

Variability-Tolerant Memristor-based Ratioed Logic in Crossbar Array

M. Escudero, A. Rubio, and F. Moll

Electronics Engineering Department
Universitat Politècnica de Catalunya
Barcelona, Spain

manuel.escudero@upc.edu, antonio.rubio@upc.edu,
francesc.moll@upc.edu

I. Vourkas

Department of Electronics Engineering
Universidad Técnica Federico Santa María
Valparaiso, Chile
ioyourkas@uc.cl

Abstract— The advent of the first TiO₂-based memristor in 2008 revived the scientific interest both from academia and industry for this device technology, with several emerging applications including that of logic circuits. Several memristive logic families have been proposed, each with different attributes, in the current quest for energy-efficient computing systems of the future. However, limited endurance of memristor devices and variations (both cycle-to-cycle and device-to-device) are important parameters to be considered in the evaluation of such logic families. In this work we build upon a well-known accurate physics-based model of a bipolar metal-oxide resistive RAM device (supporting parasitics of the device structure and variability of switching voltages and resistance states) and use it to show how performance of memristor-based logic circuits can be degraded owing to both variability and state-drift impact. Based on previous work on CMOS-like memristive logic circuits, we propose a memristive ratioed logic scheme, which is crossbar-compatible, i.e. suitable for in-/near-memory computing, and tolerant to device variability, while also it does not affect the device endurance since computations do not involve switching the memristor states. As a figure of merit, we compare such new logic scheme with MAGIC, focusing on the universal NOR logic gate.

Keywords—logic gates, memristors, logic design, variability-aware design.

I. INTRODUCTION

Leon Chua in 1971 proposed the existence of the memristor as the fourth fundamental circuit element [1]. An unprecedented attention on this new and emerging device technology was drawn much later though, after 2008, when the first demonstration of the TiO₂-based memristor by Hewlett-Packard Laboratories (HP Labs) took place [2], connecting the nature of such devices with Chua's previous theory. Owing to their analog nature, nonvolatility, high integration density potential and CMOS compatibility, memristors constitute an emerging trend in modern electronics [3], representing a very promising technology which has extended its influence beyond memory [4] to logic and in-memory computing [5], [6].

The development of several behavioral models that capture the basic characteristics of the memristors [7] (e.g. threshold-based switching and nonlinearities near the resistive boundaries,) following that of HP Labs published ten years ago

[2], contributed to the progress of this emerging research field, being adequate to demonstrate the impact and usefulness of memristors in a variety of applications [8], [9]. Among them, the memristive logic design [5], i.e. the methodology of designing logic circuits using memristors, is an emerging concept in the constant quest for energy efficient post-CMOS computing systems. Many such memristive logic families have been proposed: IMPLY, CNIMP, MAGIC, MRL [5], to name a few, use resistance to represent data, thus are suitable for crossbar-based resistive computing. The latter is considered a requirement for real in-/near-memory computing, since the topology of the logic circuits to implement should fit in the crossbar memory array.

Moreover, proposed metrics for comparison of such families so far naturally focus on latency, energy, and area efficiency [10]. However, most such relevant works omit crucial factors such as variability (both cycle-to-cycle and device-to-device, notable even by the uninitiated in all experimental works) and endurance of memristors, a major limitation to be considered if frequent switching is necessary during computations. In this context, Scouting Logic [11] was recently proposed to alleviate the endurance requirement while executing logic operations by just reading the memristor state, even though this scheme eventually suffers from device variability. It is worth mentioning though that the idea of performing logic computations without switching the states of the involved memristors was proposed much earlier by Vourkas *et al.* with the so called CMOS-like memristive logic [12], revisited recently in [13]. Nevertheless, the CMOS-like scheme was not given much attention owing to the complexity of the pull-up/down memristive networks which inhibits its implementation inside a crossbar memory array.

In this context, we present here a crossbar-compatible variation of the CMOS-like concept, namely a ratioed logic scheme, which is both variation-tolerant and does not impact the device endurance. We performed simulations of the proposed scheme using the Cadence Virtuoso and a well-known accurate physics-based model of a bipolar metal-oxide resistive RAM device [14], [15], which supports, among several other features, both parasitics of the MIM device structure and variability of switching voltages and resistance states. We highlight the importance of taking into account

device variability in circuit simulations, guiding the reader through the key model parameters, to eventually show how performance of some known memristor-based logic circuits can be degraded owing to both variability and state-drift impact. Finally, we compare the new ratioed logic scheme with MAGIC, focusing on the universal NOR gate. Simulation results show how the proposed scheme outperforms in terms of robustness and viability, getting us to the simple conclusion that “rethinking of memristive logic design from a practical point of view” is necessary if we aspire to enable *in-memory* computing.

II. INSERTING VARIABILITY IN DEVICE MODELS FOR REALISTIC CIRCUIT SIMULATIONS

Let us shortly provide the basics of the target model used in this work, which is the Stanford-PKU ReRAM device model [14]. It is a compact physics-based model which captures typical DC and AC electrical behavior of metal-oxide based ReRAM devices. The model assumes a conductive filament (CF) growth process described by a change of the CF geometry during the SET and RESET processes under various bias conditions. The core of the model is a two-dimensional description of a unique CF, which includes both the CF gap region and the CF width as control variables [4]. Most importantly, the model includes parasitic effects such as the parasitic resistance of the switching layer and the electrodes, as well as the parasitic MIM capacitance. Furthermore, the model supports intrinsic variation effects, such as statistical distribution of switching thresholds and resistance states, temperature dependency and dynamic current fluctuations for the RESET process, thus supporting literally all the ReRAM device variation effects known to date. As far as operation is concerned, a positive applied voltage produces a SET process, where the oxide layer suffers a soft-breakdown; the CF is formed and the device is found at a low resistive state (LRS or R_{ON}). On the other hand, a negative applied voltage causes a RESET process in which the CF is dissolved through ion diffusion or drift processes and the device is found at a high resistive state (HRS or R_{OFF}).

The model was used in the Cadence Virtuoso suite. The majority of the parameter values were kept at default values [14], except those directly affecting the switching thresholds, i.e. the average active energy of oxygen vacancies (E_a), the hopping barrier of O^{2-} (E_h) and the energy barrier between the electrode and the oxide (E_i). Tuning parameters is recommended to adjust the overall behavior, according to the application requirements. For instance, assuming that for a particular application the following relation is necessary: $V_{SET} > 2 \times V_{RESET}$, then the aforementioned parameters could be tuned as follows: $E_a = 0.9$ eV, $E_h = 0.9$ eV and $E_i = 0.7$ eV.

Fig. 1 demonstrates i-v curves for 20 cycles taken for a device under a triangular applied voltage. The compliance current (cc) is defined by tuning the gate voltage of a series $0.35 \mu\text{m}$ NMOS transistor. We define as SET threshold V_{SET} the voltage when the current reaches to 90% of the cc. Likewise, we define as RESET threshold V_{RESET} the voltage when the current first experiences a sudden decrease. These statistics give us the following mean values: $V_{SET} = 2$ V and $V_{RESET} = -0.5$ V, which constitute a good approximation to later

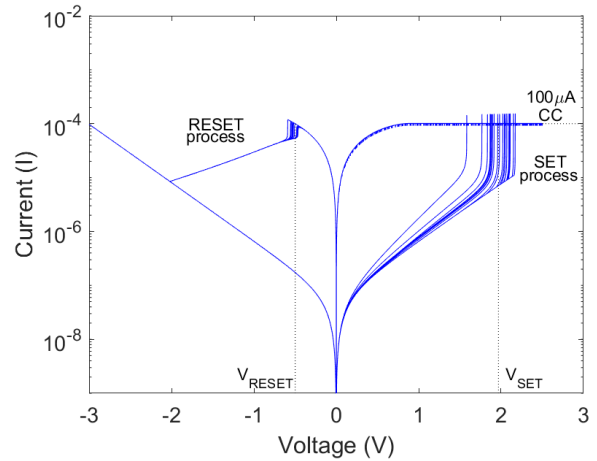


Fig. 1. I-V simulation results concerning 20 cycles with the default variability applied, using a $400 \mu\text{s}$ -period triangular voltage pulse from -3 V to 2.5 V and a $100 \mu\text{A}$ compliance current for the SET process.

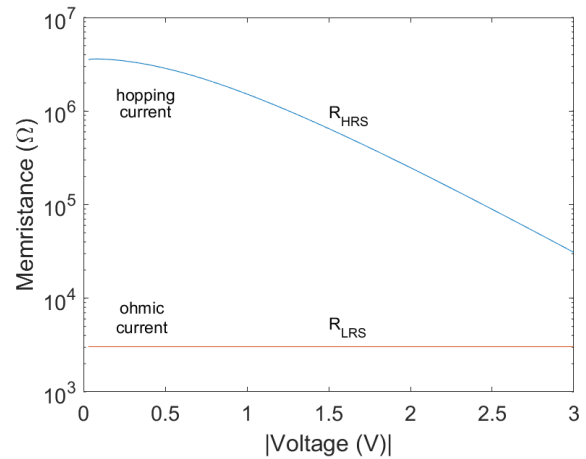


Fig. 2. R_{OFF} and R_{ON} limiting values as a function of the absolute applied voltage across the memristor (without variability).

establish the minimum required amplitude of the programming voltage pulses.

Fig. 2 shows how the effective R_{OFF}/R_{ON} ratio can change depending on the applied voltage. The device is first set to R_{ON} and then a positive voltage is applied. As expected, R_{ON} shows the ohmic conduction of the CF since the memristance cannot be lowered further beyond this value. Nevertheless, when we reset the device to R_{OFF} and then apply a negative voltage, we notice a highly nonlinear behavior of the effective R_{OFF} owing to the hopping current through the tunneling gap [4], [14]. Therefore, such dependency of the R_{OFF} state on the voltage across the device marks a significant difference compared to other device models or analyses where the R_{OFF} state is treated as purely ohmic. This behavior could have a significant impact on the efficiency of memristive applications.

Binary encoding of memristance is necessary for the target application here. Given the used parameter values, the model

exhibits a memristance range from 5 kΩ to 3 MΩ. Following the results in Fig. 2, we define values above 1 MΩ as HRS and values below 100 kΩ as LRS, whereas all values within the intermediate guard band are treated as undefined states. Reading of the memristor state is performed in voltage mode with a 100 kΩ series resistor by applying a small voltage pulse (0.5 V amplitude and 40 ns- wide). The 0.5 V amplitude is low enough to not change the memristance of the device, but lower voltages could be used if more safety in the operation or energy optimization is required. For the purposes of our circuit simulations, we implemented a simple state decoder (adapted from [16]) which provides a digital output, being either ‘0’ (HRS), ‘1’ (LRS) or ‘X’ (undefined state).

The Stanford-PKU ReRAM model supports state variability as well as voltage switching variability. During the switching process a random variable is added to the rate change of the tunneling gap distance g between the electrode and the tip of the conductive filament (CF), and that of the CF width w . Such random variable is a zero-mean Gaussian sequence $\chi(t)$ with deviation δ_g and δ_w , respectively. In our study, $\delta_g = k \times \delta_{g0}$ and $\delta_w = k \times \delta_{w0}$ ($\delta_{g0} = 10^{-4}$ m/s and $\delta_{w0} = 5 \cdot 10^{-4}$ m/s are the default values), where $k = 1, 2, 3, \dots$ is a variability factor that permits configuring easily the amount of desired variability. This state variability affects the memristance value as well as the switching thresholds (as noticed previously in Figs. 1, 2 for $k=1$).

State programming of the devices may be influenced by the past history of their state. However, since our objective here is to show the impact of device variability, we rather suppress any dependencies on the previous device history via a two-step initialization process, described as follows: when programming the device to the LRS (HRS), this is done by first performing a hard RESET (SET) and then a soft SET (RESET). Hard SET/RESET completely forms/destroys the CF to thus eliminate the previous history of the memristor and also prevents the cycle-to-cycle variability. On the other hand, the soft programming initializes the memristor to a state within the LRS or HRS ranges, thus including the desired variability effect in the initialized state. The voltage pulses applied for the HRS initialization concern: 3 V amplitude, 200 ns width and 500 μA cc for hard SET, -2 V amplitude, 100 ns width for soft RESET. On the contrary, for the LRS initialization it is: -2.5 V amplitude, 200 ns width for hard RESET, 3 V amplitude, 100 ns width and 50 μA cc for soft SET. Fig. 3 shows simulation results for the memristance distribution of R_{LRS} and R_{HRS} after the initialization.

III. VARIABILITY-AWARE ASSESSMENT OF MAGIC NOR GATES

We now proceed to the evaluation of possible impact of device variations on the performance of memristive logic gates. In this context, we focus on Memristor-Aided loGIC (MAGIC) [5], a well-known logic design scheme for its crossbar compatibility (only for the NOR operation). In MAGIC, every logic computation is evaluated in just two steps, regardless of the number of inputs. For instance, the NOR $_n$ gate consists of n -input memristors $m_{x1} \dots m_{xn}$ plus an output memristor m_y shown in Fig. 4. The logic operation is performed as follows: First, the output memristor m_y is set to LRS. Next, a voltage pulse of amplitude V_0 is applied to the top electrode (TE) of

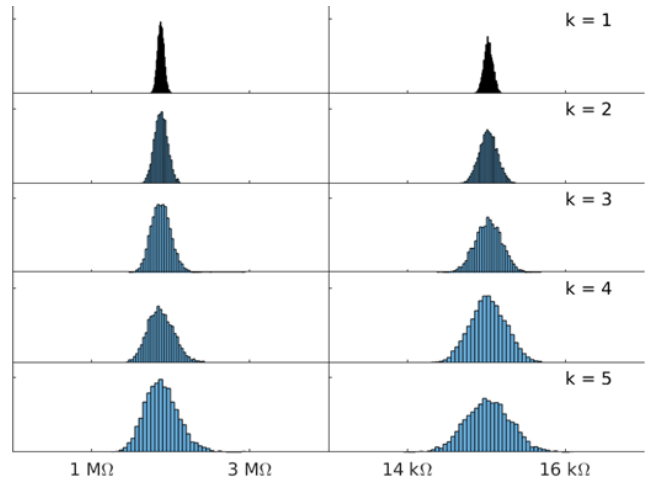


Fig. 3 5000 total HRS (left column) and LRS (right column) reading distributions for different variability factor k values.

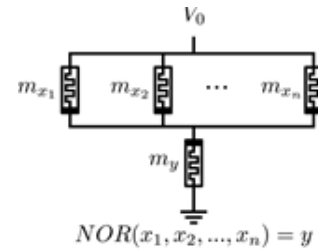


Fig. 4 MAGIC NOR $_n$ logic gate circuit.

every input memristor while the TE of the output memristor is connected to the ground. This operation is equivalent to a conditional RESET process of m_y when at least one input device has a logic ‘1’ stored. V_0 input is selected such that guarantees that m_y will switch only in the appropriate case and the operation will not be destructive, i.e. the input memristors states are not affected.

The MAGIC NOR2 logic gate was designed and simulated using the Stanford-PKU model. Considering a 200 ns- wide voltage pulse, an amplitude V_0 sweep was performed to determine which values guarantee a successful NOR2 operation. The resistance values for m_{x1} , m_{x2} and m_y were stored after every logic operation and are shown in Fig. 5(a) for each swept V_0 values. As it can be observed, there is some unintended state-drift causing either the input or the output memristor state to approximate the undefined region. In fact, the upper boundary for V_0 is defined at the 00-input case as $V_0 \approx 2.19$ V where both input memristors exceed the lowest HRS limit, so they no longer hold an acceptable ‘0’ logic level. Likewise, the lower boundary for V_0 is defined at the 11-input case as $V_0 \approx 1.89$ V where the state of the output memristor exceeds the lowest HRS limit.

Once we have located the appropriate range for V_0 , we applied variability to explore its potential impact. We assume that an error occurs if, after the logic operation, the state of any

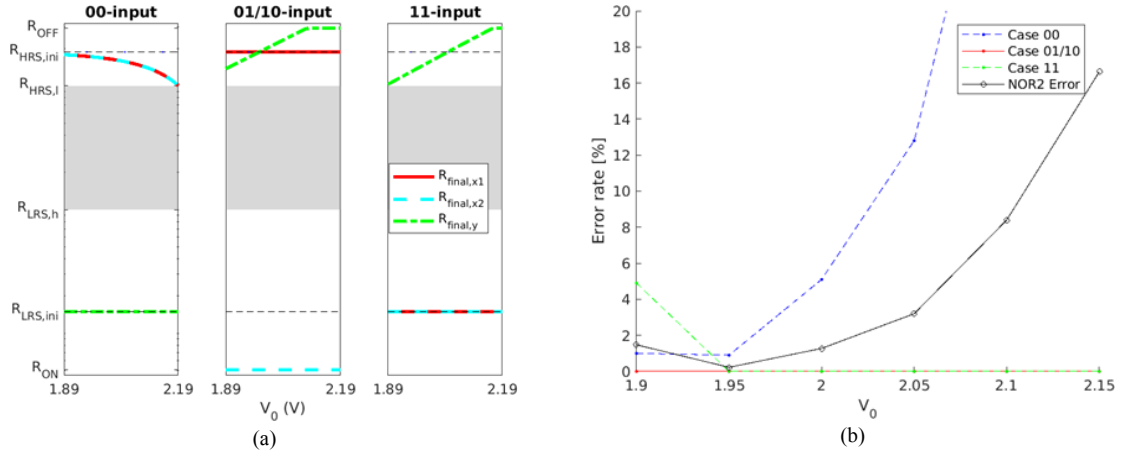


Fig. 5 (a) Resistance state of every device involved in a MAGIC NOR2 after the logic operation is complete, shown as a function of V_0 without variability. (b) MAGIC NOR2 error rate for different V_0 values when variability is included.

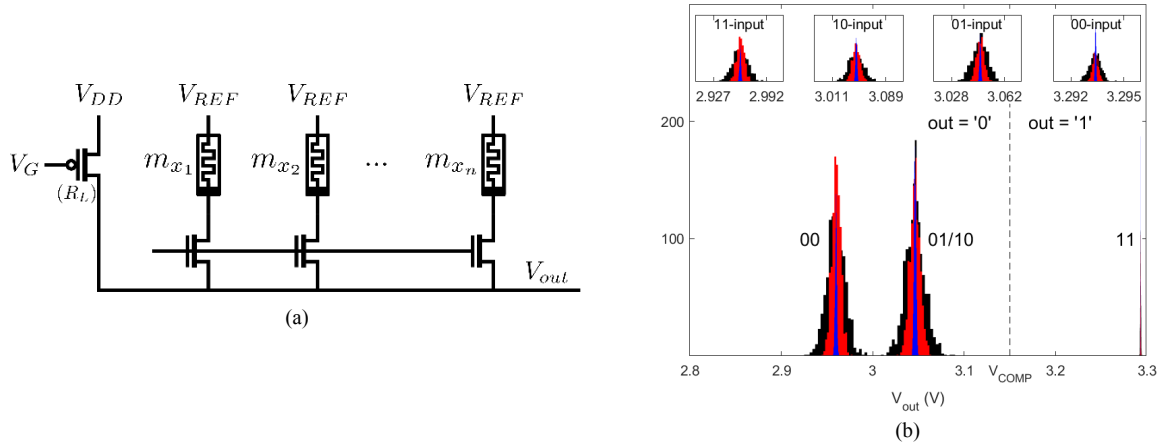


Fig. 6 (a) Proposed ratioed NOR n memristive gate circuit. (b) Output voltage of NOR2 gate in the form of histograms for each input combination (1000 evaluations per input combination) and variability factors $k = 1$ (blue), 5 (red) and 10 (black).

of the devices being involved is not the expected one. Fig. 5(b) shows the average error evolution of NOR2 for different V_0 values and a variability factor $k = 5$, concerning each time 4000 evaluations for random initialization of the input memristors. The contribution of each input combination is also shown separately with the 00-input case practically dominating as V_0 increases, but for low V_0 values it is the 11-input case that dominates; the minimum error rate is noted when $V_0 = 1.95$ V. Consequently, unless variability is properly taken into consideration in the design space exploration, high error rates can appear resulting in malfunction of MAGIC NOR gates.

IV. VARIABILITY TOLERANT RATIOED LOGIC WITH MEMRISTORS

In this Section we present an alternative crossbar-compatible logic design scheme that overcomes the limitations found due to state-drift and memristor variability. It is inspired on the pseudo-NMOS design, working very similar to the CMOS-like logic but with much less circuit complexity. Fig. 6(a) depicts a NOR n logic gate circuit in a crossbar, where memristors m_{x1} to m_{xn} store the input data, and a pull-up PMOS transistor is tuned to exhibit a channel resistance $R_L \approx R_{ON}$.

Contrary to MAGIC, the logic output corresponds to a voltage level whose value falls between V_{REF} and V_{DD} , where V_{DD} is a voltage appropriate for reading the memristor states. So the result of computation is not directly stored in the memory array, so it makes sense to talk about near-memory computing (the result can be stored back to a memory element right afterwards). Once the aforementioned voltage pulses are applied, two different cases are observed. When input is “00”, the input memristances are R_{OFF} and V_{out} is roughly V_{DD} , interpreted as logic level ‘1’. Otherwise, V_{out} falls to $(V_{DD} - V_{REF})/2 + V_{REF}$ or below that, covering the range of logic level ‘0’. Hence, the resulting V_{out} is compared to a threshold value V_{COMP} properly selected so that the output voltage levels for ‘0’ and ‘1’ are clearly distinguished.

Interestingly, the design of the logic gate is quite straightforward as it only requires that $(V_{DD} - V_{REF})$ does not change memristor states. By applying the same voltage range of 0.5 V used in the reading operation ($V_{DD} = 3.3$ V and $V_{REF} = 2.8$ V) we avoid entirely the drift phenomena that inevitably appeared in the MAGIC case. Duration of voltage pulses does not need to be as long as in MAGIC since no switching is required, so they are set to 20 ns just like in the reading

operation. Once designed, the impact of device variability was also evaluated for this logic scheme. Fig. 6(b) shows the V_{out} captures after a total of 4000 evaluations of NOR2 logic gate operation for every different input combination, while applying three different variability factors $k = 1, 5,$ and 10 shown in blue, red and black color, respectively. Regardless of the variability factor, which was set even higher for this gate, the output logic levels are unequivocally identified and error rate is zero. Therefore, such results confirm that the proposed ratioed logic scheme is very robust against variability, compared to MAGIC or Scouting Logic, being also crossbar-compatible and potentially improving further other characteristics of MAGIC gates, such as delay and unwanted state-drift phenomena. As possible drawbacks, there may be a concern in the pull-up PMOS and the delivery of the output data. In terms of area, the inclusion of PMOS increases cost, but the impact on the scaling is limited as only one transistor is required per row. Also, the fact that input data is stored in memristors but output signal is voltage implies that the result of the operation has to be stored later in a memristor or used as an intermediate results and then discarded, according to the application. This may require to enable a programming operation by means of a signal converter. Indeed, it adds an extra step in the operation but we expect little hardware overhead.

V. CONCLUSION

Simulation results concerning a well-known logic design scheme with memristors, confirmed that a variability-aware design and more realistic circuit simulations using physics-based device models, are necessary. Our analysis showed that MAGIC behavior is sensitive to design parameters, increasing significantly the error rate with little deviations. State drift is also observed in the design process using a realistic memristor model. Unless variability is properly taken into consideration in the design flow, unacceptably high error rates could certainly appear and cause malfunction. On the other hand, the proposed ratioed logic scheme with memristors was proven much more tolerant against device variability, and is crossbar-compatible, being a good candidate for in-/near-memory computing systems.

ACKNOWLEDGMENT

This work has been funded in part by the Spanish MINECO and ERDF (TEC2016-75151-C3-2-R), and by the Chilean

government under the research grant CONICYT FONDECYT Postdoctorado No. 3160042.

REFERENCES

- [1] L. O. Chua, "Memristor - The Missing Circuit Element," *IEEE Trans. Circuit Theory*, vol. 18, no. 5, pp. 507-519, 1971.
- [2] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80 - 83, 2008.
- [3] "International Technology Roadmap for Semiconductors (ITRS)," 2013. [Online]. Available: <http://www.itrs.net/>. [Accessed June 2014].
- [4] H.-S. P. Wong, et al., "Metal-Oxide RRAM," *IEEE Proc.*, vol. 100, no. 6, pp. 1951-1970, 2012.
- [5] I. Vourkas and G. Ch. Sirakoulis, "Emerging Memristor-based Logic Circuit Design Approaches: A Review," *IEEE Circ. and Syst. Mag.*, vol. 16, no. 3 (3rd quarter), pp. 15-30, 2016.
- [6] H. Li, T. F. Wu, S. Mitra, and H.-S. P. Wong, "Resistive RAM-centric computing: Design and modeling methodology," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 9, pp. 2263-2273, Sep. 2017.
- [7] Y. Pershin and M. Di Ventra "SPICE Model of Memristive Devices with Threshold," *Radioengineering*, vol. 22, no. 2, pp. 485-489, 2013.
- [8] V. Ntinis, I. Vourkas, G. Ch. Sirakoulis, and A. Adamatzky, "Modeling Physarum space exploration using memristors," *J. Phys. D: Appl. Phys.*, vol. 50 (174004), 2017.
- [9] M. A. Nugent, T. W. Molter, "Thermodynamic-RAM technology stack," *Int. Journal of Parallel, Emergent and Distributed Systems* (2017), in press, Published online: 18 Apr 2017.
- [10] J. Reuben, et al., "Memristive logic: A framework for evaluation and comparison," 27th Int. Symp. Power and Timing Modeling, Optimization and Simulation (PATMOS 2017), Thessaloniki, Greece, 25-27 Sept. 2017.
- [11] L. Xie, et al., "Scouting Logic: A Novel Memristor-Based Logic Design for Resistive Computing," 2017 IEEE Computer Society Annual Symp. VLSI (ISVLSI), , Bochum, Germany, 3-5 July 2017.
- [12] I. Vourkas, and G. Ch. Sirakoulis, "A Novel Design and Modeling Paradigm for Memristor-Based Crossbar Circuits," *IEEE Trans. Nanotechnol.*, vol. 11, no. 6, pp. 1151 - 1159, 2012.
- [13] I. Vourkas, M. Escudero, G. Papandroulidakis, G. Ch. Sirakoulis, and A. Rubio, "Variability Challenges in Emerging Memristor-based Logic Circuits," *HiPEAC'17 Workshop on Memristor Technology, Design, Automation and Computing (mDAC2017)*, Stockholm, Sweden, January 23, 2017.
- [14] Z. Jiang, H. Li, J. H. Engel, S. Yu, and X. Guan, "Stanford-PKU RRAM Model v2.0.0," [Online]. Available: <https://nano.stanford.edu/stanford-rram-model> [Accessed February 2018].
- [15] Z. Jiang, Y. Wu, S. Yu, L. Yang, K. Song, Z. Karim, and H.-S. Wong, "A Compact Model for Metal-Oxide Resistive Random Access Memory with Experiment Verification," *IEEE Trans. Electron Devices*, vol. 63, no. 5, pp. 1884-1892, 2016.
- [16] I. Vourkas and G. C. Sirakoulis, "High-Radix Arithmetic-Logic Unit (ALU) Based on Memristors," in *Memristor-Based Nanoelectronic Computing Circuits Architectures*, 1st ed. Cham, Switzerland: Springer, 2015, pp. 149-172.