A Variation-Aware CMOS Platform for Multi-Level Memristor Characterisation

Lijie Xie, Peilong Feng, Andrea Mifsud, Amir Nassibi, Christos Papavassiliou
Department of Electrical and Electronic Engineering, Imperial College London, SW7 2BT, UK

***Abstract*—Memristors could be potentially used in both multi-bit data storage and analogue computation, thanks to their programmable resistance. However, there are three common practical challenges to programme memristors to multiple resistive levels. First, the programmed resistance is unstable due to the device's intrinsic randomness and relaxation. Second, the memristor's transient resistive response to external stimuli is stochastic. Third, programming is unachievable if the target level is beyond the memristor's operational region. To overcome these challenges, this work proposes a novel characterisation methodology, which could implement multi-level memristor programming by experimentally identifying the memristor's multi-level capability and then determining the corresponding resistance ranges to accommodate its non-idealities. To automate the proposed methodology, a mixed-signal CMOS system is proposed. It mitigates device variations by providing flexibility for reconfiguring pulsing schemes. Furthermore, it provides up to $4.9\,\mathrm{V}$ voltage pulses and enhanced reading accuracy for a wide-range resistive load ($2k\Omega$ to $5M\Omega$).**

***Index Terms*—memristor, multi-level programming, closed-loop, characterisation**

I. Introduction

Memristors have been widely used in multi-bit resistive random access memory (RRAM) [1]–[3], electronic synapses for neuromorphic networks [4], [5], and reconfigurable analogue circuits [6], [7], attributing to their programmability under electrical stimuli. However, achieving multi-level programming poses challenges due to the intrinsic randomness of memristors [8]. As illustrated in Fig. 1, memristors exhibit uncertain transient resistive responses to programming pulses and manifest randomly distributed resistances when subjected to read pulses. Moreover, the programmed resistance may quickly return to a stable equilibrium due to the short-time relaxation. Consequently, precise memristor programming demands closed-loop pulsing schemes [9]–[11] as introduced in Fig. 2.

Programming an unknown memristor presents a significant challenge, particularly in determining the target resistive level and its corresponding tolerance range. If the target level exceeds the operational resistance range or is set within a narrow margin that cannot accommodate the memristor's instability, the closed-loop programming process fails. The challenge becomes more pronounced when considering device-to-device variations, which exhibit substantial variations in terms of operational resistance ranges, resistive sensitivity to voltages, tolerable voltage amplitudes, and their intrinsic temporal randomness. Therefore, it is imperative to characterise the memristor's multi-level capability before programming.

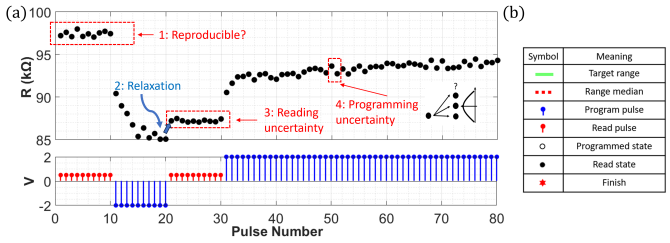


Fig. 1: Non-idealities in memristor programming. (a) 1: The non-reproducible level; 2: Relaxation; 3: Reading uncertainty; 4: Programming uncertainty. (b) Legend. The legend applies to the subsequent plots in this paper.

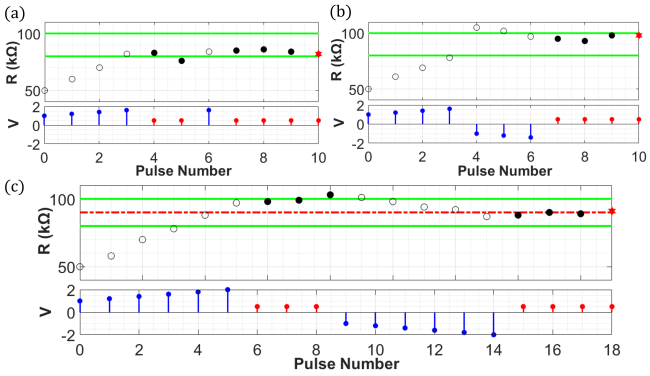


Fig. 2: A closed-loop pulsing scheme for memristor programming. The memristor is programmed using incrementally increased voltage pulses until its resistance falls within the desired region, after which read pulses are applied to assess stability. (a) The resistance gradually enters the target region. (b) The resistance overshoots the target range, requiring pulses of reversed polarity. (c) Optimised scheme used in our work to prevent the states from finishing at the region boundary. The resistance must cross the region median before read pulses are applied.

This work proposes a characterisation methodology to mitigate pragmatic non-idealities encountered in multi-level memristor programming. The characterisation is initially implemented in an experiment conducted on our TiO_2 memristor [12]. The experiment identifies the operational memristor region and determines the multiple resistive levels using a predefined pulsing scheme, paving the way for subsequent multi-level programming. Expanding upon the experiment, a CMOS system is designed to automate both the multi-level characterisation and programming. The system is adaptable to device variations by providing sufficient voltage amplitude, wide-range resistance readability, and reconfigurable pulsing schemes. The rest of this paper is organised as follows: Section II details the experimental operations and considerations for multi-level programming. Section III provides the information on designed circuits. Section IV presents the

979-8-3503-8542-7/24/$31.00 ©2024 IEEE

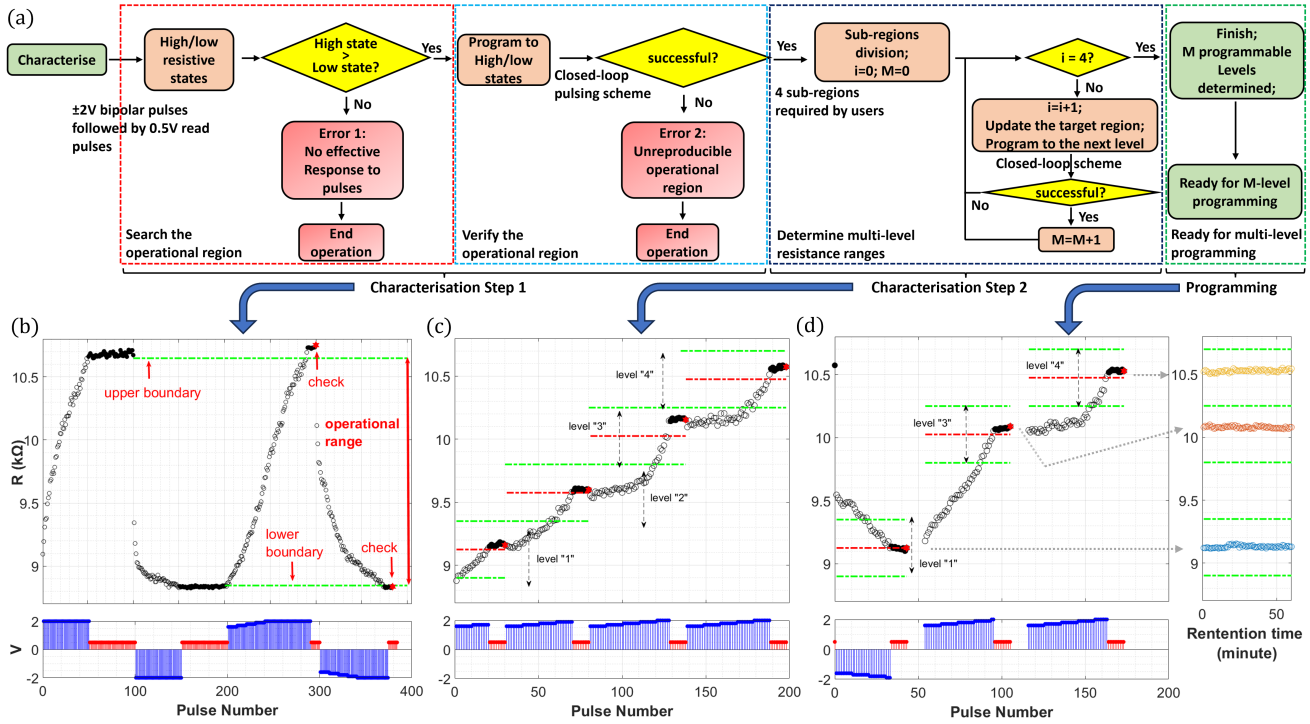


Fig. 3: Experiments of multi-level characterisation and multi-level programming. (a) The algorithm for our experiments. (b) Characterisation step 1:Determining and verifying the operational resistance range. (c) Characterisation step 2: Experimentally demonstrating the attainment of 4 programmable levels. (d) Multi-level programming. The memristor is sequentially programmed to levels "1", "3" and "4". After each finished programming, 1-hour retention measurement is carried out to read the resistance every second.

TABLE I: Predefined closed-loop pulsing scheme

Write	Voltage range	$\pm 1.6\,\mathrm{V}$ to $\pm 2\,\mathrm{V}$
	Voltage steps	4
	Pulse number	10 (50 for $\pm 2\,\mathrm{V}$)
	Pulse width	$100\,\mu\mathrm{s}$
Read	Voltage	$0.5\,\mathrm{V}$
	Pulse width	$1\,\mathrm{ms}$
	Number for stability	10

simulated results. Section V concludes this work.

II. Experiment

This section experimentally demonstrates the feasibility of the multi-level characterisation and the subsequent multi-level programming, using the instrument ArC ONE (ArC Instruments, UK [1]) to manually generate pulses and read the memristor state.

The objective of the multi-level characterisation is to determine multiple programmable resistance ranges under a predefined closed-loop pulsing scheme (Table. I). As illustrated in Fig. 3(a), it involves the sequential assessment of:

1) The responsiveness of the memristor to the predefined maximum voltage.
2) The reprogrammability of the memristor's operational resistance range using the predefined pulsing scheme.
3) The number of programmable levels and the corresponding resistance ranges.

[1]https://www.arc-instruments.co.uk/products/arc-one/

In this experiment, the programming is considered a failure if either the state overshoots the target region twice (the target region cannot accommodate variations) or the closed-loop pulsing scheme is completed (the pulsing scheme is not effective in programming the memristor).

The initial step in multi-level characterisation involves determining the operational resistance range by applying bipolar pulses with a maximum amplitude $2\,\mathrm{V}$. As seen in Fig. 3(b), achieving high and low resistive states is accomplished through 50 program pulses, followed by 50 read pulses to establish the upper and lower boundaries. Ensuring the reproducibility of states within the operational region is crucial. This requires programming the memristor to a stable level outside the boundaries.

The second step determines the ranges of multiple levels to accommodate the memristor randomness and relaxation. The experimental process of 4-level programming is illustrated in Fig. 3(c), where the entire operational region is evenly divided into four sub-regions. Feasibility is assessed by programming the memristor to each sub-region using the predefined pulsing scheme. If the state approaches the target sub-range and remains steady, the sub-range is confirmed as a programmable level. If the programming fails, the sub-range is deemed non-programmable and is merged with the higher sub-range in the next iteration. In the given example, 4-level capability is confirmed.

Once the memristor is characterised, it is ready for multi-level programming. In our experiment, the memristor is se-

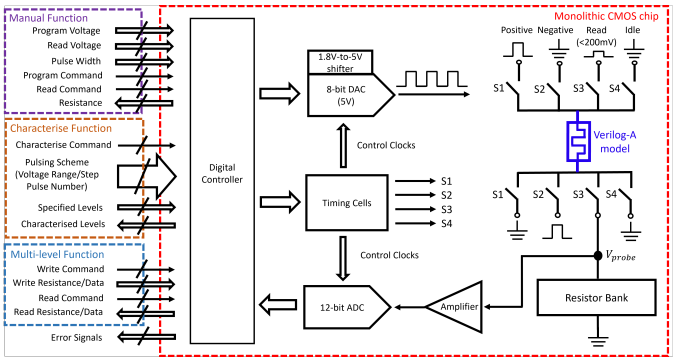


Fig. 4: System overview of the mixed-signal CMOS circuit.

quentially programmed to levels "1", "3" and "4", as illustrated in Fig. 3(d). The initiation of pulse polarity depends on the current state and the target level. With pulses featuring increased amplitudes, the state is progressively programmed to the target level and stabilises near the level median. To provide additional insight into the non-volatility of the measured memristor, a 1-hour retention test is conducted on the programmed memristor, and the results are shown in Fig. 3(d).

III. Circuit Design

Memristors have exhibited greater compatibility with CMOS circuits [13], [14], functioning as dot-product processors [15]–[17] or reconfigurable analogue units [18]. These applications demand on-chip memristor programming facilitated with high-precision and low-power peripheral circuits. Building upon the experimental methodology, a mixed-signal CMOS programming system is designed that enables automatic multi-level characterisation and programming.

The system overview is depicted in Fig. 4. A digital controller provides a fully digital interface and executes user-triggered functions. A digital-to-analogue converter (DAC) generates voltage pulses based on the predefined closed-loop pulsing scheme for either programming or reading the memristor. The read circuits, comprising a resistor bank, an amplifier, and an analogue-to-digital converter (ADC), transform the current through the memristor into an appropriate voltage amplitude for digitisation. The system allows reconfiguration of parameters in the pulsing scheme, such as pulse amplitudes and pulse numbers. Significantly, adequate pulsing is achieved by directly applying voltage potentials to both memristor terminals using switched connections, and a wide-range readout under low read voltage ($< 200\,\mathrm{mV}$) is achieved by adjusting the resistor bank. Although the system is demonstrated with only one single memristor for presentation simplicity, it is also adaptable for a memristor crossbar with minor adjustments.

The system offers the following functions:

1) Manual Pulsing and Reading: Users can manually pulse or read the measured memristor.
2) Multi-level Characterisation: Users can specify the desired number of resistive levels for achievement. The characterisation process can output the number of programmable levels and automatically determine their corresponding resistance ranges.

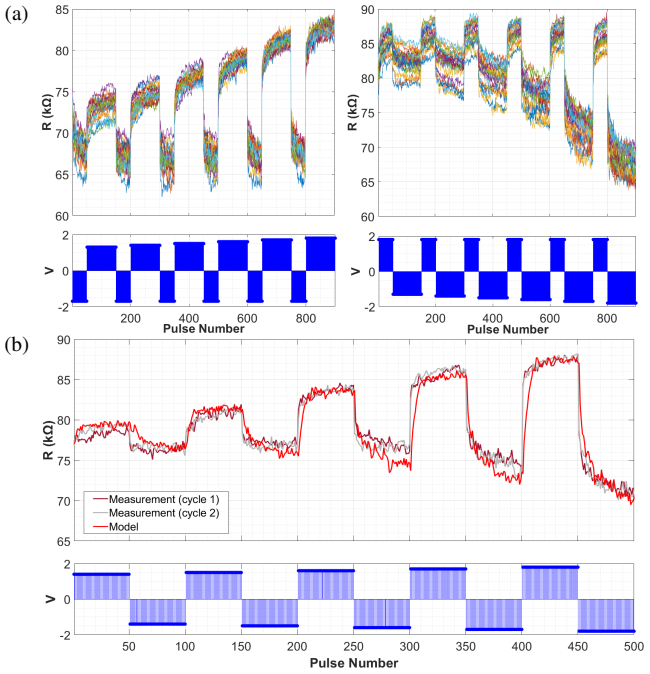


Fig. 5: The stochastic memristor model used in the simulation. (a) The training data consists of 50 iterations, with each colour representing an iteration. The memristor is programmed to a low/high state before positive/negative pulses with different amplitudes ($\pm 1.3\,\mathrm{V}$ to $\pm 1.9\,\mathrm{V}$ with $0.1\,\mathrm{V}$ increment, $100\,\mu\mathrm{s}$ pulse width) are applied. (b) The simulated model result is tested by two cycles of experimental data randomly selected, under bipolar pulses ($\pm 1.4\,\mathrm{V}$ to $\pm 1.8\,\mathrm{V}$ with $0.1\,\mathrm{V}$ increment, $100\,\mu\mathrm{s}$ pulse width).

3) Automatic Programming and Reading: Upon receiving the command, the circuit can automatically program the memristor to the required resistive level or read back its resistance.
4) Error Signals: The executed function is promptly halted by an error signal identifying the encountered problem.

The important circuits are introduced as follows:

1) Digital controller: Primarily composed of finite-state machines, the digital controller implements user-triggered functions until completion or an error occurs. When the state is idle, no voltage is applied to the memristor.

2) Timing cells: Comprising simple logical gates and delay circuits, the timing cells ensure sequential switch pair operations—turning off one pair before another turns on. Therefore, it awaits DAC output stabilised before connecting it to the memristor to avoid unexpected voltage on the memristor. For memristor digitisation, the sampling clock for ADC is delayed after the turn-on of "S3", leaving enough time for ADC sampling.

3) DAC: The 8-bit R-2R DAC generates voltage from $50\,\mathrm{mV}$ to $4.9\,\mathrm{V}$ for a wide-range resistive load ($2k\Omega$ to $5M\Omega$).

4) CMOS switches: Four pairs of CMOS switches control the memristor connections.

5) Resistor bank [19]: Offering five NMOS resistors ranging from around $0.4k\Omega$ to $4M\Omega$ to provide improved read linearity. The resistor bank facilitates current-to-voltage conversion during the read stage, ensuring the selected NMOS

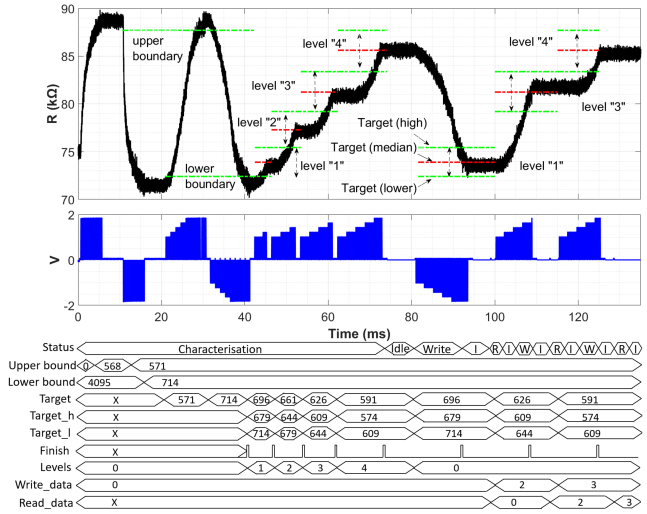


Fig. 6: Simulated results for 4-level programming. The user commands are sequentially implemented: a characterise command is followed by 3 pairs of write/read commands. The resistance response (upper) to the voltage difference across the memristor (middle), and the registered data (lower). The lower figure provides details about the finite-state machine's status, the operational region defined by upper/lower boundaries, the target resistance range (including boundaries and median), the signal of a finished programming, the characterised programmable levels, and the write/read data.

resistor allows the converted voltage to be amplified to the ADC input range.

6) ADC: The 12-bit SAR ADC based on charge redistribution operates with a sampling frequency of 250kSPS and an input range of 0.1V to 1.7V.

7) Memristor: In practical application, the memristor can either be internally integrated above the CMOS [14] or externally connected. For simulation purposes, a data-driven stochastic Verilog-A memristor model is used in this work to provide temporal stochasticity. The model is trained by the measured data in Fig. 5(a) and its performance is compared with two randomly selected experimental data sets under the same stimuli in Fig. 5(b). As a result, both the overall trend and the temporal variation of the model's resistance trajectory are highly consistent with those in the experimental data.

IV. Results

A post-layout simulation of 4-level programming is shown in Fig. 6. The initially idle system is triggered by user commands to execute characterisation, writing, and reading functions. Each function runs to completion and returns the system to the idle state. During the characterisation, the operational range (714-571) and four programmable levels (714-679, 679-644, 644-609, and 609-574) are determined. It should be noticed here that the digits represent memristor conductance rather than resistance. During the following three consecutive write/read functions, the state transitions to the target levels "0", "2", and "3" and stabilises near the level median.

Fig. 7 reveals the maximum multi-level capability when the system is given a task of 17-level programming. During the characterisation, only 8 levels are achieved due to the memristor's variations. Expanding the operational region with

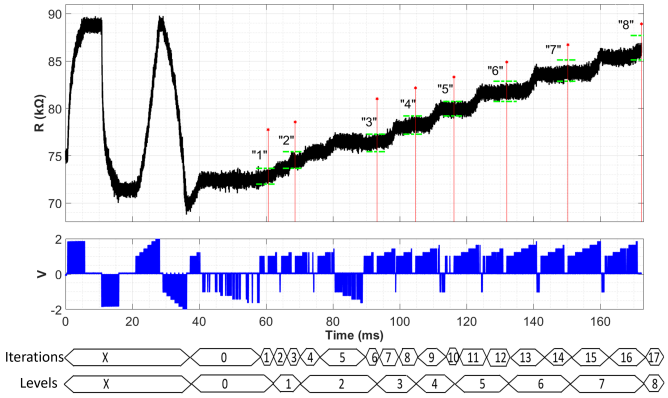


Fig. 7: Simulated results for investigating multi-level capability reveal the attainment of 8 levels from 17 divided sub-regions.

TABLE II: Comparison to other work

Features	This work	[10]	[1]	[11]	[20]	[21]
Range determination	Empirical	-	Predefined			
Closed-loop method	Yes	No	No	Yes	No	No
Temporal variation	Yes	No	No	No	No	No
Spatial variation	Yes	No	No	Yes	Yes	No
Error identification	Yes	No	No	No	No	No

higher voltage may enhance the multi-level capability but poses a greater risk of damaging the device.

Compared to other programming systems as outlined in Table. II, this system has advantages in effectively addressing temporal memristor randomness and device-to-device variations, facilitated by experimentally determining the memristor's resistive ranges. The fully digital user interface and the applied closed-loop pulsing schemes make our system a promising and practical approach for memristor programming.

V. Conclusion

This paper starts with a multi-level characterisation methodology that experimentally sweeps the memristor's operational resistance ranges and determines the multiple resistive levels to address practical programming issues, such as programming uncertainty, memristor intrinsic randomness and short-time relaxation, rendering it ready for multi-level programming. To automate the experimental process and enhance its adaptability to different memristor devices, a CMOS programming system focusing on multi-level programming is introduced. The system offers a user-friendly interface for entering commands and accessing the operation results. To validate the effectiveness of the variation-sensitive algorithm employed in the system, a stochastic compact model is used in our simulation. The simulated results vividly showcase the execution of various system functions, particularly exploring the memristor's multi-level capability.

References

[1] P. Rabbani, R. Dehghani, and N. Shahpari, "A multilevel memristor-cmos memory cell as a ReRAM," *Microelectronics Journal*, vol. 46, no. 12, pp. 1283–1290, 2015.

[2] L. Wu, H. X. Liu, J. B. Li, S. L. Wang, and X. Wang, "A multi-level memristor based on Al-doped HfO2 thin film," *Nanoscale Research Letters*, vol. 14, 2019.

[3] G. Cao, C. Gao, J. J. Wang, J. L. Lan, and X. B. Yan, "Memristor based on two-dimensional titanic nanosheets for multi-level storage and information processing," *Nano Research*, vol. 15, no. 9, pp. 8419–8427, 2022.

[4] C. H. Wang, W. He, Y. Tong, and R. Zhao, "Investigation and manipulation of different analog behaviors of memristor as electronic synapse for neuromorphic applications," *Scientific Reports*, vol. 6, 2016.

[5] M. Naqi, M. S. Kang, N. Liu, T. Kim, S. Baek, A. Bala, C. Moon, J. Park, and S. Kim, "Multilevel artificial electronic synaptic device of direct grown robust MoS2 based memristor array for in-memory deep neural network," *Npj 2d Materials and Applications*, vol. 6, no. 1, 2022.

[6] T. Ibrayev, I. Fedorova, A. K. Maan, and A. P. James, "Memristive operational amplifiers," in *5th Annual International Conference on Biologically Inspired Cognitive Architectures (BICA)*, ser. Procedia Computer Science, vol. 41, 2014, Conference Proceedings, pp. 114–119.

[7] I. Markovic, M. P. Ivanis, and D. Tosic, "The dynamic tunability of memristor-based active filters," *Micromachines*, vol. 14, no. 11, 2023.

[8] S. Gaba, P. Sheridan, J. T. Zhou, S. Choi, and W. Lu, "Stochastic memristive devices for computing and neuromorphic applications," *Nanoscale*, vol. 5, no. 13, pp. 5872–5878, 2013.

[9] F. Alibart, L. G. Gao, B. D. Hoskins, and D. B. Strukov, "High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm," *Nanotechnology*, vol. 23, no. 7, 2012.

[10] Y. Yilmaz and P. Mazumder, "A drift-tolerant read/write scheme for multilevel memristor memory," *IEEE Transactions on Nanotechnology*, vol. 16, no. 6, pp. 1016–1027, 2017.

[11] M. Ramadan, N. Wainstein, R. Ginosar, and S. Kvatinsky, "Adaptive programming in multi-level cell ReRAM," *Microelectronics Journal*, vol. 90, pp. 169–180, 2019.

[12] L. Michalas, S. Stathopoulos, A. Khiat, and T. Prodromakis, "Conduction mechanisms at distinct resistive levels of Pt/TiO2-x/Pt memristors," *Applied Physics Letters*, vol. 113, no. 14, Oct. 2018.

[13] A. El Mesoudy, G. Lamri, R. Dawant, J. Arias-Zapata, P. Gliech, Y. Beilliard, S. Ecoffey, A. Ruediger, F. Alibart, and D. Drouin, "Fully cmos-compatible passive TiO2-based memristor crossbars for in-memory computing," *Microelectronic Engineering*, vol. 255, 2022.

[14] A. Mifsud, J. Shen, P. Feng, L. Xie, C. Wang, Y. Pan, S. Maheshwari, S. Agwa, S. Stathopoulos, S. Wang, A. Serb, C. Papavassiliou, T. Prodromakis, and T. G. Constandinou, "A CMOS-based characterisation platform for emerging RRAM technologies," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2022, pp. 75–79.

[15] M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. J. Yang, and R. S. Williams, "Dot-product engine for neuromorphic computing: Programming 1t1m crossbar to accelerate matrix-vector multiplication," in *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2016, pp. 1–6.

[16] P. Yao, H. Wu, B. Gao, S. B. Eryilmaz, X. Huang, W. Zhang, Q. Zhang, N. Deng, L. Shi, H.-S. P. Wong *et al.*, "Face classification using electronic synapses," *Nature communications*, vol. 8, p. 15199, 2017.

[17] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang, J. P. Strachan, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, and Q. Xia, "In-memory computing with memristor arrays," in *2018 IEEE International Memory Workshop (IMW)*, 2018, pp. 1–4.

[18] J. Shen, S. Stathopoulos, T. Prodromakis, and C. Papavassiliou, "A reconfigurable cmos-memristor active inductor," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1–5.

[19] L. Xie, J. Shen, A. Mifsud, C. Wang, A. Alshaya, and C. Papavassiliou, "A wide dynamic range read-out system for resistive switching technology," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2022, pp. 2003–2007.

[20] N. Taherinejad, M. P. Sai, M. Rathmair, and A. Jantsch, "Fully digital write-in scheme for multi-bit memristive storage," in *2016 13th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, 2016, pp. 1–6.

[21] M. M. Adnan, S. Amer, S. Sayyaparaju, M. S. Hasan, and G. S. Rose, "A scan register based access scheme for multilevel non-volatile memristor memory," in *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2019, pp. 630–633.