Solid-State Electronics 229 (2025) 109214

Contents lists available at ScienceDirect

Solid State Electronics

journal homepage: www.elsevier.com/locate/sse

Behavioral SPICE model for memristive crosspoint arrays operating in the nonlinear transport regime☆

X. Pérez [a], R. Picos [b], J. Suñé [a], E. Miranda [a,*]

[a] Dept. Enginyeria Electrónica, Universitat Autònoma de Barcelona, 08193 Cerdanyola del Valles, Spain
[b] Industrial Engineering and Construction Department, Universitat de les Illes Balears, 07122 Palma, Spain

ARTICLE INFO

The review of this paper was arranged by Francisco F. Gamiz
Handling Editor: Francisco F. Gamiz

Keywords:
Memristor
Resistive switching
Crosspoint array
SPICE
LTspice

ABSTRACT

In this letter, a fully behavioral SPICE model for $M \times N$ memristive crosspoint arrays (CPAs) is presented. The proposed approach incorporates the current–voltage characteristics of the memdiode model for resistive switching devices, which can account for both the linear (low-voltage) and nonlinear (high-voltage) transport regimes of memristors. At low voltages, the model coincides with the conventional linear formulation based on matrix–vector multiplication (MVM) method. At high voltages, however, this algebraic operation is no longer valid. The model supports two operation modes depending on the requirements of the surrounding circuitry: voltage-controlled current source (VCCS) and voltage-controlled voltage source (VCVS). Current-controlled modes are also feasible for specific applications. Basic guidelines for applying these different modes are provided.

1. Introduction

The matrix–vector multiplication (MVM) operation lies at the core of most in-memory computing (IMC) applications for machine learning and deep learning [1–4]. The fundamental idea behind IMC is to perform numerical operations using physical processes, with the goal of minimizing the latency and energy overhead associated with conventional von Neumann architectures. MVM can be implemented using non-volatile, programmable memory devices known as memristors, which are placed at the intersections of the crosspoint array (CPA) wires (see Fig. 1). In practical applications, the resistance states of the memristors are first programmed and subsequently used as artificial synaptic weights for inference in a multilayer perceptron configuration [3,5,6]. In many cases, the input voltages are low enough to ensure operation in the linear regime, avoiding any alteration of the programmed memory state. Inference is carried out by feeding the network with vectorized input (an array of voltages) and evaluating the output (an array of currents) according to a specific rule (e.g., maximum value, distribution, etc.).

In this letter, MVM using a CPA is modeled in SPICE (*Simulation Program with Integrated Circuit Emphasis*) following a fully behavioral approach that accounts for both the linear and nonlinear conduction regimes typical of memristive devices. We assume that the memory state remains fixed; however, memory evolution can be incorporated into the model via the corresponding differential equation [7]. Within this framework, *behavioral* means that the model does not rely on conventional electrical components but instead uses a purely mathematical formulation of the devices within a suitable connection scheme. This allows us to abstract away the physical details of electron transport and ion/vacancy motion, as well as to eliminate unnecessary circuit nodes, thereby reducing both complexity and computational cost. At the same time, this behavioral modeling approach enables the evaluation of additional CPA functionalities that would otherwise require the development of extra circuitry and/or external interfaces. Our objective is to demonstrate how the behavioral CPA operates in two different modes: one based on a voltage-controlled current source (VCCS), and the other on a voltage-controlled voltage source (VCVS). Although current-controlled CPAs can also be implemented, they are not discussed in this letter. The choice of operation mode depends on the requirements of the surrounding circuitry and the intended application, as will be discussed below. We specifically address the effects of operating in the nonlinear conduction regime of memristors (nonlinear CPA) and provide examples in which simple circuits are connected to the output terminals of the array. These examples illustrate the advantages of using

☆ This article is part of a special issue entitled: 'EuroSOI-ULIS 2025' published in Solid State Electronics.

* Corresponding author.
E-mail address: enrique.miranda@uab.cat (E. Miranda).

https://doi.org/10.1016/j.sse.2025.109214

Available online 15 August 2025

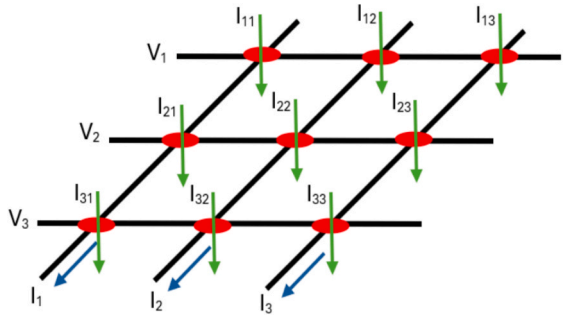


Fig. 1. Example of a 3x3 crosspoint array with memristors (red circles) at the wire intersections. In this case, the inputs are voltages (V_1, V_2, V_3) and the outputs currents (I_1, I_2, I_3). I_{ij} represents the current flowing through the device located at position *i,j*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

a circuit simulator for this kind of analysis, as opposed to relying on general-purpose programming languages or mathematical software packages.

2. Model equations

To begin with, we assume that the neural network (NN) corresponding to a particular problem or task has already been trained using appropriate software, i.e., the synaptic weights are known. No constant bias is assumed, although it can be included in the proposed model as an additional input voltage or current source. For the sake of simplicity, we consider a single-layer NN (perceptron) with positive weights; however, the method can be extended to multiple layers and negative weights, as will be discussed in Section 3. In principle, the software synaptic weights (λ) have no direct correspondence with the hardware synaptic weights (memristor conductances) that we wish to program in our CPA model, so a transformation rule is required. If the memristor is assumed to operate in the linear regime ($I = GV$), then the key parameter is its conductance, G. The transformation is straightforward: we must map the software synaptic weights in the range [λ_{min}, λ_{max}] onto the physical conductance range [G_{off},G_{on}], where G_{off} and G_{on} are the minimum and maximum conductances of the devices, respectively (measured at a low read voltage). This leads to the well-known linear transformation [8]:

$$G_{ij} = \frac{G_{on} - G_{off}}{\lambda_{max} - \lambda_{min}}\left(\lambda_{ij} - \lambda_{min}\right) + G_{off} \qquad 1 \le j \le N,\ 1 \le i \le M \tag{1}$$

Here, i denotes the row and j the column in the $M \times N$ CPA matrix (see Fig. 1). G_{ij} is the conductance of the memristor located at the intersection of row i and column j. Transformation rules other than the linear one are also feasible. Note that, according to Eq. (1), while a software weight $\lambda_{min} = 0$ does not contribute to the output summation of the perceptron, it still corresponds to a finite conductance value G_{off} at the corresponding location. From Fig. 1, the current flowing out of column j in the CPA can be expressed as:

$$I_j = \sum_{i=1}^{M} V_i G_{ij} \quad 1 \le j \le N \tag{2}$$

Equation (2) represents the standard MVM relationship between input voltages and output currents in CPAs operating in the linear conduction regime of memristors [2,3]. This equation implicitly assumes that the outputs are grounded, typically through a small sensing resistor or a virtual ground provided by a transimpedance amplifier (TIA), and that the applied voltages are sufficiently low to avoid altering the memory state of the devices. Next, by considering the memdiode model for memristors [5,7] and accounting for a possible voltage drop V_{ij} between input and output lines, Eq. (2) can be rewritten as:

$$I_j = \sum_{i=1}^{M} I_{ij} = \sum_{i=1}^{M} I_{0ij} sinh\left[\alpha\left(V_{ij} - RI_{ij}\right)\right] \quad 1 \le j \le N \tag{3}$$

where α is a model parameter and R the internal resistance of the memristor (assumed, for simplicity, to be identical for all devices). I_{ij} is the current flowing through the memristor located at intersection (*i,j*). Note that line resistance is not considered in this model. The linear conduction regime of memristors described in Eq. (2) is a particular case of Eq. (3) (see the linear region in Fig. 2).

The current factor I_{0ij} in Eq. (3) is given by the expression:

$$I_{0ij} = \left(I_{on} - I_{off}\right)w_{ij} + I_{off} \tag{4}$$

where I_{on} and I_{off} correspond to the maximum and minimum currents (they are parameters of the memdiode model). $0 \le w_{ij} \le 1$ is the memory state of the device (a variable which reflects its state of conduction). Notice that, as the current I_{ij} becomes higher in Eq. (3) because of a larger w_{ij} value in Eq. (4), V_{ij}-RI_{ij} reduces so that $sinh(x) \approx x$. Considering the two exponentials in *sinh* separately, Eq. (3) can be approximately solved as:

$$I_j \approx \frac{1}{\alpha R}\sum_{i=1}^{M}\left\{W\left[\frac{\alpha R I_{0ij}}{2} exp\left(\alpha V_{ij}\right)\right] - W\left[\frac{\alpha R I_{0ij}}{2} exp\left(-\alpha V_{ij}\right)\right]\right\} \tag{5}$$

where W is the Lambert function, *i.e.* the solution of the transcendental equation $W(x)\exp[W(x)] = x$. In this work, we use for W the Hermite-Padé approximation [9]:

$$W(x) \approx ln(x+1)\left\{1 - \frac{ln[1 + ln(x+1)]}{2 + ln(x+1)}\right\} \tag{6}$$

The use of Eq. (5) eliminates the necessity of considering an internal node in between the nonlinear device (*sinh*) and the series resistance R as expressed in Eq. (3).

Now, let's analyze the connection between the memory state of the memristor w and the software synaptic weight λ. According to the memdiode model, in the low voltage regime of a memristor, the current reads:

$$I = I_0(w) sinh[\alpha(V - RI)] \approx I_0(w)\alpha(V - RI) \tag{7}$$

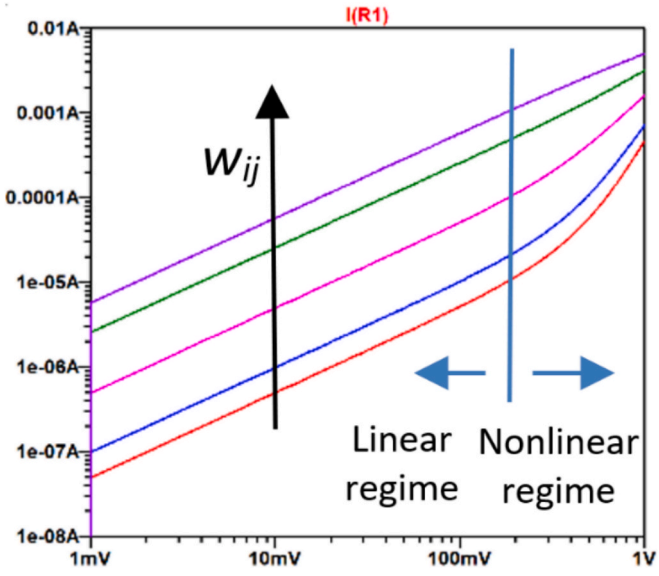


Fig. 2. Typical linear and nonlinear behavior of a single memristor. *I-V* characteristics in log–log axis obtained with SPICE using the memdiode model. As the memory state w_{ij} increases, the *I-V* curve of the memristor changes from linear-exponential to linear as experimentally observed.

so that the linear *I-V* relationship is recovered:

$$I = \frac{I_0(w)\alpha}{1 + I_0(w)\alpha R} V \tag{8}$$

Therefore, from Eq. (8):

$$G_{off} = G(w = 0) = \frac{I_{off}\alpha}{1 + I_{off}\alpha R}, \qquad G_{on} = G(w = 1) = \frac{I_{on}\alpha}{1 + I_{on}\alpha R} \tag{9}$$

and from Eqs. (1) and (4), the memory states express in general as:

$$w_{ij} = \frac{1}{I_{on} - I_{off}} \left[\frac{G_{ij}}{\alpha(1 - G_{ij}R)} - I_{off}\right] \tag{10}$$

For instance, if λ is in the range [0,1], then $w = \lambda$, regardless of the minimum and maximum conductance values of the devices. In this way, the initial normalization of the software weights directly yields the memory state values of the devices. Once the memory states are determined, they can be used to compute the current flowing out of each column according to Eqs. (4) and (5). However, note that Eq. (5) accounts for the nonlinear conduction regime as well, specifically the exponential dependence of current on voltage (see Fig. 2, nonlinear region). In the next Section, we discuss the implementation of Eqs. (3)-(6) in the SPICE simulator.

3. SPICE implementation of the CPAs and simulation results

The equations described above were implemented in LTspice XVII from Analog Devices [10], using a subcircuit structure based on behavioral components. The use of behavioral devices eliminates the need to solve the complete set of circuit equations and ensures direct compatibility with MVM. Two operation modes are considered: VCCS, as shown in Fig. 3, and VCVS, which incorporates behavioral transimpedance amplifiers (TIAs) at the outputs (see Fig. 4). Activation functions can also be implemented as behavioral components at the outputs. The model script used in the example (Fig. 3a) corresponds to a small 3 × 3 CPA, but it can be readily adapted to handle any $M \times N$ matrix. In this case, the weights are arbitrary positive values in the range [0,1]. Negative weights can be accommodated by introducing negative current terms in the output generators. The input voltage is a 3-bit digital signal cycling from 000 to 111 (Fig. 3b). In Fig. 3c, high-value resistors (1 kΩ) are connected to the CPA outputs representing an arbitrary load. This configuration allows part of the applied voltage to drop outside the CPA, meaning the outputs are no longer grounded, as was implicitly assumed in Eq. (2). Naturally, the magnitude of this effect depends on the memristor parameters, the signal levels used, and the load considered. It is important to note that in VCCS mode, any device or circuit can be connected to the outputs. The internal behavioral current sources automatically account for the potential drop across the structure. The resulting output currents as a function of time are shown in Fig. 3d.

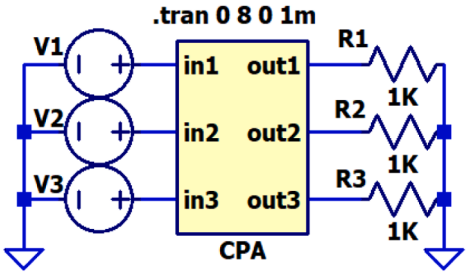
In the case of the VCVS configuration (see Fig. 4), the behavioral current sources for each column of the CPA are replaced by voltage sources. As shown in the model script (Fig. 4a), a multiplying feedback resistor RT is used as a scaling factor, and the internal output nodes are set to ground. This configuration emulates the effect of a behavioral transimpedance amplifier (TIA); that is, internal currents are converted into proportional output voltages. In this setup, the entire applied voltage (1 V) drops across the CPA, causing the device to operate primarily in the nonlinear regime (see Fig. 2). Once again, we emphasize that potentiation can be incorporated into the proposed model by including the memory differential equation (see Refs. [5,7]). Fig. 4b shows the sequence of digital input voltages (000 → 111) as a function of time; Fig. 4c depicts the CPA with open output terminals (internal nodes grounded); and Fig. 4d illustrates the resulting voltages at these open terminals.

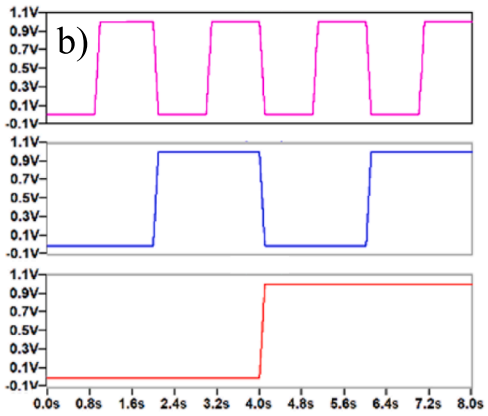
4. Two case studies

Let us now consider two specific cases that illustrate the benefits of

a)

```
.subckt CPA in1 in2 in3 out1 out2 out3
.params
+ w11=0.3 w12=0.1 w13=0.6
+ w21=0.2 w22=0.1 w23=0.3
+ w31=0.6 w32=0.1 w33=0.4
+ Ion=1e-2 Ioff=1e-5 R=200 a=5
B1 0 out1 I=I(V(in1,out1),w11)+I(V(in2,out1),w21)+I(V(in3,out1),w31)
B2 0 out2 I=I(V(in1,out2),w12)+I(V(in2,out2),w22)+I(V(in3,out2),w32)
B3 0 out3 I=I(V(in1,out3),w13)+I(V(in2,out3),w23)+I(V(in3,out3),w33)
.func I0(y) {(Ion-Ioff)*y+Ioff}
.func I(x,y) {(W(a*R*I0(y)/2*exp(a*x))-W(a*R*I0(y)/2*exp(-a*x)))/a/R}
.func W(x) {ln(x+1)*(1-ln(1+ln(1+x))/(2+ln(1+x)))}
.ends
```





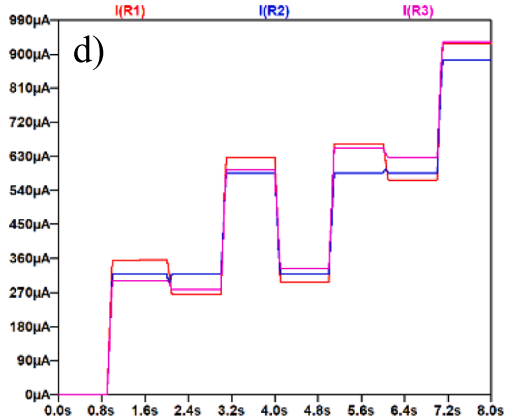


Fig. 3. A) model script for a 3x3 cpa operating in the vccs mode, b) high-voltage input signals (1 V) running from 000 to 111 as a function of time, c) high-load resistances (1KΩ) are attached to the outputs, and d) output currents in each of the output resistances as a function of time.

a)
```
.subckt CPA in1 in2 in3 out1 out2 out3
.params
+ w11=0.3 w12=0.1 w13=0.6
+ w21=0.2 w22=0.1 w23=0.3
+ w31=0.6 w32=0.1 w33=0.4
+ Ion=1e-2 Ioff=1e-5 R=200 a=5 RT=1
B1 0 out1 V=-RT*(I(V(in1,0),w11)+I(V(in2,0),w21)+I(V(in3,0),w31))
B2 0 out2 V=-RT*(I(V(in1,0),w12)+I(V(in2,0),w22)+I(V(in3,0),w32))
B3 0 out3 V=-RT*(I(V(in1,0),w13)+I(V(in2,0),w23)+I(V(in3,0),w33))
.func I0(y) {(Ion-Ioff)*y+Ioff}
.func I(x,y) {(W(a*R*I0(y)/2*exp(a*x))-W(a*R*I0(y)/2*exp(-a*x)))/a/R}
.func W(x) {ln(x+1)*(1-ln(1+ln(1+x))/(2+ln(1+x)))}
.ends
```

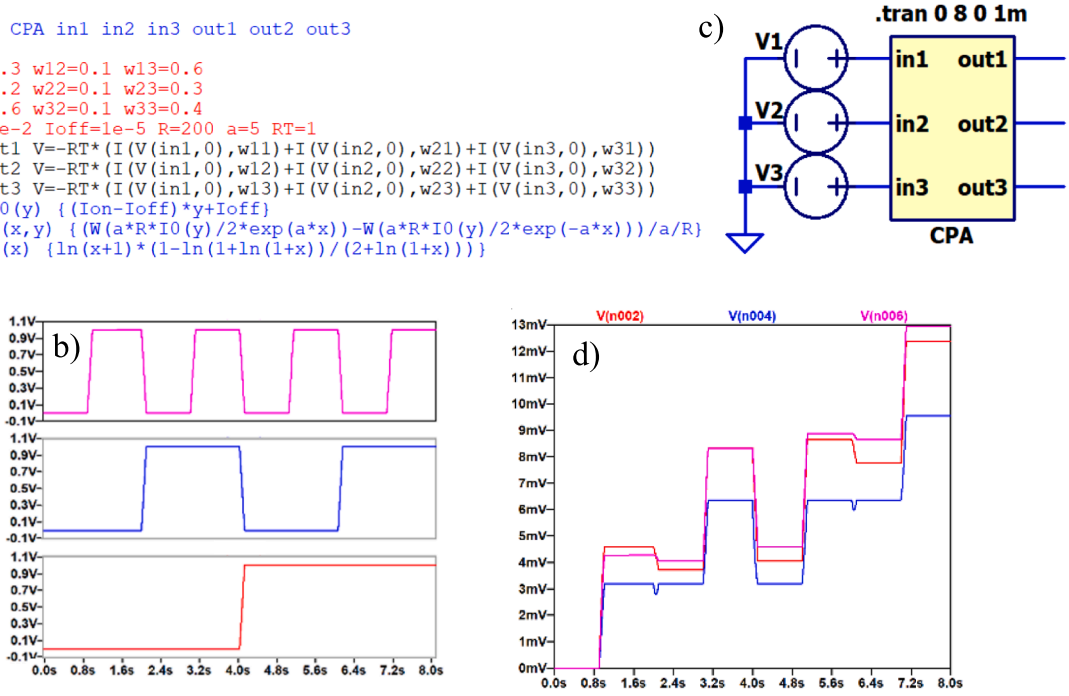


Fig. 4. A) model script for a 3x3 cpa operating in the vcvs mode. notice the internal voltage sources includes a feedback resistance rt corresponding to the behavioral tia, b) high-voltage input signals (1 V) running from 000 to 111 as a function of time, c) CPA with open terminals, and d) output voltages at each terminal as a function of time.

the CPA operation modes discussed above. First, Fig. 5a shows the CPA model operating in VCCS mode, with RC circuits connected to its outputs (example of arbitrary loads). In this configuration, the CPA sources currents determined by the voltage drops across the capacitors. Since the capacitors charge and discharge during transitions, changes in voltage drop are clearly reflected in the currents flowing through the load resistors (see Fig. 5b). This is a case where the CPA alternates between nonlinear and linear operation regimes. During abrupt transitions, the capacitive reactance drops to zero, effectively grounding the CPA outputs. In steady-state conditions, however, the capacitors become charged, and the voltage drop across the CPA no longer matches the applied voltage at the inputs.

Second, Fig. 6a illustrates the case of two identical CPAs connected in series, operating in VCVS mode. Recall that each CPA internally includes TIAs, so their output voltages are proportional to the sum of the currents flowing through the memristors in each column of the CPA. The configuration of CPAs in series is typical in deep learning neural networks [8]. Note also that the number of inputs does not necessarily have to match the number of outputs, this can be generalized to an *MxN* matrix. Each CPA represents a hidden layer of the deep neural network. In the present case, no activation functions were implemented between the structures, but they can be included within the CPA blocks or treated as separate, independent components. Fig. 6b shows the output voltages at the first and second stages of the combined system.

5. Conclusions

A fully behavioral SPICE compact model for *MxN* memristive crosspoint arrays (CPAs) has been presented. The model supports arbitrary output loads and can be readily extended to implement multilayer perceptron architectures. Two primary operation modes were described: voltage-controlled current source (VCCS) and voltage-controlled voltage

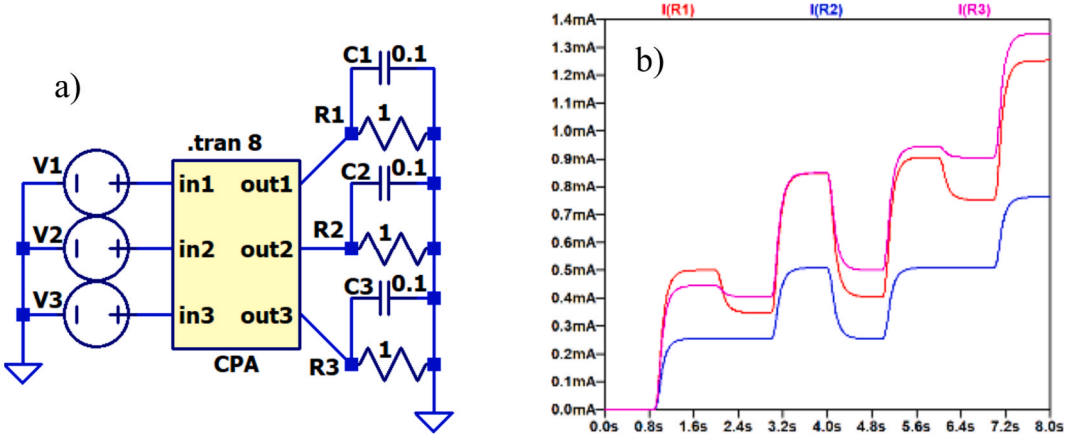


Fig. 5. A) application of the cpa in the vccs mode with rc circuits as load devices, b) output currents flowing through the attached resistances as a function of time. the input voltage is the digital sequence 000 → 111.

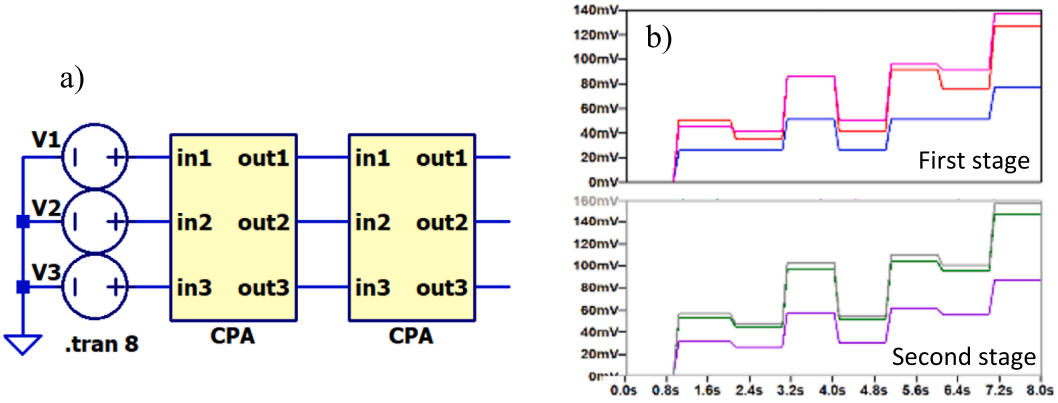


Fig. 6. A) two cpas in series operating in the vcvs mode, b) output voltages at the first and second stage as a function of time. again the input voltage is the digital sequence 000 → 111.

source (VCVS), although current-controlled configurations are also feasible. The choice between these modes depends on the intended application. In VCCS mode, arbitrary external circuits can be directly connected to the CPA outputs, enabling flexible interaction with external analog components. In contrast, the VCVS mode assumes virtually grounded outputs with embedded transimpedance amplifiers (TIAs), making it particularly well-suited for neural network and deep learning applications, where precise voltage representation of current sums is essential.

CRediT authorship contribution statement

X. Pérez: Writing – original draft, Formal analysis, Conceptualization. **R. Picos:** Supervision, Funding acquisition, Formal analysis, Conceptualization. **J. Suñé:** Investigation, Formal analysis, Conceptualization. **E. Miranda:** Writing – review & editing, Writing – original draft, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Enrique Miranda reports financial support was provided by Spanish Ministerio de Ciencia e Innovación (MCIN). If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors would like to thank the support from the Spanish Ministerio de Ciencia e Innovación (MCIN) / Agencia Española de investigación (AEI) 10.13039/501100011 033 (Under project No. PID2022-139586NB-C41).

Data availability

No data was used for the research described in the article.

References

[1] Amirsoleimani A, et al. In-memory vector-matrix multiplication in monolithic complementary metal-oxide-semiconductor-memristor integrated circuits: design choices, challenges, and perspectives. Adv Int Sys 2020;2:2000115.
[2] Lepri N, et al. In-memory computing for machine learning and deep learning. J Electron Dev Soc 2023;11:587.
[3] Pedretti G, et al. In-memory computing with resistive memory circuits: status and outlook. Electronics 2021;10:1063.
[4] Pérez-Ávila A, et al. Multilevel memristor based matrix-vector multiplication: influence of the discretization method, In: 13th Spanish Conference on Electron Devices (CDE), Sevilla, Spain, June 2021.
[5] Aguirre FL, et al. Hardware implementation of memristor-based artificial neural networks. Nat Comm 2024;15:1974.
[6] Yadav S, et al. Ultralow powered 2D MoS2-based memristive crossbar array for synaptic appications. ACS App Mater Interfaces 2025;17:26871.
[7] Aguirre FL, et al. SPICE implementation of the dynamic memdiode model for bipolar resistive switching devices. Micromachines 2022;13:330.
[8] Aguirre FL, et al. Application of the quasi-static memdiode model in cross-point arrays for large dataset pattern recognition. IEEE Access 2020;8:202174–93.
[9] Winitzki S. Uniform approximations for transcendental functions, Computational Science and Its Applications, ICCSA 2003. Lecture Notes in Computer Science, vol 2667. Springer, Berlin, Heidelberg.
[10] LTspice XVII, Analog Devices, https://www.analog.com/en/resources/design-tools-and-calculators/ltspice-simulator.html.