Memory Devices and A/D Interfaces: Design Trade-offs in Mixed-Signal Accelerators for Machine Learning Applications

Memory Devices and A/D Interfaces: Design Trade-offs in Mixed-Signal Accelerators for Machine Learning Applications

Michele Caselli, *Member, IEEE,* Peter Debacker, and Andrea Boni, *Member, IEEE*

***Abstract*—This tutorial focuses on memory elements and analog/digital (A/D) interfaces used in mixed-signal accelerators for deep neural networks (DNNs) in machine learning (ML) applications. These very dedicated systems exploit analog in-memory computation (AiMC) of weights and input activations to accelerate the DNN algorithm. The co-optimization of the memory cell storing the weights with the peripheral circuits is mandatory for improving the performance metrics of the accelerator. In this tutorial, four memory devices for AiMC are reported and analyzed with their computation scheme, including the digital-to-analog converter (DAC). Moreover, we review analog-to-digital converters (ADCs) for the quantization of the AiMC results, focusing on the design trade-offs of the different topologies given by the context.**

***Index Terms*—Analog computing, deep neural networks (DNNs), AiMC, SRAM, resistive RAM (RRAM), Indium-Gallium-Zinc-Oxide (IGZO) DRAM, Spin Orbit Torque (SOT) MRAM, A/D Converters, SAR ADCs, Flash ADCs.**

I. Introduction

DEEP Neural Networks have demonstrated great potential in a wide variety of AI/ML applications, from image classification to speech recognition. Mixed-signal accelerators aim to maximize throughput and energy efficiency during the DNN algorithm execution, exploiting Analog in-Memory Computation to reduce the data movement [1]. Moreover, they can obtain high classification accuracy, working with low precision operands, with large benefits for the energy consumption [2]. In the AiMC approach, the huge amount of Matrix-Vector Multiplications (MVMs) for the inference is realized directly inside memory computing cores. These circuits are composed of a massive number of memory cells, used to store the pre-trained DNN weights $w(i,j)$, and arranged in crossbar arrays, as shown in Fig. 1. The activations $a(i)$, representing the input data or the features extracted by a layer, are transmitted along the crossbar rows. The result output vector $Y(j)$ is accumulated on the matrix columns in analog fashion:

$$Y(j) = \sum_{i=1}^{N} a(i) \cdot w(i,j), \ \forall \ j \ \in \ [1, M] \qquad (1)$$

where N and M are the rows and the columns of the memory. To store the weights, emerging memories, like Resistive RAM (RRAM), Indium-Gallium-Zinc-Oxide (IGZO) DRAM, and Spin-Orbit Torque (SOT) MRAM, have attractive features for the AiMC context [3], [4], [5], and they can compete with standard SRAM-based implementation. Indeed, recent prototypes of compute arrays, with both emerging and standard memories, show performances close to 1000 Tera-OPerations per Watt (TOPs/W) [6]. Few simulated fully-analog accelerators have been recently reported in [7], [8], but the large majority of the architectures are mixed-signal, so including analog-digital interfaces. On the crossbar input in Fig. 1, the voltage on the rows is modulated by means of D/A converters (DACs), controlled by the input activations. At the output, the Multiply-accumulate (MAC) results are accumulated on the columns, and quantized with A/D converters (ADCs) for further processing in the digital domain. Sequential architectures, like SAR [9] or integrating ADCs [1], provide high energy efficiency and small area. For low-resolutions, Flash ADCs become energy-competitive, thanks to the higher conversion speed, guaranteed by the parallel operation [10]–[12].

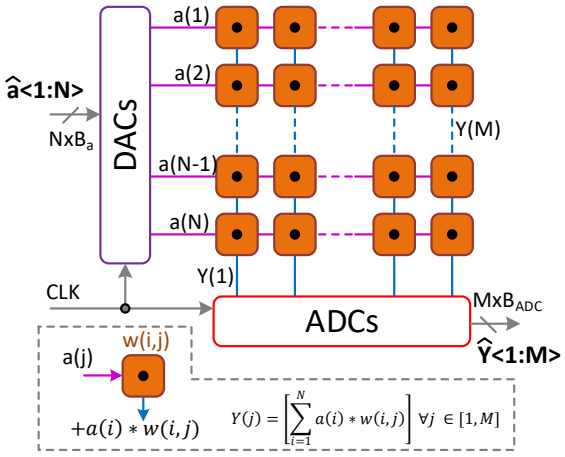

Fig. 1: AiMC accelerator architecture. Dot product and MAC equation in the dashed box.

This tutorial focuses on memory devices and A/D interfaces for mixed-signal accelerators, with AiMC approach, for convolutional neural networks. Section II proposes memory elements for analog computing, considering pros and challenges specific of the context. Then, computation schemes including the proposed memory cells and the input DACs are proposed, evaluating the trade-offs in the different schemes. Finally,

M. Caselli and A. Boni are with the Department of Engineering and Architecture, University of Parma, 42124, Parma, Italy e.mail: michele.caselli@unipr.it and andrea.boni@unipr.it

P. Debacker is with imec, Leuven, Belgium e-mail: peter.debacker@imec.be

TABLE I: Memory Devices for AiMC

	RRAM	SOT	SRAM	IGZO	Ideal
Type	Res	Res	CS	CS	-
NV	Yes	Yes	No	No	Yes
R_{on}	<0.1MΩ	>1MΩ	-	-	>1MΩ
I_{on}	-	-	<1μA	<1μA	<1μA
ON/OFF	Large	Small	Large	Large	Large
Cell Area	Medium	Medium	Huge	Small	Small
Variations	High	Low	Low	High	Low
FEOL Free	No	No	No	Yes	Yes
Multilevel	Yes	Yes	No	Yes	Yes

NV: non-volatile - FEOL: front-end of line
Memory devices in [3], [13], [5], [14], [4], [15].

in Section III, the tutorial describes three ADC topologies reported in literature to quantize the MAC result, focusing on the design trade-offs of the AiMC context.

II. MVM Computation Circuits

In mixed-signal accelerators, the memory elements for weight storage are exploited as compute cells for the analog MAC. Together with the DAC, they define the computation scheme, and impact several specifications of the output peripheral circuits. Table I reports four memory elements for AiMC, comparing their features with an ideal memory for the context. SRAM cells co-integrate well with highly scaled CMOS technology and are easy to integrate with periphery. Memories made with emerging technologies, like RRAM, IGZO DRAM, and SOT MRAM, outperform SRAM in several AiMC metrics, but with additional challenges for the design, and costs due to the co-integration with standard CMOS process. Moreover, none of these devices is available in current leading edge CMOS nodes. RRAM are potentially very dense, non volatile, and based on a relatively more mature process, compared with the other emerging memories [13]. A common RRAM is a two-terminal device, where a thin layer of insulation material separates two metallic electrodes, Fig. 2(a). This memory can switch between high resistance R_{HRS} and low resistance states R_{LRS}, by means of a reversible physical process of formation and rupture of a conductive filament in the oxide. By an external voltage applied across the device terminals, the filament is created, lowering the resistance at R_{LRS}. The reverse voltage resets the device resistance at R_{HRS}. RRAM device can potentially offer multi-bit weight capability, for increased DNN accuracy. Nonetheless, at the current maturity state of this technology the variability and reliability of the multilevel states remain an issue [3], [13]. 1-transistor 1-resistor (1T1R) RRAM cell for AiMC context integrates a selector to write the memory, partially limiting the area benefits of this very small device.

SOT MRAM is an attractive option due to the non-volatility, the small area, and the high endurance [16]. The memory is composed of two magnetic layers, the free layer (FL) and the reference layer (RL), separated by an insulating Magnesium-Oxide (MgO) tunneling barrier, Fig. 2(b). In FL, magnetization can be changed, whereas it is fixed in RL. Even if a first

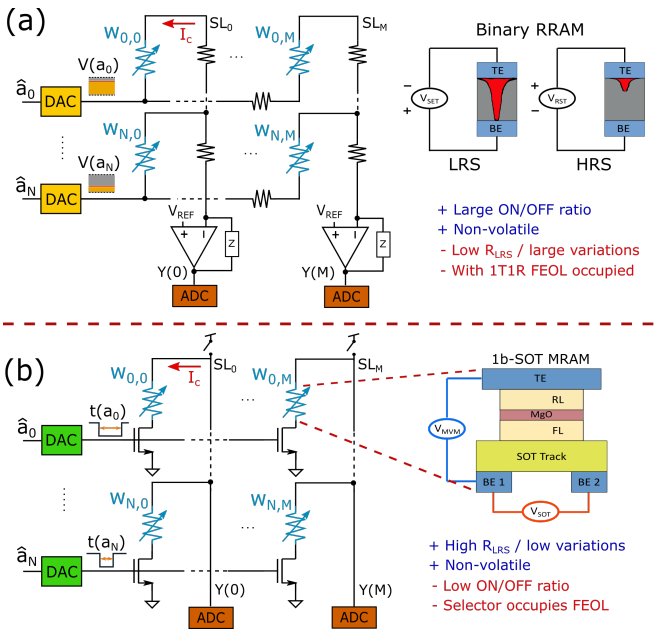

Fig. 2: Resistive cells: (a) Computation scheme with PAM DAC and output clamp - RRAM device for AiMC [3]. (b) Computation scheme with PWE DAC - Binary SOT MRAM device [5].

conceptual demonstrator for multi-bit memory is proposed in [17], SOT MRAM is binary, and it switches between R_{HRS} and R_{LRS}, where the tunneling probability through the Oxide barrier is different. Differently from the spin-transfer torque MRAM, SOT MRAM is a tripole, where the write current does not pass through device. This allows to achieve MΩ resistance values in both states.

C-Axis Aligned Crystalline IGZO transistors (CAAC-IGZO) are ultra-low leakage devices that can compose a DRAM memory for AiMC, Fig 3(b) [15]. The 2-transistors 0-capacitor (2T0C) IGZO DRAM is implemented with just two transistors, exploiting for the storage the oxide capacitance C_{OX} of the read transistor T2. Thanks to the extremely low leakage current of the access device T1, this DRAM can achieve weight retention in the order of tens of seconds, drastically limiting the costs for the periodic data refresh [4]. The data is memorized as analog voltage $V(w_{i,j})$, hence this memory can potentially store multi-bit weights. Additionally, IGZO transistors can be monolithically stacked in the back end of line (BEOL) on the peripheral circuit, minimizing the accelerator floorplan [18].

The memories for AiMC in Table I can be roughly categorized per summation approach: current source-like (CS) and resistive (R) cells. The former type exploits the memorized weight to control a transistor in saturation regime. R-cells store the weights in programmable resistance values. The levels of current/resistance of the memory cells during the computation (I_{on} and R_{on} in Table I) impact the linearity of the MVM operation in equation (1). The linearity should be maximized since it directly affects the DNN accuracy performance. Indeed, partial deviations from the ideal transfer curve can be compensated in training, but with additional modelling effort. Large deviations can drastically worsen the DNN accuracy or prevent the network to be trained [19] [20]. For example, in RRAM cells where R_{LRS} is low [13], the result of the

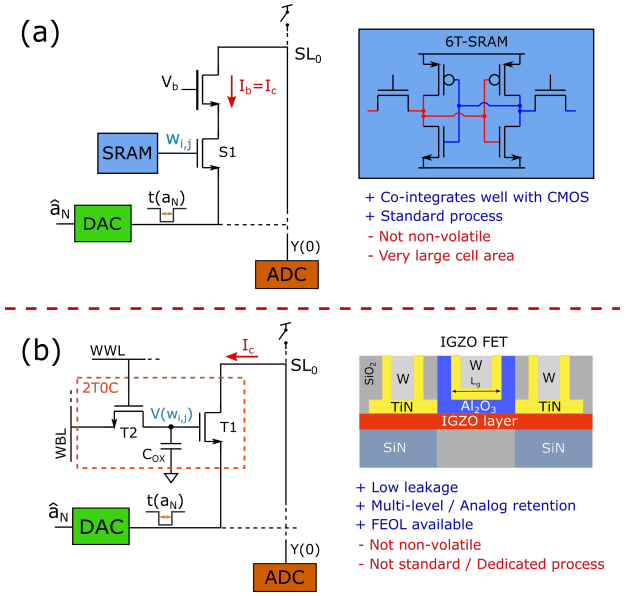

Fig. 3: CS cells: (a) Computation scheme with additional current source - SRAM device for AiMC in [14]. (b) Computation scheme with 2T0C IGZO DRAM - IGZO device cross-section in [4].

analog computation is significantly degraded by the voltage drops (IR-drop) on the parasitic resistances of summation and activation lines [21]. Together with other device non-idealities, this can cause significant losses in DNN accuracy [3]. Conversely, AiMC arrays with high resistive R-cells, like SOT MRAM, and with CS-cells, exploiting the large output impedance of the current source, are more robust against this non-ideality [14], [22]. IR-drop decreases with the number of memory cells of rows and columns, hence RRAM-based arrays in literature are usually smaller, compared to arrays including the other devices of Table I [23]. Taking into account that a size of 1000 cells per column matches well with typical convolutional layers and it balances well array and ADC power consumptions, this size-linearity trade-off favourites energy and area efficiencies of array with SRAM, DRAM, and SOT MRAM [24]. The values of I_{on} and R_{on} concur to set, with the cell capacitance C_c, the computation time t_c of the MAC operation. This latency should be small for high accelerator throughput. For computation schemes where the summation lines are precharged and then discharged by the MACs (precharge-discharge schemes), $t_c \propto C_c/I_{on}$ with CS-cells, whereas t_c is related to $\tau_c = R_{on}C_c$ with R-cells. These considerations set for scaled technology nodes the reasonable specifications $I_{on} < 1$ μA and $R_{on} > 1$ MΩ [20]. In case the summation line is clamped to a reference voltage, t_c must be compatible with the bandwidths of the downstream circuits.
The DNN accuracy can be penalized also by weight writing errors, fabrication defects, and process and mismatch variations. At the current state of maturity, emerging memories for multi-bit weights, like RRAM and IGZO DRAM, require circuits to compensate non-idealities and improve the DNN accuracy performance, with additional costs [25],[3]. Memory cells with large ON/OFF ratios, like SRAM and IGZO DRAM in Table I, provide $Y(j)$ signals with large SNR and swing V_{YS}. Computing with these devices allows a much easier design of the output ADC, with respect to memories with poor ON/OFF ratio like SOT MRAM [5],[22]. Finally, given the huge number of weights in DNNs, to store a full network in the memory array at affordable costs, the area of the ideal memory device is minimal. However, an array with one DAC/ADC per compute cell does not benefit of a small memory device, if the peripheral circuits do not scale accordingly. To circumvent this bottleneck, accelerators embedding small memory cells share the peripheral circuits on multiple cells. Strategies used in literature, like time multiplexing or multi-column summing, are discussed in Section III.

Computation Schemes for AiMC

The performance of the AiMC memory cells is tightly related to the implementation of the DAC converting the input activations a_i. Fig. 2(a) shows a computation scheme for R-cells in Table I, common in literature for RRAM memories [26], [9]. The R-cell stores the weights as conductance $R_{on} = 1/w_{i,j}$. The activations a_i are encoded as discrete voltage levels $V_{a,i}$, in a pulse-amplitude modulation (PAM). In this scheme, the summation line is clamped to a fixed voltage, to provide the exact MVM result. Each cell on SL contributes with $I_c \propto V_{a,i}/R_{on} \propto a_i \cdot w_{i,j}$. This straightforward implementation has relevant downsides. Together with the additional area, energy, and latency of the clamp, the input DAC must drive a significant amount of current during the computation. Moreover, when computing with low-resistance R-cells, the previously mentioned IR-drop negatively affects the linearity, with errors in the analog transfer function.
Fig. 2(b) shows an improved computation scheme for resistive cells. Here, the SL is precharged at a given voltage and discharged by the MAC computation (precharge-discharge scheme). In the DAC, a_i is encoded in the pulse-width t_P of a voltage pulse at full amplitude (pulse-width encoding PWE), and applied to the gate terminal of a FET switch. Each cell subtracts charges from SL:

$$Q_c \propto t_{P,i}/R_{on} \propto a_i \cdot w_{i,j} \tag{2}$$

t_P can be propagated along the rows by simple digital buffers. Compared with the previous scheme, the effect of the IR-drop on the activation line is neglected, and the high current in the DAC drastically reduced. This approach can be fruitfully exploited with SOT MRAM [5]. The large resistance values of the SOT allow good linearity and low error in the computation, avoiding the expensive clamp circuit. The computation scheme in Fig. 3(a) exploits an SRAM-based CS-cell in a precharge-discharge scheme. The binary weight memorized in the SRAM is applied to the gate of the switch S1, used to enable a current source. The cell current is set to I_b, regulated by the bias voltage V_b. The PWE DAC operates on the source of S1, and the amount of charge displaced from the summation line is:

$$Q_c \propto t_{P,i} \cdot I_b \propto a_i \cdot w_{i,j} \tag{3}$$

This scheme, proposed in [14], includes a long transistor as current source, with large increase of the cell area. However, V_b is common for all the cells and it can be tuned for the best I_b value. A similar approach is shown in Fig. 3(b) for IGZO

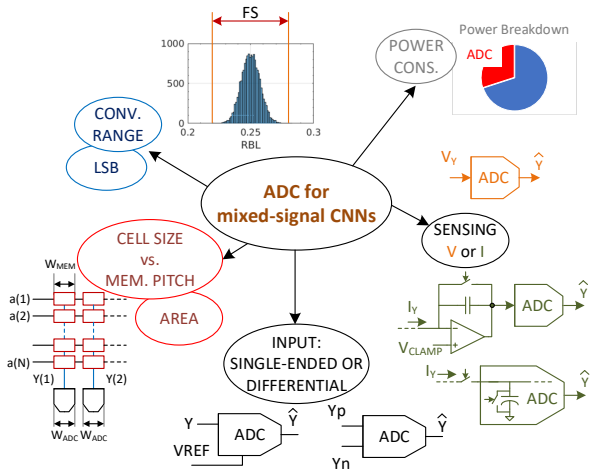

Fig. 4: ADC design constraints in mixed-signal accelerators.

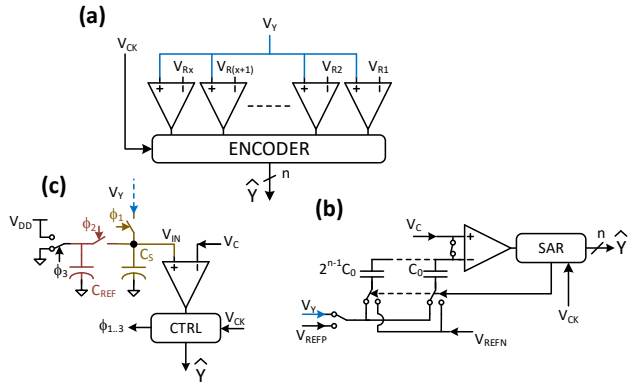

Fig. 5: ADC for AiMC accelerators: Flash (a), SAR (b), and IS (c).

DRAM cell, still in a precharge-discharge scheme. Here, the PWE activations are applied at the source of the readout IGZO FET T2. In this memory, the weight is stored as analog voltage V($w_{i,j}$) in the internal node capacitance C_{OX} and it directly sets I_c. This compute memory avoids the long FET of the SRAM-based cell, and the generation and propagation of V_b, with a significant reduction of the occupied area.

III. Output Peripheral Interfaces

In every mixed-signal AiMC accelerator, the A/D converter quantizes the analog MAC results for the digital post-processing, required by DNN algorithm. For memory arrays with single-bit output precision, complex ADC topologies are replaced by a simple latch comparator or a sense amplifier [27], [28]. With low-resolution operands and results, the DNN accuracy performance improves, and AiMC accelerators promise to be more energy efficient than fully-digital implementations [2]. However, with multi-bit output precision, the integration of the output ADCs in the memory accelerator becomes non-trivial, and several design constraints must be taken into account, as highlighted in Fig. 4. In literature, three ADC topologies have been mainly used for mixed-signal accelerators: Flash, Successive-Approximation (SAR), and Integrating-Sequential (IS) converters. Their single-ended schematic views are shown in Fig. 5. Indeed, considering the low-resolution requirement, these ADCs guarantee sampling speeds, energy consumptions, and silicon areas suitable for the AiMC implementations.

The result of the MAC can be either a voltage or a current signal, depending on the computation scheme. In the case of current sensing, a closed-loop integrator, acting as summation line clamp, can be introduced at the ADC input to perform the current-to-voltage conversion [9], [29], as discussed in Section II. Alternatively, the signal on summation line can be converted from current to charge, with a sampling capacitor or through the parasitic column capacitance [1], to improve energy and area efficiency. To benefit of the scaling of the memory cells and avoid waste of silicon area, the layout width of the ADC, and possibly of the integrator, should be within the memory-cell pitch. Taking into account the pitch size of emerging compute cells in literature, already below a micrometer [23], this additional constraint severely impacts the ADC design. If the width of the analog interface cannot be shrinked down to the memory pitch, the latter is automatically increased. For example, it is possible to argue that the column pitch in the RRAM-based accelerator in [9] ($\approx$ 20 μm) is dictated by the integrator and the ADC.

As previously mentioned, the DNN accuracy during inference is related to the operands resolution, and this algorithm specification impacts the ADC design. In Flash converters, where the MVM output is simultaneously compared to 2^n-1 references, the area depends exponentially on the number of output bits n, Fig 5(a). This relationship holds also for the area of capacitive DAC in C-SAR converters, shown in Fig 5(b), where the bottom plates of n-1 binary-scaled capacitors in the embedded DAC, after V_Y sampling, are successively rebased following a binary-search algorithm. This procedure halves the conversion range at every step and assures the convergence of the algorithm.

Other design specifications affect the ADC area occupation in the accelerator. Indeed, area-expensive capacitors, are used also in Flash ADCs to implement the offset sampling and subtraction in each comparator of the converter, to improve the non-linearity error (INL) [11]. Alternative solutions in literature to avoid this additional area are: correlated double sampling technique, made by flipping the comparator inputs, but halving the conversion speed [1], and foreground offset calibration, where the body biasing of the input devices of each comparator is trimmed at the power-on [30].

The area and pitch constraints can be overcome also by sharing among several columns a single ADC, with time multiplexing approach [30]. However, to leave the memory throughput unaffected, the sampling speed rises above 1 GS/s, just sharing the converter among few tens of columns. From these considerations, the Flash architecture is the only suitable candidate for this design option.

Another design technique, relaxing the ADC-width constraint, is proposed in [10], where the B_w-bit weights (B_w=4) are distributed over B_w successive columns. The ADC width can be B_w-times larger than the memory pitch, but a computing circuit, to combine the MAC results in the involved columns, is necessary before the conversion. A binary adder, made with binary-scaled B_w capacitors, can be used for these purposes.

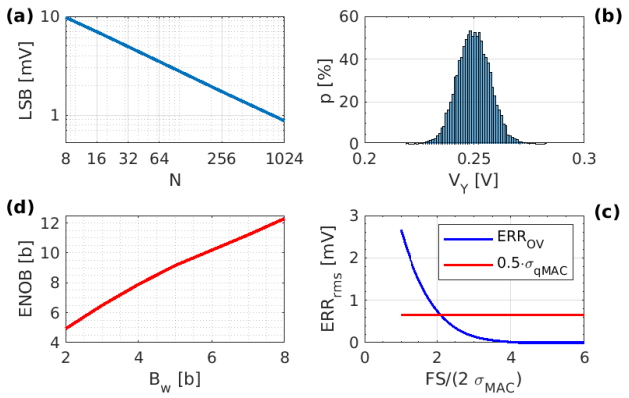

Fig. 6: From (a) to (d): Maximum LSB size vs N, PDF of V_{RBL}, over-range error compared to $0.5\,\sigma_{qMAC}$ vs. FS/σ_{MAC}, and ADC ENOB vs. B_w. If unspecified: B_a=8, B_w=2, N=1024, V_{YS}=1-V.

Differently from the other two converters, in the IS ADCs of Fig 5(c), the area is not related with the output resolution, and it is drastically reduced with a charge sharing operation based on a unit capacitor C_{REF} and a sampling capacitor C_S [1]. At the first conversion step, the MSB is obtained by comparing V_{RBL} with the mid-range reference voltage, V_C. Then, a constant voltage step, corresponding to the converter LSB, is successively added or subtracted to the input sampled voltage V_{IN}, until it reaches the mid-range voltage. The conversion of the IS ADC depends on the distance of V_Y from the mid-range. If the accelerator requires the synchronized output of all the converters, the throughput is limited by the ADC with the longest time to complete the conversion. The probability density function (PDF) of the summation signal V_Y affects also the throughput of this accelerator. Indeed, in DNN algorithms, the PDF of the MAC output often exhibits the shape of a normal distribution [31]. In case of low-energy summation signals with narrow Gaussian distribution, the IS ADC is well-suited for AiMC since it achieves higher conversion speed than SAR-ADCs, with additional advantages in terms of area and energy. However, in case of V_Y with large standard deviation, the A/D conversion requires many steps, and this sequential ADC can become the main speed-bottleneck of the accelerator.

The PDF of the output signal is also tightly related with the conversion range and the data resolutions, critical specifications for the ADC in mixed-signal accelerators. Assuming for the first DNN layer, $w(i,j)$ and $a(i)$ uncorrelated random variables, uniformly distributed (UD) over 0-to-1 range, the distribution of the MAC result $Y(j)$ is centered at μ_{MAC}=1/4, with variance σ^2_{MAC}=7/(144·N). Here, N is the number of rows in the array and $Y(j)$ is normalized to the 0-to-1 range. The variance of the quantization error, σ^2_{qMAC}, due to the limited resolution of weights (B_w bits) and activations (B_a bits), is approximated by the following equation:

$$\sigma^2_{qMAC} \approx \frac{1}{36\,N} \cdot \left(2^{-2\,B_a} + 2^{-2\,B_w}\right) \qquad (4)$$

The effect of the ADC quantization error can be neglected if the converter LSB is lower than σ_{qMAC}/2. As matter of example, with N=1024, B_a=8, B_w=2, and an $Y(j)$ swing V_{YS} =1-V, the LSB must be within approximately 0.7 mV. As shown in the graph in Fig. 6(a), a large N value leads to a severe constraint on the maximum INL and input thermal noise of the ADC. However, the conversion range FS can be tailored on the PDF of V_{RBL}, shown in Fig. 6(b), with a variance scaling with 1/N. A convenient FS is obtained by identifying the crossing point of the rms value of the over-range errors to σ_{qMAC}/2, as shown in Fig. 6(c). From the previous array specifications, the FS is only 4 σ_{MAC}, i.e. 41 mV, and centered on the V_Y mean value. In the considered case, an effective number of bits (ENOB) of 5-b is obtained, sizing the ADC for the required LSB. From this analysis, a Flash converter with reference voltages only located in the confined range is an attractive option. This solution significantly reduces the number of comparators needed, with remarkable benefits in terms of occupied silicon area and energy consumption [11], [32]. Considering the V_Y distribution, a convenient alternative is the IS topology. On the contrary, in C-SAR ADCs, an FS lower than the supply voltage requires an additional voltage references generator with enough driving capability, leading to increased area and power consumption.

Signal amplification, before the A/D conversion, is an alternative option in literature to deal with confined-FS [9], [29]. For the C-SAR topology, expanding the converter FS relaxes the INL and thermal noise requirements, with a simplification of the design. The plot in Fig. 6(d) provides the minimum ADC resolution vs. the weight precision, for 8-bit inputs and N=1024. This results, obtained from (4), can be useful for converter sizing. However, we highlight that, in DNNs, the distribution of the trained weights is approximately Gaussian with a lower variance than the UD case [33], [34]. Moreover, in the inner DNN layers, also the input activations, obtained by the digital processing of a previous layer MAC results, have gaussian-shaped or clipped gaussian-shape distributions [31], [35]. Therefore, even the variance of the accumulated MAC is lower. Nonetheless, the quantization error affecting the dot-product can still be estimated with (4). In this perspective, the maximum LSB in Fig. 6(a) is suitable for the design of the ADCs, also of the inner layers, whereas the ENOB estimation in Fig. 6(d) is a conservative design specification. Further ADC optimizations are possible for specific DNN implementations, starting from known MAC results distributions.

In some reported accelerators, the MAC signal is given by the differential voltage of two accumulation columns [1], [22]. This array design choice requires a differential-input ADC with several benefits. In particular, the weights can be ternary-quantized, providing an additional MAC resolution of 0.5 bit with respect to binary weights. This leads to better DNN accuracy performance, at the cost of slightly larger area, in case of small compute memory cells [5]. Other benefits are the increase of the MAC variance, leading to a larger LSB for the ADC, a zero-centered conversion range, and a lower memory cell activity per line, resulting in energy saving.

ACKNOWLEDGMENT

The authors thank Stefan Cosemans for the scientific contribution to this paper.

REFERENCES

[1] A. Biswas and A. P. Chandrakasan, "CONV-SRAM: An Energy-Efficient SRAM With In-Memory Dot-Product Computation for Low-Power Convolutional Neural Networks," *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 217–230, 2019.

[2] B. Murmann, "Mixed-Signal Computing for Deep Neural Network Inference," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 1, pp. 3–13, 2021.

[3] S. Yu, W. Shim, X. Peng, and Y. Luo, "RRAM for Compute-in-Memory: From Inference to Training," vol. 68, no. 7, 2021, pp. 2753–2765.

[4] S. Subhechha, N. Rassoul, A. Belmonte, R. Delhougne, K. Banerjee, G. L. Donadio, H. Dekkers, M. J. van Setten, H. Puliyalil, M. Mao, S. Kundu, M. Pak, L. Teugels, D. Tsvetanova, N. Bazzazian, L. Klijs, H. Hody, A. Chasin, J. Heijlen, L. Goux, and G. S. Kar, "First demonstration of sub-12 nm L_g gate last IGZO-TFTs with oxygen tunnel architecture for front gate devices," in *2021 Symposium on VLSI Technology*, 2021, pp. 1–2.

[5] J. Doevenspeck, K. Garello, B. Verhoef, R. Degraeve, S. Van Beek, D. Crotti, F. Yasin, S. Couet, G. Jayakumar, I. A. Papistas, P. Debacker, R. Lauwereins, W. Dehaene, G. S. Kar, S. Cosemans, A. Mallik, and D. Verkest, "SOT-MRAM Based Analog in-Memory Computing for DNN Inference," in *IEEE Symp. on VLSI Technology*, 2020, pp. 1–2.

[6] P. Houshmand, S. Cosemans, L. Mei, I. Papistas, D. Bhattacharjee, P. Debacker, A. Mallik, D. Verkest, and M. Verhelst, "Opportunities and Limitations of Emerging Analog in-Memory Compute DNN Architectures," in *2020 IEEE International Electron Devices Meeting (IEDM)*, 2020, pp. 29.1.1–29.1.4.

[7] J. Lim, M. Choi, B. Liu, T. Kang, Z. Li, Z. Wang, Y. Zhang, K. Yang, D. Blaauw, H.-S. Kim, and D. Sylvester, "AA-ResNet: Energy Efficient All-Analog ResNet Accelerator," in *2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2020, pp. 603–606.

[8] K. Zhou, C. Zhao, J. Fang, J. Jiang, D. Chen, Y. Huang, M. Jing, J. Han, H. Tian, X. Xiong, Q. Liu, X. Xue, and X. Zeng, "An Energy Efficient Computing-in-Memory Accelerator With 1T2R Cell and Fully Analog Processing for Edge AI Applications," vol. 68, no. 8, 2021, pp. 2932–2936.

[9] Q. Liu, B. Gao, P. Yao, D. Wu, J. Chen, Y. Pang, W. Zhang, Y. Liao, C.-X. Xue, W.-H. Chen, J. Tang, Y. Wang, M.-F. Chang, H. Qian, and H. Wu, "A Fully Integrated Analog ReRAM Based 78.4TOPS/W Compute-In-Memory Chip with Fully Parallel MAC Computing," in *IEEE Int. Solid- State Circuits Conf. (ISSCC)*, 2020, pp. 500–502.

[10] Q. Dong, M. E. Sinangil, B. Erbagci, D. Sun, W.-S. Khwa, H.-J. Liao, Y. Wang, and J. Chang, "A 351TOPS/W and 372.4GOPS Compute-in-Memory SRAM Macro in 7nm FinFET CMOS for Machine-Learning Applications," in *IEEE Int. Solid- State Circuits Conf. (ISSCC)*, 2020, pp. 242–244.

[11] Z. Jiang, S. Yin, J.-S. Seo, and M. Seok, "C3SRAM: An In-Memory-Computing SRAM Macro Based on Robust Capacitive Coupling Computing Mechanism," *IEEE J. Solid-State Circuits*, vol. 55, no. 7, pp. 1888–1897, 2020.

[12] J.-H. Yoon, M. Chang, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, and A. Raychowdhury, "A 40-nm, 64-Kb, 56.67 TOPS/W Voltage-Sensing Computing-In-Memory/Digital RRAM Macro Supporting Iterative Write With Verification and Online Read-Disturb Detection," *IEEE J. Solid-State Circuits*, vol. 57, no. 1, pp. 68–79, 2022.

[13] Z. Chen, H. Zhou, and J. Gu, "R-Accelerator: An RRAM-Based CGRA Accelerator With Logic Contraction," vol. 27, no. 11, 2019, pp. 2655–2667.

[14] 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 *2021 IEEE Custom Integrated Circuits Conference (CICC)*, 2021, pp. 1–2.

[15] S. R. S. Raman, S. Xie, and J. P.Kulkarni, "Compute-in-eDRAM with Backend Integrated Indium Gallium Zinc Oxide Transistors," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021, pp. 1–5.

[16] K. Garello, F. Yasin, and G. S. Kar, "Spin-Orbit Torque MRAM for ultra-fast embedded memories: from fundamentals to large scale technology integration," in *2019 IEEE 11th International Memory Workshop (IMW)*, 2019, pp. 1–4.

[17] J. Doevenspeck, K. Garello, S. Rao, F. Yasin, S. Couet, G. Jayakumar, A. Mallik, S. Cosemans, P. Debacker, D. Verkest, R. Lauwereins, W. Dehaene, and G. Kar, "Multi-pillar SOT-MRAM for Accurate Analog in-Memory DNN Inference," in *2021 Symposium on VLSI Technology*, 2021, pp. 1–2.

[18] A. Belmonte, H. Oh, N. Rassoul, G. Donadio, J. Mitard, H. Dekkers, R. Delhougne, S. Subhechha, A. Chasin, M. J. van Setten, L. Kljucar, M. Mao, H. Puliyalil, M. Pak, L. Teugels, D. Tsvetanova, K. Banerjee, L. Souriau, Z. Tokei, L. Goux, and G. S. Kar, "Capacitor-less, Long-Retention ($>$ 400s) DRAM Cell Paving the Way towards Low-Power and High-Density Monolithic 3D DRAM," in *2020 IEEE International Electron Devices Meeting (IEDM)*, 2020, pp. 28.2.1–28.2.4.

[19] J. Doevenspeck, R. Degraeve, S. Cosemans, P. Roussel, B.-E. Verhoef, R. Lauwereins, and W. Dehaene, "Analytic variability study of inference accuracy in RRAM arrays with a binary tree winner-take-all circuit for neuromorphic applications," in *2018 48th European Solid-State Device Research Conf. (ESSDERC)*, 2018, pp. 62–65.

[20] S. Cosemans, B. Verhoef, J. Doevenspeck, I. A. Papistas, F. Catthoor, P. Debacker, A. Mallik, and D. Verkest, "Towards 10000 TOPS/W DNN Inference with Analog in-Memory Computing – A Circuit Blueprint, Device Options and Requirements," in *IEEE Int. Electron Devices Meeting (IEDM)*, 2019, pp. 22.2.1–22.2.4.

[21] C. Huang, N. Xu, K. Qiu, Y. Zhu, D. Ma, and L. Fang, "Efficient and Optimized Methods for Alleviating the Impacts of IR-Drop and Fault in RRAM Based Neural Computing Systems," vol. 9, 2021, pp. 645–652.

[22] M. Caselli, I. A. Papistas, S. Cosemans, A. Mallik, P. Debacker, and D. Verkest, "Charge sharing and charge injection a/d converters for analog in-memory computing," in *2021 19th IEEE Int. New Circuits and Systems Conference (NEWCAS)*, 2021, pp. 1–4.

[23] S. Yin, X. Sun, S. Yu, and J.-S. Seo, "High-Throughput In-Memory Computing for Binary Deep Neural Networks With Monolithically Integrated RRAM and 90-nm CMOS," vol. 67, no. 10, 2020, pp. 4185–4192.

[24] K. Ueyoshi, I. A. Papistas, P. Houshmand, G. M. Sarda, V. Jain, M. Shi, Q. Zheng, S. Giraldo, P. Vrancx, J. Doevenspeck, D. Bhattacharjee, S. Cosemans, A. Mallik, P. Debacker, D. Verkest, and M. Verhelst, "15.6 DIANA: An End-to-End Energy-Efficient DIgital and ANAlog Hybrid Neural Network SoC," in *2022 IEEE International Solid- State Circuits Conference - (ISSCC) - Accepted*, 2022.

[25] M. Caselli, S. Subhechha, P. Debacker, A. Mallik, and D. Verkest, "Write-Verify Scheme for IGZO DRAM in Analog in-Memory Computing," in *IEEE Int. Symposium on Circuits and Systems (ISCAS) - Accepted*, 2022.

[26] C.-X. Xue, T.-Y. Huang, J.-S. Liu, T.-W. Chang, H.-Y. Kao, J.-H. Wang, T.-W. Liu, S.-Y. Wei, S.-P. Huang, W.-C. Wei, Y.-R. Chen, T.-H. Hsu, Y.-K. Chen, Y.-C. Lo, T.-H. Wen, C.-C. Lo, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, and M.-F. Chang, "A 22nm 2Mb ReRAM Compute-in-Memory Macro with 121-28TOPS/W for Multibit MAC Computing for Tiny AI Edge Devices," in *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, 2020, pp. 244–246.

[27] X. Si, W.-S. Khwa, J.-J. Chen, J.-F. Li, X. Sun, R. Liu, S. Yu, H. Yamauchi, Q. Li, and M.-F. Chang, "A Dual-Split 6T SRAM-Based Computing-in-Memory Unit-Macro With Fully Parallel Product-Sum Operation for Binarized DNN Edge Processors," *IEEE Trans. Circuits Syst. I*, vol. 66, no. 11, pp. 4172–4185, 2019.

[28] H. Valavi, P. J. Ramadge, E. Nestler, and N. Verma, "A 64-Tile 2.4-Mb In-Memory-Computing CNN Accelerator Employing Charge-Domain Compute," *IEEE J. Solid-State Circuits*, vol. 54, no. 6, pp. 1789–1799, 2019.

[29] S. Zhang, K. Huang, and H. Shen, "A Robust 8-Bit Non-Volatile Computing-in-Memory Core for Low-Power Parallel MAC Operations," *IEEE Trans. Circuits Syst. I*, vol. 67, no. 6, pp. 1867–1880, 2020.

[30] A. Boni, F. Frattini, and M. Caselli, "Time-Multiplexed Flash ADC for Deep Neural Network Analog in-Memory Computing," in *Proc. of IEEE Int. Conf. on Electronics Circuits and Systems (ICECS)*, 2021, pp. 1–4.

[31] Z. Cai, X. He, J. Sun, and N. Vasconcelos, "Deep Learning with Low Precision by Half-Wave Gaussian Quantization," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5406–5414.

[32] S. Yin, Z. Jiang, J.-S. Seo, and M. Seok, "XNOR-SRAM: In-Memory Computing SRAM Macro for Binary/Ternary Deep Neural Networks," *IEEE J. Solid-State Circuits*, vol. 55, no. 6, pp. 1733–1743, 2020.

[33] Z. He and D. Fan, "Simultaneously Optimizing Weight and Quantizer of Ternary Neural Network Using Truncated Gaussian Approximation," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 430–11 438.

[34] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight Uncertainty in Neural Networks," 2015, arXiv:1505.05424.

[35] H. Kim, J. Park, C. Lee, and J.-J. Kim, "Improving Accuracy of Binary Neural Networks using Unbalanced Activation Distribution," 2021, arXiv:2012.00938.