A drift-resilient hardware implementation of neural accelerators based on phase change memory devices

Irene Muñoz-Martín*, *Student Member, IEEE*, Stefano Bianchi*, *Student Member, IEEE*, Octavian Melnic, *Student Member, IEEE*, Andrea Giovanni Bonfanti, *Senior Member, IEEE*, and Daniele Ielmini, *Fellow, IEEE*

***Abstract*—Memory devices, such as the phase change memory (PCM), have recently shown significant breakthroughs in terms of compactness, 3D stacking capability and speed up for deep learning neural accelerators. However, PCM is affected by the conductance drift, which prevents a precise definition of the synaptic weights in artificial neural networks. Here, we propose an efficient system-level methodology to develop drift-resilient multi-layer perceptron (MLP) networks. The procedure guarantees high testing accuracy under conductance drift of the devices and enables the use of only positive weights. We validate the methodology using MNIST, rand-MNIST and Fashion-MNIST datasets, thus offering a roadmap for the implementation of integrated non-volatile memory-based neural networks. We finally analyse the proposed architecture in terms of throughput and energy efficiency. This work highlights the relevance of robust PCM-based design of neural networks for improving the computational capability and optimizing the energetic efficiency.**

Keywords: Neural networks, integrated neural networks, analogue circuit, digital circuit, phase-change-memory (PCM), conductance drift.

I. Introduction

In the last few years, artificial intelligence (AI) has shown rapid progress, demonstrating unprecedented ability in tasks like pattern recognition, natural language processing and playing games [1], [2]. Despite the recent advances at software level, hardware implementations of AI still show significant limits. In particular, CMOS-based circuits store most of the weights in dynamic random-access memory (DRAM) while data are processed in the central processing unit (CPU), thus forcing a major data transfer between DRAM and CPU [3], [4]. Such data transfer causes significant latency and power consumption, which highlights the memory bottleneck of Von Neumann architectures [3]. On the other hand, memory devices, such as resistive-random-access memory (RRAM) and the phase change memory (PCM), have recently demonstrated significant progress for AI acceleration [5]–[8]. Along with the reduced area consumption, memory arrays are the best candidates for efficient matrix-vector multiplication (MVM) via Ohm's and Kirchhoff's laws [9].

However, memories present some key non-idealities which currently hinder the realization of a full in-memory computing hardware. In particular, PCM suffers from conductance drift, which causes a time variation of the synaptic weights in neural computation, thus affecting the recognition accuracy

I. Muñoz-Martín*, S. Bianchi*, O. Melnic, A. G. Bonfanti, and D. Ielmini are with the Dipartimento di Elettronica, Informazione e Bioingegneria and Italian Universities Nanoelectronics Team, Politecnico di Milano, Milan 20133, Italy (e-mail: irene.munoz@polimi.it; stefano1.bianchi@polimi.it; octavian.melnic@polimi.it; andrea.bonfanti@polimi.it; daniele.ielmini@polimi.it). *Authors contributed equally.

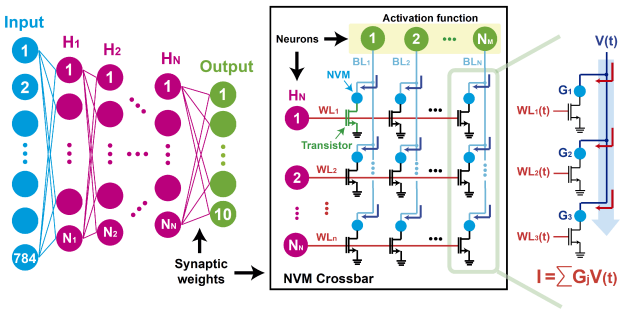

Fig. 1. Standard multilayer perceptron network (MLP) with crossbar memory arrays in order to take advantage of the matrix-vector multiplication (MVM) procedure. The weights can be implemented with analogue or digital programming of the PCM devices.

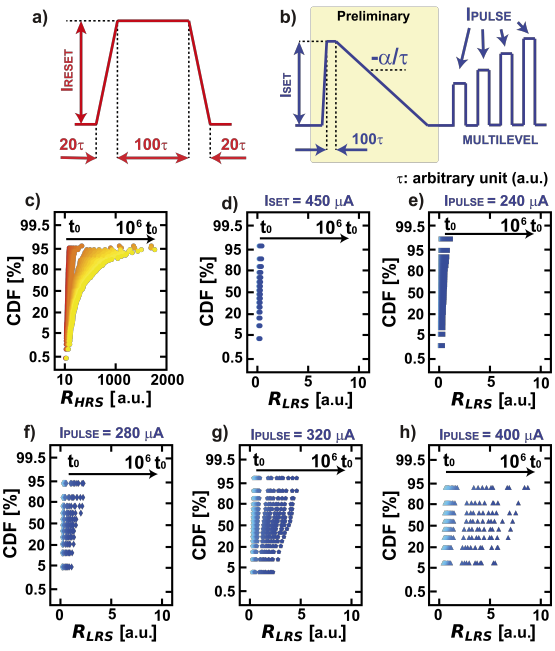

Fig. 2. Typical (a) set and (b) reset signals for programming the LRS and the HRS of the PCM devices. The multilevel states of the PCMs can be obtained by applying rectangular pulsed signals after a full set. (c) HRS and (d) LRS drift for a pure reset/set programming condition (from a time t_0 to a time $10^6 t_0$ after the experiment). (e-h) Drift in time for PCM multilevel states at increasing I_{PULSE}.

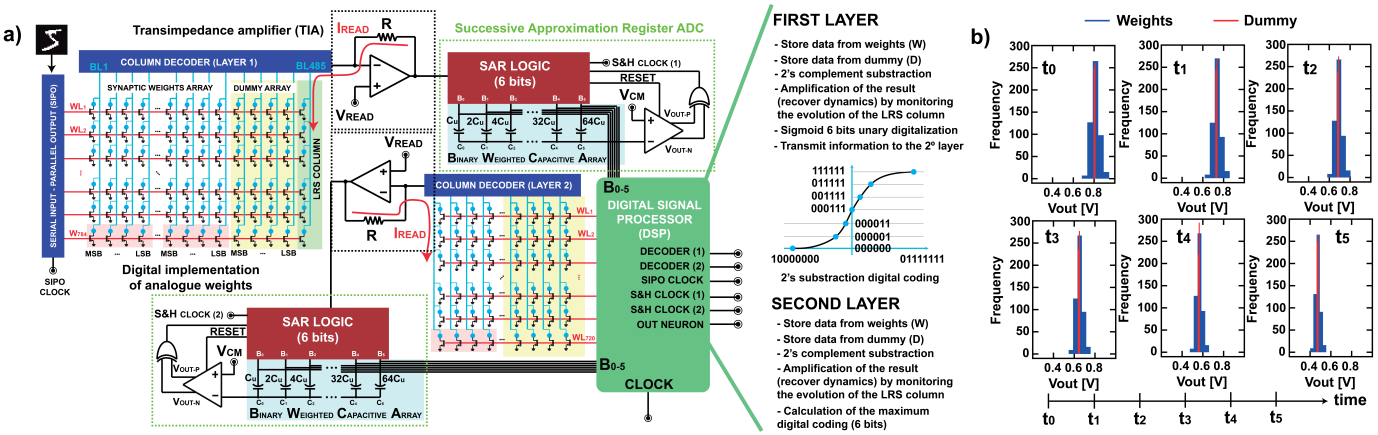

Fig. 3. (a) Block scheme of the proposed architecture for a drift-resilient neural network. The input corresponds to the dataset to classify during inference. The PCM arrays are used to map the synaptic weights and for compensating the drift ("dummy array"). The result of the MVM is collected by transimpedance amplifiers, converted using the ADCs and then elaborated into the DSP. Note that the evolution of the LRS is monitored in time by reading a further "LRS column", in order to maintain the same voltage dynamics of the initial programmed distribution and to reduce the drift effect of the weights. (b) Evolution in time of the MVM voltages of the first layer. Note that the distributions get smaller and narrower as a consequence of the drift of the PCMs. As observed, both distributions (dummy array and synaptic weights array) maintain the same relative distance, which is a key point to perform a correct 2's complement computation. Since the distributions become also narrower, in order to correctly apply the sigmoid function with the proper voltage dynamics, the digital voltage must be corrected by applying a digital amplification proportional to the evolution in time of the average LRS, which is actively measured from a column of the array.

[10], [11]. Several approaches have been introduced to limit this error, such as: (I) the use of multi-cells in order to reduce the variation of the devices [12]; (II) the recovery of the PCM state by means of partial set pulses [13]; (III) the analytical analysis of the drift evolution [14]–[16]; (IV) the design of specific neural networks robust against the drift of the weights [17]–[20]. However, these results are not enough when accurate inference is required in parallel with low area and power consumption [21], [22].

This work proposes a drift-resilient implementation of hardware neural networks with PCM arrays implementing the MVM procedure. Fig. 1 shows a multi-layer-perceptron (MLP) network trained with backpropagation that is here analysed as a case-study by both analytical and experimental standpoints. Simulations in Cadence Virtuoso with embedded PCM devices have been also performed in order to provide useful insights for the integrated design of PCM-based MLPs. In particular, the drift of the PCM devices can be compensated by a self-referential hardware, where the conductance decrease is monitored by dummy arrays. This approach enables the use of strictly positive synaptic weights and of only positive rails [23] without accuracy loss for MNIST, rand-MNIST and fashion-MNIST datasets. The results are proposed by experimental validation of neural networks whose weights are programmed in multilevel-mode [24] or in binary-mode [5], [25]. We finally discuss the performance metrics in terms of throughput and energy efficiency. This work highlights the benefits of non-volatile memory-based design for neural networks.

II. PCM DEVICES

Embedded PCMs were reported in 90 nm [26] and 28 nm nodes [27], thus enabling the design of in-memory computing hardware integrated in the usual CMOS platform. Fig. 2 shows that the PCM can be programmed between the low resistive state (LRS) and the high resistive state (HRS) by (a) set (with current I_{SET}) and (b) reset transitions, respectively [10]. The set transition relies on a gradual decaying current slope signal in order to induce the crystallization of the phase change material, while the reset transition shows a fast quenching to induce the high-resistive amorphous state. However, the PCM resistance suffers from drift at increasing time, which causes a progressive decrease of the initially programmed conductance. The drift affects both HRS, Fig. 2(c), and LRS, Fig. 2(d), distributions. Note that the initial LRS distribution can be modulated by a proper choice of I_{PULSE}, Fig. 2(e-h). Starting from a full set state, for instance, incremental resetting pulses can be applied in order to provide a distribution of incremental resistive states. Note that the conductance drift induces a final resistive value that is dependent on the initially programmed resistive state. This is a key limitation for the implementation of neural accelerators, since deep neural networks rely on a clear definition of the multilevel synaptic weights [28], [29].

III. DRIFT-RESILIENT NEURAL NETWORK

Since the drift of the resistive values is detrimental for AI applications, a hardware realization immune to such non-ideality would be significant for the PCM-based design of neural networks. Fig. 3(a) shows a drift-resilient approach to multi-layer perceptron (MLP) to overcome the limitations due to the PCM drift. The proposed MLP consists of two synaptic layers with N_{IN} input neurons, H_1 hidden neurons and 10 output neurons for classification. We chose N_{IN} = 784 neuron (accordingly to the size of MNIST, Rand-MNIST and Fashion-MNIST datasets) and H_1 = 120 neurons. Since the hidden layer is fed by the results of the MVM computation of the first layer, we implemented a unary coding of 6 bits of the sigmoidal activation function, thus needing a total of 720 wordlines (WLs). The MVM is executed in parallel in

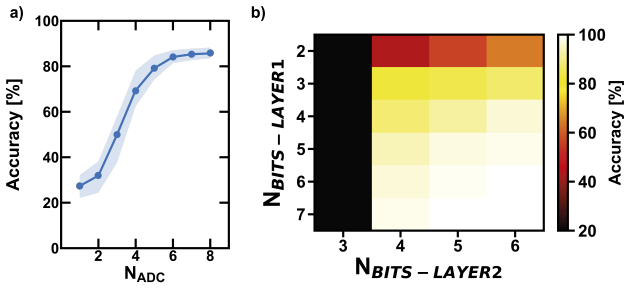

Fig. 4. Accuracy of the MNIST dataset as a function of the resolution of the ADC (number of bits) for an MLP neural network formed by one synaptic layer (a) and two synaptic layers (b), respectively (1000 Monte Carlo simulations with I_{SET} = 450 μA have been performed). The latter case shows the dependence of the accuracy on the number of conversion bits for both the synaptic layers. Note that the best result is achieved for 8 bits for the 1-synaptic-layer MLP and for 6 bits for the 2-synaptic-layer MLP.

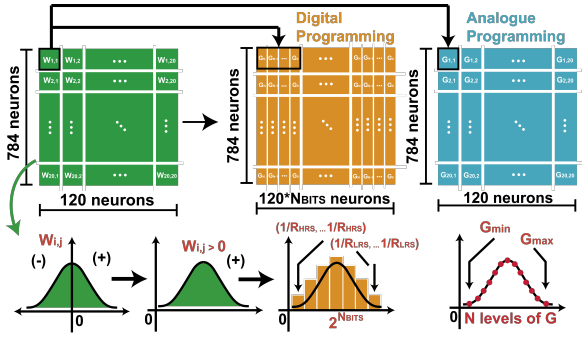

Fig. 5. Digital and analogue implementation of the matrix of the weights using PCM devices. In the digital approach, the synaptic weights are mapped by using combinations of binary PCM conductances, while in the analogue case the weights are mapped by exploiting the multilevel capability of the PCM devices. Note that for both the cases the weights are only positive since a rigid shift is applied to the distributions.

the PCM arrays to speed-up the computation. The output of the MVM is then converted by analogue-to-digital circuits (ADCs). The use of ADCs enables further data processing by a digital signal processor (DSP), which is co-integrated with the PCM arrays. The DSP implements the non-linear activation (sigmoidal) function σ mapped within the internal registers by look-up-tables (LUTs) and it enables the recovery of the narrowing effect of the MVM distributions reported in Fig. 3(b). Other non-linear functions can be also implemented, such as the rectifying linear unit (ReLU) or the clipped-ReLU [30], [31].

Note that the number of bits for the conversion of the ADC is related to the type and to the depth of the neural network. For instance, as shown in Fig. 4, a simple 1-synaptic-layer MLP (a) is more sensitive with respect to the ADC precision (8 bits are needed for the best computation) with respect to a 2-synaptic-layer MLP (b), where 6 bits are enough for providing the best result. This suggests that, given a neural network configuration (such as MLPs), deeper architectures require less effort for the analogue to digital conversion in order to achieve the best results.

The network can be designed with positive analogue weights or digital weights. In the first case, each synaptic device of the MLP is programmed until a specific synaptic value is achieved [7]. Digital weights 0 and 1 are instead written as HRS and LRS, respectively, thus resulting in larger memory area. In this latter case, N-bit MVM is then conducted by a shift-and-add approach [32]. The final idea is to create a differential configuration of the MVM, and not only at the synaptic level [33], in order to improve the immunity against drift and to increase the area and power efficiency.

A. Neural network with only positive weights

In this section, the drift-resilience at the system-level is going to be analytically demonstrated. From now on, we take into consideration as case study an MLP network with N_{IN} = 784 and H_1 = 120 neurons. The bias contribution will be neglected for simplicity.

The ideal output of each layer is given by:

$$Out_{IDEAL} = \sigma(Out_{L-1} * W_L), W_L \lesseqgtr 0, \qquad (1)$$

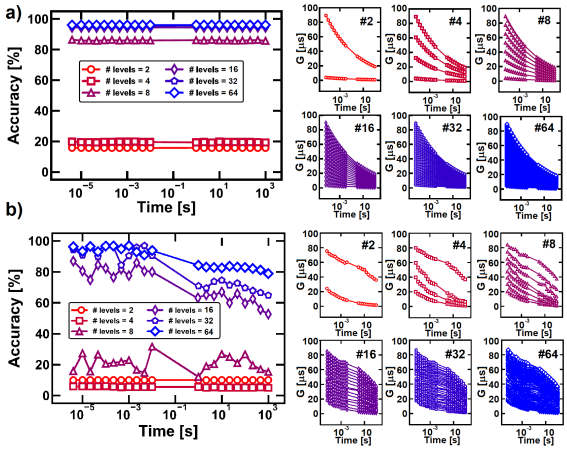

Fig. 6. (a) Simulated accuracy of the network as a function of the ideal drift of the analogue weights implemented with the PCM devices (from 2 to 64 levels). Note that the accuracy of the MNIST dataset remains constant in time, since the recovering algorithm perfectly retrieves the initial combination of the weights. (b) Monte Carlo simulations for the accuracy of the MNIST considering the best programming of the analogue weights (from 2 to 64 levels). The algorithm is initially able to recover the relative separation between the weights. After a certain time, however, the accuracy suffers from a significant drop.

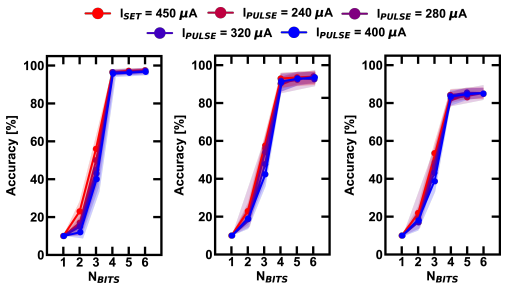

Fig. 7. Measured accuracies at time t_0 of MNIST (a), rand-MNIST (b), fashion-MNIST (c) datasets for different LRS programming conditions as a function of the number of bits (N_{BIT}) per single synaptic weight. Note that, for all the datasets, the accuracy remains more or less constant after 4 bits.

where Out_{L-1} is the vector of the outputs of the previous layer while W_L is the matrix of synaptic weights. As schemat-

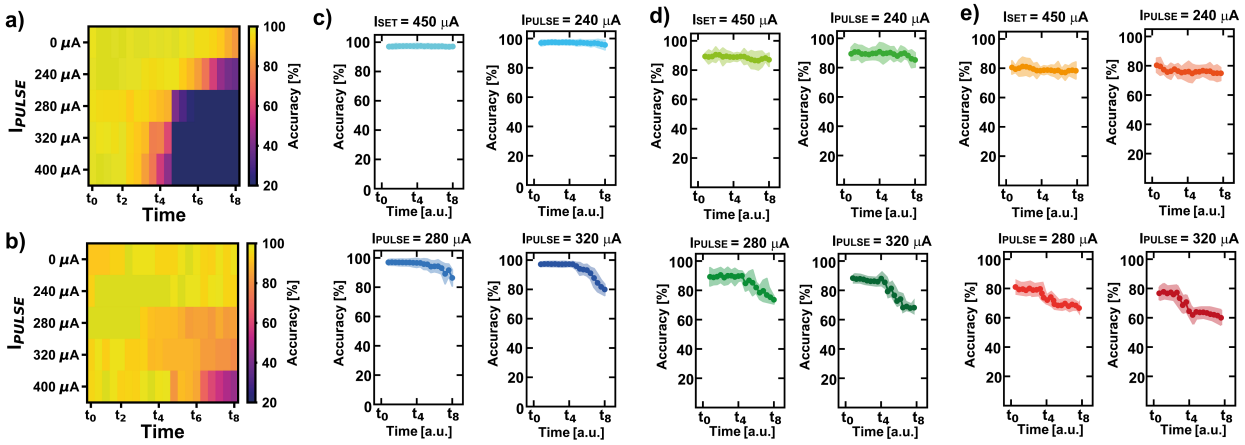

Fig. 8. Average evolution of the accuracy with time considering (a) just the use of dummy arrays as compensation method for the recovering the drift or (b) considering also the monitoring of the average value of the LRS. (c) Measured accuracy in time considering the drift of the PCMs for the MNIST (c), rand-MNIST(d), fashion-MNIST (e) at different programming conditions.

ically shown in Fig. 5, the weight distributions are then shifted by the absolute value of the lowest negative value $|W_{min}| > 0$, thus leading to a new weight $W_L^* = W_L + |W_{min}| > 0$ and to a different output:

$$Out_{PCM_{ARRAY}} = \sigma(Out_{L-1} * W_L^*), W_L^* > 0. \quad (2)$$

Eq. (1) can be recovered from Eq. (2) within the DSP by using the 2's complement mathematical definition to implement the negative values, Fig. 3. Note that more synapses are needed with respect the full analogue implementation, Fig. 5 [33]. The operation carried out by the DSP consists of the application of the sigmoidal function to the difference between (i) the MVM of the input with the positive weights and (ii) the MVM of the input with $|W_{min}|$. The output of the DSP is thus easily obtained as:

$$Out_{DSP} = \sigma(Out_{L-1} * W_L^* - Out_{L-1} * |W_{L,min}|), \quad (3)$$

$$Out_{DSP} = \sigma(Out_{L-1} * (W_L^* - |W_{L,min}|)) = Out_{IDEAL}, \quad (4)$$

The $|W_{L,min}|$ can be physically implemented by just a limited number of PCM bitlines, since $|W_{L,min}|$ maps the same digital value for all the neurons, as highlighted in green in Fig. 3. Note that $|W_{L,min}|$ is usually close to half of the dynamic range of the positive weights, as shown in Fig. 5.

B. Drift compensation

While allowing to map positive and negative weights as positive conductance values, Eq. (4) intrinsically introduces a drift-recovery procedure. In fact, since both W_L^* and $|W_{L,min}|$ drift in the same direction, the difference of the two MVM outcomes is approximately constant. Actually, the drift of the PCM devices makes the distributions of the MVM results narrower, Fig. 3(b). Thus, a continuous re-normalization in time of the MVM is needed to recover the full-scale range dynamics and accuracy of the initial programmed states. The digital implementation of the weights enables an efficient hardware procedure to recover the ideal distributions, since the PCM synaptic weights only contain LRS and HRS values, with the latter two orders of magnitude more resistive. The drift of the HRS is good, since the drift in time makes more negligible its contribution as '0' digital value. Ideally, each PCM-based digital synaptic weight should give the same voltage contribution to MVM as if it were digital, namely:

$$\frac{V_{FSR}}{2}\sum_{n=0}^{N-1} b_n \frac{1}{2^n} = K\frac{V_{Read}R_{Feed}}{2}\sum_{n=0}^{N-1} G_n \frac{1}{2^n}, \quad (5)$$

Where V_{FSR} is the ideal full scale range of the activation function, V_{Read} is the read voltage applied to the bitline for MVM (see Fig. 3), b_n is the digital word of the synaptic weight, R_{Feed} is the feedback resistance of the TIA collecting the MVM currents, and G_N is the digital word written as a function of the combination of LRS and HRS of the memory devices. Neglecting the HRS contribution in the MVM, Eq. (5) can be rewritten as:

$$\frac{V_{FSR}}{2}\sum_{n=0}^{N-1} b_n \frac{1}{2^n} = K\frac{V_{Read}R_{Feed}}{2} < G_{LRS} > \sum_{n=0}^{N-1} b_n \frac{1}{2^n}, \quad (6)$$

Where $< G_{LRS} >$ is the nominal conductance of the LRS value. Thus, from Eq. (6) we obtain the final factor K to be sampled in time for then carrying out a correct computation in the DSP:

$$K = \frac{V_{FSR}}{V_{Read}R_{FEED}} < R_{LRS} > . \quad (7)$$

As a result, a full dynamic compensation is achieved just by sensing and compensating the LRS drift in time, obtained by an additional array of LRS weights highlighted in green in Fig. 3(a). Such compensation is read by the ADC and then processed within the DSP. Thus, this approach enables power saving since the hardware is self-referential and does

not require cyclic refresh of the synaptic weights to track the drift of the PCM devices [13].

IV. Results

To test the drift-compensation algorithm of the MLP, we studied the testing accuracy of the network with N_{IN} = 784 and H_1 = 120 neurons using MNIST, rand-MNIST (randomness of 10%) and fashion-MNIST datasets.

A. Drift resilience of hardware neural networks

Assuming an ideal multilevel implementation of the weights and the same drifting coefficient for each programmed state, the compensation algorithm can fully recover the best accuracy, as shown in Fig. 6(a). However, the real PCM devices show different evolutions for each programmed synaptic value depending on the initial resistive state [10], which leads to a significant accuracy loss at increasing time, since the initial weights mix up, Fig. 6(b).

On the other hand, considering the discretization of the synaptic weights without errors, it is possible to get the best results using 4 bits per weight for all the datasets, Fig. 7. Thus,taking into consideration the digital programming of the PCMs implementing the synaptic words, Fig. 5, it is possible to overcome the drift limitation accordingly to the drift-recovery algorithm described in Section III. Fig. 8 shows the accuracy of the MNIST dataset considering only the differential MVMs expressed in Eqs. 1-4 (a) and with the recovery of the distribution narrowing described in Eqs. 5-7 (b). Fig. 8 also shows the accuracy as a function of time for MNIST (c), rand-MNIST (d) and fashion-MNIST (e), considering different programming conditions of the PCMs. The accuracy remains constant for a correct digital programming of the devices, while it drops if the LRS is modulated by I_{PULSE}.

B. Hardware performance

Although the hardware implementation with digital weights allows for a better drift recovery, additional area and energy consumption are required. Considering that the goal is to test the MNIST in less than 0.5 s, the ops/s (expressed in giga operations per second, GOPS) are calculated by computing the bit-level calculation that the hardware is able to provide during the inference of 10000 testing images. Considering (I) the read procedure of the synaptic and dummy arrays, (II) the tracking of the average LRS drift in time, (III) the number of implemented arrays, (IV) the master clock of 10 MHz, a total of 30 GOPS is obtained in the case of a 4-bits synaptic array, Fig. 9(a). Fig. 9(a) shows that the accuracy, as well as the GOPS, are functions of the number of bits used per each weight (from 1-bit to 6-bits are reported).

Note that, since the additional area of the dummy arrays is negligible with respect to the peripheral electronics [6], the specific throughput (TOPS/mm^2) increases at increasing number of bits, Fig. 9(b). On the other hand, Fig. 9(c) shows that the energy efficiency (TOPS/W) decreases with the increasing number of bits, since further computation is needed at increasing PCM array size. The limiting factor in terms of efficiency is the peripheral circuitry for the MVM signal management. All these results are in line with previous works for analogue based MLPs [6], [34] and have been computed in relation to hardware estimations using Cadence Virtuoso.

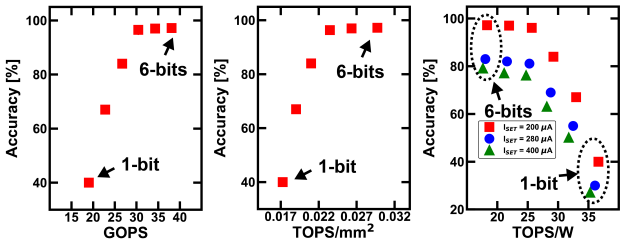

Fig. 9. (a) Accuracy of the MLP as a function of the ops/s for testing the MNIST in $<$ 0.5s for different number of bits (from 1 bits to 6-bits per synaptic weight). (b) Accuracy of the MLP as a function of the ops/s/mm^2. (c) Accuracy of the MLP as a function of the ops/s/W for three values of I_{SET}.

TABLE I
Summary of the Results

	Hardware with digital PCM devices		Software
Number of bits/levels	16 levels (4 bits)		Software training with analogue values
Accuracy (t_0)	I_{SET} = 200µA	97.1%	97.4%
Accuracy ($t_f = 10^6 t_0$)	I_{SET} = 200µA	96.8%	97.4%
GOPS (to get MNIST < 0.5s)	40		Software training and testing (no hardware implementation)
TOPS/mm²	0.03		
GOPS/W	20000		
Number of cycles for programming the weights	1		

Table I shows a comparison for the MNIST dataset between the pure software and the digital hardware approach. Note that the drift-resilient algorithm with digital weights has an accuracy near to the theoretical value, thus offering software-like results in terms of robustness.

V. Conclusions

This work introduced a methodology for the integrated hardware design of PCM-based neural networks immune to drift. The methodology relies on strictly positive synaptic weights, where drift is monitored and properly subtracted by using reference PCM arrays. The accuracy of the network is studied using a digital implementation of the weights, demonstrating high accuracies for MNIST, rand-MNIST and Fashion-MNIST datasets. Furthermore, the throughput and the energy efficiency are analysed, thus providing a useful guideline for designing fast and efficient neural accelerators based on PCM devices.

VI. Acknowledgements

The authors thank STMicroelectronics for experimental data and extensive discussions. This work received funding

from: European Research Council (ERC-2018-PoC-842472-CIRCUS) and Italian Minister of University and Research (R164TYLBZP).

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015. doi: 10.1038/nature14539.

[2] D. Silver et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, p. 484–489, January 2016. doi: 10.1038/nature16961.

[3] M. A. Zidan, J. P. Strachan, and W. D. Lu, "The future of electronics based on memristive systems," *Nature Electronics*, vol. 1, pp. 22–29, January 2018. doi: 10.1038/s41928-017-0006-8.

[4] S. Bianchi, I. M. Martín, and D. Ielmini, "Bio-inspired techniques in a fully digital approach for lifelong learning," *Frontiers in Neuroscience*, vol. 14, pp. 379–393, April 2020.

[5] S. Bianchi, G. Pedretti, I. Muñoz-Martín, A. Calderoni, N. Ramaswamy, S. Ambrogio, and D. Ielmini, "A compact model for stochastic spike-timing-dependent plasticity (STDP) based on resistive switching memory (RRAM) synapses," *IEEE Transaction on Electron Devices*, vol. 67, pp. 2800 – 2806, July 2020. doi: 10.1109/TED.2020.2992386.

[6] S. Ambrogio, P. Narayanan, H. Tsai, R. Shelby, I. Boybat, C. Nolfo, S. Sidler, M. Giordano, M. Bodini, N. Farinha, B. Killeen, C. Cheng, Y. Jaoudi, and G. Burr, "Equivalent-accuracy accelerated neural-network training using analogue memory," *Nature*, vol. 558, pp. 60–67, June 2018. doi: 10.1038/s41586-018-0180-5.

[7] I. Muñoz Martín, S. Bianchi, G. Pedretti, O. Melnic, S. Ambrogio, and D. Ielmini, "Unsupervised learning to overcome catastrophic forgetting in neural networks," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 5, pp. 58–66, June 2019. doi: 10.1109/JXCDC.2019.2911135.

[8] G. W. Burr, R. M. Shelby, A. Sebastian, S. Kim, S. Kim, S. Sidler, K. Virwani, M. Ishii, P. Narayanan, A. Fumarola, L. L. Sanches, I. Boybat, M. L. Gallo, K. Moon, J. Woo, H. Hwang, and Y. Leblebici, "Neuromorphic computing using non-volatile memory," *Advances in Physics: X*, vol. 2, December 2016. doi: 10.1080/23746149.2016.1259585.

[9] M. Hu et al., "Dot-product engine for neuromorphic computing: programming 1t1m crossbar to accelerate matrix-vector multiplication," pp. 1–6, 06 2016. doi: 10.1145/2897937.2898010.

[10] H. S. P. Wong, S. Raoux, S. Kim, J. Liang, J. P. Reifenberg, B. Rajendran, M. Asheghi, and K. E. Goodson, "Phase change memory," *Proceedings of the IEEE*, vol. 98, pp. 2201–2227, December 2010. doi: 10.1109/JPROC.2010.2070050.

[11] S. Kim, B. Lee, M. Asheghi, F. Hurkx, J. P. Reifenberg, K. E. Goodson, and H. S. P. Wong, "Resistance and threshold switching voltage drift behavior in phase-change memory and their temperature dependence at microsecond time scales studied using a micro-thermal stage," *IEEE Transaction on Electron Devices*, vol. 58, pp. 584–592, January 2011. doi: 10.1109/TED.2010.2095502.

[12] Y. Hwang, C. Um, J. Lee, C. Wei, H. Oh, G. Jeong, H. Jeong, C. Kim, and C. Chung, "Mlc pram with slc write-speed and robust read scheme," *2010 Symposium on VLSI Technology*, pp. 201–202, 2010. doi: 10.1109/VLSIT.2010.5556227.

[13] I. Boybat, S. R. Nandakumar, M. L. Gallo, B. Rajendran, Y. Leblebici, A. Sebastian, and E. Eleftheriou, "Impact of conductance drift on multi-pcm synaptic architectures," *2018 Non-Volatile Memory Technology Symposium (NVMTS)*, pp. 1–4, 2018. doi: 10.1109/NVMTS.2018.8603100.

[14] W. Kim, R. L. Bruce, T. Masuda, G. W. Fraczak, N. Gong, P. Adusumilli, S. Ambrogio, H. Tsai, J. Bruley, J. . Han, M. Longstreet, F. Carta, K. Suu, and M. BrightSky, "Confined pcm-based analog synaptic devices offering low resistance-drift and 1000 programmable states for deep learning," pp. T66–T67, 2019. doi: 10.23919/VLSIT.2019.8776551.

[15] D. Ielmini, S. Lavizzari, D. Sharma, and A. L. Lacaita, "Physical interpretation, modeling and impact on phase change memory (PCM) reliability of resistance drift due to chalcogenide structural relaxation," *2007 IEEE International Electron Devices Meeting (IEDM)*, pp. 939–942, January 2008. doi: 10.1109/IEDM.2007.4419107.

[16] S. Ambrogio, M. Gallot, K. Spoon, H. Tsai, C. Mackin, M. Wesson, S. Kariyappa, P. Narayanan, C. . Liu, A. Kumar, A. Chen, and G. W. Burr, "Reducing the impact of phase-change memory conductance drift on the inference of large-scale hardware neural networks," *2019 IEEE International Electron Devices Meeting (IEDM)*, pp. 6.1.1–6.1.4, 2019. doi: 10.1109/IEDM19573.2019.8993482.

[17] I. Boybat, M. L. Gallo, S. R. Nandakumar, T. Moraitis, T. Parnell, T. Tuma, B. Rajendran, Y. Leblebici, A. Sebastian, and E. Eleftheriou, "Neuromorphic computing with multi-memristive synapses," *Nature communications*, vol. 9, p. 15, 2018. doi: 10.1038/s41467-018-04933-y.

[18] S. Bianchi, I. Muñoz-Martin, G. Pedretti, O. Melnic, S. Ambrogio, and D. Ielmini, "Energy-efficient continual learning in hybrid supervised-unsupervised neural networks with PCM synapses," pp. T172–T173, June 2019. doi: 10.23919/VLSIT.2019.8776559.

[19] Y. Lin, C. Wang, M. Lee, D. Lee, Y. Lin, F. Lee, H. Lung, K. Wang, T. Tseng, and C. Lu, "Performance impacts of analog reram non-ideality on neuromorphic computing," *IEEE Transactions on Electron Devices*, vol. 66, no. 3, pp. 1289–1295, 2019. doi: 10.1109/TED.2019.2894273.

[20] I. Muñoz-Martin, S. Bianchi, S. Hashemkhani, G. Pedretti, O. Melnic, and D. Ielmini, "A brain-inspired homeostatic neuron based on phase-change memories for efficient neuromorphic computing," *Frontiers in Neuroscience*, vol. 15, p. 1054, 2021. doi: 10.3389/fnins.2021.709053.

[21] A. Sebastian, M. L. Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nature Nanotechnology*, vol. 15, pp. 529–544, 2020. doi: 10.1038/s41565-020-0655-z.

[22] Z. Wang, C. Li, W. Song, M. Rao, D. Belkin, Y. Li, P. Yan, H. Jiang, P. Lin, M. Hu, J. P. Strachan, N. Ge, M. Barnell, Q. Wu, A. G. Barto, Q. Qiu, R. S. Williams, Q. Xia, and J. J. Yang, "Reinforcement learning with analogue memristor arrays," *Nature electronics*, vol. 2, pp. 115–124, March 2019. doi: 10.1038/s41928-019-0221-6.

[23] 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 European Solid State Circuits Conference*, pp. 135–138, November 2019. doi: 10.1109/ESSCIRC.2019.8902656.

[24] M. Suri, V. Sousa, L. Perniola, D. Vuillaume, and B. DeSalvo, "Phase change memory for synaptic plasticity application in neuromorphic systems," *The 2011 International Joint Conference on Neural Networks*, pp. 619–624, 2011. doi: 10.1109/IJCNN.2011.6033278.

[25] I. M. Martin, S. Bianchi, E. Covi, G. Piccolboni, A. Bricalli, A. Regev, G. Molas, J. F. Nodin, E. Nowak, and D. Ielmini, "A SiOx RRAM-based hardware with spike frequency adaptation for power-saving continual learning in convolutional neural networks," *2020 Symposia on VLSI Technology and Circuits*, June 2020. doi: 10.1109/VLSITechnology18217.2020.9265072.

[26] P. Zuliani, E. Palumbo, M. Borghi, G. Dalla Libera, and R. Annunziata, "Engineering of chalcogenide materials for embedded applications of phase change memory," *Solid-State Electronics*, vol. 111, pp. 27 – 31, 2015. https://doi.org/10.1016/j.sse.2015.04.009.

[27] F. Arnaud et al., "Truly innovative 28nm fdsoi technology for automotive micro-controller applications embedding 16mb phase change memory," *2018 IEEE International Electron Devices Meeting (IEDM)*, pp. 18.4.1–18.4.4, 2018. doi: 10.1109/IEDM.2018.8614595.

[28] H. Tsai, S. Ambrogio, P. Narayanan, R. Shelby, and G. W. Burr, "Recent progress in analog memory-based accelerators for deep learning," *Journal of Physics D: Applied Physics*, vol. 51, June 2018. doi: 10.1088/1361-6463/aac8a5.

[29] V. Joshi, M. L. Gallo, S. Haefeli, I. Boybat, S. R. Nandakumar, C. Piveteau, M. Dazzi, B. Rajendran, A. Sebastian, and E. Eleftheriou, "Accurate deep neural network inference using computational phase-change memory," *Nature Communications*, vol. 11, May 2020. doi: 10.1038/s41467-020-16108-9.

[30] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv*, vol. cs.NE, 2019. doi: 1803.08375.

[31] Z. Cai, X. He, J. Sun, and N. Vasconcelos, "Deep learning with low precision by half-wave gaussian quantization," July 2017. doi: 10.1109/CVPR.2017.574.

[32] G. W. Reitwiesner, "Binary arithmetic," *Elsevier, Advances in Computers*, vol. 1, pp. 231 – 308, 1960.

[33] G. W. Burr, R. M. Shelby, S. Sidler, C. di Nolfo, J. Jang, I. Boybat, R. S. Shenoy, P. Narayanan, K. Virwani, E. U. Giacometti, B. N. Kurdi, and H. Hwang, "Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element," *IEEE Transactions on Electron Devices*, vol. 62, no. 11, pp. 3498–3507, 2015. doi: 10.1109/TED.2015.2439635.

[34] H. Mujtaba, "NVIDIA Volta GV100 12nm FinFET GPU detailed-Tesla V100 specifications include 21 billion transistors, 5120 CUDA Cores, 16 GB HBM2 with 900 GB/s bandwidth," *WCCFTECH*, May 2017. https://wccftech.com/nvidia-volta-gv100-gpu-tesla-v100-architecture-specifications-deep-dive/.