A Multilayer Neural Accelerator With Binary Activations Based on Phase-Change Memory

Marco Bertuletti, *Student Member, IEEE*, Irene Muñoz-Martín, *Member, IEEE*, Stefano Bianchi, *Member, IEEE*, Andrea G. Bonfanti, *Senior Member, IEEE*, and Daniele Ielmini, *Fellow, IEEE*

***Abstract*— Novel in-memory computing circuits, based on arrays of emerging nonvolatile memories, such as the phase-change memory (PCM), can boost cutting-edge performances of artificial intelligent applications. However, the spread of PCM-based circuits is currently hindered by the lack of a design framework enabling fast, efficient, and low-power neural networks. In this work, a novel approach to the conceptual and technical design of integrated neural networks is proposed. In particular, to relax the power hunger and complexity of state-of-the-art solutions, we propose a fully analog computing approach where the analog-to-digital converter (ADC) is replaced by a simple comparator. The analog building blocks of the accelerator are presented and validated in Cadence Virtuoso. The major nonidealities, such as PCM conductance variability, conductance drift, IR drop, and readout threshold, are studied by considering their impact on accuracy.**

***Index Terms*— Artificial intelligence, hardware accelerator, in-memory computing, neural network, nonvolatile memory, phase-change memory (PCM).**

I. Introduction

MACHINE learning (ML) has emerged as one of the most disruptive sciences of the latest years demonstrating human-like skills in tasks, such as image recognition and natural language processing [1]. ML algorithms usually rely on intensive matrix-vector multiplication (MVM), which results in time- and energy-consuming transfer of input data and model parameters between the dynamic random access memories (DRAMs) and the CPUs [2], [3]. On the other hand, in-memory computing emulates the parallel computation of the brain [4], [5], [6], thus overcoming the main limitations of conventional digital computing systems, also because of high-density, back end of the line nonvolatile memories, such as resistive random access memory (RRAM) and phase-change memory (PCM) [5], [7], [8], [9], [10]. In particular, as shown in Fig. 1(a), the MVM is the most intensive operation for implementing hardware accelerators of large neural networks. MVM can be executed efficiently by in-memory computing hardware because of the use of memory arrays of nonvolatile memories [11], which are shown in Fig. 1(b). These cross-point arrays inherently perform parallel multiply-and-accumulate (MAC) operations by direct application of Ohm's and Kirchhoff's laws [2], [12]. Several mixed-signal accelerators integrating a memory array have been proposed for neural network inference [13], [14], [15], [16], [17], [18], [19], [20], [21], [22]. The performance of some of these accelerators in terms of throughput, energy efficiency, and input, output, and weights precision is reported in Table I. Hardware accelerators based on in-memory computing usually rely on a mixed analog-digital approach with analog-to-digital

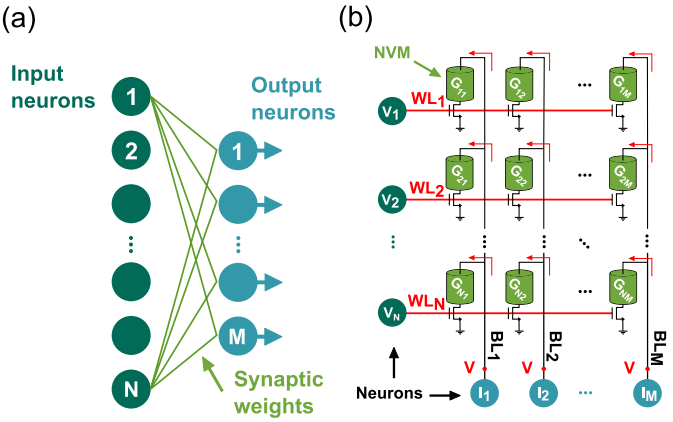

Fig. 1. (a) General concept of a neural network accelerator and (b) implementation of the synaptic weights using 1T1R PCM devices. Note that presynaptic and postsynaptic neurons are also depicted in the figure.

Manuscript received 1 December 2022; accepted 19 December 2022. Date of publication 30 January 2023; date of current version 24 February 2023. This work was supported by the Electronics Components and Systems for European Leadership (ECSEL) Joint Undertaking (JU) under Grant 101007321. The JU was supported by the European Union's Horizon 2020 Research and Innovation Program and France, Belgium, Czech Republic, Germany, Italy, Sweden, Switzerland, and Turkey. The review of this article was arranged by Editor P.-Y. Du. *(Corresponding author: Daniele Ielmini.)*

Marco Bertuletti was with the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, 20133 Milan, Italy. He is now with the Integrated Systems Laboratory (IIS), ETH Zürich, 8092 Zürich, Switzerland (e-mail: marco.bertuletti@mail.polimi.it).

Irene Muñoz-Martín and Stefano Bianchi were with the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, 20133 Milan, Italy. They are now with Infineon Technology, 9500 Villach, Austria (e-mail: irene.munoz@polimi.it; stefano1.bianchi@polimi.it).

Andrea G. Bonfanti and Daniele Ielmini are with the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano and IU.NET, 20133 Milan, Italy (e-mail: andrea.bonfanti@polimi.it; daniele.ielmini@polimi.it).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TED.2022.3233292.

Digital Object Identifier 10.1109/TED.2022.3233292

TABLE I
ANALOG MIXED-SIGNAL CIRCUITS FOR NEURAL NETWORK INFERENCE BASED ON EMBEDDED NONVOLATILE MEMORIES

		[13]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]
Device		RRAM	RRAM	RRAM	PCM	PCM	RRAM	RRAM	RRAM	RRAM
# input bits		1	1-4	1	1	1-5	1-8	1	8	1-4
# weight bits		8	2-4	3	4	4-11	1-8	1	4	1-4
# output bits		1	6-11	1-8	N/A	8-11	3-19	1-4	8	6
GOPS		660	28.91-13.99	0.02	1.7	3900-475	5120-142	N/A	N/A	5242
TOPs/W		20.7	146.21-36.16	78.4	N/A	718-65	21.6-416.5	26.56	95.8	238
Accuracy	MNIST	90.8%	N/A	94.4%	97.13%	N/A	N/A	N/A	N/A	N/A
	CIFAR-10	N/A	90.88%	N/A	N/A	91.89%	91.74%	N/A	N/A	90%
	CIFAR-100	N/A	65.71%	N/A	N/A	67.53%	67.11%	N/A	N/A	N/A

converter (ADC) for two main reasons: 1) achieving sufficient signal integrity and 2) enabling digital processing of the information, including activation functions, shift-and-add, and normalization operations, which might not be straightforward in the analog domain. Note that an ADC consumes more than 80% and 60% of the circuit power and area, respectively [23], [24]. Thus, simplifying or even removing the ADC would result in a strong improvement for the integrated design of neural networks accelerators.

This work addresses the hardware design of a PCM-based multilayer neural network with comparator-based nonlinear activation functions. The circuit speeds up the workload by avoiding the use of ADCs, and it is tested for the recognition of Modified National Institute of Standards and Technology (MNIST) and Fashion-MNIST datasets. An activation-slope aware training method is used, and the accuracy loss is minimized in the tests using step function activations and quantized 4-bits weights. The network is implemented on a cross-point array of real conductances leveraging a multibit resistive weight mapping. This technique is extended from the array of RRAM cells use case presented in [14] and [15] to our PCM-based network. Previous works investigated the impact of PCM variability on the classification accuracy. In [25], the variability and drift of PCM devices were simulated by rescaling the network weights by the maximum conductance of an experimental distribution. Programming variation with zero mean and variance obtained from the Gaussian fitting of the distribution was added. In our work, we extend these results by considering the effects of PCM variability on a quantized network with multibit resistive weights, thus offering a comprehensive picture on the joint effect of weight quantization and variability. We focus on the interaction between these nonidealities and the Heaviside activation of our network. We also consider the impact of IR drop on the network accuracy. Differently from the analysis carried on in previous works [26], we evaluate the effects of IR drop on a complete inference task. The higher average conductance of the cells adopted in our work also forces to address a more severe IR-drop configuration.

Our work presents a fully analog hardware implementation of the bit-line (BL) readout and neural activation, including a transimpedance stage, a weighting star of resistors, an integration stage, and a dynamic StrongARM comparator. The joint effect of conductance variability, drift, IR drop, and readout threshold on the network accuracy is carefully investigated.

II. CROSS-POINT ARRAY FOR MVM

Fig. 1(b) shows the cross-point memory array based on one-transistor/one-resistor (1T1R) PCM cells, with the top electrodes (TEs) of the PCM devices connected to the BLs and the bottom electrodes to the drain of MOS transistors, which work as selectors. The gate of the transistor is increased by applying a voltage signal at the corresponding word line (WL), while the TE is kept at a relatively low read voltage V_{READ}. The resulting BL current is proportional to the dot product between the input vector of the gate voltages applied to the WLs (binary on/off voltages) and the vector of analog conductance values stored in the memory cells of the BL.

III. SLOPE-UPDATE TRAINING

The network we propose in this work has 784 binary inputs, ten outputs, and 150 hidden neurons. An element always set to one is also appended to the input vector of each layer, so that the bias can be implemented [1]. After an MVM operation, the outputs of each layer undergo a nonlinear activation. We used the logistic function

$$\sigma(x) = \frac{A}{1 + e^{-Bx}} \tag{1}$$

where A and B are free parameters, as nonlinear activation function. Fig. 2(a) shows the logistic function for increasing B, which describes the slope of the activation function. During training, the input patterns are feedforwarded across the network. Errors can be evaluated by computing the squared difference between the one-hot code of ten elements associated with the input image and the actual output of the network. The network weights are modified according to the product between this error and a model parameter, called learning rate. Backpropagation and gradient descent are used during the process [27]. The hyperparameters of the model, namely, A and B in (1), the number of epochs and the learning rate η, were selected to obtain, at the same time, the largest accuracy and a high slope for the activation function, thus allowing to implement the logistic function in (1), as an analog comparator. At high slope coefficients B, none of the combinations of the other parameters gave high accuracy. The slope of the activation function was then increased gradually during training, as shown in Fig. 2(a). This method was called *slope-update* training. Two variants of the method are considered to optimize the training process: in the derivative function $\sigma'(x) = (A'B'e^{-B'x})/(1 + e^{-B'x})$ used for backpropagation,

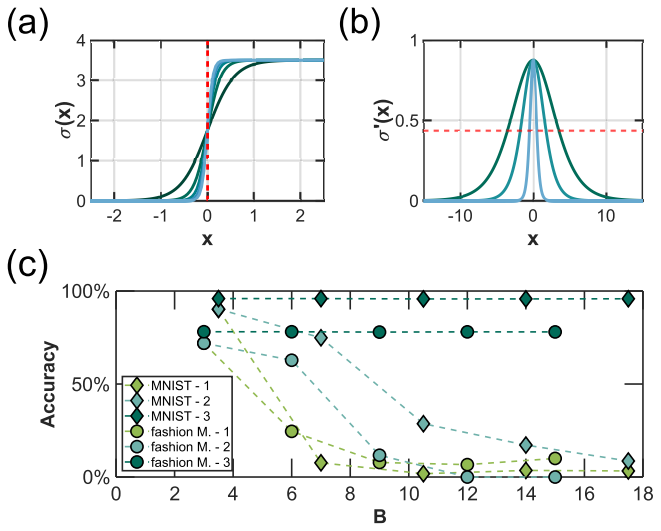

Fig. 2. (a) Activation function of the neural network at incremental slope. (b) Normalized derivative of the slope of the activation function. (c) Software accuracy on MNIST and Fashion-MNIST for different training schemes.

TABLE II
TOP ACCURACIES FOR DIFFERENT TRAINING SCHEMES

	High FWHM	Slope update	Quantization	MNIST	Fashion MNIST
1	no	no	no	90%	71.9%
2	no	yes	no	90.3%	72%
3	yes	yes	no	96%	78.6%
4	yes	yes	yes	95.5%	78%

parameters A' and B' are either chosen equal to A and B of the function used in the forward propagation or selected to obtain a larger full-width half-maximum (FWHM), as shown in Fig. 2(b). Fig. 2(c) shows the top accuracy achieved for the different training methods as a function of parameter B. In these calculations, weights were assumed with floating-point precision, and a test was performed with comparators as activations. Various training schemes were assumed, as summarized in Table II. The training using *slope update* minimizes the accuracy loss when the final slope of the activation function increases. *Slope update* coupled with a large FWHM $\sigma'(x)$ allows efficient backpropagation of the errors, thus ensuring the highest accuracy.

IV. Quantization

Nonvolatile memory devices, such as PCM cells, are not suitable to represent high-precision analog weights [28]. The 64-bits full-precision weights obtained by the supervised training algorithm must then be quantized to limited-precision levels. Fig. 3 schematically illustrates the quantization and mapping steps. For each network layer, the full-precision weights generally have a zero-mean distribution as in Fig. 3(a). Note that 4 bits are used for quantization; thus, 16 quantized levels can be represented. The 16 equally spaced values selected between $\pm(3.5 \times \sigma)$ of the full-precision distribution are chosen for this purpose. In order to minimize the accuracy loss, the incremental-quantization method proposed in [29] has been adopted. As shown in Table II, this approach coupled to the *slope update*, and the use of a large FWHM backpropagation function ensured a small accuracy loss.

In a 1T1R memory array, single PCM cells cannot represent positive and negative weights, because the current is sourced in a single direction. To avoid this limitation, we rely on the multibit resistive weight approach presented in [9], [14], and [15]. As shown in Fig. 3(a), the quantized distribution of weights is shifted on a positive range only. The product between the input vector and a positive and negative matrix is recovered by subtracting a midpoint reference, as shown in Fig. 3(b). The positive quantized weights are mapped on the conductance of the PCM array, programed in either a low-resistive state (LRS) or a high-resistive state (HRS). By combining these binary (LRS/HRS) PCM conductances, 4-bit binary codes are obtained to describe each one of the 16 quantized levels. The value of a dot-product operation is obtained by binary weighting of the currents of four PCMs.

V. Simulations With PCM Distributions

The effect of the variability and drift of PCM devices on the network accuracy is simulated considering the conductance distributions obtained from previous experimental measurements [30]. The PCM devices were initialized by a forming operation to uniformize [31] the Ge-rich composition of the phase change material, as shown in Fig. 4(a). Higher forming currents can be used to create a larger conductive region in the PCM device. Fig. 4(b) shows the average value and the variance of the LRS conductance G_{LRS}, as a function of the forming current [32]. Fig. 4(c) shows the computed network accuracy versus I_{FORM}. Higher forming currents lead to a better accuracy, although at the expense of a higher power consumption. In fact, by programming the cells with $I_{\mathrm{FORM}} = 150\ \mu$A, a BL draws at maximum 165 μA, while the maximum current is 317 μA for $I_{\mathrm{FORM}} = 550\ \mu$A, roughly corresponding to a two times larger power dissipation. Since accuracy and power dissipation are trade-offs, applications that require a low power consumption may have accept a relatively low classification accuracy.

The performance of the network is evaluated at increasing time, to account for the effect of the drift. Fig. 5(a) shows the PCM conductance as a function of time after the programming pulse. The measured distributions of PCM devices are fit to extract their mean value and variance. The conductances used in network simulations are extracted by the obtained distributions. Fig. 5(b) and (c) shows the accuracy obtained by considering the time evolution of the PCM conductance. In the simulations corresponding to the red plot, the mean values of the LRS and the HRS distributions decrease over time, as well as the dynamic range of the differential analog dot product. The accuracy of the network is not affected; because of the self-referential implementation of the analog circuitry, the sign of this difference with respect to an infinite slope activation threshold is preserved. The conductance variance increases with time for both the HRS and the LRS. The evolution in time of the conductance variance causes the statistical spread of the analog dot products around the Heaviside activation

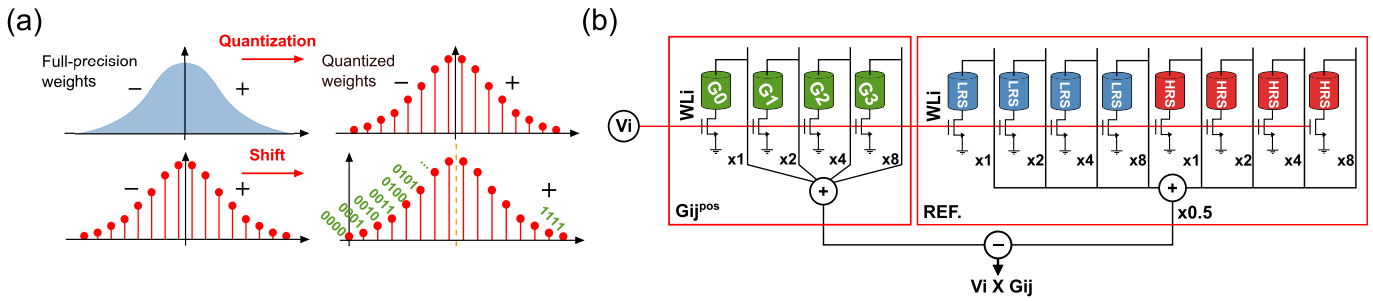

Fig. 3. (a) Network is trained by incremental quantization with 4-bit precision: then, weights are shifted to positive and mapped in four binary PCM cells. (b) Illustration of weight mapping with four binary PCM cells and eight reference cells, producing the average between 0000 and 1111.

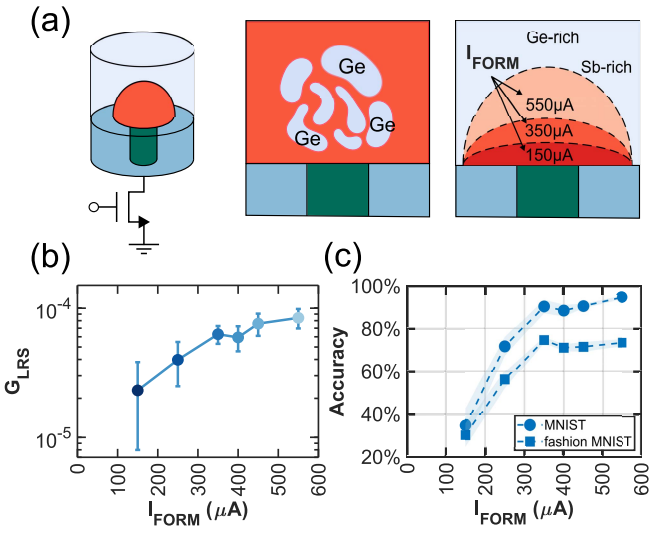

Fig. 4. (a) Scheme of the forming process in a virgin PCM cell. (b) Average LRS conductance as a function of forming current. (c) Simulated accuracy for MNIST and Fashion-MNIST datasets.

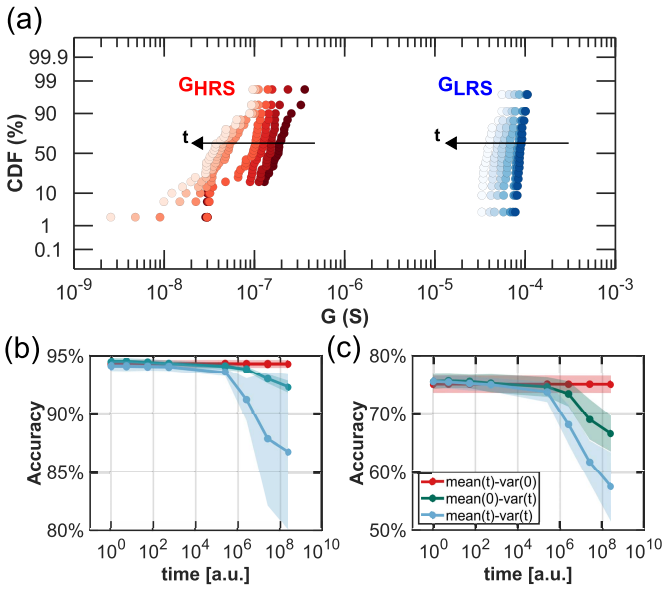

Fig. 5. (a) Distributions of conductance for HRS and LRS of PCM cells at increasing time. (b) Simulated accuracy for MNIST. (c) Simulated accuracy for Fashion-MNIST.

threshold. As shown in the green plot of Fig. 5(b) and (c), when the time evolution of the conductance variance only is considered, a moderate accuracy drop is observed. On the contrary, when both the mean value and the variance of the PCM conductance distribution evolve in time, the conductance variance has a larger impact on accuracy. In this case, as shown in the light-blue plot of Fig. 5(b) and (c), the statistical spread superimposed to the lower dynamic range dot-product analog signals causes a larger accuracy drop. Nevertheless, the simulations results demonstrate a good resilience to drift by the network because of the self-compensation between the weights in the array and in the reference combined by the step-like activation function.

VI. Simulations With IR Drop

The current flowing in the BL causes a voltage drop across the BL wire resistance and across the resistance of the BL decoder, as shown in Fig. 6(a). A decoder circuit is indeed necessary to share the readout among different BLs of the array. This IR-drop effect superimposes an input dependent error to the current of each cell, thus affecting the linearity of the MVM operation [26], [33]. The network was simulated to take into account the voltage drop across the decoder resistance and the IR drop accumulated along the BL, assuming a cell-to-cell wire resistance $R_{\text{wire}} = 0.5\ \Omega$. To reduce the impact of IR drop, the readout of a BL was divided into different steps by reading 16, 32, 64, or 128 cells. The current obtained from different read steps was then integrated in the analog domain, because the step function used as activation needs to operate on the result of a full analog dot product. Fig. 6(b) and (c) shows the simulation results of the impact of IR drop. The results indicate that good performance can be obtained if the voltage drop across the decoder resistance is kept below 10% of the read voltage, with a reading of 32 out of 128 WLs for MNIST and of 16 out of 128 WLs for Fashion-MNIST. Based on the evidence that a higher forming current results in a higher accuracy and yet a higher voltage drop on the BL, the compensation of the IR drop on the network accuracy has to take into account the average conductance of the PCM cells in the array. In the framework of a hardware–software codesign, the PCM array and the reading circuit could be jointly reprogrammed. When higher PCM conductances, hence higher network accuracies, are targeted, the number of BL cells read at a time can be reduced, without

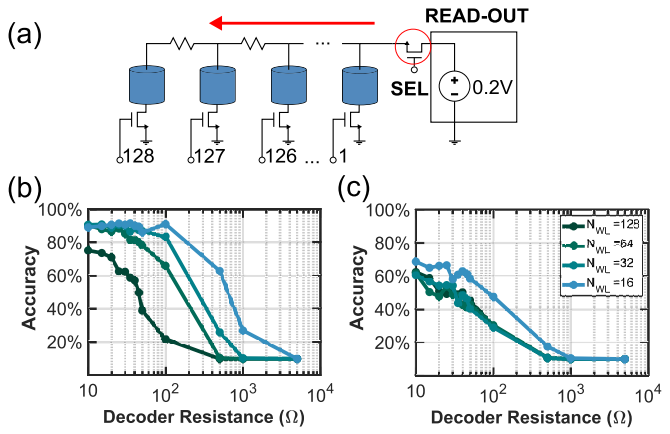

Fig. 6. (a) Sketch of the parasitic resistance across the wire and the decoder. (b) Simulated accuracy as a function of the decoder resistance for MNIST. (c) Simulated accuracy for Fashion-MNIST.

enforcing expensive retraining processes to minimize the IR drop.

VII. Readout Circuit

Fig. 7(a) schematically shows the first layer of the network. Input signals are applied to the WLs of the array in parallel. The BLs share the readout circuits of each array by means of a BL decoder, whose input dimensions depend on the number of outputs of the network layer. A readout and weighting circuit like the one represented in Fig. 7(b) is used to assign a binary weight to each of the four BLs, while clamping the BL read voltage to 0.2 V. The readout circuit includes four transimpedance stages to convert the BL current in a voltage. The outputs of these amplifiers are connected using binary weighted resistors, having resistance R, $2R$, $4R$, and $8R$. A second noninverting stage is used to amplify the signal. The output of the second stage is, thus, given by

$$0.2\ \text{V}\left[\left(1+\frac{R_B}{R_A}\right)\sum_{k=0}^{3}\frac{8}{15}\left(1+\frac{R_T}{2^k R_{\text{BL},k}}\right)-\frac{R_B}{R_A}\right] \quad (2)$$

which is equivalent to the result of a dot product between a 16-, 32-, 64-, or 128-WL input and a four-BLs column vector. The same readout integrates both the positive array and the reference, thus avoiding any potential mismatch due to process variations. Since the synaptic weights are split in different arrays and only 32 out of 128 WLs are activated at a time, analog integrators are used to accumulate the partial dot products from different arrays and different read steps. Connecting a different BL segment to the transimpedance inverting input at each read step modifies the circuit linear response and causes transient effects. A two-step integration avoids the influence of these effects on the integrated charge. Fig. 7(c) shows the integration circuit. In the first integration step, the output of the readout reaches steady state, and in the second, the charge stored across C_1 is transferred to the feedback capacitor C_2. While one channel integrates the BL currents, the other one integrates the two reference columns, and the latter requiring a 1/2 gain factor in the ratio (C_1/C_2) to set the reference voltage to half of the dynamic range. Since, given an input pattern, the reference is the same for any of the equivalent four-BLs columns, the charge integrated on the reference side is maintained during the integration of all the BLs in the positive array. The integrated outputs on the array and the reference side are used as input voltages of the dynamic StrongARM comparator, which is designed based on a differential stage driving a latch, as shown in Fig. 7(d) [34]. The output bits of the comparison are then used as binary inputs for the next network layer. A control block is used to set the timing of the circuit and to manage the communication between layers. When a layer completes the dot product and activation operations for the following one, the inputs are propagated forward in parallel mode. With the proposed readout and integrator, reading a group of four BLs takes two clock cycles. To reduce the impact of IR drop, a 128-cells BL is divided in four sections of 32 cells. The 2×4 clock cycles are then used to integrate a full BL, and two additional clock cycles are needed to generate the comparator output and reset the integrator. The classification of the full MNIST dataset in a network with 150 hidden neurons requires to read 152×4 BLs, including the reference, for 10 000 patterns. The circuit design assumes a clock period of 50 ns, and the classification can, thus, be completed in less than 1 s.

VIII. Network Simulations

Fig. 8 shows a summary of the simulated accuracy for MNIST [Fig. 8(a)] and Fashion-MNIST [Fig. 8(b)], where the various nonidealities are separately considered. The baseline accuracy is reported for training with binary inputs, 64-bits weights, and testing by using step functions instead of sigmoids. The following items show the accuracy for different nonidealities taken into account. We considered quantization, conductance variability, drift, and IR drop as well as the nonideal properties of the readout, namely, the comparator offset, and the mismatch between the integration channels. These statistical variables are obtained from Monte Carlo circuit simulations on Cadence Virtuoso. The statistical variations of the PCM conductance affect the accuracy by introducing a statistical uncertainty around the result of the analog dot product, which is presented at the comparator input. IR drop and drift instead lead to an overall reduction of the signal full-scale range. The combination of these effects results in lower accuracies.

IR drop is caused by the current flowing in a BL and increases with the number of PCM synapses and their respective current. Fashion-MNIST is strongly affected by IR drop, as a result of the lower sparsity of the input pattern. When drift is included, the accuracy slightly increases with respect to the time-zero test, as the average PCM resistance increase causes a lower current to flow in the array, which suppresses the IR drop. Drift and IR drop can, thus, positively affect each other.

The strive for energy efficiency of edge devices pushes toward the binarization of convolutional neural networks (CNNs) [35]. Long short-term memory (LSTM) networks with binary activations and states and binary graph neural networks (GNNs) were proposed too [36], [37]. We, therefore, consider

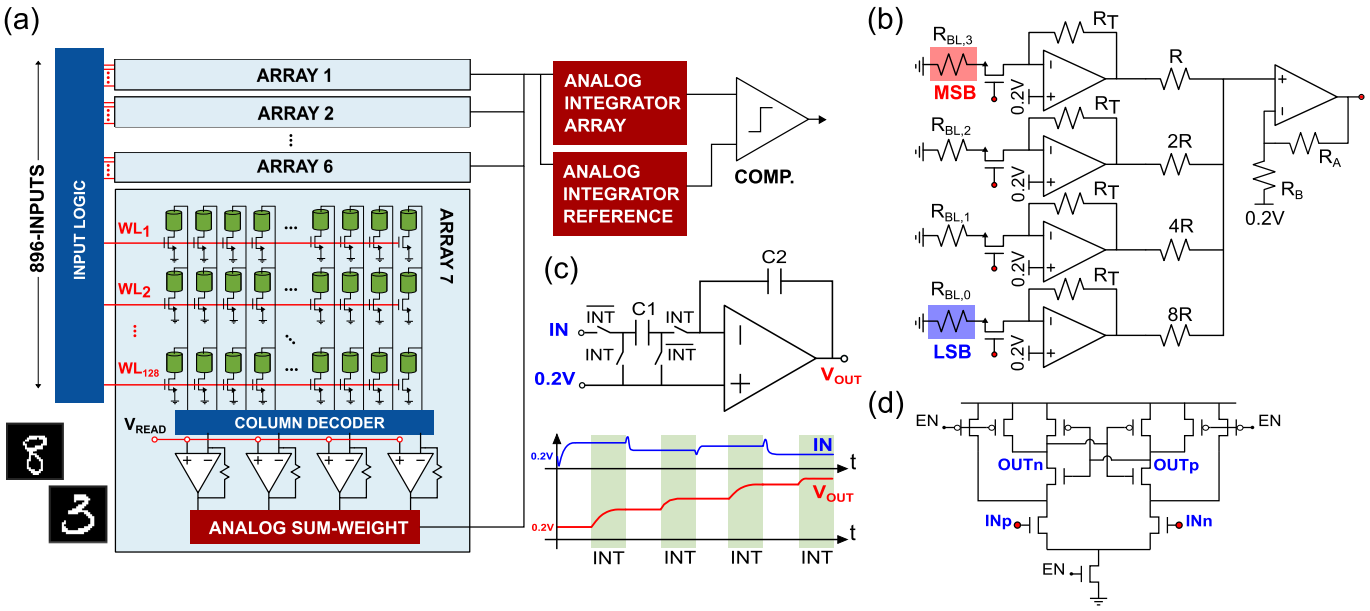

Fig. 7. (a) Illustration of the analog/digital implementation of the neural network for a single layer. (b) Readout circuit for weighted summation of the synaptic currents in the analog domain. (c) Integrator circuit and evolution over time of the output voltage during the integration steps. (d) Dynamic StrongARM comparator circuit.

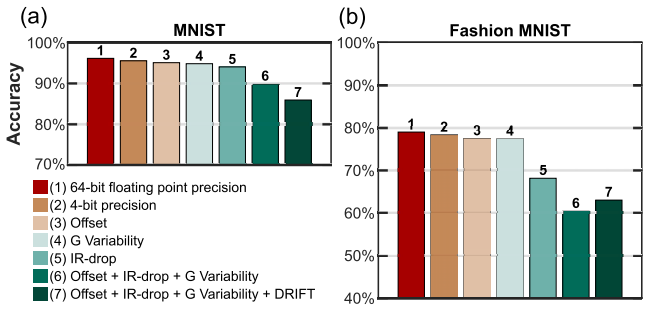

Fig. 8. Assessment of accuracy for the hardware accelerator, including the impact of limited precision, comparator offset, PCM variability, IR drop, and drift for (a) MNIST and (b) Fashion-MNIST, respectively.

our approach transferrable to other network architectures and problem sizes. The dynamic interaction of PCM variance, PCM drift, IR drop, and their effect on the decision performed by the binary threshold of the comparator presented in this article can indeed serve as a reference for the hardware implementation of any binary activated network.

To address the deployment of networks requiring higher precision of the activation, an extension of the proposed circuit is possible. In our work, we compared the result of a weighted analog dot product to a midpoint reference, generating a binary activation. In applications that require activation with higher precision, the analog dot product could be compared with multiple reference levels, generated from additional reference BLs, similar to what was discussed in [38]. If each one of the new voltage references is held by a replica of the reference integrator, the area occupied by the readout circuit on the in-memory computing chip increases. Using a single reference integrator and scheduling in time the comparisons with different references require more time to infer an input pattern. In both cases, the power consumption of the circuit increases. The analysis of these trade-offs is left to future work.

The works on quantization of attention-based GNNs show that the quantization of the attention layers is critical for accuracy performances, requiring up to 32b in a full integer inference [39]. Nevertheless, these network models are mainly intended to be deployed on the cloud, whereas the in-memory computing circuit proposed targets edge applications, where the exploration could be restricted to lower precision ranges.

IX. Conclusion

This work presented the study of a multilayer hardware neural accelerator based on PCM synapses. The communication between different layers of the network was simplified by the use of comparators instead of ADCs. The effect of the variability and drift of PCM binary state on accuracy was addressed. IR drop was minimized by introducing an integrator, to read smaller portions of a BL in different time steps. The network was simulated on the MNIST and Fashion-MNIST datasets showing good robustness. This work enhances the relevance of PCM-based circuits for artificial intelligence applications, and it paves the way for accurate and small-area hardware neural networks.

Acknowledgment

The authors would like to thank STMicroelectronics, Agrate Brianza, Italy, for experimental data and extensive discussions.

References

[1] Y. Lecun, Y. Bengio, and H. Geoffrey, "Deep learning," *Nature*, vol. 512, pp. 436–444, May 2015.
[2] M. A. Zidan, J. P. Strachan, and W. D. Lu, "The future of electronics based on memristive systems," *Nature Electron.*, vol. 1, no. 1, pp. 22–29, Jan. 2018.

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

[4] 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 Neurosci.*, vol. 15, p. 1054, Aug. 2021, doi: 10.3389/fnins.2021.709053.

[5] S. Bianchi et al., "A compact model for stochastic spike-timing-dependent plasticity (STDP) based on resistive switching memory (RRAM) synapses," *IEEE Trans. Electron Devices*, vol. 67, no. 7, pp. 2800–2806, Jul. 2020, doi: 10.1109/TED.2020.2992386.

[6] S. Bianchi, I. Munoz-Martin, S. Hashemkhani, G. Pedretti, and D. Ielmini, "A bio-inspired recurrent neural network with self-adaptive neurons and PCM synapses for solving reinforcement learning tasks," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Oct. 2020, pp. 1–5.

[7] S. Ambrogio et al., "Equivalent-accuracy accelerated neural-network training using analogue memory," *Nature*, vol. 558, no. 7708, pp. 60–67, Jun. 2018, doi: 10.1038/s41586-018-0180-5.

[8] I. Munoz-Martin, S. Bianchi, G. Pedretti, O. Melnic, S. Ambrogio, and D. Ielmini, "Unsupervised learning to overcome catastrophic forgetting in neural networks," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 5, pp. 58–66, 2019, doi: 10.1109/JXCDC.2019.2911135.

[9] I. Muñoz-Martín, S. Bianchi, O. Melnic, A. G. Bonfanti, and D. Ielmini, "A drift-resilient hardware implementation of neural accelerators based on phase change memory devices," *IEEE Trans. Electron Devices*, vol. 68, no. 12, pp. 6076–6081, Dec. 2021.

[10] I. Muñoz-Martín, S. Bianchi, S. Hashemkhani, G. Pedretti, and D. Ielmini, "Hardware implementation of PCM-based neurons with self-regulating threshold for homeostatic scaling in unsupervised learning," in *Proc. ISCAS*, Oct. 2020, pp. 1–5.

[11] M. Hu et al., "Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication," in *Proc. 53rd ACM/EDAC/IEEE Design Automat. Conf. (DAC)*, Jun. 2016, pp. 1–6, doi: 10.1145/2897937.2898010.

[12] D. Ielmini and H.-S.-P. Wong, "In-memory computing with resistive switching devices," *Nature Electron.*, vol. 1, no. 6, pp. 333–343, Jun. 2018.

[13] R. Mochida et al., "A 4M synapses integrated analog ReRAM based 66.5 TOPS/W neural-network processor with cell current controlled writing and flexible network architecture," in *Proc. IEEE Symp. VLSI Technol.*, Jun. 2018, pp. 175–176.

[14] C.-X. Xue et al., "A 1 Mb multibit ReRAM computing-in-memory macro with 14.6 ns 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.

[15] C.-X. Xue et al., "A CMOS-integrated compute-in-memory macro based on resistive random-access memory for ai edge devices," *Nature Electron.*, vol. 4, no. 5, pp. 173–195, 2021.

[16] Q. Liu et al., "A fully integrated analog ReRAM based 78.4 TOPS/W compute-in-memory chip with fully parallel MAC computing," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 500–502.

[17] P. Narayanan et al., "Fully on-chip MAC at 14 nm enabled by accurate row-wise programming of PCM-based weights and parallel vector-transport in duration-format," in *Proc. Symp. VLSI Technol.*, 2021, pp. 1–2.

[18] W.-S. Khwa et al., "A 40-nm, 2M-cell, 8b-precision, hybrid SLC-MLC PCM computing-in-memory macro with 20.5–65.0 TOPS/W for tiny-ai edge devices," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2022, pp. 180–181.

[19] J.-M. Hung et al., "An 8-Mb DC-current-free binary-to-8b precision ReRAM nonvolatile computing-in-memory macro using time-space-readout with 1286.4–21.6 TOPS/W for edge-AI devices," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2022, pp. 182–183.

[20] S. D. Samuel et al., "A 40 nm 64 kb 26.56 TOPS/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*, Feb. 2022, pp. 268–269.

[21] Z. Xuan, Y. Zhang, Y. Li, C. Liu, and Y. Kang, "HPSW-CIM: A novel ReRAM-based computing-in-memory architecture with constant-term circuit for full parallel hybrid-precision-signed-weight MAC operation," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2022, pp. 3274–3278.

[22] Y. He, Y. Huang, J. Yue, W. Sun, L. Zhang, and Y. Liu, "C-RRAM: A fully input parallel charge-domain RRAM-based computing-in-memory design with high tolerance for RRAM variations," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2022, pp. 3279–3283.

[23] D. V. Christensen et al., "2022 roadmap on neuromorphic computing and engineering," *Neuromorphic Comput. Eng.*, vol. 2, no. 2, 2022, Art. no. 022501.

[24] I. Chakraborty et al., "Resistive crossbars as approximate hardware building blocks for machine learning: Opportunities and challenges," *Proc. IEEE*, vol. 108, no. 12, pp. 2276–2310, Dec. 2020.

[25] S. Kariyappa et al., "Noise-resilient DNN: Tolerating noise in PCM-based AI accelerators via noise-aware training," *IEEE Trans. Electron Devices*, vol. 68, no. 9, pp. 4356–4362, Sep. 2021.

[26] C. Liu et al., "An 1-bit by 1-bit high parallelism in-RRAM macro with co-training mechanism for DCNN applications," in *Proc. Int. Symp. VLSI Design, Autom. Test (VLSI-DAT)*, Apr. 2022, pp. 1–4.

[27] Y. Lecun, "A theoretical framework for back-propagation," in *Proc. Connectionist Models Summer School*, D. Touretzky, G. Hinton, and T. Sejnowski, Eds. Pittsburgh, PA, USA: Morgan Kaufmann, 1988, pp. 21–28.

[28] Z. Wang et al., "Resistive switching materials for information processing," *Nature Rev. Mater.*, vol. 5, no. 3, pp. 173–195, Jan. 2020.

[29] A. Zhou et al., "Incremental network quantization: Towards lossless CNNs with low-precision weights," *CoRR*, vol. abs/1702.03044, pp. 1–14, Feb. 2017.

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

[31] M. Baldo et al., "Modeling of virgin state and forming operation in embedded phase change memory (PCM)," in *IEDM Tech. Dig.*, Dec. 2020, pp. 267–270, doi: 10.1109/IEDM13553.2020.9372089.

[32] O. Melnic, M. Borghi, E. Palumbo, P. Zuliani, R. Annunziata, and D. Ielmini, "Monte Carlo model of resistance evolution in embedded PCM with Ge-rich GST," in *Proc. Symp. VLSI Technol.*, Jun. 2019, pp. T64–T65.

[33] N. Lepri et al., "Modeling and compensation of IR drop in crosspoint accelerators of neural networks," *IEEE Trans. Electron Devices*, vol. 69, no. 3, pp. 1575–1581, Mar. 2022.

[34] B. Razavi, "The StrongARM latch [a circuit for all seasons]," *IEEE Solid-State Circuits Mag.*, vol. 7, no. 2, pp. 12–17, Spring 2015.

[35] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or −1," 2016, *arXiv:1602.02830*.

[36] Z. Li et al., "Towards binary-valued gates for robust LSTM training," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, J. Dy and A. Krause, Eds., Jul. 2018, pp. 2995–3004.

[37] H. Wang et al., "Binarized graph neural network," *World Wide Web*, vol. 24, no. 3, pp. 825–848, Apr. 2021.

[38] W.-H. Chen et al., "A 65 nm 1 Mb nonvolatile computing-in-memory ReRAM macro with sub-16 ns multiply-and-accumulate for binary DNN AI edge processors," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018, pp. 494–496.

[39] S. Kim, A. Gholami, Z. Yao, M. W. Mahoney, and K. Keutzer, "I-BERT: Integer-only BERT quantization," in *Proc. 38th Int. Conf. Mach. Learn.*, vol. 139, M. Meila and T. Zhang, Eds., Jul. 2021, pp. 5506–5518.

Open Access funding provided by 'Politecnico di Milano' within the CRUI CARE Agreement