IEEE Journal on Exploratory Solid-State Computational Devices and Circuits

Received 16 May 2023; revised 7 July 2023 and 14 August 2023; accepted 23 August 2023.
Date of publication 29 August 2023; date of current version 5 December 2023.

Digital Object Identifier 10.1109/JXCDC.2023.3309713

Boosting RRAM-Based Mixed-Signal Accelerators in FD-SOI Technology for ML Applications

ANDREA BONI (Member, IEEE), FRANCESCO MALENA (Student Member, IEEE), FRANCESCO SACCANI (Student Member, IEEE), MICHELE AMORETTI (Senior Member, IEEE), and MICHELE CASELLI (Member, IEEE)

Department of Engineering and Architecture, University of Parma, 43124 Parma, Italy

CORRESPONDING AUTHOR: A. BONI (andrea.boni@unipr.it)

This work was supported by the European Union under Grant PON Research and Innovation 2014-2020 DM n.1062-2021.

This article has supplementary downloadable material available at https://doi.org/10.1109/JXCDC.2023.3309713, provided by the authors.

ABSTRACT This article presents the flipped (F)-2T2R resistive random access memory (RRAM) compute cell enhancing the performance of RRAM-based mixed-signal accelerators for deep neural networks (DNNs) in machine-learning (ML) applications. The F-2T2R cell is designed to exploit the features of the FD-SOI technology and it achieves a large increase in cell output impedance, compared to the standard 1-transistor 1-resistor (1T1R) cell. The article also describes the modeling of an F-2T2R-based accelerator and its transistor-level implementation in a 22-nm FD-SOI technology. The modeling results and the accelerator performance are validated by simulation. The proposed design can achieve an energy efficiency of up to 1260 1 bit-TOPS/W, with a memory array of 256 rows and columns. From the results of our analytical framework, a ResNet18, mapped on the accelerator, can obtain an accuracy reduction below 2%, with respect to the floating-point baseline, on the CIFAR-10 dataset.

INDEX TERMS Analog in-memory computing, convolutional neural networks, FD-SOI, mixed-signal accelerators, resistive random access memory (RRAM).

I. INTRODUCTION

Deep neural networks (DNNs) have achieved large success in a wide variety of machine-learning (ML) applications, from image classification to object detection and speech recognition [1]. In recent years, mixed-signal accelerators have emerged as a valuable solution to maximize throughput and energy efficiency, in DNN algorithm execution. These processing units operate with low-precision operands, obtaining high classification accuracy, with a great reduction of energy consumption [2]. Accelerators exploit the concept of analog in-memory computation (AiMC) to reduce data movement between memory and processor [3]. Matrix–vector multiplications (MVMs) for DNN inference are realized directly in computing memory cores, where multiplications and accumulations (MACs) of weights w and input activations a are realized in the analog domain. The memory cells, used to store the pretrained $w_{i,j}$, are arranged in crossbar arrays, and the activations a_i, representing the input data of each layer, are transmitted along the crossbar rows, as shown in Fig. 1. The MVM results Y_j are stored as analog quantities on the array columns, called summation lines (SLs), and digitized by means of analog to digital converters (ADCs), for nonlinear operation in the digital domain. In the last years, several nonstandard memory devices have been exploited for weight storage [1]. By the inclusion of these memories, mixed-signal accelerators can outperform the standard SRAM-based

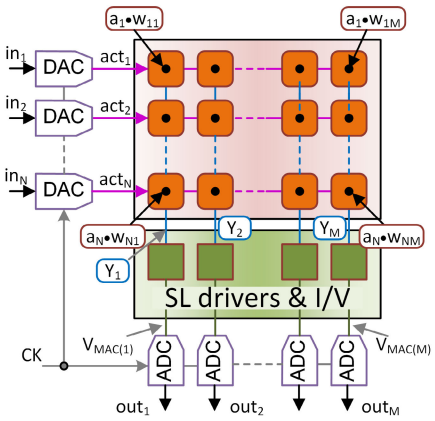

FIGURE 1. Block diagram of the memory accelerator for DNN with I/O interfaces.

implementations, at the cost of additional challenges in process integration and circuit design [2]. Among nonstandard devices, resistive random access memory (RRAM) is nonvolatile, with potential high memory density, and can offer multibit weight capability [4]. Mixed-signal accelerators embedding RRAM are often based on the 1-transistor 1-resistor (1T1R) cell, which integrates the resistive memory and a selector for the write. Several challenges, related to the relatively low RRAM resistance values, the small programming range, and the variability of the resistance levels, affect this compute cell [5], [6]. The low value of the programmable resistance is particularly critical for the computation linearity, making the MVM result severely affected by the IR drop along the column. This forces countermeasures in hardware, including integrator circuit or transimpedance amplifier (TIA) to clamp SL, with penalties for accelerator area and energy consumption [2]. Reducing the number of cells in computation can also alleviate this issue, but it also reduces the energy efficiency [7]. A compute cell, based on the position switching of the RRAM device and the access transistor to increase the cell output impedance, was recently proposed in [8]. Despite the significant reduction of the cell current, the accelerator proposed for this cell still exploits the clamp on the SL, operating in the continuous time domain. This leads to high values of energy per computation and penalizes the computation accuracy, due to low resolution of the activations and the integration of the compute-cell output noise over the whole clamp bandwidth.

In this article, we describe and analyze an accelerator for DNNs, based on the novel flipped (F)-2T2R compute cell. By design, the F-2T2R has high output impedance in processing, allowing the removal of costly clamping circuits and activation in parallel of a large number of cells, without penalty for the MVM linearity. Moreover, it enables multibit activations and accelerators operating in the discrete-time domain, with large benefits in terms of computation efficiency. The F-2T2R compute cell, exploiting the features of the FD-SOI technology for the RRAM writing tasks, is described and analyzed in Section II. In Section III, we propose the design in 22-nm technology of a memory accelerator, integrating the new F-2T2R. A MATLAB model of the accelerator, including the most significant circuits nonidealities, has been developed for a system-level optimization and it is described in Section IV. The model and the accelerator hardware metrics are verified with simulations at the transistor level, reported in Section V. The F-2T2R accelerator performance has been also validated on an ML benchmark, with an in-house analytical framework. The results are reported in Section VI.

II. F-2T2R: A NOVEL CELL FOR TIA-LESS AND HIGHLY PARALLEL RRAM-BASED ACCELERATOR

Mixed-signal accelerators in CMOS technology, with embedded RRAM, are typically based on the 1T1R compute cell scheme, where a MOS transistor is used as a switch for cell selection, during the write and the computation phases, and the weight is stored as the resistance value R_r of a RRAM memory device [7]. This approach is common in literature, and Fig. 2(a) shows an example array column of a charge-domain RRAM accelerator based on the 1T1R compute cell. R_r can be programed over L_r values, from

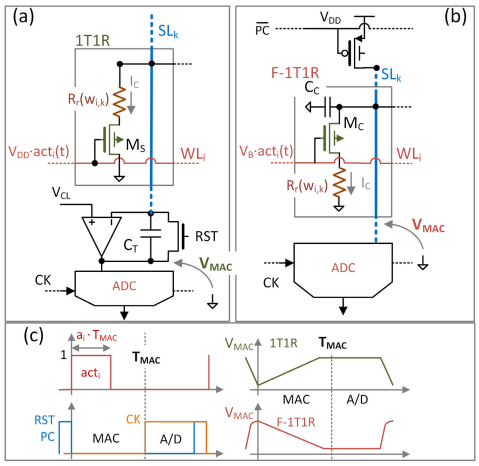

FIGURE 2. (a) 1T1R cell RRAM accelerator column, (b) proposed F-1T1R compute cell column, and (c) timing diagrams of the two architectures.

the minimum R_L to the maximum R_H, to obtain equally spaced resistance values. At constant SL voltage, this corresponds to equally spaced current levels in the compute cell. Typically, L_r ranges from 2 to 8 values [9]. The scheme in Fig. 2(a) includes a digital-to-time D/A converter (DAC) to drive the gates of the MOS switches of the 1T1R cells, on the same row, with the pulsewidth modulated (PWM) signal $V_{\mathrm{DD}} \cdot \mathrm{act}_i(t)$[1] [2]. The length of the pulse is proportional to the activation value, that is, $T_{\mathrm{MAC}} \cdot a_i$ with $a_i \in [0, 1]$. The voltage of the SL is clamped by means of a TIA, implementing the charge-to-voltage conversion [10]. Thus, the current sunk by the ith cell $I_{c(i)}$ in the MAC phase is, in first approximation, proportional to the conductance of the RRAM device and enabled for a duration equal to $T_{\mathrm{MAC}} \cdot a_i$. The voltage V_{MAC} at the output of the TIA represents the MAC result. Despite the good computation linearity, this architecture is severely penalized by the high power consumption and the large silicon area of the TIA. Both of them are noticeably dependent on the current on SL, which is proportional to the number of rows in the array. Additionally, the TIA should not be a computation speed bottleneck, but the bandwidth of the amplifier and its power consumption are directly related.

Fig. 2(b) shows a novel structure to exploit the RRAM memory in AiMC. In the column, the flipped (F)-1T1R compute cell replaces the conventional 1T1R. The NMOS device, acting here as a current generator, is biased in saturation, when on-state. To this aim, the amplitude of the PWM signal, V_B, sets the bias of the transistor during the activation pulse, that is, with $\mathrm{act}_i = 1$. Thus, the cell current I_c is set by the programed RRAM resistance value and by V_B. The new configuration of the NMOS device guarantees a much larger output resistance than the 1T1R cell. This allows the removal of the TIA for a drastic circuit simplification and area reduction, with additional benefits in terms of energy consumption and computation speed. This column scheme is now based on the precharge–discharge approach, where SL

[1] $\mathrm{act}_i(t)$ is a PWM-modulated signal with normalized unitary amplitude.

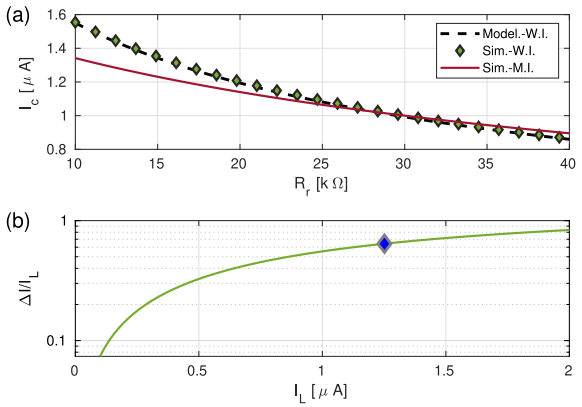

FIGURE 3. F-1T1R compute cell. (a) I_C versus R_r with M_C in W.I. (green diamonds) and in M.I. (red line), and I_C from (1) (black dashed line). (b) Green line: $\Delta I/I_L$ versus I_L with $R_L = 10$ kΩ, $R_H = 30$ kΩ, M_C in W.I. region; diamond: F-1T1R design point for a 256-row accelerator from our analytical framework.

is first precharged and then discharged by the analog MAC. The SL capacitance C_{SL} is used to integrate the output current while computing and the SL voltage represents the MAC result.

In the F-1T1R of Fig. 2(b), with the transistor biased in weak inversion (W.I.) and saturation region, the cell current I_c is

$$I_c = \frac{n \cdot v_{\mathrm{th}}}{R_r} \cdot W\left(\frac{I_{c0} \cdot R_r}{n \cdot v_{\mathrm{th}}}\right) \tag{1}$$

where $W(x)$ is the Lambert W function solving the equation $y \cdot \exp(y) = x$ [11], v_{th} the thermal voltage, n the slope factor, and I_{c0} the equivalent current of M_C with its source connected to ground. At the end of the MAC computation, the SL voltage (corresponding to V_{MAC}) is

$$V_{\mathrm{SL}(k)} = V_{\mathrm{DD}} - \frac{\Delta I \cdot T_{\mathrm{MAC}}}{C_{\mathrm{SL}}} \sum_{i=1}^{N} \left(a_i \cdot w_{i,k}\right) + K_a \tag{2}$$

$$K_a \equiv -\frac{N \cdot I_L \cdot T_{\mathrm{MAC}}}{C_{\mathrm{SL}}} \cdot \mu\,(a) \tag{3}$$

where $\Delta I \equiv I_H - I_L$ is the maximum current step, with $I_L \equiv I_c(R_H)$ and $I_H \equiv I_c(R_L)$, $\mu(a)$ is the mean value of the activations, and $C_{\mathrm{SL}} = N \cdot C_C$, with C_C the load capacitance, embedded in the compute cell. The RRAM device is programed to obtain L_r equally spaced cell current $I_c = I_L + \Delta I \cdot w_i$, where w_i is the normalized unipolar weight, that is, $w_i \in [0, 1]$. Starting from these equations, the plots in Fig. 3 were obtained, using a 1.5-V SOI MOS transistor, available in the 22-nm technology, with a gate area of 0.7 μm^2. Moderate inversion (M.I.) and W.I. regions were considered for the transistor, with g_M/I_D of 26 V^{-1} and 16 V^{-1}, respectively, at $I_c = 1$ μA. Fig. 3(a) shows the simulated I_c versus R_r, while Fig. 3(b), the F-1T1R normalized current variation $\Delta I/I_L$ versus I_L. A large ΔI corresponds to a larger signal on the SL, which relaxes the ADC specifications [2]. From the simulations, $\Delta I/I_L$ is maximized in the W.I. regime and increases with I_L.

A. F-2T2R DIFFERENTIAL COMPUTE CELL

The F-2T2R compute cell, made with two F-1T1R cells in a differential scheme, is shown in Fig. 4. This configuration can

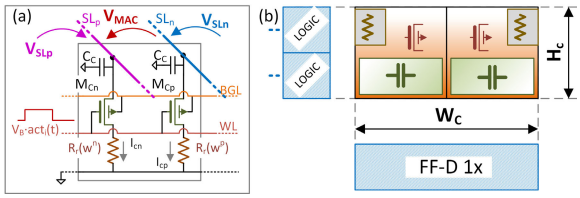

FIGURE 4. F-2T2R differential cell. (a) Schematic and (b) layout floorplan.

provide several benefits in the MAC computation. In fact, the MAC output voltage in (2) is not proportional to the MAC result, due to the term K_a depending on I_L. K_a cannot be neglected due to the limited ratio R_H/R_L, achievable by the current RRAM technologies. Hence, this issue is intrinsic in every RRAM compute cell. The correct MAC value could be restored either by digital postprocessing or by setting the center of the ADC range at $V_{\mathrm{DD}} + K_a$, but both options exhibit major drawbacks. As a matter of fact, the former requires a larger ADC range and a higher converter resolution, whereas the latter requires the ADC range to track K_a value in every column of the array. The F-2T2R compute cell with differential output overcomes this problem. As shown in Fig. 4, two memristors are used to store the weight in bipolar format, increasing the resolution to $L_w = 2 \cdot L_r - 1$. The weight w_i, ranging over $[-1, 1]$, is split between the memristors, as w_i^p and w_i^n, with $w_i = w_i^p - w_i^n$

$$w_i^p = \frac{\mathrm{sgn}(w_i) + 1}{2} \cdot w_i, \quad w_i^n = \frac{\mathrm{sgn}(w_i) - 1}{2} \cdot w_i. \tag{4}$$

With the cell in computation, the values of the positive and negative output currents are

$$I_{cp} = I_L + \Delta I \cdot w_i^p, \quad I_{cn} = I_L + \Delta I \cdot w_i^n. \tag{5}$$

The single-ended voltage of the two SLs, $V_{\mathrm{SL}p}$ and $V_{\mathrm{SL}n}$, is given by (2), whereas the differential voltage is proportional to the MAC result

$$V_{\mathrm{MAC}(k)} = \frac{\Delta I \cdot T_{\mathrm{MAC}}}{C_{\mathrm{SL}}} \sum_{i=1}^{N} \left(a_i \cdot w_{i,k}\right). \tag{6}$$

With the F-2T2R cell, a differential ADC, with a symmetric conversion range can be used, thereby resulting in another major design simplification [12]. At the cell level, the proposed F-2T2R clearly requires a larger area, when compared with the 1T1R cell. However, as explained in Section IV, this penalty becomes almost negligible if the whole memory accelerator is considered.

B. RRAM PROGRAMMING

The writing procedure of the RRAM memories severely impacts the integration of this technology with scaled CMOS processes. RRAM devices are programed through the forming, set, and reset operations [13]. Forming is performed to change the RRAM device from the pristine condition to a low resistance state (LRS). The resistance value is then programed by iterating between set and reset phases. Currently, RRAM devices embedded in CMOS nodes require a forming voltage higher than 3 V, exceeding the safe operating area (SOA) of core devices in sub-100-nm technologies [4], [14]. The voltage required in the other operations is lower, but still not compatible with devices of scaled

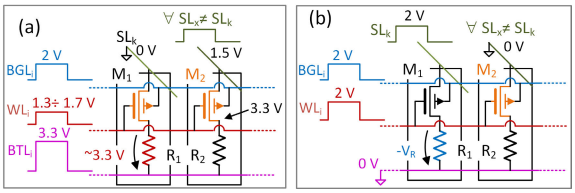

FIGURE 5. RRAM accelerator under forming/set (a) and reset (b), with a cell in the *k*-column addressed (a subsection with only two cells in the same row is considered). Devices in orange are subject to the maximum voltage stress.

CMOS technologies [13], [15]. The current is another key aspect in the writing phase since it must be limited to avoid damaging the RRAM device during forming and constrained to set the resistance value in the set-programming procedure. These issues are typically addressed with dedicated writing circuits, based on I/O transistors. Considering the small sizes of the RRAM tiles, the additional area usually turns out to be very large [7]. For safe writing of each F-1T1R in the differential cell, our design exploits the four voltage levels (from 0.8 to 1.8 V) available in the 22-nm FD-SOI technology and threshold voltage control, through the back-gate, of the MOS transistor M_i. The proposed implementations of the forming/set and reset operations for a differential cell are shown in Fig. 5(a) and (b), respectively, with the programming voltage waveforms. In the forming phase, the transistor M_1 is configured in a common-source configuration to maximize the voltage drop across the RRAM device. Hence, WL is driven with a voltage pulse at V_{WFS}, the SL is tied to the ground, and a pulse with a voltage up to 3.3 V drives BTL. The other cells are excluded from the programming operation, by setting the corresponding SLs to 1.5 V and their BTL and WL to ground. A 1.5-V SOI device, featuring an SOA rating voltage of 2 V, is used in the compute cell, to be compliant with a BTL voltage of 3.3 V and a WL voltage in the 1.3- to 1.7-V range. The set phase of the programming procedure is similar to the forming phase, but with lower RRAM voltage and current bounds, thus the BTL and WL voltages are reduced accordingly.

In reset, the polarity of the voltage across the RRAM device is reversed by setting both SL and WL at 2 V and BTL to ground. To mitigate the effect of the gate–source voltage of M_1 on the effective voltage across the RRAM, the back-gate of the flipped-well SOI device is exploited, by raising the BGL line voltage of the specific cell to 2 V. With this solution, a reset voltage V_R of 1.5 V, at 30 μA can be obtained, with a leakage current well below 1 nA, for the other cells in the column. It is worth noting that, differently from standard writing procedures for 1T1R cells, this programming technique guarantees a maximum SL voltage much lower than the 3-V forming voltage. This allows the design of the ADC front-end with SOI devices, avoiding high-voltage transmission gates at the converter input, with a substantial saving in silicon area per column.

III. F-2T2R-BASED MEMORY ACCELERATOR

This section describes the design of a mixed-signal accelerator, tailored for the F-2T2R cell computing memory, and exploiting the features of the SOI technology. The block diagram, including the I/O interfaces and the SL driving circuits, is shown in the red dashed box of Fig. 6(a). The circuit in Fig. 6(b) is the row-interface for input activations, including a PWM DAC [16] and an analog multiplexer, based on the NMOS M_{B3} and the transmission gate TG_{B1}, converting the output signal of the DAC from logic levels to the 0-V_B range. SOI devices, with 1.8-V nominal voltage supply (and 2-V strength) and compliant with the programming procedure in Fig. 5, drive the WLs. TG_{B1} is also driven with 1.8-V signals for low on-resistance in MVM mode and the full PMOS switch-off in write mode. A level shifter is used to convert the DAC output signal from 0.85 to 1.8 V. Only SOI devices are used in the block, keeping the cell height within the vertical pitch of the memory array, for a compact layout. The SL driver of each differential column is implemented with a pair of PMOS transistors, connected to the precharge voltage V_{PC}, rail, and with the common-mode control (CMC) circuit. The PMOS device of each SL in Fig. 6(a) is switched on, for a short time interval before the MAC phase, to precharge the SL to V_{PC}. On the other hand, the CMC is activated only during the MAC phase to reduce the spread and drop of the common-mode voltage of the SL. This circuit is designed to reduce the operand K_a in (2) and ensure the maximum values of I_L and of ΔI in Fig. 3, compatible with the lower bound of $V_{\mathrm{SL}p,n}$. Indeed, the minimum V_{SL} in the MAC is lower bounded by the minimum ADC common-mode input voltage and by the requirements for the saturation regime of the M transistors in the compute cells. In short, the CMC enables a larger swing of V_{MAC}, a larger MAC LSB, and relaxes the specifications for the column ADC. Moreover, combined with the large output impedance of the F-2T2R, allows the simultaneous exploitation of all the cells in the column, practically removing the limit on the maximum number of cells enabled, common in the 1T1R implementations [7]. The schematic of the CMC circuit is shown in Fig. 6(c). The current generator, made with the transistors M_{C1}-M_{C2}, injects the current I_{CM} into each SL, introducing an additive term in (2), to partially cancel the term K_a. M_{C3} and M_{C4} are used as cascode devices when the CMC circuit is activated, and as switches when the current generators are switched off. This compensation circuit is driven by a PWM DAC, as those used for the act_i signals. We propose two common-mode compensation strategies, both technology-agnostic. In Type-1 CMC, the length of the CM current pulse, always within the T_{MAC} phase, is digitally calibrated to reduce the $\mu(a)/C_{\mathrm{SL}}$-dependent term of V_{SL} and kept constant over all the MAC periods. With Type-2 CMC, the pulselength is adjusted at each MAC period and held proportional to the sample mean of $\mu(a)$ in each input frame. Thus, the compensating CM current, injected by the pulsed current generator, in Fig. 6(c), is highly correlated with the CM charge removed from the SL capacitance by the activated cells. This causes, a narrower probability density function (pdf) distribution for type-2 CMC, and a generally better performance, but it requires the sample mean $\mu(a)$ to be computed in real-time in the digital domain. The performance boost, provided by the CMC, is discussed in Section IV and assessed with simulations. V_B in MVM and writing is provided by the reference generator in Fig. 6(d), based on a replica of the F-1T1R cell biased with I_L or the maximum current in forming/set, I_W.

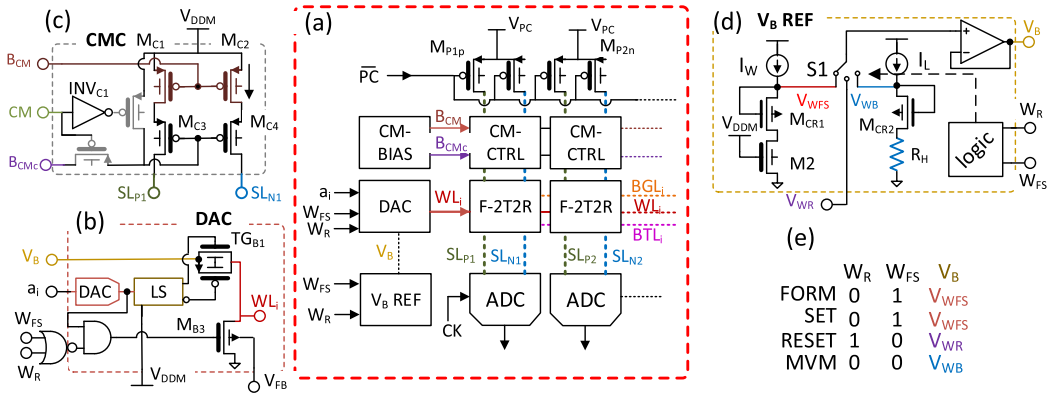

FIGURE 6. **(a) Accelerator block diagram (core supply voltage V_{DD} = 0.85 V, $V_{PC} = V_{DD}$, V_{DDM} = 1.8 V). (b) CM control. (c) DAC with interface circuit to the crossbar array. (d) V_B reference generator. (e) Truth table for W_R, W_{FS}, and $S1$ switch for V_B setting.**

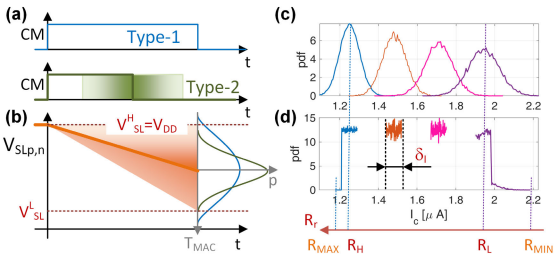

FIGURE 7. **(a) CM waveforms. (b) $V_{SLp,n}$ average trajectory (orange) and pdf at T_{MAC}, with type-1 (blue) and type-2 (green) CMC. PDF of I_C including MOS mismatch w/o (c) and with (d) mismatch cancellation (L_r = 4).**

IV. MIXED-SIGNAL ACCELERATOR MODEL

A MATLAB model has been developed to optimize the design of the F-2T2R-based mixed-signal accelerator, considering for the analysis the multilevel RRAM device described in [9]. For accurate modeling, it includes the main sources of error affecting the computation of V_{MAC} in the analog domain: the over-range clipping of $V_{SLp,n}$, the mismatch of the current-source MOS transistor in the compute cell, the quantization noise and the clipping errors introduced by the ADC, and the finite output resistance of the compute cell. To assess the impact of these error sources, they must be compared to the MAC error baseline e_{awQ}, derived by the digital quantization

$$e_{awQ} = V^*_{MAC} - V_{MAC} \tag{7}$$

$$\sigma_{awQ} \equiv \sigma\left(e_{awQ}\right) \tag{8}$$

where V^*_{MAC} is the MAC result computed with floating-point activations and weights, and σ_{awQ} is the standard deviation of e_{awQ}. The SL voltage should not exceed the lower-bound V^L_{SL}, set by the ADC and by the saturation limits of $M_{Cp,n}$, and the upper-bound $V^H_{SL} = V_{PC}$ set by the precharge PMOS device on the top of each SL in Fig. 6(a). The waveform of the CM signal with both types of CMC is depicted in Fig. 7(a). The plot in Fig. 7(b) shows the mean trajectory of $V_{SLp,n}$, obtained with random extractions of activations and weights, and its distribution after the analog computation. Type-2 CMC results in a narrower distribution at the same I_L, leading to a lower occurrence of out-of-range errors.

Local mismatch variations affect $M_{Cp,n}$, and they lead to random shifts of I_c, from the nominal value. Fig. 7(c) shows the pdf of I_c for a four-level RRAM case. This error source can be reduced by calibrating the RRAM value to compensate for the mismatch affecting the cell current. The calibration range is limited by the maximum RRAM variation [R_{MIN}, R_{MAX}] and by the resistance spread, setting the minimum achievable resistance step ΔR in the mismatch calibration. The plot in Fig. 7(d) shows the results of an example I_c calibration, obtained with the developed MATLAB framework, with [R_{MIN}, R_{MAX}] = [8, 35] kΩ, [R_L, R_H] = [10, 30] kΩ, and ΔR = 400 Ω, extracted from [7], a gate area of 0.77 μm^2 for the MOS transistor, and the mismatch Pelgrom coefficient of a 22-nm technology. The calibration reshapes I_c pdf of the inner levels to narrow uniform distributions, but for the lowest and highest levels, due to the saturation occurring in the resistance calibration. The normalized weight variability is defined as $\epsilon_c = \delta_I/\Delta I$, where δ_I is the residual spread of the cell current after mismatch cancellation. It is worth noticing that a tradeoff stands between the programming range and the headroom left for mismatch calibration, that is, R_{MAX}-R_H and R_L-R_{MIN}: the larger the programming range, the higher ΔI and the lower the calibration range for the errors affecting the first and last cell-current level. The mismatch-induced current variation leads to MAC error e_M, with standard deviation σ_M.

The quantization noise and the clipping errors introduced by the ADC are considered in the model of the accelerator. To keep these errors conveniently below the σ_{awQ} baseline, the ADC LSB and conversion range are sized as

$$\text{LSB} = \frac{\sigma_{awQ}}{\alpha_Q} \cdot \sqrt{12} \tag{9}$$

$$\sigma_{OV}(V_R) = \sigma_{awQ}/\alpha_{OV} \tag{10}$$

where σ_{OV} is the rms value of the ADC clipping error, and α_Q and α_{OV} are design optimization coefficients, setting the ratio of σ_{awQ} to the quantization and overrange errors, respectively [17]. The adverse effect of the finite-cell output resistance can be reduced below the ADC LSB by increasing the length of the transistor $M_{cP,N}$, in the compute cell. In a

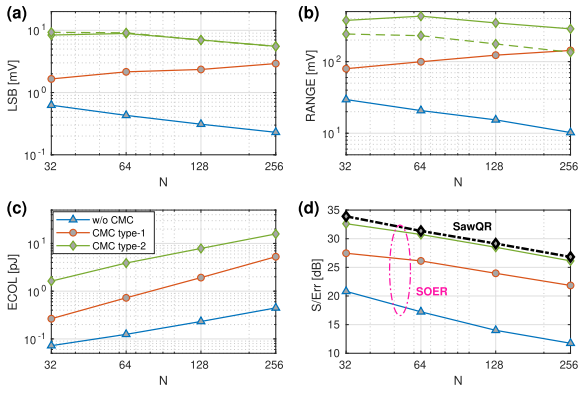

FIGURE 8. ADC and accelerator performance versus N. Blue, orange, and green lines: no CMC, type-1 CMC, and type-2 CMC. Dashed line: weights with normal pdf. (a) and (b) ADC LSB and conversion range. (c) Energy consumption per column. (d) SOER (solid lines) and ideal SawQR (black dotted-dashed).

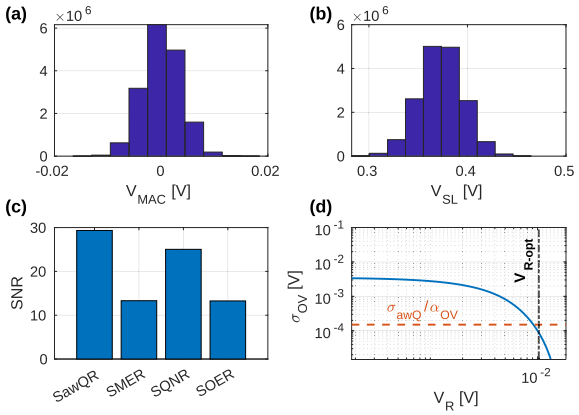

FIGURE 9. Accelerator without CM compensation for $N = 128$. (a) and (b) Distribution of V_{MAC} and $V_{SLp,n}$, (c) SNR metrics, and (d) ADC range optimization.

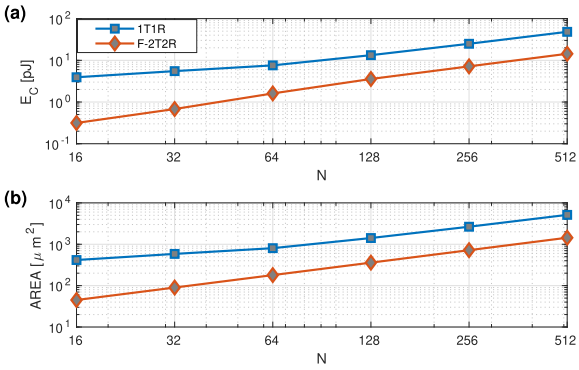

FIGURE 10. Comparison of 1T1R and F-2T2R memory accelerators versus N at $L_r = 2$ and $V_{CL} = V_{SL}^{L} = 0.3$ V. (a) Energy consumption per column and MAC operation and (b) estimated area of one column, without A/D and D/A converters.

fixed-area design, this leads to a smaller $\Delta I/I_L$ in Fig. 3, due to the lower g_M/I_D value of $M_{cP,N}$.

Design Optimization: From the previous analysis, we developed a MATLAB framework [17] for the optimized design of the accelerator in Fig. 6(a), with the F-2T2R cell. The optimization loop operates on the CMC, the A/D interface, and the parameters I_L and g_M/I_D of $M_{Cn,p}$. The plots in Fig. 8 show the result of a design optimization, with $L_r = 8$ and 7-bit activations, sweeping the number of memory rows N [9]. Mismatch calibration is enabled with $\alpha_Q = 10$ dB and $\alpha_{OV} = 20$ dB. I_L is optimized for the maximum V_{MAC} swing, but constrained within 1.2 μA, and the SL precharge voltage is set to $V_{PC} = 0.85$ V, at the core voltage supply. The simulation results reported in Fig. 8(a) and (b) demonstrate that the proposed common-mode compensation is beneficial for maintaining large ADC conversion range and LSB, relaxing the converter design and shrinking the area. Type-2 compensation provides better performance with respect to Type-1, resulting in the largest LSB value and the lowest loss of the signal-to-overall error ratio (SOER), with respect to the baseline, SawQR $= \sigma(V_{MAC})/\sigma_{awQ}$

$$\text{SOER} \equiv \frac{\sigma\,(V_{\text{MAC}})}{\sqrt{\sigma_{awQ}^2 + \sigma_M^2 + \sigma_{\text{ADC}}^2}} \tag{11}$$

where σ_{ADC} is the ADC quantization noise including clipping. Fig. 8(c) also reports the energy consumption per column E_C, excluding the A/D and D/A interfaces. From the graphs, the CM compensation leads to a higher energy consumption, due to the additional CM current I_{CM} and the larger compute cell current. This penalty is mitigated by the benefits derived by a larger ADC LSB [2]. Without the CM compensation, SOER is almost 15 dB below the baseline. This performance can be understood by looking at the graphs in Fig. 9, showing the distributions of V_{MAC} and V_{SL}, for $N = 128$, and I_L optimized without the CM compensation. Fig. 9(c) shows the SNR metrics of the accelerator, where SMER and SQNR are defined as the signal-to-mismatch-only error and the signal-to-ADC quantization noise: SMER $\equiv \sigma(V_{MAC})/\sigma_M$ and SQNR $\equiv \sigma(V_{MAC})/\sigma_{ADC}$, respectively. From the comparison, SMER achieves the lowest SNR, being penalized by the low I_L, required to limit the under-range clipping occurrence of $V_{SLp,n}$, when CMC is not enabled. As shown in Fig. 3, this leads to a small ΔI and, consequently, to larger overlap of the I_c pdfs, in Fig. 7(c). The design point of V_R, at the target over-range error $\sigma_{OV}(V_R) = \sigma_{awQ}/\alpha_{OV}$ is shown in Fig. 9(d).

Fig. 10 shows the comparison, in terms of energy and area, of an accelerator based on the F-2T2R with type-2 CMC, against an F-1T1R accelerator with SL clamp. $I_L = 1.25$ μA, returned by the optimization framework and shown in Fig. 3, is one order of magnitude lower than the typical current of 1T1R cells in accelerators operating in the continuous-time domain [5], and more than three times lower than I_L of the compute cell proposed in [8]. This allows the proposed accelerator to significantly reduce the energy consumption metric, with respect to conventional 1T1R designs, as shown in Fig. 3(a). The F-2T2R accelerator outperforms also on the area metric, due to the use of a TIA per column in the standard approach. For the area of the 1T1R, we used 0.07 μm^2, reported in [18].

V. ACCELERATOR TRANSISTOR-LEVEL IMPLEMENTATION

In this section, we propose the transistor-level design of an F-2T2R accelerator, in an FDSOI 22-nm technology node,

based on the information collected with the optimization framework. By comparison with the transistor-level accurate modeling, it is possible to verify the quality of our high-level accelerator model and assess its practical feasibility, when peripheral circuits are included. The design targets the minimum area occupation and it considers the memory accelerator in Fig. 6(a), with a number of rows $N = 256$, one input DAC per array row, and one output ADC per column. At the state of the technology, peripheral circuits based on silicon cannot scale at the size of the memory devices, due to fabrication imperfections, which penalize their performance with aggressive scaling [2]. Therefore, the area optimization of the accelerator starts from the minimum pitch that the I/O interfaces (PWM-DAC and ADC) can sustain, in the technology node used for the design, and following the floor plan proposed in Fig. 4(b). A PWM-DAC, based on a digital delay line, with a resolution in the 5-to-7-bit range, can be laid out on two rows of digital standard cells [16]. In 22-nm FD-SOI, this approximately corresponds to a vertical array pitch $H_c = 1.2\ \mu$m. The width of the column ADC sets the width of the column W_c and the horizontal memory pitch. Usually, SAR and Flash A/D converters are common choices in mixed-signal accelerators [12]. These ADCs use logic gates, and at least one D flip-flop (D-FF) is embedded in both converters. In scaled technology nodes, the digital cells must be laid out with homogeneous poly-gate orientation across all the die. Therefore, the minimum ADC width is set by the width of the D-FF, which is approximately 2.3 μm. From the vertical and horizontal pitches dictated by the peripheral circuits is obtained the upper bound of the gate area of $M_{Cp,n}$. The F-2T2R cell floor plan proposed in Fig. 4(b) is implemented with almost the whole cell area used for the current-source transistor since the RRAM device is implemented at the BEOL step, and $C_c = 2.2$ fF is a metal–oxide–metal (MOM) capacitor, involving the highest metal layers. The aspect ratio of $M_{Cp,n}$ is obtained from the minimum g_M/I_D bias, providing the current ratio $\Delta I/I_L$ optimized with the MATLAB framework. The device length is set by the output resistance specification. Considering the vertical poly orientation, $M_{Cp,n}$ is arranged as a dual-finger device, with a finger width close to H_c and a length of 350 nm. The CM compensation circuit shown in Fig. 6(c) is laid out with M_{C1}-to-M_{C4} as four-finger devices fitting the horizontal memory pitch. The area of the devices for the column precharge is 3% of the memory column area, while the CM-compensation cascode mirror requires an equivalent area from 3% to 2.2%, with N increasing from 64 to 256. The F-2T2R allows the implementation of the SL precharge–discharge. This makes the overall area of the SL driving circuits no larger than 6% of the whole accelerator area, with a massive area saving with respect to the conventional TIA-based column driving. Fig. 11 reports the results of the Spectre transistor-level simulations with type-2 CMC and without CMC. In our implementation, the computation latency is dictated by the PWM DAC, considering 7-bit activations less than 13 ns are required to generate the MVM results [16]. The system period for the precharge, the MAC computation, and the output quantization is 50 ns. The plot in Fig. 11(a) shows V_{MAC}, with 256 input activations set at the same value, normalized in the range [0; 1], and increased at each MAC step.

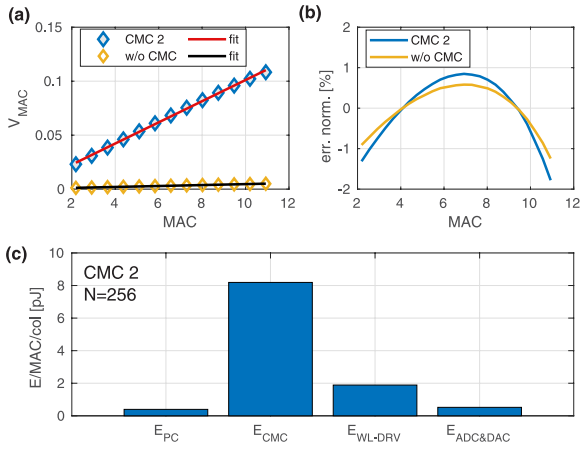

FIGURE 11. Transistor-level simulation results with $N = 256$, $L_r = 8$, without CMC and with type-2 CMC. (a) Simulated V_{MAC} (diamonds) and fit line versus MAC $= \sum a_i \cdot w_i$. (b) Relative linearity errors. (c) Energy consumption per column.

The weights, normalized in the range $[-1; 1]$, were randomly extracted with a nonnull average value, to have V_{MAC} spanning the ADC range shown in Fig. 8(b). Given that the MAC signal range decreases with $\sqrt{N}$, simulations are carried out for covering three standard deviations of the normalized V_{MAC} output distribution [19]. The results highlight the need for the CMC circuit to boost the dynamic range of V_{MAC}. The linear fitting of the simulated point returns a linearity error below 2%, as shown in Fig. 11(b), which is expected not to affect the DNN inference accuracy, being well below the equivalent noise floor of a 256-row accelerator, as shown in the SawQR plot of Fig. 8(d). The bar chart in Fig. 11(c) reports the breakdown of the energy consumption in the MAC phase, per single column, excluding the PWM-DAC. The main contributors are the precharge devices, E_{PC}, the CMC, E_{CMC}, and the DAC-to-memory interface of Fig. 6(b), $E_{\mathrm{WL\text{-}DRV}}$. Fig. 12 compares the computation efficiency of an F-2T2R accelerator with state-of-the-art RRAM accelerators [6], [7], [8], [20], [21], including only the consumption of the memory array and the interfaces, for a fair comparison. In our implementation, we consider the energy consumption of the DAC and of the ADC for mixed-signal accelerators, both in 22 nm, reported in [16] and [12]. An accelerator combining the F-2T2R cell with CMC type-2 is expected to achieve an energy efficiency, normalized to 1-bit MAC, of 1260 1 bit-TOPS/W [22], at $L_r = 8$, corresponding to 3.9-bit weight resolution, and 7-bit activations. This value is approximately ten times larger than the efficiency reported for the state-of-the-art 1T1R accelerators [5], [7], [10]. The result derives from the increased output impedance of the F-2T2R compute cell, which allows the precharge–discharge operation, the reduction of I_L compared to the 1T1R cell, and the simultaneous activation of hundreds of cells in parallel. Type-1 CMC achieves an efficiency comparable with type-2, whereas up to 4500 1 bit-TOPS/W could, in principle, be achieved without CMC. However, the almost complete loss of V_{MAC} range makes this option impracticable.

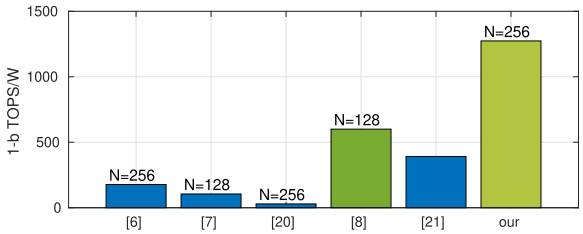

FIGURE 12. Computation efficiency of the F-2T2R RRAM accelerator with type-2 CMC versus state-of-the-art RRAM accelerators. Only energy consumption of the MVM array, interfaces, and SL driving was considered, and TOPS/W data are normalized at 1-bit MAC. Blue bars: measurements; green bars: simulations.

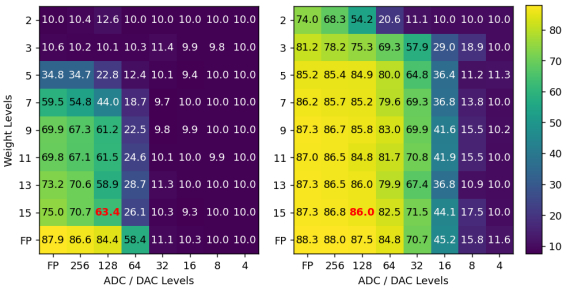

FIGURE 13. Accuracy of the model as the ADC/DAC and weights resolutions vary, considering a constant normalized weight variability of 2%. In red is the accelerator design point in the transistor-level implementation.

VI. F-2T2R ACCELERATOR PERFORMANCE ON BENCHMARK APPLICATIONS

To simulate the deployment of a DNN on a memory accelerator and evaluate the hardware impact on the software performance, an analytical framework was developed, based on the popular deep-learning library PyTorch [23]. This framework implements discretization, variability, and bounds management for model weights and activations that can be applied differently for each DNN parameter or module. More in detail, to emulate the writing process of weights into memory, values are mapped into a finite number of levels distributed linearly in a given interval, corresponding to the linearly spaced I_L levels, in the F-2T2R compute cell. Additive Gaussian noise is applied, whose standard deviation represents the mismatch current error and it is related to the whole discretization interval. Similarly, before and after each module, discretization, bounds, and noise can be applied to the activations to emulate the nonidealities derived from hardware implementation of digital-to-analog and analog-to-digital conversions. The framework assumes time-independent weight variability and that each layer can be fully contained in a single tile. The framework was tested using the ResNet-18 model [24] on the CIFAR-10 dataset [25], which consists of 50 000 training images and 10 000 test images classified into ten classes. After the hyperparameter tuning phase, the ResNet-18 achieved an accuracy of 88.4% on the test set, by training the model for up to 300 epochs using stochastic gradient descent (SGD), with 0.01 as the initial learning rate, 0.9 as momentum, 0.001 as weight decay, and cosine learning rate schedule.

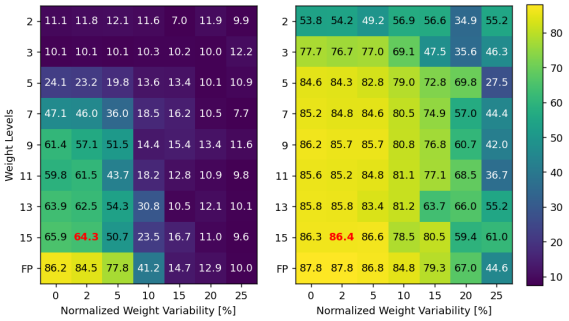

FIGURE 14. Accuracy of the model versus normalized weight variability and number of levels per weight. ADC/DAC resolution: 7 bits.

Several data augmentation techniques were applied to mitigate model over-fitting during the training phase. Trained weight distributions were studied to define the bounds of discretization intervals for each layer, which appeared to be critical for the model performance. In Fig. 13, the accuracy of the model is reported versus weight, ADC, and DAC resolutions, for $\epsilon_c = 2\%$. On the left, the complete DNN is mapped on the accelerator and so it is entirely affected by the analog nonidealities. On the right, Fig. 13 shows the results of a partial mapping scenario, where only the fourth ResNet-18 convolutional block, made of four layers, containing approximately 75% of the model parameters, is mapped onto the accelerator. For the full mapping, with w above seven levels of resolution, the algorithm already achieves a good classification accuracy, while 128 levels, corresponding to 7 bits of data resolution, seem to be the minimum for the peripheral circuits. For partial mapping, the hardware specifications can be drastically relaxed. The comparison between the two scenarios is shown also in Fig. 14, which reports the classification accuracy versus the weight resolution and ϵ_c, for ADC and DAC resolutions set at 7 bits. The results show the heavy impact of the weight variations on the network performance and the need for accurate weight writing, as the RRAM calibration procedure, reported in [7]. The partial mapping approach can achieve accuracy close to the baseline, still mapping the layers that contain most of the parameters. With ResNet-18, this approach should be preferred when the target is the best classification performance. It is worth underlining that the largest portion of the algorithm would still benefit from the high computation efficiency ensured by the AiMC approach. On the other hand, a full mapping guarantees the most efficient use of the F-2T2R accelerator, at the cost of a limited accuracy reduction. This approach should be preferred in computing systems targeting the best TOPS/W performance.

VII. CONCLUSION

This article has presented the F-2T2R compute cell, conceived for boosting the performance of RRAM-based mixed-signal accelerators. The cell exploits the FD-SOI technology to ease the RRAM programming and exhibits a large value of output impedance. A mixed-signal accelerator, embedding the F-2T2R compute cell, could obtain up to 1260 1 bit-TOPS/W and a classification accuracy of 86% on CIFAR-10, with a partial mapping of ResNet-18.

REFERENCES

[1] S. Yu, H. Jiang, S. Huang, X. Peng, and A. Lu, "Compute-in-memory chips for deep learning: Recent trends and prospects," *IEEE Circuits Syst. Mag.*, vol. 21, no. 3, pp. 31–56, 3rd Quart., 2021.

[2] M. Caselli, P. Debacker, and A. Boni, "Memory devices and A/D interfaces: Design tradeoffs in mixed-signal accelerators for machine learning applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 7, pp. 3084–3089, Jul. 2022.

[3] 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, Jan. 2019.

[4] A. Grossi et al., "Experimental investigation of 4-kb RRAM arrays programming conditions suitable for TCAM," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 12, pp. 2599–2607, Dec. 2018.

[5] S. Yu, W. Shim, X. Peng, and Y. Luo, "RRAM for compute-in-memory: From inference to training," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 7, pp. 2753–2765, Jul. 2021.

[6] C.-X. Xue et al., "Embedded 1-Mb ReRAM-based computing-in-memory macro with multibit input and weight for CNN-based AI edge processors," *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 203–215, Jan. 2020.

[7] W. Li, X. Sun, S. Huang, H. Jiang, and S. Yu, "A 40-nm MLC-RRAM compute-in-memory macro with sparsity control, on-chip write-verify, and temperature-independent ADC references," *IEEE J. Solid-State Circuits*, vol. 57, no. 9, pp. 2868–2877, Sep. 2022.

[8] T. Xie, S. Yu, and S. Li, "A high-parallelism RRAM-based compute-in-memory macro with intrinsic impedance boosting and in-ADC computing," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 9, no. 1, pp. 38–46, Jun. 2023.

[9] E. Esmanhotto et al., "High-density 3D monolithically integrated multiple 1T1R multi-level-cell for neural networks," in *IEDM Tech. Dig.*, Dec. 2020, pp. 36.5.1–36.5.4.

[10] C.-X. Xue et al., "24.1 A 1Mb multibit ReRAM computing-in-memory macro with 14.6ns parallel MAC computing time for CNN based AI edge processors," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 388–390.

[11] M. Caselli, C. van Liempd, A. Boni, and S. Stanzione, "A low-power native NMOS-based bandgap reference operating from −55°C to 125°C with Li-ion battery compatibility," *Int. J. Circuit Theory Appl.*, vol. 49, no. 5, pp. 1327–1346, May 2021.

[12] M. Caselli, D. Bhattacharjee, A. Mallik, P. Debacker, and D. Verkest, "Tiny ci-SAR A/D converter for deep neural networks in analog in-memory computation," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2022, pp. 1823–1827.

[13] P.-Y. Chen and S. Yu, "Compact modeling of RRAM devices and its applications in 1T1R and 1S1R array design," *IEEE Trans. Electron Devices*, vol. 62, no. 12, pp. 4022–4028, Dec. 2015.

[14] F. M. Puglisi, C. Wenger, and P. Pavan, "A novel program-verify algorithm for multi-bit operation in HfO_2 RRAM," *IEEE Electron Device Lett.*, vol. 36, no. 10, pp. 1030–1032, Oct. 2015.

[15] B. Q. Le et al., "RADAR: A fast and energy-efficient programming technique for multiple bits-per-cell RRAM arrays," *IEEE Trans. Electron Devices*, vol. 68, no. 9, pp. 4397–4403, Sep. 2021.

[16] F. Malena, A. Boni, and M. Caselli, "A PWM-DAC for analog in-memory computing in mixed-signal accelerators," in *Proc. 18th Conf. Ph.D Res. Microelectron. Electron. (PRIME)*, Jun. 2023, pp. 273–276.

[17] M. Caselli and A. Boni, "Modelling and optimization of a mixed-signal accelerator for deep neural networks," in *Proc. 19th Int. Conf. Synth., Model., Anal. Simul. Methods Appl. Circuit Design (SMACD)*, Jul. 2023, pp. 1–4.

[18] L. Grenouillet et al., "16 kbit 1T1R OxRAM arrays embedded in 28 nm FDSOI technology demonstrating low BER, high endurance, and compatibility with core logic transistors," in *Proc. IEEE Int. Memory Workshop (IMW)*, May 2021, pp. 1–4.

[19] B. Murmann, "Mixed-signal computing for deep neural network inference," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 1, pp. 3–13, Jan. 2021.

[20] S. D. Spetalnick et al., "A 40 nm 64 kb 26.56TOPS/W 2.37 Mb/mm^{2}RRAM binary/compute-in-memory macro with 4.23× improvement in density and >75% use of sensing dynamic range," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 1–3.

[21] C.-X. Xue et al., "16.1 A 22 nm 4 Mb 8 b-precision ReRAM computing-in-memory macro with 11.91 to 195.7TOPS/W for tiny AI edge devices," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 64, Feb. 2021, pp. 245–247.

[22] N. R. Shanbhag and S. K. Roy, "Benchmarking in-memory computing architectures," *IEEE Open J. Solid-State Circuits Soc.*, vol. 2, pp. 288–300, 2022.

[23] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, H. Wallach et al., Eds. Red Hook, NY, USA: Curran Associates, 2019, pp. 8024–8035.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[25] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.

Open Access funding provided by 'Università degli Studi di Parma' within the CRUI CARE Agreement