An embedded PCM Peripheral Unit adding Analog MAC In-Memory Computing Feature addressing Non-linearity and Time Drift Compensation

Alessio Antolini[1*], Andrea Lico[1], Eleonora Franchi Scarselli[1], Antonio Gnudi[1], Luca Perilli[1],
Mattia Luigi Torres[2], Marcella Carissimi[2], Marco Pasotti[2], Roberto Antonio Canegallo[2]
[1]DEI – ARCES, University of Bologna, viale Carlo Pepoli 3/2, 40123 Bologna, Italy.
[2]STMicroelectronics, via Camillo Olivetti 2, 20864 Agrate Brianza, Italy.
*Correspondence: alessio.antolini2@unibo.it

***Abstract*—This paper presents an integrated peripheral unit interfaced to an embedded Phase-change Memory (ePCM) macrocell, with the aim of adding Analog In-memory Computing (AIMC) feature without any modifications to the internal structure of the memory array. The testchip has been designed and manufactured in a 90-nm STMicroelectronics CMOS technology. The unit allows the execution of signed Multiply and Accumulate (MAC) operations at the edge of the memory array exploiting the physical characteristics of memory devices. I-V characteristic non-linearity and transconductance time drift of PCM cells are overcome through a regulated bitline readout circuitry with time-coded inputs, along with a drift compensation technique based on a conductance ratio. Measurements results show 1-σ accuracy of 95.56% in MAC operations, whose decrease in time is roughly negligible at room temperature and is less than 1% after 24 hours at 85°C bake.**

***Keywords*—Analog in-memory computing (AIMC), drift compensation, multiply and accumulate operation (MAC), Phase-change memory (PCM).**

I. Introduction

In the field of Beyond-von Neumann architectures, Analog In-memory Computing (AIMC) has been proposed as a valid strategy to reduce the amount of energy consumption and latency due to internal data transfers. The aim of AIMC is performing computations within the memory unit, typically leveraging the physical features of the memory devices [1]. Among resistive non-volatile memories (NVMs), Phase-change Memory (PCM) is a promising technology in this field, thanks to the intrinsic capability of its memory cells to store multilevel data [2]. A relevant AIMC task is the Multiply and Accumulate (MAC) operation, which is a kernel of the Matrix Vector Multiplication (MVM). MAC can be implemented on PCM arrays mapping the matrix coefficients on memory cells conductances, but, despite recent advances, computation accuracy is deeply affected by PCM devices I-V non-linearity and conductance time drift [2],[3].

This paper presents a peripheral unit interfaced with a 128-kB embedded PCM (ePCM) array in a 90-nm STMicroelectronics CMOS technology, with the purpose of executing one-step MAC operations with both signed inputs and coefficients. The developed testchip is mainly intended to demonstrate a readout technique for non-linearity and time drift compensation, which differs from solutions based on empirical models and post-processing compensation [4],[5],[6]. Moreover, the unit is conceived to avoid any changes of the internal structure of the memory. This paper is organized as follows: Section II describes the implementation of the proposed unit; Section III illustrates experimental results; Section IV concludes the paper.

II. AIMC unit implementation

A. Testchip structure and interface to the ePCM array

Figure 1 shows a simplified schematic of the 128-kB ePCM array architecture [7] and AIMC unit interface. The AIMC unit is directly connected to the main bitlines (MBL) and during AIMC computation standard ePCM read and program operations are disabled. To perform MAC tasks, the AIMC unit sets the voltage of each MBL and reads the current of the cells belonging to the addressed word line (WL). Unlike other works [5],[6],[8], where MVM is performed in a single step, the proposed solution implements a MVM with multiple consecutive MACs; this requires a sequential activation of different WLs, but prevents the row decoder from being modified, so that the ePCM can be employed as a binary memory as well.

B. MAC computation architecture

The proposed architecture, shown in Fig. 2, is designed to perform a single signed MAC operation $Z = \boldsymbol{w} \cdot \boldsymbol{x} = \sum_i w_i x_i$. The input array $\boldsymbol{x} = [x_1, \dots, x_n]$, where x_i are 5-bit signed data, is stored in the control unit. A set of DACs converts the 4-bit absolute value of each x_i to analog value V_i, while the sign bits $x_{i,sign}$ are directly connected to the readout circuit. Each element w_i of the array $\boldsymbol{w} = [w_1, \dots, w_n]$ is expressed through the absolute value of its conductance g_i and its sign g_{Si}, which are stored in two different PCM cells.

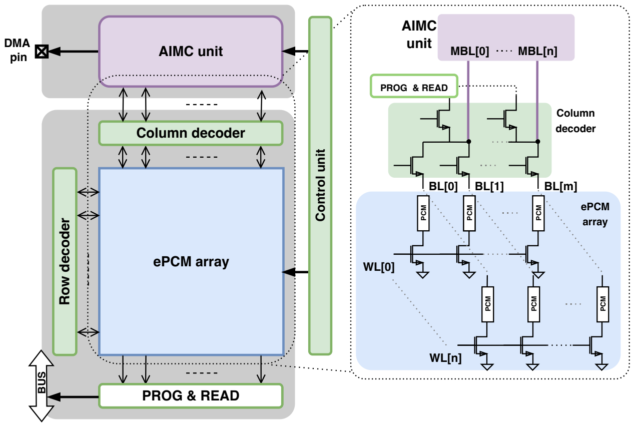


Fig. 1. Left: Block diagram. Right: Simplified schematic of the array architecture and column decoder. DMA pin is used to access various internal analog signals.

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 101007321. The JU receives support from the European Union's Horizon 2020 research and innovation programme and France, Belgium, Czech Republic, Germany, Italy, Sweden, Switzerland, Turkey.

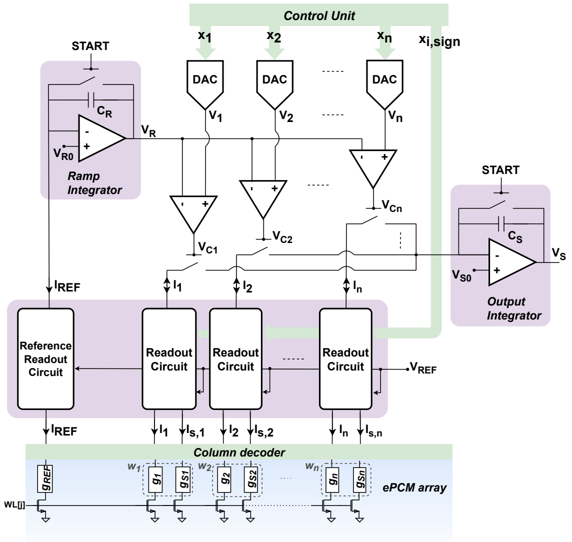


Fig. 2. Block diagram of the proposed MAC architecture.

The Reference Readout Circuit sets the read voltage V_{REF} across the reference conductance g_{REF}. According to Fig. 2 and 3, when the START signal switches to logic low, a current $I_{REF} = g_{REF}V_{REF}$ is integrated on capacitance C_R, generating a ramp signal V_R starting from V_{R0}:

$$V_R(t) = \frac{I_{REF}}{C_R}t + V_{R0}. \tag{1}$$

The same reference read voltage V_{REF} is applied across each weight cell g_i through n Readout Circuits. Each current $I_i = g_iV_{REF}$ is then sourced to or sunk from the output integrator circuit according to the sign of the product w_ix_i, which is obtained combining the corresponding sign cell g_{Si} value and $x_{i,sign}$ sign bit, as described in Paragraph D. Current I_i is then integrated on capacitance C_S for a time window T_{ON_i}, which begins at the START falling edge and ends when the output $V_{C,i}$ of the i-th comparator switches to logic low, i.e., when $V_R(t) = V_i$. According to (1):

$$T_{ON_i} = \frac{(V_i - V_{R0})\,C_R}{I_{REF}} = \frac{(V_i - V_{R0})\,C_R}{g_{REF}V_{REF}}, \tag{2}$$

is the time-coded version of V_i. Summing all the n currents $\pm I_i$, the output variation $\Delta V_S = V_S - V_{S0}$ is:

$$\Delta V_S = \sum_{i=1}^{n}\left[\pm\frac{I_iT_{ON,i}}{C_S}\right] = \frac{C_R}{C_S}\sum_{i=1}^{n}\left[\pm\frac{g_i}{g_{REF}}(V_i - V_{R0})\right]. \tag{3}$$

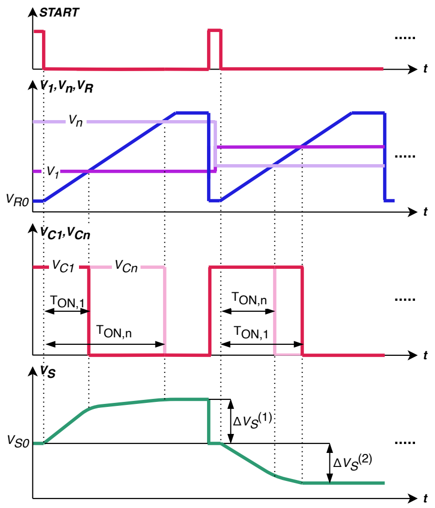


Fig 3. Sketch of waveforms showing two consecutive MAC operations. The first one represents a positive MAC, while the second a negative one.

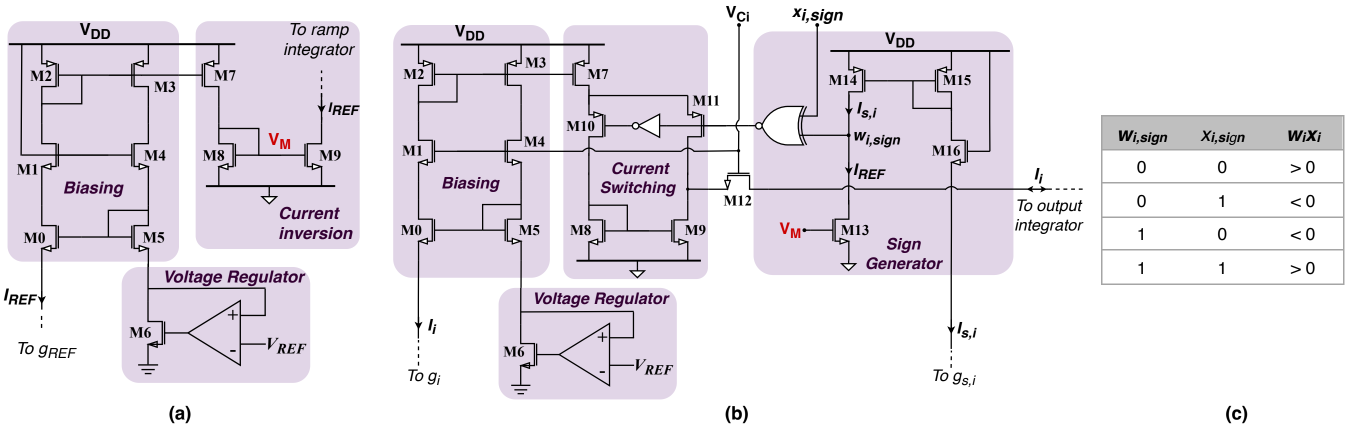
Considering $V_i - V_{R0}$ and g_i/g_{REF} as the absolute values of x_i and w_i, respectively, $\Delta V_S = (C_R/C_S)\sum_i w_ix_i = (C_R/C_S)Z$ is therefore proportional to the signed MAC operation Z.

C. Drift compensation

The drift of a generic cell conductance $g(t)$ has been shown to follow the power law $g(t) = g_0(t/t_0)^{-\alpha}$ [9], where g_0 is the conductance at arbitrary initial time t_0, and α is the drift coefficient, which is positive and cell-to-cell variable. The MAC result is proportional to the conductance ratio g_i/g_{REF}, and combining the drift model of $g(t)$ with (3), the MAC operation evaluated at time t_1 after t_0 becomes:

$$\Delta V_S(t_1) = \frac{C_R}{C_S}\sum_{i=1}^{n}\left[\pm\frac{g_{0,i}}{g_{0,REF}}\left(\frac{t_1}{t_0}\right)^{-(\alpha_i-\alpha_{REF})}(V_i - V_{R0})\right], \tag{4}$$

where $g_{0,i}$ and $g_{0,REF}$ are the weight and reference cells conductance at t_0, respectively. Therefore, each resulting drift coefficient is reduced to $\alpha_i - \alpha_{REF}$, and drift is partially compensated. In other words, the slope of ramp $V_R(t)$ decreases accordingly to the reference cell conductance drift; this leads to an increase in integration time T_{ON_i}, which compensates for the drift-induced drop of weight cells currents. Moreover, the adoption of time-coded inputs T_{ON_i}, along with cells being read at fixed voltage, addresses cells I-V characteristic non-linearity issue.



Wi,sign	Xi,sign	WiXi
0	0	> 0
0	1	< 0
1	0	< 0
1	1	> 0

Fig. 4. (a) Schematic of reference readout circuit. (b) Schematic of the cells readout circuit, with sign generation system. (c) Table summarizing the management of signed inputs and coefficients.

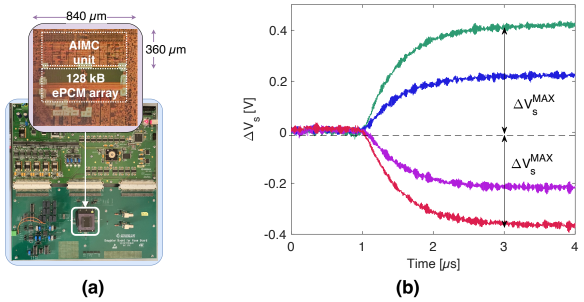


Fig 5. (a) Die micrograph and evaluation board. (b) Waveforms showing four different MAC operations.

D. Reference and Readout circuit with sign management

The detailed schematic of the Reference Readout Circuit is shown in Fig 4.a. The biasing circuit along with the voltage regulator allows to read the reference cell at a fixed reference voltage level V_{REF}. The reference voltage V_{REF} applied to the source terminal of transistor M5 is generated using a voltage regulator circuit, composed of an operational amplifier and transistor M6. Feedback from the source of transistor M5 is provided to the non-inverting input of the amplifier, while the inverting input is connected to V_{REF}, which is generated from a band-gap circuit. Current mirrors M5-M0 and M2-M3 provide voltage feedback that forces the gate-to-source voltages of transistors M0 and M5 to be equal. Thus, neglecting voltage drop through column decoder shown in Fig. 1 right, reference voltage V_{REF} is applied to the reference cell too. The current mirroring ratio of both current mirrors is 10:1, which is the same used between transistors M2 and M7 to provide I_{REF} to the ramp integrator.

Fig. 4.b shows one of the n Readout Circuits. The biasing and voltage regulator circuits are equal to those previously described and apply V_{REF} to the selected cells. Moreover, the n biasing circuits share a single voltage regulator. The current switching and sign generator circuits allow to manage both the input and the weight signs. Sign cell current I_{Si}, which is the bit line current from the weight bit cell g_{Si}, is mirrored by current mirror M14-M15 and compared to reference current I_{REF}, mirrored from reference readout circuit to M13 by means of voltage V_M. The result of the current comparison is a logic signal $w_{i,sign}$ that represents the sign of w_i. When $I_{Si} < I_{REF}$, this being indicative of a positive sign, $w_{i,sign}$ voltage value is logic low. Whereas, when $I_{Si} > I_{REF}$ (i.e., a negative sign), $w_{i,sign}$ is logic high. $w_{i,sign}$ is then combined with the sign bit $x_{i,sign}$ to produce a control signal applied to transistors M10 and M11 of the current switching circuit. Thus, the sign of the multiplication, as summarized in the Table of Fig. 4.c, determines the direction of current I_i applied to the integrator.

III. Characterization Results

The testchip includes a single 128-kB ePCM array interfaced with the AIMC unit, and it is mainly intended to validate the proposed drift compensation technique. In this first prototype, the dimension of the input and coefficient arrays is n = 12. Circuits have been designed with V_{DD} = 1.2 V and V_{REF} = 0.3 V, leading to PCM cells currents ranging from hundreds of nA to 10 μA. The minimum time required to perform a single MAC operation is 150 ns, and depends on the reference conductance value, as shown in (2), while the maximum output voltage ΔV_S^{MAX} is ±400 mV. According to the intention of this work, this Section presents the AIMC performances in terms of MAC accuracy. The result of the generic operation was obtained measuring on DMA pin the output voltage ΔV_S, as defined in (3). Data were digitally converted using a 16-bit ADC available on a dedicated evaluation board. Examples of measured ΔV_S are reported in Fig. 5.b. In the following Paragraphs, results are related to the normalized MAC output z, defined as:

$$z \doteq \frac{Z}{Z^{MAX}} = \frac{\boldsymbol{w} \cdot \boldsymbol{x}}{\max\{\boldsymbol{w} \cdot \boldsymbol{x}\}}, \tag{5}$$

which is obtained measuring $\Delta V_S / \Delta V_S^{MAX}$.

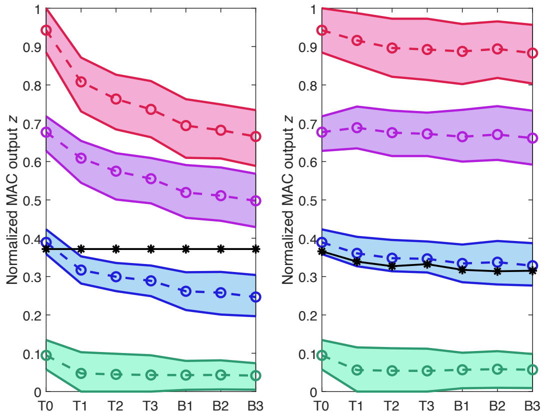


Fig. 6. Measured normalized outputs z after programming (T0), T1 = 1 day, T2 = 4 days and T3 = 7 days at room temperature, and then after B1 = 1 hour, B2 = 5 hours and B3 = 24 hours bake at 85°C. Left: constant reference current (black line); right: PCM reference current (black line). Dashed lines identify the mean measured values, while areas borders identify the minimum and the maximum.

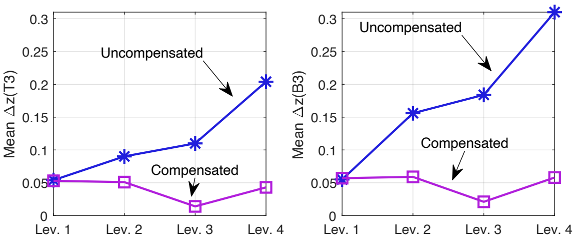


Fig. 7. Normalized mean drift error Δz as a function of the PCM cells levels, in T3 (left) and B3 (right).

A. Evaluation of conductance time drift compensation

The drift compensation technique on individual cells has been tested evaluating a set of normalized MAC outputs depending on a single weight w_i. To this purpose, 960 PCM cells, belonging to 80 different WLs, have been programmed with a dedicated iterative algorithm [10] with four different conductance levels. Then, in accordance with (5), all but the i-th input x_i have been set to 0, whereas x_i was chosen equal to the maximum x^{MAX}; then, the normalized MAC output $z = w_i / w^{MAX}$ was measured. This operation has been repeated for all the 960 cells after T1 = 1 day, T2 = 4 days, and T3 = 7 days from initial time T0. Then, the testchip has been baked at 85°C in a controlled climate chamber to accelerate cells drift phenomena [11]. Measures have been repeated after B0 = 1 hour, B1 = 5 hours and B3 = 24 hours bake.

All measurements have been performed first with a ramp signal V_R generated with an external source constant in time, thus expecting no drift compensation (Fig. 6-left); then, V_R was generated by a PCM reference cell g_{REF} (Fig. 6-right). As expected, in the first case, normalized outputs z tend to

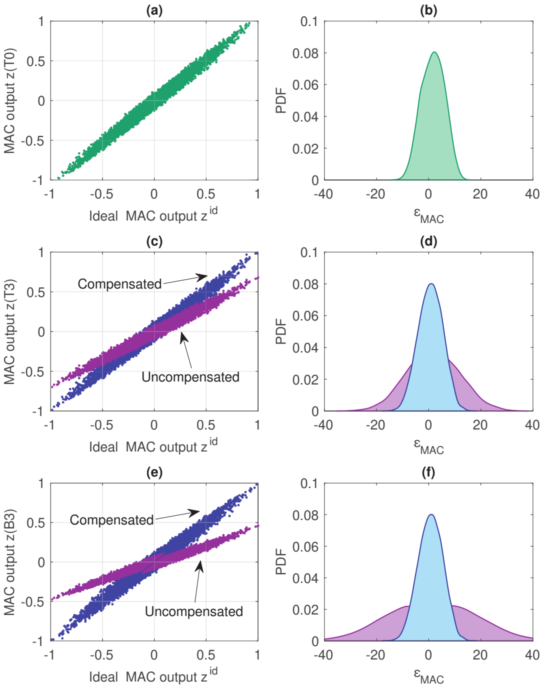


Fig. 8. Experimental results of MAC operations and PDFs of corresponding MAC error: (a-b) after programming, (c-d) after T3 = 7 days, and (e-f) after 24 hours bake at 85°C, with constant reference current (purple data) and with PCM reference (blue data).

decrease in time under the effect of cells conductance drift, which becomes even stronger after the bake. On the contrary, in the second case, the compensation mechanism successfully reduces the drop of results in time, in agreement with (4), keeping the four considered output levels widely separated. Nonetheless, the spread of MAC operations belonging to the same level is unaffected by drift compensation, as it is related to programming precision and to PCM cell-to-cell drift variability. In this example, g_{REF} was programmed to the second conductance level. Similar results were obtained when g_{REF} was programmed with the third or the fourth level. To quantify the effect of compensation, the normalized drift errors after 7 days at room temperature $\Delta z(T3) \doteq z(T0) - z(T3)$, and after 24-hours bake $\Delta z(B3) \doteq z(T0) - z(B3)$ have been calculated for each multiplication in both uncompensated and compensated case. Results of Fig. 7 show that the proposed technique keeps mean drift error under 6%, even after bake.

B. *MAC accuracy*

To ultimately evaluate the MAC accuracy, PCM cells of several WLs have been randomly programmed with the same conductance levels of the previous test, along with RESET state. Moreover, weights were chosen half with positive sign and half with negative sign. The employed value of g_{REF} was the same as in the previous test. Once a WL had been fully programmed, a set of 10000 different input vectors $\boldsymbol{x}$ of length $n = 12$ was applied to measure the normalized MAC outputs z. In Fig. 8.a the whole set of outputs immediately after programming is reported with dots as a function of the output $\boldsymbol{z}^{id}$; the latter is defined considering the nominal programming values of $\boldsymbol{x}$ and target values of $\boldsymbol{w}$. The distribution of error $\varepsilon_{MAC} = 100(\boldsymbol{z}^{id} - \boldsymbol{z})$ is plotted in Fig. 8.b. The error values lie between –12.17% and +13.9%, and its standard deviation $\sigma(\varepsilon_{MAC})$ is 4.44%. Thus, the AIMC unit mean accuracy, defined as $100 - \sigma(\varepsilon_{MAC})$, is equal to 95.56%. Evidently, this performance is limited by the programming algorithm finite tolerance and the circuit non-idealities.

The same operations have been repeated after T3 = 7 days, both with a constant reference current and with PCM variable reference current. Results are reported in Fig. 8.c, where all the measured normalized MAC outputs z are plotted in the two cases, together with the distributions of the corresponding MAC error (Fig. 8.d). In the uncompensated case, ε_{MAC} varies between –30.2% and +32.81%, with a standard deviation equal to 10.58%, while ε_{MAC} range is reduced to –16.02 and +15.52% and $\sigma(\varepsilon_{MAC})$ to 4.66%, when using PCM reference. The same test was finally carried out after B3 = 24 hours bake at 85°C and results are reported in Fig. 8.e and 8.f. In this case, ε_{MAC} varies from –42.42% to 41.8%, with $\sigma(\varepsilon_{MAC}) = 17.71\%$ with no compensation, while, when the PCM reference is used, ε_{MAC} and $\sigma(\varepsilon_{MAC})$ in B3 are similar to the T3 case. It is evident that the compensation feature of the AIMC unit keeps the computation error fairly invariant in both cases. Accordingly, the MAC accuracy in T3 is equal to 95.34%, and to 94.97% in B3, with a decrease of 0.22% and 0.59% only with respect to the initial one.

IV. Conclusion

In this paper a peripheral unit adding analog in-memory multiply and accumulate (MAC) computing function to an embedded phase-change memory (ePCM) macrocell has been presented. The unit exploits an innovative readout scheme to address non-linearity of I-V characteristic and time drift of cells conductances. The unit is conceived to operate with signed inputs and coefficients and does not require any modification to the internal structure of the ePCM. MAC operations are performed with a 1-σ accuracy of 95.56%, which is not significantly affected in time by drift effects, even after 24-hours bake at 85°C.

V. Acknowledgment

The Authors wish to thank Chantal Auricchio and Laura Capecchi from STMicroelectronics Italy for their fundamental contribution to the testchip design and development.

VI. References

[1] W. Haensch, T. Gokmen, and R. Puri, "The Next Generation of Deep Learning Hardware: Analog Computing," *Proc. IEEE*, vol. 107 , 2019.

[2] G. W. Burr, B. N. Kurdi, J. C. Scott, C. H. Lam, K. Gopalakrishnan, and R. S. Shenoy, "Overview of candidate device technologies for storage-class memory," *IBM J. Res. Dev.*, vol. 52, no. 4–5, 2008.

[3] G. W. Burr *et al.*, "Neuromorphic computing using non-volatile memory," *Adv. Phys. X*, vol. 2, no. 1, Jan. 2017.

[4] C. Paolino *et al.*, "Compressed Sensing by Phase Change Memories: coping with encoder non-linearities,", *ISCAS* 2021.

[5] V. Joshi *et al.*, "Accurate deep neural network inference using computational phase-change memory," *Nat. Commun.*, vol. 11, 2020.

[6] X. Sun *et al.*, "PCM-Based Analog Compute-In-Memory: Impact of Device Non-Idealities on Inference Accuracy," *IEEE Trans. Electron Devices*, vol. 68, no. 11, 2021.

[7] M. Carissimi *et al.*, "2-Mb Embedded Phase Change Memory with 16-ns Read Access Time and 5-Mb/s Write Throughput in 90-nm BCD Technology for Automotive Applications," *ESSCIRC 2019 - IEEE 45th Eur. Solid State Circuits Conf.*, 2019.

[8] R. Khaddam-Aljameh *et al.*, "HERMES Core-A 14nm CMOS and PCM-based In-Memory Compute Core using an array of 300ps/LSB Linearized CCO-based ADCs and local digital processing," *IEEE Symp. VLSI Circuits, Dig. Tech. Pap.*, vol. 2021-June, 2021.

[9] D. Ielmini, A. L. Lacaita, and D. Mantegazza, "Recovery and drift dynamics of resistance and threshold voltages in phase-change memories," *IEEE Trans. Electron Devices*, vol. 54, no. 2, 2007.

[10] A. Antolini *et al.*, "Characterization and programming algorithm of phase change memory cells for analog in-memory computing," *Materials (Basel).*, vol. 14, no. 7, 2021.

[11] F. G. Volpe, A. Cabrini, M. Pasotti, and G. Torelli, "Drift induced rigid current shift in Ge-Rich GST phase change memories in low resistance state," *2019 26th IEEE Int. Conf. Electron. Circuits Syst. ICECS*, 2019.