

On the Variability-aware Design of Memristor-based Logic Circuits

M. Escudero¹, I. Vourkas², A. Rubio¹, and F. Moll¹

¹ Dept. of Electronic Engineering, Universidad Politécnica de Cataluña, Barcelona, Spain, email: manuel.escudero@upc.edu

²Dept. of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile

Abstract— Ever since the advent of the first TiO₂-based memristor and the respective linear model published by Hewlett-Packard Labs, several behavioral models of memristors have been published. Such models capture the fundamental characteristics of resistive switching behavior through simple equations and rules, so they received a lot of attention and contributed significantly to the fast progress of research in this new and emerging device technology field. However, while this technology is maturing, accurate physics-based models are being developed, which go deeper into the device dynamics and capture more details than what just would be the fundamentals: i.e. parasitics of the device structure, variability of threshold voltages and resistance states, temperature dependency, dynamic current fluctuations, etc. In this work we build upon such a physics-based model of a bipolar metal-oxide resistive RAM device, showing how to take into account device variability and its significance in evaluation of processing circuits. With the Cadence Virtuoso suite, we focus on a family of memristive logic gate implementations showing that read & write errors can emerge due to both variability and state-drift impact, features rarely seen so far in results shown in other relevant published works.

I. INTRODUCTION

The existence of the memristor as the fourth fundamental circuit element was postulated by Leon Chua in 1971 [1]. However, an unprecedented attention on this new and emerging device technology has been drawn only ever since 2008 and the first demonstration of the well-known TiO₂-based memristor by Hewlett-Packard Laboratories (HP Labs) [2], who managed to connect the nature of such devices with Chua's previous theory. Owing to their analog nature, potential nonvolatility, high integration density and CMOS compatibility [3], memristors constitute an emerging trend in modern electronics, representing a promising technology with a variety of applications including memory [4], logic [5] and computing [6].

The 2008 HP Labs invention also concerned the development of a simple device model, which has been ever since the basis for several more behavioral and SPICE-compatible memristor models published later [7-9]. Such models usually capture the very basic behavioral characteristics of the devices (e.g. threshold-based switching and nonlinearities near the resistive boundaries through window functions [10], to name a few), being generally adequate to demonstrate the impact and usefulness of using memristors in a variety of applications [11], [12] as well as for the development

of memristor emulators [13]. However, they omit device physics and thus, leave a lot of space for potential errors or malfunctions that could be seen in real circuit implementations. Thankfully, while this technology is continuously maturing, the first commercially available devices were released by *Knowm Inc.* [14] who provided affordable access to real memristors. Therefore, since variability is something easily observed in such devices in laboratory experiments, even by the uninitiated, it should be by default included in circuit simulations. Moreover, there are some physics-based device models published recently, that go deeper into the device dynamics and take into account features such as variation of voltage thresholds and resistance states, temperature dependencies, parasitics owing to the device structure [e.g. metal-insulator-metal (MIM)], etc.

In this work we build upon such a physics-based model of a bipolar metal-oxide resistive RAM (ReRAM) device [15], [16]. The main contributions of this work concern a) showing how to take into account device variability in circuit simulations and b) its importance in the design and evaluation of memristor-based circuits and systems. More specifically, using the Cadence Virtuoso suite we focus on one of the most promising applications of memristors, i.e. logic design, and we select a popular target logic family as an example to show that both read and write errors can emerge due to variability and state-drift impact. Such features are rarely considered so far in other relevant publications in this field and the presented results highlight that variability can be critical for proper device selection and circuit design quality/viability assessment.

II. TARGET ReRAM DEVICE MODEL DETAILS

A. General Description of the Target ReRAM Device Model

The target model used in this work is the Stanford-PKU ReRAM device model [6]. 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. Most importantly, the model includes parasitic effects such as the parasitic resistance of the switching layer and the electrodes, the parasitic MIM capacitance, it supports intrinsic variation effects, temperature dependency, etc., thus supporting literally all the ReRAM device variation effects known to date. Operation is very similar to that of other models. A positive applied voltage produces a SET process, where the oxide layer suffers a soft-breakdown; the CF is formed and the

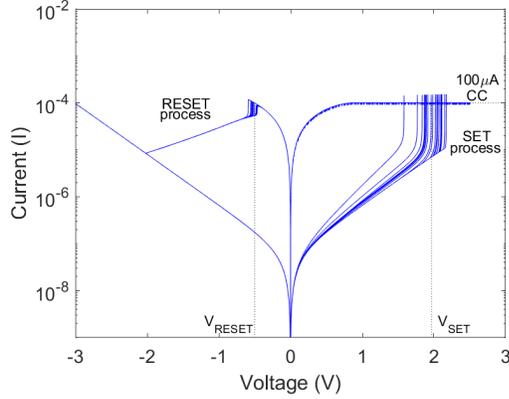


Fig. 1. i - v simulation results concerning 20 cycles with the default variability applied, using a 400 μ s- period triangular voltage pulse from -3 V to 2.5 V and a 100 μ A compliance current (cc) for the SET process.

device is at a low resistive state (LRS or R_{ON}). On the other hand, a negative applied voltage causes a RESET in which the CF is dissolved through ion diffusion or drift processes and the device is at a high resistive state (HRS or R_{OFF}).

B. Exploring Important Features & Variability

The model was adapted for use in the Cadence Virtuoso suite and the majority of the parameter values were kept at their default values [15], except the ones that directly affect the voltage 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 of these parameters is recommended to adjust the overall device behavior, according to the application requirements. For instance, assuming that for a particular application the following relation between the voltage thresholds is necessary: $V_{SET} > 2 \times V_{RESET}$, then to achieve such behavior the aforementioned parameters could be tuned as follows: $E_a = 0.9$ eV, $E_h = 0.9$ eV and $E_i = 0.7$ eV.

Several sets of simulations were done to observe the behavior of the model, understand and control the variability and further explore its impact. Fig. 1 demonstrates i - v curves for 20 cycles taken for a device under a triangular applied voltage. The compliance current (cc) was defined by tuning the gate voltage of a 0.35 μ m NMOS transistor connected in series. 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. Statistics give us the following mean values: $V_{SET} \approx 2$ V and $V_{RESET} \approx -0.5$ V.

Fig. 2 shows the way in which the effective ratio between R_{OFF} and R_{ON} can be modified by the applied voltage. We first set the device to R_{ON} and then apply a positive voltage. As expected, R_{ON} shows the ohmic conduction of the CF since the memristance cannot be lowered further beyond the R_{ON} value. However, when we reset the device to R_{OFF} and apply a negative voltage, interestingly we notice a highly nonlinear behavior of the effective R_{OFF} owing to the hopping current through the

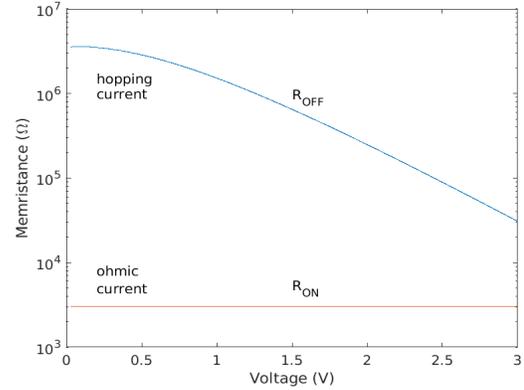


Fig. 2. R_{OFF} and R_{ON} limiting values as a function of the absolute applied voltage across the memristor (without variability).

tunneling gap [4], [15]; i.e. although the memristance does not change beyond R_{OFF} , the conducting behavior does depend on the state and on the absolute value of the applied voltage. 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. As shown next, this behavior could have a significant impact on the efficiency of memristive applications.

C. Important Considerations for the Target Memristive Application

Since memristor-based digital logic is on the focus in this paper, binary encoding of memristance is necessary. Encoding is comprised by two stages, a sensing and a comparing stage. Memristance of a target device is sensed in voltage mode, i.e. with a 100 k Ω series resistor by applying a voltage pulse low enough (0.5 V amplitude and 40 ns- wide) to not affect the device state. According to Fig. 2, our memristor model exhibits a memristance range from 5 k Ω to 3 M Ω and the binary correspondence of the memristance is shown in Fig. 3a. In fact, we defined values above 1 M Ω as HRS and values below 100 k Ω as LRS, whereas all values within the guard band (the dark area) are undefined states. While memristance is being sensed, the corresponding voltage is driven to the state decoder shown in Fig. 3b (adapted from [17]), in which the comparators and the rest of digital components compare the sensed value and provide a valid digital output, this being either '0' (HRS), '1' (LRS) or 'X' in case of an undefined read state.

D. Variability Handling

The Stanford-PKU ReRAM model [15] supports state variability as well as voltage switching variability. Equations for the tunneling gap distance g and the conductive filament (CF) width w , are as follows:

$$g = \int_0^t \left(\frac{dg}{dt} + \delta g \cdot \chi(t) \right) dt \quad (1)$$

$$w = \int_0^t \left(\frac{dw}{dt} + \delta w \cdot \chi(t) \right) dt \quad (2)$$

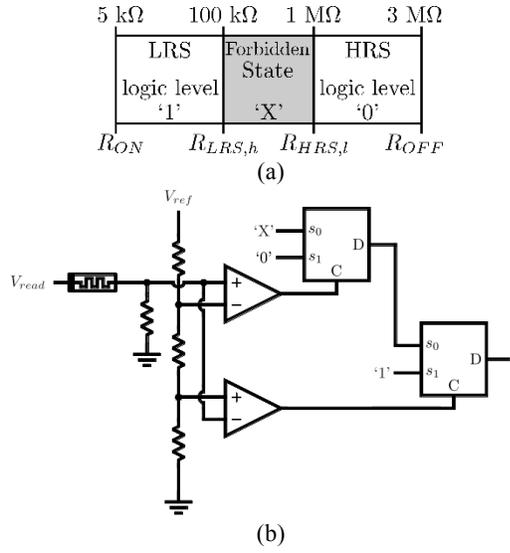


Fig. 3. (a) Memristance binary coding. (b) General form of the reading circuit that senses the memristance in voltage mode and compares with different reference voltages to decode the memristance as logic '0', '1' or undefined (badly written) 'X'.

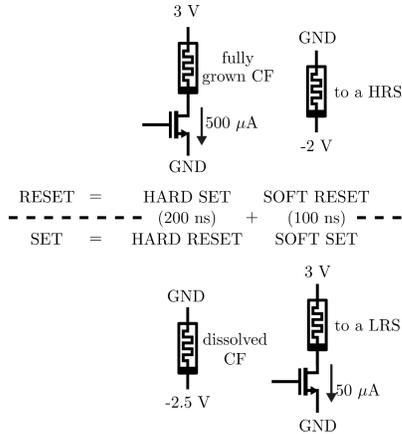


Fig. 4. Two-step programming details for RESET and SET processes.

Given (1) and (2) from [15], 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 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 Fig. 1 where $k=1$ was used).

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 (see Fig. 4), described as follows: when programming the device to the LRS (HRS), this is done by first

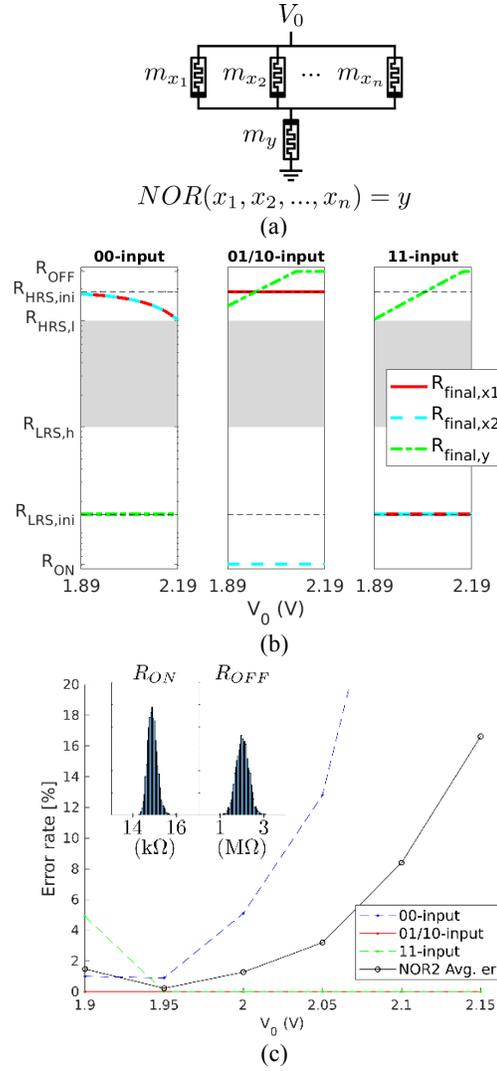


Fig. 5. (a) MAGIC NOR_n circuit schematic showing basic configuration. (b) Memristance of every device involved in a MAGIC NOR₂ after the logic operation, shown as a function of V_0 without variability. (c) MAGIC NOR₂ average error for different V_0 values when variability is included.

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, according to Fig. 3a, thus including the desired variability effect in the initial state. More specifically, 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, the voltage pulses applied for the LRS initialization concern: -2.5 V amplitude, 200 ns width for hard RESET, 3 V amplitude, 100 ns width and 50 μA cc for soft SET, explained in Fig. 4.

III. VARIABILITY IMPACT ON MEMRISTIVE LOGIC GATES

In this Section we show that variability can impact on the performance of memristive logic gates. We select Memristor-Aided loGIC (MAGIC) [5], a promising logic design scheme

for its crossbar compatibility which could result in real in-memory computations. Particularly, the MAGIC NOR n gate consists of n -input memristors $m_{x1} \dots m_{xn}$ plus an output memristor m_y as depicted in Fig. 5a. The logic operation is performed as follows: The output memristor m_y which will hold the logic result, is first set to LRS. Next, a voltage pulse of amplitude V_0 is applied to the top electrode (TE) of every input memristor with the TE of the output memristor being grounded. This is equivalent to a conditional RESET process of m_y when at least one input device has a logic '1'. V_0 is selected such that guarantees that m_y will switch only in the appropriate case and the operation will not be destructive for the input memristors.

The MAGIC NOR2 gate was designed and simulated using the Stanford-PKU model, first without the effect of variability. Considering a 200 ns- wide voltage pulse, an amplitude V_0 sweep was performed to determine which values guarantee a successful NOR2 operation. The memristance values for m_{x1} , m_{x2} and m_y for different V_0 values were stored after every logic operation and are shown in Fig. 5b. As it can be observed, there is some unintended state-drift owing to hopping current conduction, 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 (no longer hold an acceptable '0'.) Likewise, the lower boundary for V_0 is defined at the 11-input case as $V_0 \approx 1.89$ V where the output memristor state exceeds the lowest HRS limit.

Once we decided on the appropriate range for V_0 , we applied variability to further explore its impact. We define that an error occurs in the logic operation if, after the logic operation, the state of anyone of the devices being involved is not the expected one. Fig. 5c shows the average error evolution of NOR2 concerning 4000 evaluations for random initialization of the input memristors, for different V_0 values and variability with a std. dev. of 215 k Ω and 282 Ω for R_{OFF} and R_{ON} distributions (see the inset of Fig. 5c). The contribution of each input case is also shown separately with the 00-input case practically dominating as V_0 increases. However, for low V_0 values it is the 11-input case that distinguishes. Depending on V_0 , the average error is different and in this case a minimum is found when $V_0 = 1.95$ V. Such results confirm the requirement for variability-aware analysis of such circuits involving several interconnected memristors. Unless variability is taken into consideration in the design space exploration, high error rates can appear resulting in unexpected malfunction.

IV. CONCLUSIONS

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 absolutely necessary. Our analysis showed that error rate is sensible on the valid design space. Also, potential state-drift after a logic operation should be a design concern when the memristors involved are subsequently used in other logic operations. Moreover, unless variability is

properly taken into consideration in the design flow, unacceptably high error rates could certainly appear and cause malfunction. Future work will thus focus on the definition of variability-aware design space for different memristive logic families.

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. 3160042/2016.

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.itrs2.net/>. [Accessed June 2018].
- [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] I. Vourkas, A. Batsos, and G. Ch. Sirakoulis, "SPICE modeling of nonlinear memristive behavior," *Int. J. Circ. Theor. Appl.*, vol. 43, no. 5, pp. 553-565, 2015
- [8] C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, "Memristor SPICE Modeling," in *Advances in Neuromorphic Memristor Science and Applications*, 1st Ed. The Netherlands: Springer, 2012, pp. 211-244
- [9] Y. Pershin and M. Di Ventra "SPICE Model of Memristive Devices with Threshold," *Radioengineering*, vol. 22, no. 2, pp. 485-489, 2013
- [10] J. Zha, H. Huang, and Y. Liu, "A Novel Window Function for Memristor Model with Application in Programming Analog Circuits," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 5, pp. 423-427, 2016
- [11] 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
- [12] 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
- [13] D. Biolek, "Memristor emulators," in *Memristor Networks*, A. Adamatzky and L. Chua, Eds. Cham, Switzerland: Springer, 2014, pp. 487-503
- [14] Knowm Inc., "Neuromemristive Artificial Intelligence," [Online]. Available: <https://knowm.org> [Accessed June 2018].
- [15] 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].
- [16] 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
- [17] 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