA. Then, it focuses on the AIMC output quantization parameters control. It finds the circuit parameters that can modulate the output quantization. The effects of these parameters on quantization will be modeled in the following sections.

A. AIMC MACRO STRUCTURE

The macro is an 1152 × 512 array of analog processing elements (APE). When the macro is fully utilized, 1152 7-bit activations are converted to PWM signals. Then, each is fed to all 512 APEs in a row. APEs multiply activations by ternary weights (+1, 0, -1) stored in two standard 6T SRAM cells. The product is accumulated at summation lines, which connect the APEs in the same column. Finally, 512 6-bit ADCs convert

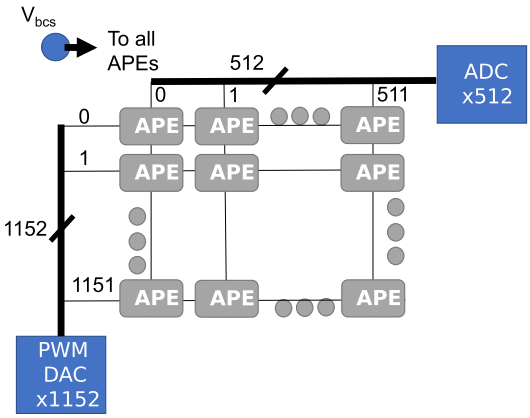


FIGURE 1. Block diagram of the AIMC macro; each of 1152 activations goes to 512 APEs in a row (horizontal lines). Outputs of APEs are accumulated in summation lines (vertical lines).

voltage on the summation lines to digital. Fig. 1 illustrates the simplified diagram of the AIMC.

Fig. 2 shows the transistor-level schematic of an APE. PWM DAC produces two active-low signals, Act- and Act+. Each signal is used to modulate the activations with the corresponding signs. Act+ and Act- are connected to the source of two transistors. Two SRAMs store weight (W+) and negated weight (W-); each is connected to the gate of two transistors with different PWM signals at sources. The drains of the transistors with concordant sign signals (W+ and Act+ or W- and Act-) go to the positive summation line; Sum+, and the other two (discordant signs) go to the negative summation line; Sum-.

At the beginning of a processing cycle, summation lines are pre-charged to VDD. For non-zero weights and activations, one transistor turns on and determines the connected summation line and, consequently, the product sign. PWM width dictates the magnitude of the product. Outputs of APEs are in the form of current. They discharge the connected summation line proportional to the pulse width. Thus, accumulation is conducted at the summation lines. The readout circuit deals with Sum+ and Sum- as differential signals and sends the results to ADCs.

Two current-limiting transistors connect each APE to the summation lines. It adds more flexibility to the design and mitigates the channel length modulation effect. The bias voltage of these transistors (hereafter V_{bcs} or bias) and PWM unit time determine the quantization parameters and control the resolution and dynamic range of the output activations.

B. DIANA'S AIMC OUTPUT QUANTIZATION

The ADCs' voltage thresholds are fixed. Therefore, output activations (O) are related to the summation line voltages (V_{adc}) as (5). The summation line voltage is proportional to the result of the MAC operation. We define the unit voltage (V_u) as the summation line voltage corresponding to the result of a MAC equal to one. With the assumption that the cell currents are DC, as the summation lines are capacitive,

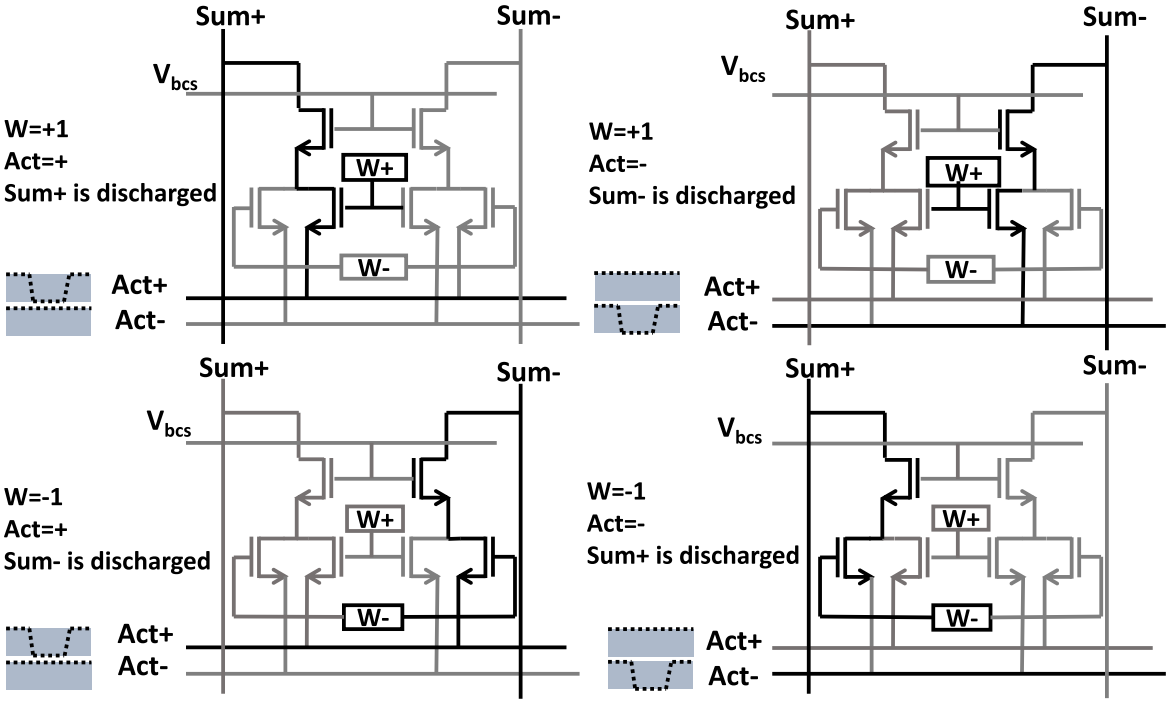


FIGURE 2. AIMC macro transistor-level schematic; active paths are shown with black lines. Activation and weight signs determine the product sign. The result magnitude is controlled by that of the activation.

the summation line voltage is equal to:

$$V_u = \frac{t_u.I_{cell}(V_{bcs})}{C_{line}} \tag{7}$$

I_{cell} is the cell current that is a function of bias voltage. t_u and C_{line} are respectively unit time and summation line capacitance. V_{adc} corresponding to other MAC operation results are proportional to the unit voltage:

$$V_{adc} = V_u.O_{mac} \tag{8}$$

combining (5), (7), and (8), we have:

$$O = int(\frac{t_u.I_{cell}(V_{bcs})}{\delta_{adc}.C_{line}}.O_{mac}) \tag{9}$$

With a comparison between (3) and (9) scale factor is obtained.

$$S = \frac{t_u.I_{cell}(V_{bcs})}{\delta_{adc}.C_{line}} \tag{10}$$

C_{line} and δ_{abc} are fixed. Thus, quantization scale factor control is possible through bias voltage and unit time. The bias voltage is applied to DIANA externally, and the unit time can be changed among 16 values by programming a register runtime.

Unit time and bias voltage have different effects on DIANA performance. So the designer has the freedom to make tradeoffs and optimize these two parameters with respect to each other to achieve the desirable quantization setup and overall performance. For example, unit time controls the AIMC's total cycle time and speed. Unit time affects power consumption more than bias voltage does [17]. However, it is shown in the next section that small unit times may cause scheduling problems. Therefore, it is possible to make tradeoffs on speed, power, and scheduling sanity with these two parameters.

In this section, we introduced the DIANA's AIMC circuit and showed the relation between output quantization configurations and unit time and bias voltage. The AIMC quantization control can now be achieved by a model that connects bias voltage and unit time to the output quantization parameters to enable the implementations of optimization techniques like [28]. The relationship between quantization and circuit parameters is not straightforward as it is nonlinear and correlated. For example, bias voltage affects MOSFET switching time, which changes the effective unit time. Therefore, it is important to obtain the model via experiment rather than theory.

The following section will present the experimental results. These results will be used to develop the model and also to provide guidelines for the best chip setup and design improvements.

IV. EXPERIMENTAL RESULTS

The characterization campaign aims to incrementally model the AIMC behavior in order to connect output quantization to circuit parameters. It also shows the non-idealities. Information on non-idealities can be used for compensation or improvement of the next AIMC generations.

In the experiment setup, DIANA is installed on a custom motherboard. A ZedBoard™ Zynq®-7000 ARM/FPGA SoC development board is connected to the motherboard via FPGA Mezzanine Card (FMC) Low Pin Count (LPC) connector. The ZedBoard performs the DIANA's booting procedure and provides the clock signal. A PC programs and

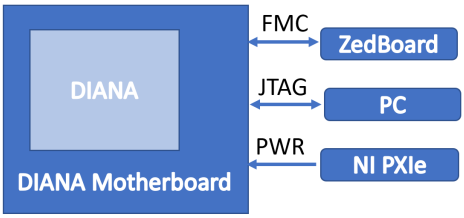


FIGURE 3. Measurement setup diagram; ZedBoard boots the chip, PC programs and reads the results, and NI PXIe powers DIANA.

loads the inputs and weights into the chip through a JTAG interface. The PC reads the results from DIANA through the same interface. The motherboard and DIANA's power, as well as the bias voltage, is provided by two NI PXIe-4145 4-channel source-measure units mounted on a NI PXIe-1088 chassis. Fig. 3 shows the diagram of the experiment setup.

The first investigation examines the accumulation function linearity. The summation linearity is crucial because it allows the AIMC model to break down into the addition of small models of APEs using the additive property.

Then, mismatches between APEs are evaluated. Two design imperfections, bias voltage drop and interconnect delay, are investigated later. These three experiments give chip users guidelines to avoid non-ideality effects and provide chip designers with suggestions for improvement.

The last experiment shows that nonlinearity error is a function of the output rather than the input. This observation is utilized to develop the linear and fine-grain models combination. Eventually, the results of this part are used to plan an experimental exploration of unit time and bias that leads to a linear model in section IV.

A. ACCUMULATION LINEARITY

AIMC performs multiplications inside APEs and accumulations at the summation lines. If the accumulation operation is linear, the AIMC model decomposes to the addition of APEs models by utilizing the additivity property. So, before modeling APEs in section V, we must show that the accumulator is linear.

The number of activations is increased in an experiment to analyze the linearity of addition. Fig. 4 shows the AIMC output for different numbers of activations. In this specific example, non-zero activations are set to 5, while weights are all positive (1). Fixed APEs inputs isolate the experiment from APE's nonlinearities. However, their mismatches still reflect in the results. In each experiment, 128 activations are added. The set of iterations is then repeated for ten values of unit time ([50ns: 140ns, 10ns]). The bias voltage in this experiment is 0.61.

Fig. 4 visualizes the linearity of the accumulation operation up to output saturation. As the ADCs output range is [-31: +31], the AIMC output is saturated on 31. A part of the nonlinearity is rooted in the APEs mismatches, as shown in the following subsection.

Thus, the rest of the experiments try to demystify the APEs assuming accumulation at the summation lines is linear.

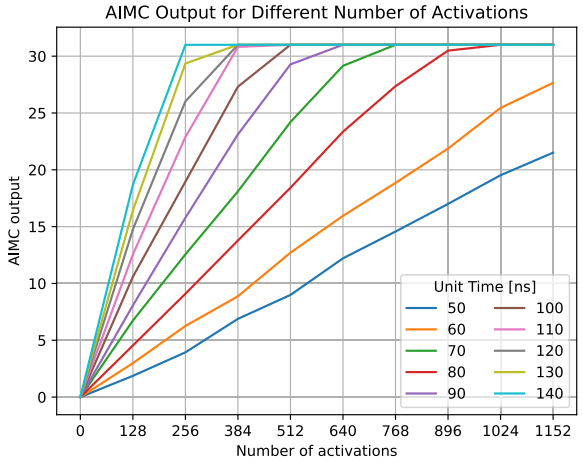


FIGURE 4. Addition linearity; the number of activations and addends is linearly proportional to the output.

B. APE MISMATCH

Dealing with device mismatches is a burdensome challenge for analog designers, and DIANA's AIMC is no exception. Mismatches occur due to differences in devices that are designed identically. Coping with mismatches after tape-out is not possible. Their analysis needs a statistical approach that is out of the scope of this paper. Fig. 5 illustrates the mismatches as differences in the AIMC output for different APEs with the same activation and weight. Here, 20 APEs are examined in each experiment, as for a single APE, the output might be weak and noisy.

Mismatches should be avoided as much as possible to achieve good linearity. The best stage to deal with mismatches is during the layout design. Matching guidelines can be found in analog layout books including [41].

C. BIAS VOLTAGE DROP

Experiments show that, for bigger workloads, the sensitivity of the summation lines with higher indices decreases. An experiment isolated the effect by feeding the whole array with equal activations and weights. Fig. 6 illustrates the output of the summation lines. The output decreases for summation lines with higher indices.

The loading effect is related to the drop in the current limiter transistors bias voltage in the AIMC macro. Although MOSFET gates do not ideally draw any current, an array of more than one million long-channel transistors introduces big gate current leakage in scaled FDSOI technologies due to thin oxide and direct tunneling effect [42].

The design level fix is to add more bias voltage contact on the die in distant locations. At the application level, it is better if high utilization is avoided. Otherwise, post-processing compensation necessitates for chip users.

D. INTERCONNECT DELAY

The AIMC output is quantized in the range [-31,31]. So, the output is saturated when it reaches 31. However, experiments show that the outputs become saturated sooner when the

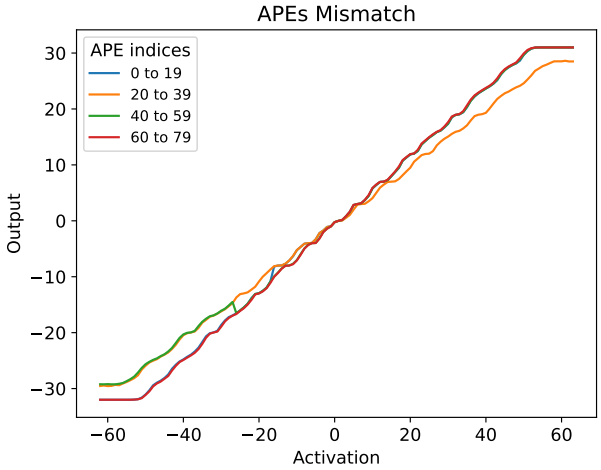


FIGURE 5. APE mismatch; the outputs of the experiments with the same inputs but on different APEs are slightly off due to the mismatch.

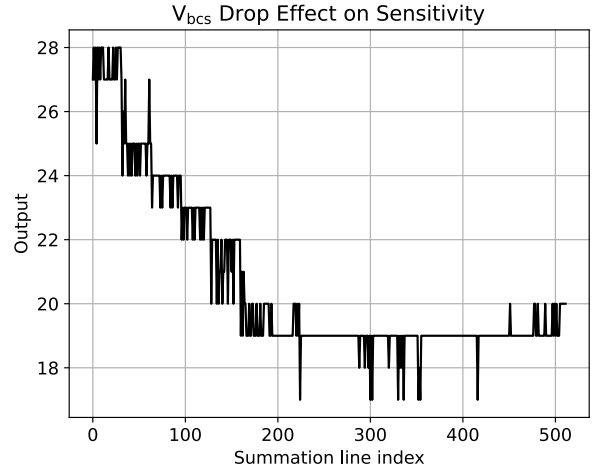


FIGURE 6. Current limiting transistor bias voltage (V_{bcs}) drop causes a sensitivity reduction for high-index summation lines.

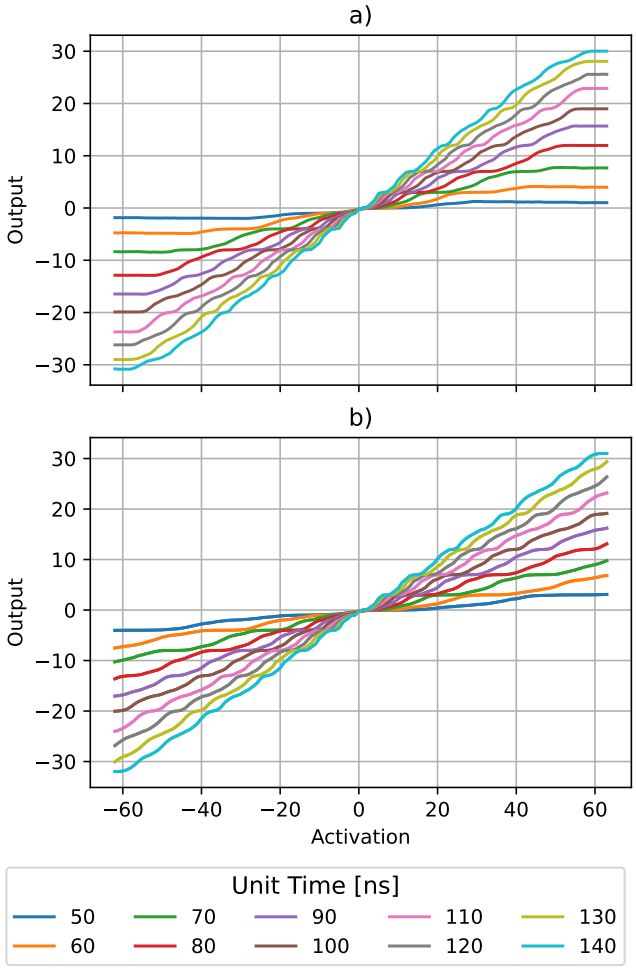


FIGURE 7. Large outputs linearity; a) early saturation happens for low-index APEs, and b) high-index APEs stay linear in [-31, +31].

low-index APEs are utilized. The early saturation is shown in Fig. 7. The effect is only visible for low-index APEs and is proportional to unit time; the output becomes saturated in lower values for smaller unit times.

Interconnect delay causes early saturation. The control and timing unit (CTU) is located at the bottom of the AIMC array, closer to the high-index APEs. There is an interconnect delay (t_{id}) for the PWM DAC enable signal from CTU to the top of the AIMC array where low-index APEs and their DACs are located. Thus, these DACs start their operations later, leading to a delay in low-index APEs' PWM signals. When unit time decreases, the total AIMC cycle time decreases proportionally while the delay remains constant. So, for small unit times and big activations, a big part of the PWM signal does not overlap with the AIMC active time and is ineffective. That leads to an early saturation. Fig. 8 shows the timing diagram in a) high- and b) low-index APEs. Due to interconnect delays, for big activations, the end of the PWM signal does not overlap with the AIMC active time. So, the output is saturated as increases in activation just increase the futile part of the PWM signal.

Hopefully, it is possible to lengthen the AIMC active window by setting DIANA's control registers. So, the users should make sure to set these registers correctly while using small unit times. However, some buffers on control signals improve the design, so increasing active window time will be unnecessary. This will improve the chip's speed and power.

E. ERROR AS A FUNCTION OF OUTPUT

Fig. 9 shows that error is a function of the output for different unit times and bias voltages. It does not include errors rooted in voltage drop or interconnect delay. It is utilized in the next section to develop linear and fine-grain model serialization.

This section delivered observations used in the next section to develop a linear model of AIMC, the non-idealities that one should take into account during the application of the chip, and an observation that will be used for model nonlinearity adjustment. The following section will model the AIMC macro with a look-up table that translates the quantization parameters to unit time and bias voltage.

V. MODEL

This section proposes an approach to include nonlinearities in a linear AIMC model. Then, we present a model for the

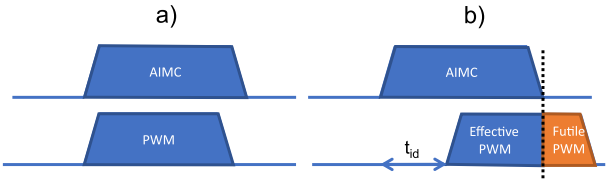


FIGURE 8. Timing diagram of PWM and AIMC in a) high- and b) low-index APEs. PWM signal slides off the AIMC active time due to interconnect delay and causes early saturation.

DIANA's AIMC macro that connects its circuit parameters with scale factors and quantization configurations. At the end of the section, the model is used to evaluate the DIANA's AIMC in the implementation of output quantization calibrations from the literature.

A. NONLINEARITY ADJUSTMENT FOR LINEAR MODEL

In the previous section, the error of a linear model was presented as a function of the output. Thus, one can model the AIMC macro as a linear and nonlinearity adjustment model in series if voltage drop and interconnect delay errors are neglected. This approach is depicted in Fig 10.

The nonlinearity adjustment is sample-specific and cannot be achieved generally.

B. MODEL PRESENTATION

The characterization was planned as an exploration of DIANA's operating points; bias voltage and unit time as their effects on quantization are shown in Section II. Bias voltage has values between 0.5V to 0.8V with steps of 0.01V. The DAC conversion unit time ranges between 50 ns to 200 ns with 10 ns step granularity. The experiment is conducted on 40 APEs with the highest indices in each summation line. APEs with high indices are selected to avoid the interconnect delay effect. Forty APEs are used to average out the spatial variations due to mismatches while avoiding the voltage drop effects resulting from overloading. These effects should be eluded by the user with suggestions provided in the previous section or compensated in the post-processing.

The output quantization is fixed to 6-bit symmetric and uniform by design. Only the quantization scale factor and clipping range can be set during the application phase. We couple each combination of unit time and bias voltage with a quantization scale factor. For this purpose, activations are swept from -63 to +63 for Each operating point. The output is normalized by the number of APEs. A line is fitted into the normalized output versus activation graph using linear regression. As weights are one, the slope of the fitted line is the scale factor. The linear regression standard error is also calculated and provided as a measure of accuracy in the look-up table. The user can apply this error along with the observations from the previous section to favor one combination over others.

If the clipping range is obtained from quantization calibration, the scale factor can then be converted to clipping range

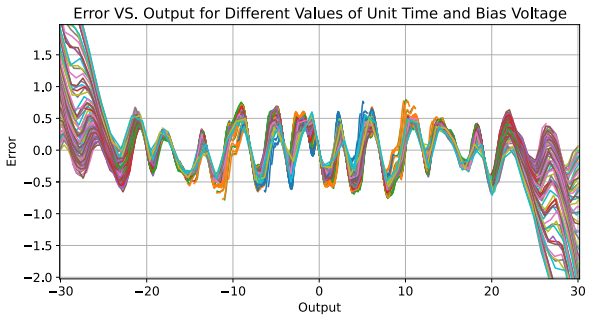


FIGURE 9. Error is a function of the expected linear output for different unit times and bias voltages.

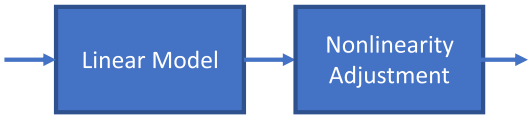


FIGURE 10. Linear model can be followed with a fine-grained sample-specific model for fine adjustment.

$[-\beta, \beta]$ by the following equation.

$$\beta = \frac{2^6 - 1}{2S} \tag{11}$$

The minimum value for the scale factor in the look-up table is 0.00044, and the maximum is 0.0477. This means clipping ranges between [-660, 660] and [-71590, 71590] are possible. MAC operations results conducted on DIANA's AIMC can be in the range of [-72576, 72576] (1152×63) that almost fit inside the wider clipping range. The smallest possible step size equals 20.6.

TABLE 1 is a part of the model look-up table. Let's assume the calibration leads to a layer with a scale equal to 0.01. If the layer can be fitted in high-index APEs, interconnect delay is insignificant. Thus, the user can select combination 1, which has a smaller unit time and consequently less power consumption and higher speed [14]. However, small unit times should be accompanied by large AIMC active windows for layers that utilize the low-index APEs to mitigate the interconnect delay error. Combination 5 is a good choice to reduce the interconnect delay errors at the cost of lower speed and higher power consumption for layers that use low-index APEs if increasing the AIMC active window is not an option.

C. DIANA'S OUTPUT QUANTIZATION CAPABILITY FOR ACCEPTING CALIBRATED PARAMETERS

To validate our method, we use quantization parameters from [28] to execute ResNet-20 for CIFAR10 and ResNet-18 for ImageNet on AIMC. These parameters are obtained via two quantization calibration approaches; network-wide and layer-wise.

In the network-wide quantization, they selected a single scale factor for the whole network by which the network accuracy is maximized. The scale factors were 0.031 and 0.018 for Resnet-20 and Resnet-18, respectively. These scale factors were implementable by DIANA with our model. Nevertheless, they showed that with layer-wise quantization,

TABLE 1. Model look-up table example.

Nr.	Scale	Bias Voltage	Unit Time [ns]	Standard Error
1	0.010264	0.76	80	$2.16e^{-5}$
2	0.010265	0.63	130	$2.08e^{-5}$
3	0.010379	0.69	100	$2.18e^{-5}$
4	0.010419	0.72	90	$2.17e^{-5}$
5	0.010494	0.62	140	$2.05e^{-5}$

in which the quantization setup is unique in each layer, a 0.5% higher network accuracy is achievable (90.1% for Resnet 20 and 64.7% for Resnet18). Therefore, we analyze layer-wise quantization in more detail.

For each layer, we use the look-up table to generate the optimal unit time and bias voltage for efficiency in DIANA. The first observation is that some scale factors are beyond the hardware capabilities (e.g., need a higher unit time). There are different solutions to this problem: during design, one can either allow a wider range of legal unit times or make changes at the circuital level, such as reducing the summation line capacitance. Increasing the ADC resolution is also an option that comes with the power cost. At runtime, it is possible to execute the computation with half the required scale factor and multiply the result by two in the digital domain. There is a post-processing SIMD unit in DIANA that can be utilized for this purpose. Using post-processing is a power-efficient and straightforward solution. However, it only mimics the scale factor of the calibrated quantization setup, and its step size is twice bigger, which can degrade the accuracy. To avoid this degradation by consuming more power, a scaling unrolling scheme can increase the obtainable scale factor; if weights and activations are unrolled twice, it is like a gain of two, and all scale factors in the look-up table are doubled. Finally, it could be possible to constrain the neural network training to lower scales, but the viability of this last option is out of the scope of our work.

To further assess the benefit of using AIMC for ML workloads, we measured the performance and power consumption for the different workloads in the ResNet20 and ResNet18, changing the unit time. Fig. 11 shows the relationship between efficiency/performance and unit time. Longer pulse widths result in higher power consumption and a slower computation cycle. The dots on the line represent the mapped scale factors for different layers. The dots on the extreme right, marked with a star, exceed the macro operating limits and have been mapped with the highest gain possible.

In the plots, we reported peak performance and efficiency from the digital core of DIANA [17] and TinyVers [43], a digital SoC that targets extreme edge and efficient inference, to compare analog and digital computation paradigms for real workloads. When considering efficiency, the DIANA analog macro dominates its digital counterparts: only in limited cases TinyVers is comparable with AIMC but with two orders of magnitude degradation in performance. Focusing on performance, TinyVers is bounded by a 10 MHz clock and low

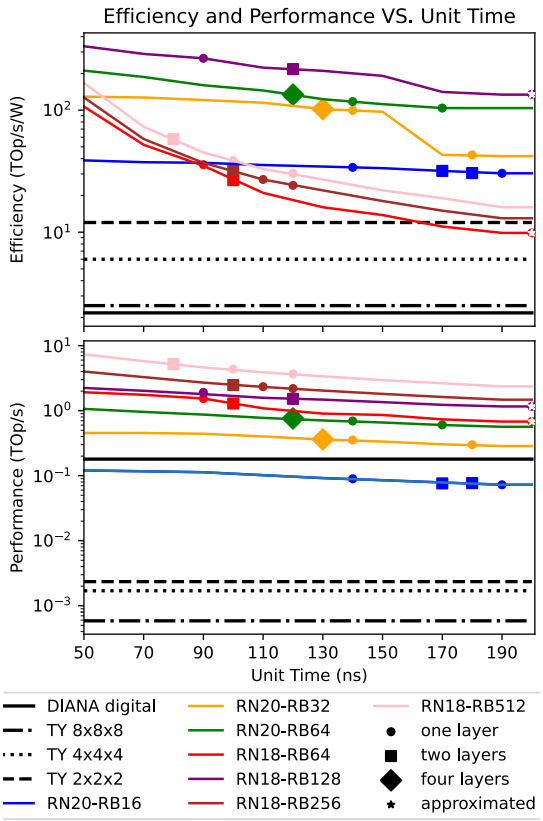


FIGURE 11. Efficiency and performance of DIANA's AIMC for different layer structures along with the digital baselines and mapped output quantization operating points from [28].

throughput, while DIANA digital core shows better performance than analog on the early, small layers from ResNet20. The comparison shows the performance, efficiency, and accuracy dilemma. The digital platforms exchange performance and efficiency, TinyVers in favor of efficiency, and DIANA in favor of performance. AIMC achieved higher figures in both merits, however, by sacrificing deterministic computation accuracy. We also noticed that unit time affects performance and efficiency for the analog macro in different degrees; while performance can only degrade by a factor of 2, efficiency can decrease by one order of magnitude when unit time increases over its legal values.

The developed model can translate the calibrated output quantization parameters into DIANA's AIMC's operating points; bias voltage and unit time. Applying output quantization calibration is essential to achieve high accuracies. It is shown that DIANA can implement network-wide calibrations reported in the literature [28]. However, The current design needs minor changes to support the higher scale factors that are required for layer-wise quantizations that offer more accuracy. The overall standard error of the model is always below $3.4e^{-4}$, suggesting good linearity of the DIANA's AIMC. However, a nonlinearity adjustment model is proposed that can be utilized to study AIMC variability impact or in training or compensation.

VI. CONCLUSION

The primary purpose of this paper was to bridge the gap between theoretical works on AIMC output quantization calibration and the practical difficulties of working with AIMC analog circuits. Hardware imposition of optimized quantization parameters is important for achieving high accuracy. The aim is fulfilled by studying the quantization from both network and circuit perspectives. The analog gain in AIMCs should be controlled in order to set the quantization parameters. As, a case study, the method is applied to DIANA's AIMC output quantization calibration. We coupled the calibrated quantization parameters with the chip's operating points that determine its analog gain in a look-up table. Thus, a dynamic quantization control on DIANA for implementing layer-wise quantization calibration is possible by using the look-up table.

It is also learned from the case study that more AIMC analog gain range improves the control over quantization parameters, enables more quantization implementations, and increases the achievable accuracies.

This paper also spots the design improvement points in DIANA and suggests solutions. More bias voltage contacts solve the voltage drop problem, and early saturation is remedied by interconnect delay reduction. These minor fixes can benefit the chip performance by a significant amount. The improvement points can be important for other AIMC designs to avoid the trial and error phase.

REFERENCES

[1] A. Ng, *AI is the New Electricity*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2018.
[2] H. Zhu, Q. Wei, F. Qiao, Y. Yang, X. Liu, S. Xu, and H. Yang, "CMOS image sensor data-readout method for convolutional operations with processing near sensor architecture," in *Proc. IEEE Asia Pacific Conf. Circuits Syst. (APCCAS)*, Oct. 2018, pp. 528–531.
[3] T. Shaik, X. Tao, Y. Li, C. Dann, J. McDonald, P. Redmond, and L. Galligan, "A review of the trends and challenges in adopting natural language processing methods for education feedback analysis," *IEEE Access*, vol. 10, pp. 56720–56739, 2022.
[4] S. Herbreteau and C. Kervrann, "DCT2net: An interpretable shallow CNN for image denoising," *IEEE Trans. Image Process.*, vol. 31, pp. 4292–4305, 2022.
[5] L. Yang, Q. Song, Z. Wang, M. Hu, and C. Liu, "Hier R-CNN: Instance-level human parts detection and a new benchmark," *IEEE Trans. Image Process.*, vol. 30, pp. 39–54, 2021.
[6] X. Zhong and D. Enke, "Predicting the daily return direction of the stock market using hybrid machine learning algorithms," *Financial Innov.*, vol. 5, no. 1, p. 24, Dec. 2019.
[7] C. Wu, M. Wang, X. Chu, K. Wang, and L. He, "Low-precision floating-point arithmetic for high-performance FPGA-based CNN acceleration," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 15, no. 1, pp. 1–21, Mar. 2022.
[8] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022.
[9] Y. Hu, Y. Liu, and Z. Liu, "A survey on convolutional neural network accelerators: GPU, FPGA and ASIC," in *Proc. 14th Int. Conf. Comput. Res. Develop. (ICCRD)*, Jan. 2022, pp. 100–107.
[10] J. Jo, S. Jeong, and P. Kang, "Benchmarking GPU-accelerated edge devices," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2020, pp. 117–120.
[11] M. Zainab, A. R. Usmani, S. Mehrban, and M. Hussain, "FPGA based implementations of RNN and CNN: A brief analysis," in *Proc. Int. Conf. Innov. Comput. (ICIC)*, Nov. 2019, pp. 1–8.
[12] D. Moolchandani, A. Kumar, and S. R. Sarangi, "Accelerating CNN inference on ASICs: A survey," *J. Syst. Archit.*, vol. 113, Feb. 2021, Art. no. 101887.
[13] M. Verhelst and B. Murmann, *Machine Learning at the Edge*. Cham, Switzerland: Springer, 2020, pp. 293–322.
[14] I. Dadras, M. H. Ahmadilivani, S. Banerji, J. Raik, and A. Abloo, "An efficient analog convolutional neural network hardware accelerator enabled by a novel memoryless architecture for insect-sized robots," in *Proc. 11th Int. Conf. Modern Circuits Syst. Technol. (MOCAST)*, Jun. 2022, pp. 1–6.
[15] D. Kim, C. Yu, S. Xie, Y. Chen, J.-Y. Kim, B. Kim, J. P. Kulkarni, and T. T. Kim, "An overview of processing-in-memory circuits for artificial intelligence and machine learning," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 12, no. 2, pp. 338–353, Jun. 2022.
[16] S. Kwon, K. Lee, Y. Kim, K. Kim, C. Lee, and W. W. Ro, "Measuring error-tolerance in SRAM architecture on hardware accelerated neural network," in *Proc. IEEE Int. Conf. Consum. Electronics-Asia (ICCE-Asia)*, Oct. 2016, pp. 1–4.
[17] P. Houshmand, G. M. Sarda, V. Jain, K. Ueyoshi, I. A. Papistas, M. Shi, Q. Zheng, D. Bhattacharjee, A. Mallik, P. Debacker, D. Verkest, and M. Verhelst, "DIANA: An end-to-end hybrid digital and analog neural network SoC for the edge," *IEEE J. Solid-State Circuits*, vol. 58, no. 1, pp. 203–215, Jan. 2023.
[18] I. A. Papistas, S. Cosemans, B. Rooseleer, J. Doevenspeck, M.-H. Na, A. Mallik, P. Debacker, and D. Verkest, "A 22 nm, 1540 TOP/s/W, 12.1 TOP/s/mm^2 in-memory analog matrix-vector-multiplier for DNN acceleration," in *Proc. IEEE Custom Integr. Circuits Conf.*, 2021, pp. 1–2. [Online]. Available: https://ieeexplore.ieee.org/document/9431575/
[19] K. Roy, I. Chakraborty, M. Ali, A. Ankit, and A. Agrawal, "In-memory computing in emerging memory technologies for machine learning: An overview," in *Proc. 57th ACM/IEEE Design Autom. Conf. (DAC)*, Jul. 2020, pp. 1–6.
[20] H.-Y. Chen, S. Brivio, C.-C. Chang, J. Frascaroli, T.-H. Hou, B. Hudec, M. Liu, H. Lv, G. Molas, and J. Sohn, "Resistive random access memory (RRAM) technology: From material, device, selector, 3D integration to bottom-up fabrication," in *Resistive Switching: Oxide Materials, Mechanisms, Devices and Operations*. Cham, Switzerland: Springer, 2022, pp. 33–64.
[21] A. Ehrmann, T. Blachowicz, G. Ehrmann, and T. Grethe, "Recent developments in phase-change memory," *Appl. Res.*, vol. 1, no. 4, Dec. 2022, Art. no. e202200024.
[22] P. Jangra and M. Duhan, "A review on emerging spintronic devices: CMOS counterparts," in *Proc. 7th Int. Conf. Commun. Electron. Syst. (ICCES)*, Jun. 2022, pp. 90–99.
[23] S. Jain and A. Raghunathan, "CxDNN: Hardware–software compensation methods for deep neural networks on resistive crossbar systems," *ACM Trans. Embedded Comput. Syst.*, vol. 18, no. 6, pp. 1–23, Nov. 2019, doi: 10.1145/3362035.
[24] C. Yu, T. Yoo, K. T. C. Chai, T. T. Kim, and B. Kim, "A 65-nm 8T SRAM compute-in-memory macro with column ADCs for processing neural networks," *IEEE J. Solid-State Circuits*, vol. 57, no. 11, pp. 3466–3476, Nov. 2022.
[25] B. Zhang, J. Saikia, J. Meng, D. Wang, S. Kwon, S. Myung, H. Kim, S. J. Kim, J.-S. Seo, and M. Seok, "A 177 TOPS/W, capacitor-based in-memory computing SRAM macro with stepwise-charging/discharging DACs and sparsity-optimized bitcells for 4-bit deep convolutional neural networks," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr. 2022, pp. 1–2.
[26] E. Choi, I. Choi, C. Jeon, G. Yun, D. Yi, S. Ha, I.-J. Chang, and M. Je, "A 133.6 TOPS/W compute-in-memory SRAM macro with fully parallel one-step multi-bit computation," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr. 2022, pp. 1–2.
[27] P.-C. Wu, J.-W. Su, Y.-L. Chung, L.-Y. Hong, J.-S. Ren, F.-C. Chang, Y. Wu, H. Y. Chen, C.-H. Lin, H.-M. Hsiao, S.-H. Li, S.-S. Sheu, S.-C. Chang, W.-C. Lo, C.-C. Lo, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, C.-I. Wu, and M.-F. Chang, "A 28 nm 1Mb time-domain computing-in-memory 6T-SRAM macro with a 6.6 ns latency, 1241 GOPS and 37.01 TOPS/W for 8b-MAC operations for edge-AI devices," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 1–3.
[28] N. Laubeuf, J. Doevenspeck, I. A. Papistas, M. Caselli, S. Cosemans, P. Vrancx, D. Bhattacharjee, A. Mallik, P. Debacker, D. Verkest, F. Catthoor, and R. Lauwereins, "Dynamic quantization range control for analog-in-memory neural networks acceleration," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 27, no. 5, pp. 1–21, Sep. 2022.

[29] S. Spetalnick and A. Raychowdhury, "A practical design-space analysis of compute-in-memory with SRAM," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 4, pp. 1466–1479, Apr. 2022.
[30] J. Klein, I. Boybat, Y. Qureshi, M. Dazzi, A. Levisse, G. Ansaloni, M. Zapater, A. Sebastian, and D. Atienza, "ALPINE: Analog in-memory acceleration with tight processor integration for deep learning," *IEEE Trans. Comput.*, vol. 72, no. 7, pp. 1985–1998, Jul. 2023.
[31] A. Gholami, S. Kim, D. Zhen, Z. Yao, M. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," in *Proc. Comput. Vis. Pattern Recognit.*, Jan. 2022, pp. 291–326.
[32] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan, "PACT: Parameterized clipping activation for quantized neural networks," 2018, *arXiv:1805.06085*.
[33] S. K. Esser, J. L. McKinstry, D. Bablani, R. Appuswamy, and D. S. Modha, "Learned step size quantization," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–12.
[34] A. Basumallik, D. Bunandar, N. Dronen, N. Harris, L. Levkova, C. McCarter, L. Nair, D. Walter, and D. Widemann, "Adaptive block floating-point for analog deep learning hardware," 2022, *arXiv:2205.06287*.
[35] J. L. McKinstry, S. K. Esser, R. Appuswamy, D. Bablani, J. V. Arthur, I. B. Yildiz, and D. S. Modha, "Discovering low-precision networks close to full-precision networks for efficient embedded inference," 2018, *arXiv:1809.04191*.
[36] NVIDIA. (2017). *8-Bit Inference With Tensorrt*. [Online]. Available: http://on-demand.gputechconf.com/gtc/2017/presentation/ s7310-8-bit-inference-with-tensorrt.pdf
[37] C. Yakopcic, M. Z. Alom, and T. M. Taha, "Extremely parallel memristor crossbar architecture for convolutional neural network implementation," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1696–1703.
[38] P. Yao, H. Wu, B. Gao, J. Tang, Q. Zhang, W. Zhang, J. J. Yang, and H. Qian, "Fully hardware-implemented memristor convolutional neural network," *Nature*, vol. 577, no. 7792, pp. 641–646, Jan. 2020.
[39] M. Yamaguchi, G. Iwamoto, Y. Nishimura, H. Tamukoh, and T. Morie, "An energy-efficient time-domain analog CMOS BinaryConnect neural network processor based on a pulse-width modulation approach," *IEEE Access*, vol. 9, pp. 2644–2654, 2021.
[40] S.-T. Lee and J.-H. Lee, "Neuromorphic computing using NAND flash memory architecture with pulse width modulation scheme," *Frontiers Neurosci.*, vol. 14, Sep. 2020, Art. no. 571292.
[41] J. Lienig and J. Scheible, *Fundamentals of Layout Design for Electronic Circuits*. Berlin, Germany: Springer, 2020. [Online]. Available: https://books.google.ee/books?id=qICgzAEACAAJ
[42] M. Vermeer, "Interface trap density extraction from the subthreshold slope of FDSOI devices," Univ. Twente, Twente, The Netherlands, Tech. Rep., May 2019.
[43] V. Jain, S. Giraldo, J. D. Roose, L. Mei, B. Boons, and M. Verhelst, "TinyVers: A tiny versatile system-on-chip with state-retentive eMRAM for ML inference at the extreme edge," *IEEE J. Solid-State Circuits*, vol. 58, no. 8, pp. 2360–2371, Aug. 2023.

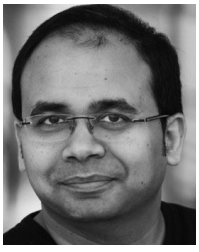
IMAN DADRAS received the B.S. and M.S. degrees in electronic engineering from Shahid Rajayi Teacher Training University, Tehran, Iran, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the University of Tartu. From 2014 to 2017, he worked on wide-band and low-noise trans impedance amplifiers for optical links. Since 2019, he has been working on ASIC design implementation for soft robots. His current research interests include analog and mixed-signal IC design, front-end amplifiers, radio frequency circuits, analog computing, neural network accelerators, soft robotics, and image processing for biomedical applications.

GIUSEPPE M. SARDA received the B.Sc. and M.Sc. degrees in electrical engineering from Politecnico di Torino, Turin, Italy, in 2018 and 2020, respectively. He carried out the master's thesis from Technische Universit Wien (TU Wien), Vienna, Austria. In September 2020, he joined imec, Leuven, Belgium, and the MICAS Group, Katholieke Universiteit Leuven, Leuven, as a Ph.D. Researcher. His current research focuses on in-memory design for efficient embedded machine learning.

NATHAN LAUBEUF received the dual master's degree in microelectronics and computer science from École des Mines de Saint Étienne, Gardanne, France, and Aix-Marseille University, Marseille, France, in 2017. He is currently pursuing the Ph.D. degree with imec, Leuven, Belgium, supported by Katholieke Universiteit Leuven, Leuven.

His research focuses on efficient deployment of artificial neural networks on analog in-memory computing-based systems and their execution with low precision arithmetic.

DEBJYOTI BHATTACHARJEE received the B.Tech. degree in computer science and engineering from the West Bengal University of Technology (WBUT), Kolkata, West Bengal, India, in 2013, the M.Tech. degree in computer science from the Indian Statistical Institute, Kolkata, in 2015, and the Ph.D. degree in computer science and engineering from Nanyang Technological University, Singapore, in 2019. He was a Research Fellow with Nanyang Technological University, for a year. During the Ph.D. study, he worked on design of architectures using emerging technologies for in-memory computing (IMC). He has developed novel technology mapping algorithms, technology-aware synthesis techniques, and proposed novel methods for multi-valued logic realization. He is currently a Research and Development Engineer with the Compute System Architecture Unit, imec, Leuven, Belgium. His current research interests include machine learning accelerator using analog hardware, hardware design automation tools, and application-specific accelerator design, with an emphasis on emerging technologies.



ARINDAM MALLIK received the M.S. and Ph.D. degrees in electrical engineering and computer science from Northwestern University, Evanston, IL, USA, in 2004 and 2008, respectively. He is currently a technologist in semiconductor research with 20 years of experience. He leads the Future System Exploration (FuSE) Group, Compute System Architecture (CSA) Research and Development Unit. He has authored or coauthored more than 100 articles in international journals and conference proceedings. He holds number of international patents. His research interests include novel computing systems, design–technology co-optimization, and economics of semiconductor scaling.