A Binary Pattern Matching Task Performed in an ePCM-Based Analog In-Memory Computing Unit

A Binary Pattern Matching task performed in an ePCM-based Analog In-Memory Computing Unit*

Francesco Zavalloni[1], Alessio Antolini[1], Andrea Lico[1], Eleonora Franchi Scarselli[1], Mattia Luigi Torres[2], Riccardo Zurla[2], and Marco Pasotti[2]

[1] ARCES-DEI, University of Bologna, Italy
[2] STMicroelectronics, Agrate Brianza, Italy
francesco.zavalloni2@unibo.it, alessio.antolini2@unibo.it

Abstract. This paper shows the results related to a Binary Pattern Matching (BPM) task executed on an Analog In-memory Computing (AIMC) unit based on an embedded Phase-Change Memory (ePCM), both designed in a 90-nm CMOS STMicroelectronics technology. The Hit Rate in pattern recognition is characterized and modeled in different scenarios, in order to evaluate the influence of cells Conductance Time Drift (CTD) in a real application. In particular, two PCM multilevel programming algorithms and different cells conductances are considered, and their effects on CTD are experimentally observed. Results suggest that the adoption of a SET staircase (SSC) sequence implies a lower CTD on PCM cells with respect to a RESET Staircase (RSC) sequence, as well as an increased Hit Rate, even with lower levels of employed conductance.

Keywords: Analog In-memory Computing (AIMC) · Phase-change Memory (PCM) · Binary Pattern Matching.

1 Introduction

Analog In-memory Computing (AIMC) based on Phase-Change Memory (PCM) has recently been gaining interest due to its potentiality in accelerating computations for a plethora of data-centric applications [1], [2]. In this context, PCM has established as a promising technology thank to its capability to store multilevel conductance levels. However, retention of PCM devices is sill a notable challenge, as their compound suffers from conductance time drift (CTD) [3].

This work aims at the characterization of two possible PCM cells multilevel programming algorithms in terms of their impact on CTD. The employed test

* 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.

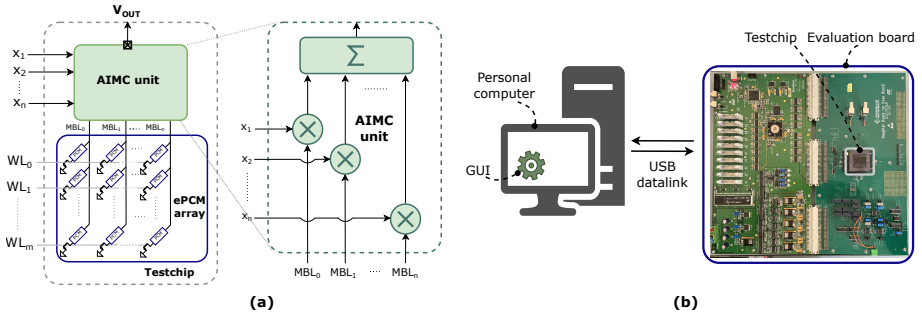

Fig. 1. (a) Schematic representation of the employed test vehicle and of the AIMC unit. (b) Sketch of the experimental setup.

vehicle is a PCM-based AIMC unit [5], which performs one-step signed Multiply-and-Accumulate (MAC) operations, designed in a 90-nm CMOS STMicroelectronics technology. Moreover, the same integrated prototype will be used to perform operations of Binary Pattern Matching (BPM), a technique used in computer science to identify patterns or similarities between data sets, which is well suited to the AIMC paradigm [4].

This paper is organized as follows: in Section 2 the employed test vehicle and experimental setup are described; Section 3 illustrates the methods and the results related to PCM devices programming algorithms and CTD; Section 4 reports the execution of BPM tasks on the AIMC unit also including the effects of cells CTD; Section 5 concludes the paper.

2 AIMC unit and experimental setup

The test vehicle employed in the following analyses is the AIMC testchip presented in [5]. The system includes a peripheral AIMC unit interfaced with an embedded PCM (ePCM) IP [6], with the purpose of performing one-step signed Multiply-and-Accomulate (MAC) operations. The entire testchip is manufactured in a 90-nm STMicroelectronics CMOS technology, which features a specifically optimized Ge-rich GeSbTe (GST) alloy for PCM cells.

As depicted in Fig. 1(a), the ePCM stores the coefficients required during the computations, while the AIMC unit performs each MAC operation in an analog fashion reading the cells currents. The AIMC unit takes as input a vector $\mathbf{x} = [x_1, ..., x_n]$ and executes the MAC operation $\mathbf{x} \cdot \mathbf{G}_j$, where $\mathbf{G}_j = [G_{j1}, ..., G_{jn}]$ are the conductances stored in the j-th wordline (WL) of the ePCM array. To this purpose, the AIMC unit applies a constant read voltage V_{REF} on the cells involved in the MAC operations. The result of the operation is mapped on the AIMC unit output voltage value by integrating the sum of the cells currents. Each single cell current $I_i = G_i V_{\mathrm{REF}}$ is integrated for a time window $T_{\mathrm{ON}_i} \leq T_{\mathrm{MAC}}$ which is proportional to the corresponding input x_i. Accordingly, the expression

of the output V_{OUT_j} at the end of the integration period is:

$$V_{\mathrm{OUT}_j} = k \left[\sum_{i=1}^{n} \left(\pm \frac{G_{j,i}}{G_{\mathrm{REF}}} V_i \right) \right], \tag{1}$$

where $G_{j,i}$ is the conductance of the PCM cell representing a element of the vector stored in the j-th WL of the array, G_{REF} is the reference conductance, V_i is the analog voltage level of the input vector element x_i, and k is a dimensionless constant value accounting for the effects of circuit parameters. In particular, the reference conductance is meant to normalize the whole MAC result in order to maximize the output swing of its possible analog values.

The testchip is experimentally characterized through the employment of a dedicated evaluation board (see Fig. 1(b)), which allows to program PCM cells with customized programming sequences, as well as to measure the conductance of each device. Furthermore, MAC operations can be invoked through a specific Graphic User Interface (GUI), which also collects and elaborates any type of measurement result.

3 Drift characterization

One of the main open challenges in the field of PCM-based AIMC is the conductance time drift (CTD) of PCM devices, namely, the cells amorphous phase tends to randomly increase its resistivity in time due to internal structural relaxation phenomena [7]. Among the elements that influence the cells CTD, the programming technique adopted to program PCM cells has been shown to play an important role [8], [9]. Consequently, two well-known PCM programming algorithms are characterized in the following in terms of their impact on cells CTD, as well as of their programming success rate.

3.1 PCM cells programming algorithms: SSC and RSC

In this work the SET-Staircase (SSC) and the RESET-Staircase (RSC) iterative algorithms are considered. Both algorithms modulate the conductance of the PCM cell to target level $\hat{G}$ using a specific sequence of current pulses, namely, the trapezoidal (SET) or rectangular (RESET) pulse, which cause an increase and a decrease of the cell conductance, respectively. Specifically, the SSC sequence starts with a high-amplitude Start RESET pulse and, using a sequence of intermediate SET pulses, chases the target conductance level by increasing the pulse amplitude at each step. Conversely, RSC starts with a high-amplitude SET pulse, followed by an increasing-amplitude RESET pulse sequence. The conductance is measured after the application of each pulse, and the algorithm stops once the target level $\hat{G}$ is obtained up to a tolerance ΔG, otherwise the cell is brought back to its original state through a start SET or RESET pulse in case of RSC or SSC, respectively. The procedure is repeated until a maximum number of iteration is reached. The programming sequences are represented with

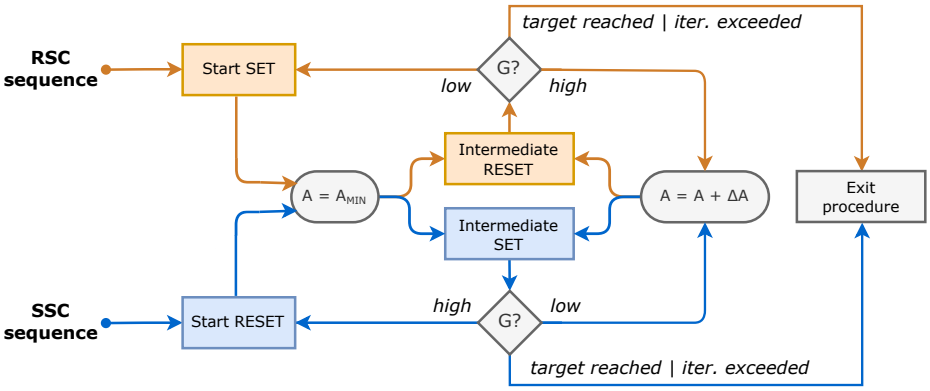

Fig. 2. Diagram of the RSC and SSC programming sequences (represented with red and blue paths, respectively). Parameter A identifies the current pulse amplitude, whereas ΔA and A_{MIN} represent the amplitude increase and its minimum value.

the diagram of Fig. 2, where the blue paths represent the steps of the SSC algorithm, whereas the red ones are related to the RSC sequence. The delay between the application of any current pulse and the following cell conductance measure is estimated to be in the order of milliseconds.

3.2 Experimental results

To evaluate the success rate of each algorithm to successfully impose an arbitrary level of conductance to a single PCM cell and to estimate the cells CTD, eight sets of 250 cells were programmed at four increasing level of conductance, namely: $\hat{G}_1$, $\hat{G}_2$, $\hat{G}_3$ and $\hat{G}_4$, using for the first half of sets the SSC algorithm, then the RSC algorithm for the remaining ones. A programming tolerance $\Delta G = \hat{G}_1/20$ has been used for all targets, and the maximum number of allowed iterations have been set to 255. The success rate reached by each algorithm for each conductance level are reported in Table 1. As can be seen from the results, the SSC algorithm exhibits a higher success rate with respect of the RSC algorithm, especially for the intermediate levels $\hat{G}_2$ and $\hat{G}_3$. This behaviour could be ascribed to the programming curves of SSC and RSC sequences. Specifically, as pointed out in [8], RSC leads to an abrupt programming curve, whereas SSC programming allows for a smoother control of the conductance by means of the SET amplitude.

In order to investigate the effects of each algorithm on cells CTD, devices have been monitored first for 2 hours at room temperature, and then after 50 minutes at an annealing temperature of 150°C. Cells whose programming procedure failed have been excluded from this characterization.

Collected data have been employed to extrapolate the drift coefficients υ of each cell conductance $G(t)$, i.e., the power factors of the empirical law [10] which

Table 1. Success rate of multilevel programming achieved by SSC and RSC algorithms on 250 PCM devices.

Target level	**SSC sequence**	**RSC sequence**
$\hat{G_1} = \hat{G_4}/4$	100%	100%
$\hat{G_2} = \hat{G_4}/2$	99.1%	86.4%
$\hat{G_3} = 3\hat{G_4}/4$	96.5%	88.5%
$\hat{G_4}$	98.8%	99.4%

quantifies the PCM behavior in time:

$$G(t) = G_0 \left(\frac{t}{T_0} \right)^{-\upsilon} , \tag{2}$$

where G_0 is the cell conductance measured at the initial arbitrary time T_0 after the end of the programming procedure.

The υ-coefficients at room temperature (υ_0) and at 150°C (υ_1) have been extrapolated according to the following equations [7]:

$$\upsilon_0 = \frac{\ln G_0 - \ln G_1}{\ln(\frac{T_1}{T_0})} , \tag{3}$$

$$\upsilon_1 = \frac{\ln(\frac{G_1}{G_3}) + \upsilon_0 \ln(\frac{T_1}{T_0})}{\ln(\frac{T_3 - T_2}{T_0})} , \tag{4}$$

where G_0 represents the conductance value obtained right after the programming phase at time T_0; G_1 and G_2 are the conductance measured after 50 minutes (T_1) and 2 hours (T_2) at room temperature, whereas G_3 is taken at the end of the annealing (T_3). All measurements were performed at room temperature, and a schematic representation of the process is depicted in the insets (a) and (b) of Fig. 3.

The obtained υ-coefficients are shown in insets (c) and (d) of Fig. 3, where the drift coefficients are displayed as function of the target conductance level (normalized to $\hat{G_4}$); different curves refers to the SSC and RSC algorithms. It is evident that the υ-coefficients are inversely proportional to the conductance level both at room temperature and at 150°C, in according with [10]. Moreover, cells programmed with the SSC algorithm exhibit a lower value of υ in all conditions. As a conclusion, this result underlines the capability of cells programmed with SSC to be more resilient to conductance drift, as it will be further stressed out in the following Section.

4 Binary Pattern Matching task

Among large-scale data analytic applications, Pattern Matching (PM) stands for one of the most demanding, since it involves repetitive search over very

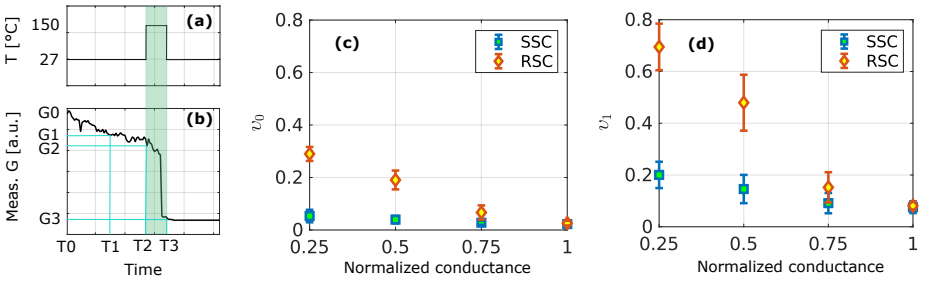

Fig. 3. (a), (b): cell conductance evolution: T_0 is immediately after the programming phase, T_1 and T_2 after 50 minutes and 2 hours at room temperature, and T_3 after additional 50-minutes bake at 150°C. (c), (d): drift coefficients v_0 and v_1 as a function of target conductance and programming algorithm.

extensive reference data sets stored in memory [4]. In this framework, the Binary Pattern Matching (BPM) task consists in a bit-to-bit comparison between an unknown length-fixed binary pattern and all its possible combinations, aiming to its recognition. Thanks to its bit-wise nature and the possibility to employ an opportune coding strategy, BPM well suites the AIMC paradigm [11].

In this Section, an empirical implementation of the aforementioned BPM task will be presented, making use of the integrated prototype described in Section 2. Empirical results will be presented, taking consideration of several operating conditions, such as the conductance level at which PCM cells where programmed to, different bit-length patterns and the algorithm employed during the programming phase. Furthermore, simulations based on the results of the previous Section has been developed with the purpose to expand the in-memory BPM task analysis.

4.1 Implementation using AIMC prototype and experimental results

In this analysis, the BPM task is performed on the AIMC unit with logic '0' and '1' elements coded as -1 and $+1$ coefficients, respectively (e.g.: $[0, 1, 0]$ becomes $[-1, +1, -1]$). The n-bit input pattern is compared to all its 2^n possible combinations, each one stored in a different WL of the PCM array. Each comparison with the j-th string is provided by a MAC operation on the same WL, and the correct match is identified by the maximum MAC.

As far as the experimental implementation is concerned, according to (1), the input binary coefficients are mapped with input voltage levels $\pm V_0$ (for '0' and '1' bits, respectively), whereas coefficients in each WL are represented by the conductance level of two PCM cells. In particular, as described in [5], coefficient signs are coded by cells programmed in a rough full RESET or full SET state (i.e., with no need of any accurate programming procedure), whereas absolute values are obtained through cells programmed to the same target conductance level $\hat{G}$:

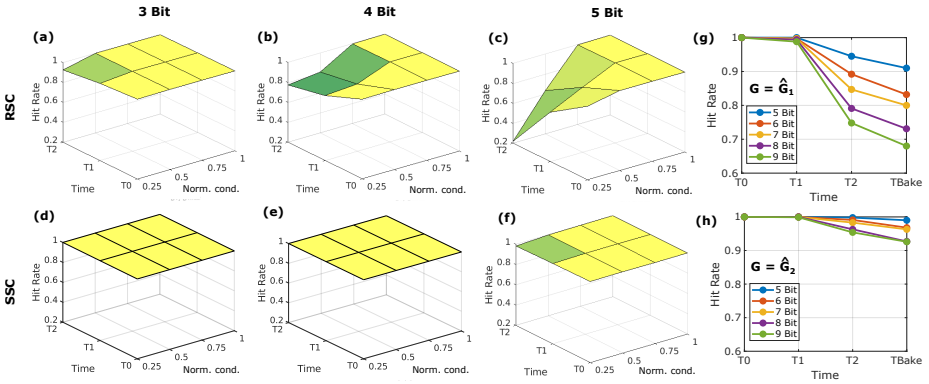

Fig. 4. Measured Hit Rate versus normalized weight conductance and time for pattern length n equal to 3, 4 and 5 bits: (a), (b), (c) with RSC and (d), (e), (f) with SSC algorithm; measures consider BPM executed after 6 (T_1) and 12 (T_2) hours from programming (T_0); (g), (h): simulated Hit Rate using SSC increasing the pattern length (up to 9 bit) and considering additional 50-minutes 150°C bake after T_2 (T_{bake}); levels $\hat{G}_1$ and $\hat{G}_2$ are used for weight conductances, respectively.

as a result, each '0' and '1' element of all stored binary strings is coded with $\pm\hat{G}$, respectively. The same $\hat{G}_1$, $\hat{G}_2$, $\hat{G}_3$ and $\hat{G}_4$ of Section 3.2 have been considered in the following to compare the impact of SSC and RSC algorithms on BPM tasks. Moreover, computations have been executed right after the programming phase (T_0), then after 6 and 12 hours (T_1 and T_2) at room temperature, and considering patterns of lenghts $n = 3$, 4 and 5 bits.

The accuracy of BPM operations has been tested, at a given size n, by identifying each one of the 2^n possible inputs five consecutive times in T_1 and T_2. Results have been evaluated through the Hit Rate H, namely the correct pattern matching percentage averaged on the five attempts and on all the input strings. The plots (a)-(f) of Fig. 4 report H as a function of time T_i and target conductance $\hat{G}_i$. In case $n = 3$, the Hit Rate is above 90% in all tested conditions. When $n > 3$, the RSC algorithm grants a significant level of H only if the most two conductive states $\hat{G}_3$ and $\hat{G}_4$ are used, otherwise the accuracy considerably drops under the effects of CTD. The accuracy loss is worsen with an increased length of patterns. On the other hand, the SSC algorithm allows to keep $H >$ 90% in all conditions, as the effect of CTD is sensibly lower with respect to the RSC scenario, in accordance with the previous results. As a result, SSC allows the employment of lower levels of PCM cells conductance, with a consequent gain in terms of energy efficiency.

4.2 Extended analysis based on conductance monitoring

This Section extends the analysis of BPM operations executed on the testchip considering more severe constraints in terms of CTD, as well as arbitrary length of the input pattern. Since cells programmed with SSC algorithm have been

shown to exhibit less CTD, as well as to reach better accuracy in BPM tasks, this investigation is limited to the SSC scenario.

To this purpose, measures of cells conductances collected for CTD characterization of Section 3.2 have been used to simulate the execution of the same previous tasks. In particular, values of υ_0 and υ_1 obtained through (3) and (4), along with all cell values G_0 measured in T_0, have been exploited to describe the cells conductance behavior at arbitrary time at room temperature and after the annealing of Section 3.2. This values have been then exploited to reproduce BPM tasks just as they have been implemented on the AIMC unit. For this reason, simulations have been tailored also taking into account of the non-negligible aspects of the circuital implementation of the testchip, such as, for example, the finite precision of the output analog-to-digital conversion.

The model has been employed first to infer the Hit Rate H of BPM attempts in T_0 and T_1 to validate its coherence with the previous empirical results, then to assess the accuracy in T_{bake}. Then, it has been used to estimate H including patterns with length from 5 to 9 bits in the same three time instants. In Fig. 4(g) and (h) the estimated H is shown, and results show that SSC-programmed $\hat{G}_2$ conductance levels grant H to be above 90% in all scenarios, thus validating the trend outlined by experimental results.

5 Conclusion

This paper compares two different Phase-change Memory (PCM) cells programming algorithms, the SET-Staircase (SSC) and RESET-Staircase (RSC) procedure, in the framework of a Binary Pattern Matching (BPM) task executed on a PCM-based Analog In-memory Computing (AIMC) testchip. A characterization of the two programming algorithms in terms of drift-induced conductance drop has been proposed, and their effect is further analyzed in pattern recognition executed on an AIMC prototype. Moreover, simulations exploiting measures of cells conductance have allowed to extend experimental results by means of increased pattern lengths and harsher operative conditions in terms of conductance drift. Experimental data prove the capability of SSC algorithm to grant a higher Hit Rate in all conditions with respect to the RSC one, even with PCM cells programmed at the lowest conductance level. A Hit Rate above 90% is achievable by SSC cells programmed to an intermediate level of conductance in the simulated approach as well, even under effect of a high-temperature induced drift and with increased length of binary patterns.

References

1. N. Verma et al., In-Memory Computing: Advances and Prospects, IEEE SSC Magazine (2019).
2. M. Le Gallo et al., Brain-inspired computing using phase-change memory devices, J. Appl. Phys. (2018).

3. G W. Burr et al., Phase-change memory technology, Journal of Vacuum Science and Technology, vol. 28, issue 2 (2010).
4. I. Giannopoulus et al., In-memory Database Query, Adv. Intell. Sys. (2020).
5. A. Antolini et al., An embedded PCM Peripheral Unit adding Analog MAC In-Memory Computing Feature addressing Non-linearity and Time Drift Compensation, IEEE 48th ESSCIRC (2022).
6. M. Pasotti et al., A 32-KB ePCM for Real-Time Data Processing in Automotive and Smart Power Applications, IEEE JSSC (2018).
7. D. Ielmini et al., Physical mechanism and temperature acceleration of relaxation effects in phase-change memory cells, IEEE 46th AIRPS (2008).
8. A. Antolini et al., Characterization and Programming Algorithm of Phase Change Memory Cells for Analog In-Memory Computing, Materials (2021).
9. N. Papandreou et al., Programming algorithms for multilevel phase-change memory, IEEE ISCAS (2011).
10. N. Ciocchini et al., Modeling Resistance Instabilities of Set and Reset States in Phase Change Memory With Ge-Rich GeSbTe, IEEE Transaction on Electron Devices, vol. 61, issue 6 (2014).
11. Z. I. Chowdhury et al., Spintronic In-Memory Pattern Matching Using Computational RAM (CRAM), IEEE JXCDC (2019).