Accurate program/verify schemes of resistive switching memory (RRAM) for in-memory neural network circuits

Valerio Milo, *Member, IEEE*, Artem Glukhov, Eduardo Pérez,
Cristian Zambelli, *Member, IEEE*, Nicola Lepri, Mamathamba K. Mahadevaiah,
Emilio Perez-Bosch Quesada, Piero Olivo, Christian Wenger, and Daniele Ielmini, *Fellow, IEEE*

***Abstract*— Resistive switching memory (RRAM) is a promising technology for embedded memory and their application in computing. In particular, RRAM arrays can provide a convenient primitive for matrix-vector multiplication (MVM) with strong impact on the acceleration of neural networks for artificial intelligence (AI). At the same time, RRAM is affected by intrinsic conductance variations which might cause a degradation of accuracy in AI inference hardware. This work provides a detailed study of the multilevel-cell (MLC) programming of RRAM for neural network applications. We compare three MLC programming schemes and discuss their variations in terms of the different slope in the programming characteristics. We test the accuracy of a 2-layer fully-connected neural network (FC-NN) as a function of the MLC scheme, the number of weight levels, and the weight mapping configuration. We find a trade-off between the FC-NN accuracy, size and current consumption. This work highlights the importance of a holistic approach to AI accelerators encompassing the device properties, the overall circuit performance, and the AI application specifications.**

***Index Terms*— Resistive switching memory (RRAM); multilevel cell (MLC) operation; artificial neural network (ANN); in-memory computing (IMC).**

I. Introduction

RESISTIVE switching memory (RRAM) has recently gained increased interest for its application in novel computing concepts called in-memory computing (IMC) [1], [2]. A major advantage of IMC is the capability to execute matrix-vector multiplication (MVM) in parallel on multiple rows and columns of a memory array, which allows for a strong acceleration of neural networks [3]–[7]. The recent demonstration of embedded RRAM devices at Mbit capacity [8] enables the design and integration of IMC circuits [9]–[11], thus paving the way for energy efficient RRAM-based accelerators of artificial intelligence (AI).

A potential issue for RRAM-based IMC is the limited precision of conductance, which is affected by programming variations [12], [13], random telegraph noise (RTN) [14], drift [15], and other types of random fluctuations [16], [17]. Additionally, reliability concerns at array-level such as conductance relaxation over time [18] and temperature instability [19] also challenge the achievement of a stable and high accuracy in RRAM-based IMC. Multilevel-cell (MLC) program/verify techniques have been proposed to overcome the variability effects and improve the precision of conductance in RRAM [20]–[23]. Still, the optimization of MLC precision and its impact on the overall performance of the IMC accelerators in terms of precision and energy efficiency is not fully understood.

This work compares three different MLC program/verify schemes for a 4kb RRAM array used to accelerate MVM in a 2-layer fully-connected neural network (FC-NN). We

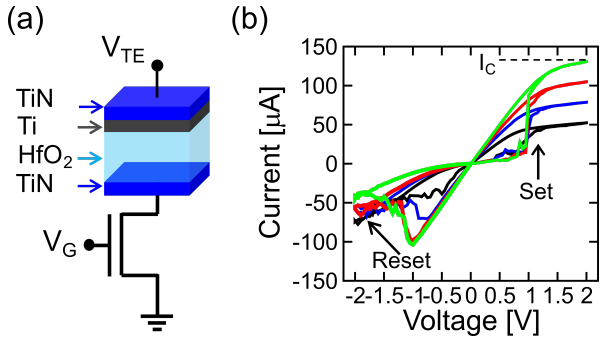

Fig. 1. (a) Schematic of a 1T1R RRAM device of the 4-kbit array used in this work. The RRAM has a stack consisting of a Ti-based oxygen reservoir and an amorphous HfO_2 switching layer sandwiched between a TiN TE and a TiN BE. (b) Multilevel $I-V$ characteristics of 1T1R RRAM device measured for increasing V_G.

This article has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 648635), by the Deutsche Forschungsgemeinschaft (German Research Foundation) with Project-ID 434434223-SFB1461 and by the Federal Ministry of Education and Research of Germany under grant number 16ES1002.

V. Milo was with the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano and IU.NET, 20133 Milan, Italy (e-mail: valerio.milo@polimi.it). Now he is with Applied Materials Italia Srl, 42124 Reggio Emilia, Italy.

A. Glukhov, N. Lepri, and D. Ielmini are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano and IU.NET, 20133 Milan, Italy (e-mail: daniele.ielmini@polimi.it).

C. Zambelli and P. Olivo are with the Dipartimento di Ingegneria, Università degli Studi di Ferrara, 44122 Ferrara, Italy.

E. Pérez, M. K. Mahadevaiah, E. Perez-Bosch Quesada, and Ch. Wenger are with IHP-Leibniz-Institut für innovative Mikroelektronik, 15236 Frankfurt (Oder), Germany.

Ch. Wenger is also with BTU Cottbus-Senftenberg, 01968 Cottbus, Germany.

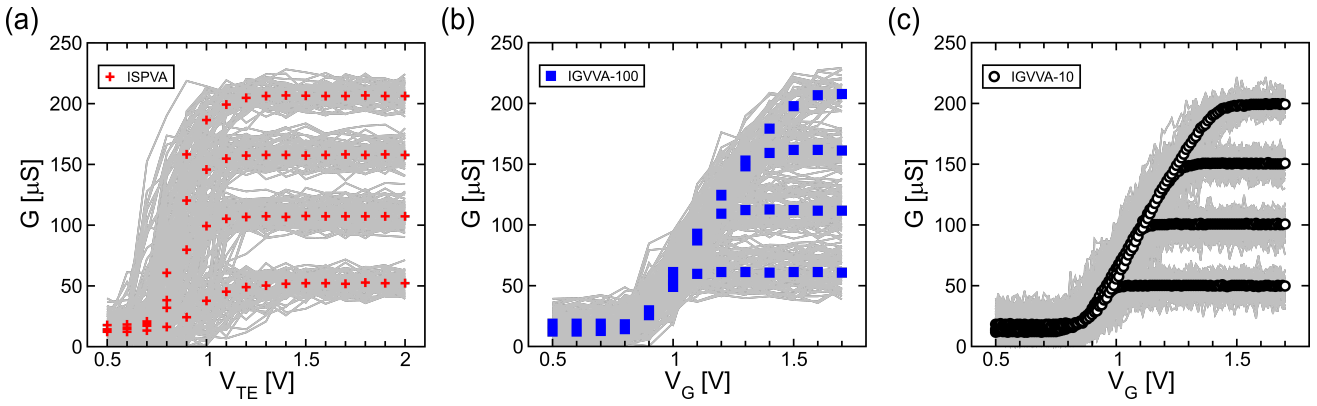

Fig. 2. Measured single-cell (gray line) and median (symbol) conductance of 4 LRS levels programmed in the 4-kbit array by (a) ISPVA, (b) IGVVA-100, and (c) IGVVA-10. ISPVA shows abrupt conductance transitions in correspondence of the set voltage while IGVVA-100 and IGVVA-10 provide a more gradual conductance increase as a result of better current modulation achieved by V_G control than V_{TE} control. Smaller voltage step ΔV_G makes IGVVA-10 programming more accurate than IGVVA-100.

show that gate-based program/verify techniques, where the compliance current is increased at each programming step, display the best accuracy thanks to relatively shallow characteristics of conductance vs. number of pulses. Thanks to this optimized control of MLC conductance, we program the RRAM array with synaptic weights obtained from an offline training and quantization technique for the recognition of handwritten characters. The results are discussed in terms of the tradeoff between inference testing accuracy and current consumption in the array. These results show that a multiscale approach, ranging from weight precision at device level to overall circuit performance, is essential in the design of IMC accelerators of AI.

Preliminary results about this work were reported in [24]. In this work, we extend the number of MLC states up to 9 resistive levels, resulting in 19 synaptic weights. We also include the impact of fluctuations on the FC-NN accuracy, by comparing the conductance immediately after verify to the one at the end of the algorithm. Finally, we include a comprehensive study of accuracy as a function of the number/choice of conductance levels and the number of hidden neurons, also including the impact of IR drop due to parasitic wire resistance in the RRAM array.

II. Multilevel 1T1R RRAM device

The 4-bit array used as test vehicle in this work includes 64x64 RRAM cells based on the one-transistor/one-resistor (1T1R) structure shown in Fig. 1(a). This structure consists of the serial connection of a TiN/Ti/HfO_2/TiN RRAM with a n-channel MOS in 0.25 μm CMOS technology, which is introduced to select the cell and limit the current to the compliance current I_C. Fig. 1(b) shows the measured $I-V$ curves for increasing I_C which was controlled by the gate voltage V_G. These characteristics present abrupt set transitions from the high resistance state (HRS) to the low resistance state (LRS) and gradual reset transitions from the LRS to the HRS for positive/negative voltages, respectively. These curves clearly support the ability of our 1T1R RRAM devices to achieve MLC operation by tuning of V_G.

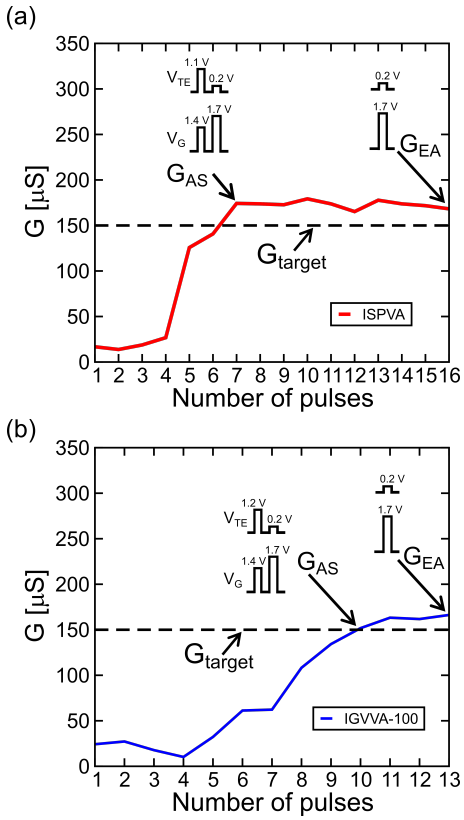

Fig. 3. Measured conductance of a single RRAM device evidencing the after-switching conductance G_{AS} and the end-algorithm conductance G_{EA} for (a) ISPVA and (b) IGVVA-100. The inset shows the pulse amplitude of V_{TE} and V_G for AS and EA conditions.

III. MLC algorithm characterization

To achieve an accurate MLC programming of the 4-kbit RRAM array, we compared two program/verify algorithm approaches based on the modulation of top electrode voltage V_{TE} and gate voltage V_G, respectively. The first algorithm, referred to as incremental step pulse program and verify algorithm (ISPVA), was proposed in [25] and allows multilevel programming via step-by-step application of set pulses (pulse

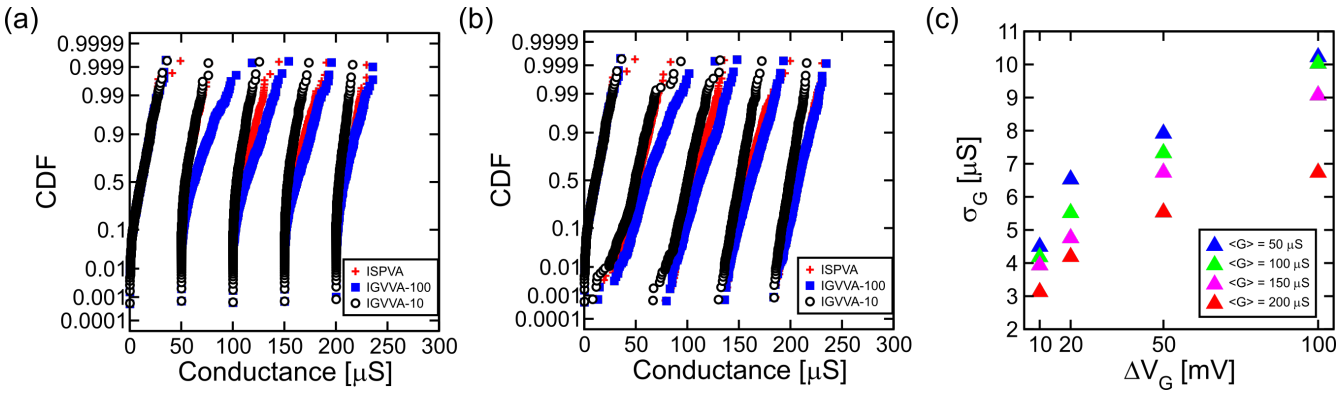

Fig. 4. Conductance CDFs of HRS and 4 LRS levels measured (a) after the switching event (AS) and (b) at the end of the algorithm (EA) by application of ISPVA, IGVVA-100, and IGVVA-10. IGVVA-10 CDFs display the lowest D2D variability for both AS and EA condition, followed by ISPVA and IGVVA-100. (c) Conductance variability of LRS levels for decreasing programming voltage step in IGVVA.

width t_{pulse} = 1 μs) with increasing V_{TE} while V_G is set to achieve the I_C corresponding to the desired target level, and the source of the transistor is grounded. Fig. 2(a) shows the conductance of 4 LRS levels measured by application of ISPVA on a quarter of the 4-kbit 1T1R RRAM devices initially prepared in HRS. V_{TE} was increased from 0.5 V to 2 V with a voltage step ΔV_{TE} = 100 mV by keeping the amplitude of V_G pulses fixed at 1 V, 1.2 V, 1.4 V, and 1.6 V to achieve the target level currents of 10 μA, 20 μA, 30 μA, and 40 μA, respectively. Note that, after any programming operation, a read operation was performed by application of V_G = 1.7 V and V_{TE} = 0.2 V. By considering both single cell conductance G and its median value <G> for each LRS level, it can be noted that abrupt changes take place as soon as V_{TE} becomes larger than $V_{set} \approx 0.9$ V, which evidences that ISPVA is not suitable to finely modulate the device conductance. In particular, the median characteristics show faster transitions for increasing V_G.

To overcome the ISPVA limitation, we designed and investigated a V_G-based programming algorithm called incremental gate voltage and verify algorithm (IGVVA) [24]. Unlike V_{TE}-controlled ISPVA, IGVVA consists of the application of programming pulses (pulse width t_{pulse} = 1 μs) with increasing amplitude V_G from 0.5 V to 1.7 V. On the other hand, the amplitude of V_{TE} programming pulses is kept equal to 1.2 V, which is larger than V_{set} to allow for the set transition. Fig. 2(b) and (c) show the measured G and <G> as a function of V_G by IGVVA based on the voltage steps ΔV_G = 100 mV (IGVVA-100) and ΔV_G = 10 mV (IGVVA-10), respectively, which exhibit a more gradual increase compared to ISPVA. This is due to the higher accuracy in the device current control arising from the tight relation between V_G and I_C [26]. Also, it can be noted that the level programming precision of IGVVA-10 is higher than IGVVA-100 thanks to the smaller ΔV_G allowing a finer control of current flowing in the device during the programming algorithm operation.

To compare the programming accuracy of these MLC algorithms, we programmed 5 levels into the 4-kbit RRAM array by measuring the after-switching (AS) and the end-algorithm (EA) conductance, namely the conductance values measured immediately above the verify threshold and the value

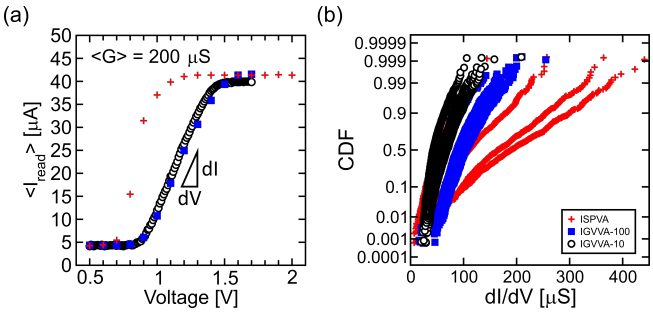

Fig. 5. (a) Comparison among ISPVA, IGVVA-100, and IGVVA-10 median $\boldsymbol{I - V}$ characteristics of the LRS with <G> = 200 $\boldsymbol{\mu}$S in terms of maximum slope $\boldsymbol{g}$ = dI/dV. (b) CDFs of ISPVA, IGVVA-100, and IGVVA-10 dI/dV. According to Fig. 2, ISPVA provides dI/dV CDFs with increasing median and standard deviation variability for increasing level. On the other hand, IGVVA-100 and IGVVA-10 show dI/dV with small median and standard deviation variability. In particular, IGVVA-10 dI/dV CDFs show the lowest median variability, confirming its finer conductance tuning capability.

measured at the end of the whole algorithm, respectively [23]. For example, Fig. 3 shows the AS conductance G_{AS} and the EA conductance G_{EA} for (a) ISPVA and (b) IGVVA-100 program/verify pulses in the case of the conductance level with G_{target} = 150 μS. It should be noted that in ISPVA, no program pulses are applied to the device after G_{AS} is measured, while the read pulse is applied until the number of pulses reaches 16, which would be needed to increase the theoretical V_{TE} to the maximum value of 2 V. Similar to ISPVA, no additional programming pulses are applied to the device between the AS state and the EA state in IGVVA-100. This allows to evidence post-programming fluctuations of the conductance. Note also that in the case of IGVVA-10 (not shown), we adopted the same scheme used for IGVVA-100, while applying a total sequence of 121 pulses as a result of the smaller ΔV_G. This highlights the tradeoff between the higher precision obtained by reducing ΔV_G and the larger number of pulses, hence longer programming time.

Fig. 4(a) shows the (a) AS and (b) EA cumulative distributions (CDFs) of HRS and 4 programmed LRS levels. Compared to AS, the EA distributions show a conductance relaxation for any level, thus resulting in the conductance

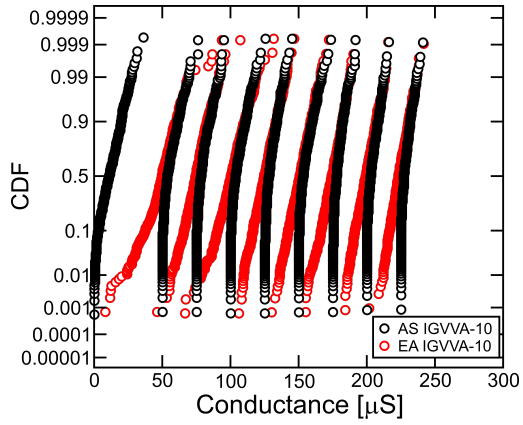

Fig. 6. Measured AS and EA CDFs of 9 IGVVA-10 conductance levels used to implement the synaptic weights into the 4-kbit RRAM array.

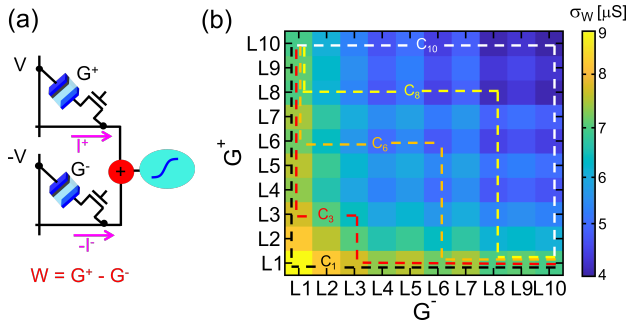

Fig. 7. (a) Schematic representation of a synaptic weight W implemented using the differential configuration of two 1T1R RRAM devices. (b) Color plot of standard deviation σ_W of 19 differential weights calculated via all the IGVVA-10 CDF differences. Based on σ_W, we selected 10 combinations of 19 weights (C1-C10) for mapping the synaptic weights of the neural network, where C10 features the lowest variability.

of some programmed cells decreases below the conductance target [23]. Both figures indicate that IGVVA-10 shows the lowest device-to-device (D2D) variability, followed by ISPVA and IGVVA-100. Also, to better evaluate the impact of programming step on D2D variability in IGVVA, we measured the standard deviation of LRS conductance for decreasing ΔV_G. As shown in Fig. 4(c), reducing ΔV_G from 100 mV to 10 mV allows to significantly decrease the D2D variability of LRS levels as a result of finer control of current during program/verify algorithm operation.

To gain more insight about the control of CDFs by the algorithms, we also studied the maximum slope g = dI/dV of the experimental $I-V$ characteristics, which is explained in Fig. 5(a) for the case of the highest programmed level. Fig. 5(b) shows the CDFs of g for each of the 4 LRS levels programmed by ISPVA, IGVVA-100, and IGVVA-10. From these data, ISPVA features increasing slope with increasing level as opposed to IGVVA-100 and IGVVA-10, where a small increase of g with negligible dependence on ΔV_G can be noted. These results evidence the better control of CDFs with IGVVA-10 compared with IGVVA-100 and ISPVA, thus supporting IGVVA-10 as the most accurate approach for programming the synaptic weights of a neural network in our RRAM array.

IV. Synaptic Weight Mapping by IGVVA-10

To test the accuracy of IGVVA-10 for encoding synaptic weights in a neural network, we programmed 8 LRS levels corresponding to the target conductances from 50 μS to 225 μS by IGVVA-10. Fig. 6 shows the experimental CDFs of the HRS and 8 LRS levels. Both the AS and the EA distributions are reported, evidencing a small D2D variability and relatively small EA relaxation tails affecting all the levels.

The 9 conductance CDFs in Fig. 6 were used to implement the 4-kbit synaptic weights of the 2-layer FC-NN investigated in [24]. This neural network was trained off-line by back-propagation rule for recognizing a simplified 14x14 version of the handwritten digit images of Modified National Institute of Standards and Technology (MNIST) dataset [27]. The network consists of an input layer including 197 neurons, a hidden layer with 20 neurons, and an output layer with 10 neurons. The after-training weight quantization scheme proposed in [28] was implemented to take advantage of the quantized levels in the RRAM array. To maximize the inference accuracy of the neural network in the 4-kbit array, here we implemented the FC-NN synaptic weights using the 9 IGVVA-10 CDFs in Fig. 6 combined with the differential scheme illustrated in Fig. 7(a), namely by encoding the weight as the difference of two 1T1R conductances G^+ and G^- [3]. Note that, to obtain a certain weight W, there are various possible combinations of G^+ and G^- from the distributions of Fig. 6. Fig. 7(b) shows the 100 combinations of G^+ and G^- for mapping 19 weights in the network. For instance, a weight of 100 μS can be obtained as the difference between G^+ = 100 μS and G^- = 0, or as the difference between G^+ = 200 μS and G^- = 100 μS. Note that L2, which corresponds to $< G >$ = 25 μS, was not experimentally measured, but calculated in simulation by differences of CDFs. The figure also shows the standard deviation σ_W of the differential levels obtained by all the possible differences of the 9 IGVVA-10 CDFs. It can be noted that the combination of 19 differential weights with the lowest σ_W, called C10, is found at the top row (G^+ = L10) and the rightmost column (G^- = L10) as a result of the lowest σ_G of L10 shown in Fig. 6. Also, Fig. 7(b) shows other examples of weight mapping combinations (C1, C3, C6, and C8), which, despite the higher σ_W, allow to implement the 19 differential weights by using smaller conductance levels. This poses a significant trade-off for the design of our network: while the weight precision is maximized in correspondence of the highest conductance levels, the relatively large current results in a larger area and energy of the periphery circuits as well as a higher IR drop, causing additional errors. The IR drop impact might be minimized provided that the interconnect resistances are much smaller than the device resistances [29].

V. Impact of Synaptic Weight Mapping on Neural Network Design

To better understand the impact of the various weight mapping combinations, Fig. 8(a) shows the calculated inference accuracy η of the 2-layer FC-NN with 100 hidden

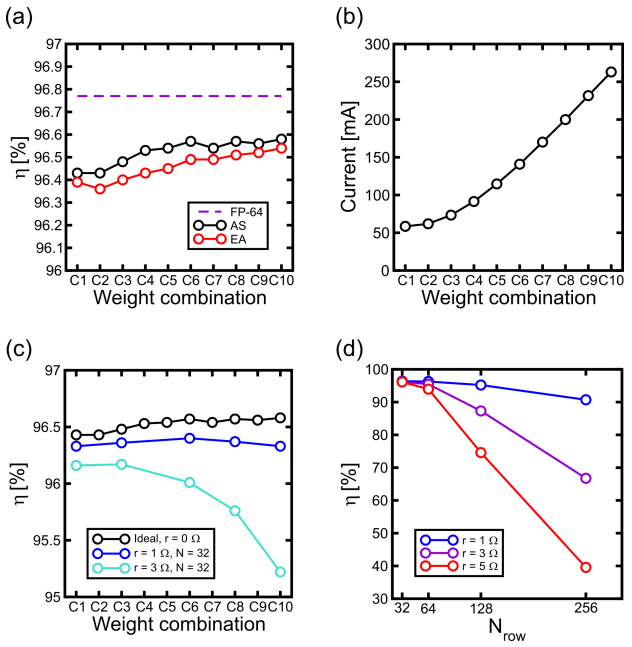

Fig. 8. (a) Calculated inference accuracy using differential weights based on AS and EA IGVVA-10 CDFs and (b) corresponding current consumption as a function of the weight mapping combination. EA relaxation has a small impact on both figures of merit of our network implementation. (c) Impact of the IR drop on the inference accuracy of the calculated FC-NN based on crossbar arrays of 32x32 RRAM devices as a function of the weight combination for increasing parasitic resistance r from 0 to 3 Ω. (d) Impact of the IR drop on the calculated inference accuracy as a function of the size of the crossbar arrays for increasing r in the case of weight combination C1.

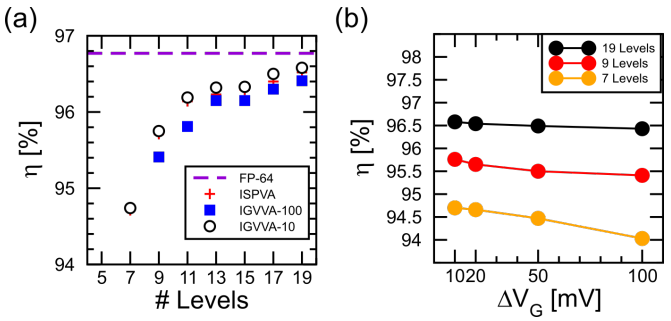

Fig. 9. (a) Calculated inference accuracy of the 2-layer FC-NN with N_H = 100 as a function of the number of synaptic weight levels by IGVVA-100, ISPVA, and IGVVA-10 CDFs. The higher number of levels combined with IGVVA-10 programming leads η close to FP-64 accuracy. (b) Calculated inference accuracy as a function of ΔV_G for increasing number of synaptic weight levels.

neurons (N_H = 100) and 19 differential weight levels based on IGVVA-10 on MNIST test dataset as a function of the weight combination Ci, where the index i ranges from 1 to 10. In agreement with the color plot in Fig. 7(b), the testing accuracy of the network increases with increasing i, supporting C10 as the best combination to increase η (96.58%) closer to the software accuracy calculated using real-valued weights with 64-bit floating point (FP-64) precision (96.77%). Note that the improvement in terms of inference accuracy from C1 to C10 for both AS and EA is by 0.15%. Fig. 8(b) shows the current consumption during the inference phase as a function of Ci. This was calculated as the sum of all the column currents of the network during the testing of the MNIST images. We obtained that the consumed current increases with Ci as a result of the increasing conductances used to implement the differential weights, thus leading to a maximum value at C10 which is about five times the C1-based dissipation. These results clearly illustrate the trade-off between the inference accuracy of the FC-NN and the current consumption. In addition to the current consumption, the impact of the IR drop, namely the voltage drop due to the parasitic wire resistance, was evaluated. Fig. 8(c) shows the inference accuracy of our network as a function of the combinations indicated in Fig. 7(b). The network was broken into 6 individual tiles consisting of 32x32 crossbar arrays of RRAM devices where both terminals of each device are connected to row and column wires with a finite non-zero resistance r between two contacts. The simulation results show that the inference accuracy decreases with increasing r from 0 to 3 Ω evidencing an increasing drop at higher Ci because of the larger currents. These results further highlight the importance of adopting relatively low conductance levels, despite their slightly larger variation. An approach to achieve low conductance levels to minimize the IR drop issues could be the application of an algorithm based on incremental reset. However, partial reset was shown to lead to higher conductance variation compared to gradual set [30], suggesting that combined algorithms based on partial set/reset pulses need further studies. Also, Fig. 8(d) shows the FC-NN inference accuracy as a function of size N_{row} of crossbar arrays for r = 1 Ω, 3 Ω, and 5 Ω for combination C1, evidencing an increasing accuracy drop for increasing r due to the larger impact of the IR drop for increasing array size.

While accuracy is only barely improved by increasing the conductance levels (Fig. 8(a)), it can be more heavily impacted by the conductance precision in terms of the number of levels of the synaptic weight. This is shown in Fig. 9(a) where we report the calculated η of the 2-layer FC-NN with N_H = 100 as a function of the number of discrete weight levels programmed by ISPVA, IGVVA-100, and IGVVA-10. First, the increasing number of weight levels from 9 to 19 leads to increasing inference accuracy values for all the algorithms, supporting the need for memory devices with accurate MLC operation. Also, accordingly with Fig. 4(a), IGVVA-10 provides the highest improvement followed by ISPVA and IGVVA-100 thanks to its lower D2D variability. To better address the impact of the programming voltage step, Fig. 9(b) shows the FC-NN inference accuracy as a function of ΔV_G in the case of 7, 9, and 19 weight levels with ΔW = 25 μS, by evidencing a small variation for increasing ΔV_G in all the cases and the best accuracy achieved by IGVVA-10.

In addition to the number of levels, mapping a wider range of real-valued weights calculated in software is also essential to achieve higher inference accuracies. This is shown in Fig. 10(a), where we report that η can be achieved by mapping the weights using levels with a step ΔW = 50 μS rather than a smaller step of 25 μS in the case of 9 discrete levels.

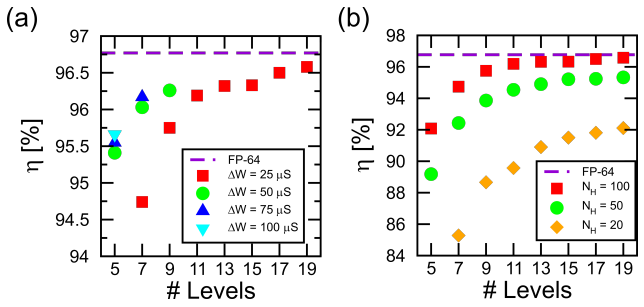

Fig. 10. Calculated inference accuracy of the 2-layer FC-NN as a function of the number of synaptic weight levels for (a) various steps ΔW in weight mapping and (b) increasing size of hidden layer N_H.

Obviously, this choice has the drawback of requiring a larger current consumption, and a larger number of discrete MLC states in the memory device. Fig. 10(b) shows the increase in inference accuracy of the 2-layer FC-NN as a function of the number of weight levels programmed by IGVVA-10 AS CDFs for increasing N_H. As expected, a larger number of weights enables a significant improvement in test accuracy. In particular, if 19 levels are adopted for weight mapping, η increases from 92% with N_H = 20 to 96.2% with N_H = 100. However, increasing the number of hidden neurons, namely the size of FC-NN, also results in a larger area of the memory array, thus evidencing a trade-off between accuracy and area consumption.

VI. Conclusions

We investigated 3 MLC algorithms to optimize the synaptic weight implementation for RRAM-based FC-NNs. IGVVA-10 allows to program 9 conductance levels exhibiting the lowest D2D variability thanks to the highly accurate slope control of $I-V$ characteristics. Combining the differential encoding scheme and IGVVA-10, we mapped the weights of a 2-layer FC-NN demonstrating high inference accuracy for increasing number of levels, weight mapping step, and hidden layer size. This study also allows to evidence key trade-offs between the improvement of inference accuracy and current/area consumption, with a focus on the impact of the IR drop. The results discussed in this work support the need for a co-optimization at device and system level to bring the array-level neural network implementations close to the accuracy achieved by neural networks operated in software.

References

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

[2] D. Ielmini and H.-S. P. Wong, "In-memory computing with resistive switching devices," *Nat. Electron.*, vol. 1, pp. 333–343, 2018, DOI: 10.1038/s41928-018-0092-2.

[3] G. W. Burr *et al.*, "Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element," *IEEE Trans. Electron Devices*, vol. 62, no. 11, pp. 3498–3507, 2015, DOI: 10.1109/TED.2015.2439635.

[4] T. Gokmen and Y. Vlasov, "Acceleration of deep neural network training with resistive cross-point devices: Design considerations," *Front. Neurosci.*, vol. 10, p. 333, 2016, DOI: 10.3389/fnins.2016.00333.

[5] S. Yu, "Neuro-inspired computing with emerging nonvolatile memory," *Proc. IEEE*, vol. 106, no. 2, pp. 260–285, 2018, DOI: 10.1109/JPROC.2018.2790840.

[6] C. Li *et al.*, "Analogue signal and image processing with large memristor crossbars," *Nat. Electron.*, vol. 1, pp. 52–59, 2018, DOI: 10.1038/s41928-017-0002-z.

[7] M. Hu *et al.*, "Memristor-based analog computation and neural network classification with a dot product engine," *Adv. Mater*, vol. 30, no. 1705914, 2018, DOI: 10.1002/adma.201705914.

[8] C.-C. Chou *et al.*, "An N40 256K44 embedded RRAM macro with SL-precharge SA and low-voltage current limiter to improve read and write performance," *IEEE Int. Solid-State Circ. Conf. (ISSCC)*, pp. 478–480, 2018, DOI: 10.1109/ISSCC.2018.8310392.

[9] W.-H. Chen *et al.*, "A 65nm 1Mb nonvolatile computing-in-memory ReRAM macro with sub-16ns multiply-and-accumulate for binary DNN AI edge processors," *IEEE Int. Solid-State Circ. Conf. (ISSCC)*, pp. 494–496, 2018, DOI: 10.1109/ISSCC.2018.8310400.

[10] C.-X. Xue *et al.*, "A 1Mb multibit ReRAM computing-in-memory macro with 14.6ns parallel MAC computing time for CNN based AI edge processors," *IEEE Int. Solid-State Circ. Conf. (ISSCC)*, pp. 388–390, 2019, DOI: 10.1109/ISSCC.2019.8662395.

[11] C.-X. Xue *et al.*, "A CMOS-integrated compute-in-memory macro based on resistive random-access memory for AI edge devices," *Nat. Electron.*, vol. 4, pp. 81–90, 2021, DOI: 10.1038/s41928-020-00505-5.

[12] A. Fantini *et al.*, "Intrinsic switching variability in HfO_2 RRAM," *IEEE Int. Mem. Workshop (IMW)*, pp. 1–4, 2013, DOI: 10.1109/IMW.2013.6582090.

[13] S. Ambrogio *et al.*, "Statistical fluctuations in HfO_x resistive-switching memory: Part I – Set/reset variability," *IEEE Trans. Electron Devices*, vol. 61, no. 8, pp. 2912–2919, 2014, DOI: 10.1109/TED.2014.2330200.

[14] S. Ambrogio *et al.*, "Statistical fluctuations in HfO_x resistive-switching memory: Part II – Random telegraph noise," *IEEE Trans. Electron Devices*, vol. 61, no. 8, pp. 2920–2927, 2014, DOI: 10.1109/TED.2014.2330202.

[15] Y. Lin *et al.*, "Performance impacts of analog ReRAM non-ideality on neuromorphic computing," *IEEE Trans. Electron Devices*, vol. 66, no. 3, pp. 1289–1295, 2019, DOI: 10.1109/TED.2019.2894273.

[16] S. Ambrogio *et al.*, "Noise-induced resistance broadening in resistive switching memory – Part I: Intrinsic cell behavior," *IEEE Trans. Electron Devices*, vol. 62, no. 11, pp. 3805–3811, 2015, DOI: 10.1109/TED.2015.2475598.

[17] S. Ambrogio *et al.*, "Noise-induced resistance broadening in resistive switching memory – Part II: Array statistics," *IEEE Trans. Electron Devices*, vol. 62, no. 11, pp. 3812–3819, 2015, DOI: 10.1109/TED.2015.2477135.

[18] W. He *et al.*, "Characterization and mitigation of relaxation effects on multi-level RRAM based in-memory computing," *IEEE Int. Reliab. Phys. Symp. (IRPS)*, pp. 1–7, 2021, DOI: 10.1109/IRPS46558.2021.9405228.

[19] Y.-F. Chang *et al.*, "Embedded emerging memory technologies for neuromorphic computing: temperature instability and reliability," *IEEE Int. Reliab. Phys. Symp. (IRPS)*, pp. 1–5, 2021, DOI: 10.1109/IRPS46558.2021.9405120.

[20] E. Pérez *et al.*, "Toward reliable multi-level operation in RRAM arrays: Improving post-algorithm stability and assessing endurance/data retention," *J. Electron. Dev. Soc.*, vol. 7, pp. 740–747, 2019, DOI: 10.1109/JEDS.2019.2931769.

[21] V. Milo *et al.*, "Multilevel HfO_2-based RRAM devices for low-power neuromorphic networks," *APL Mater.*, vol. 7, p. 081120, 2019, DOI: 10.1063/1.5108650.

[22] Y. Luo *et al.*, "Array-level programming of 3-bit per cell resistive memory and its application for deep neural network inference," *IEEE Trans. on Electron Devices*, vol. 67, no. 11, pp. 4621–4625, 2020, DOI: 10.1109/TED.2020.3015940.

[23] E. Pérez *et al.*, "Variability and energy consumption tradeoffs in multilevel programming of RRAM arrays," *IEEE Trans. on Electron Devices*, 2021, DOI: 10.1109/TED.2021.3072868.

[24] V. Milo *et al.*, "Optimized programming algorithms for multilevel RRAM in hardware neural networks," *IEEE Int. Reliab. Phys. Symp. (IRPS)*, pp. 1–6, DOI: 10.1109/IRPS46558.2021.9405119.

[25] E. Pérez *et al.*, "Reduction of the cell-to-cell variability in $Hf_{1-x}Al_xO_y$ based RRAM arrays by using program algorithms," *IEEE Electron Device Lett.*, vol. 38, no. 2, pp. 175–178, 2017, DOI: 10.1109/LED.2016.2646758.

[26] D. Ielmini, "Modeling the universal set/reset characteristics of bipolar RRAM by field- and temperature-driven filament growth," *IEEE Trans. on Electron Devices*, vol. 58, no. 12, pp. 4309–4317, 2011, DOI: 10.1109/TED.2011.2167513.

[27] Y. LeCun *et al.*, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, DOI: 10.1109/5.726791.

[28] A. Zhou *et al.*, "Incremental network quantization: towards lossless CNNs with low-precision weights," *arXiv:1702.03044*, 2017.

[29] D. Ielmini and G. Pedretti, "Device and circuit architectures for in-memory computing," *Adv. Intell. Syst.*, vol. 2, no. 7, p. 2000040, 2020, DOI: 10.1002/aisy.202000040.

[30] G. Pedretti, E. Ambrosi, and D. Ielmini, "Conductance variations and their impact on the precision of in-memory computing with resistive switching memory (RRAM)," *IEEE Int. Reliab. Phys. Symp. (IRPS)*, pp. 1–8, 2021, DOI: 10.1109/IRPS46558.2021.9405130.