IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 25, 2019, accepted November 8, 2019, date of publication November 20, 2019,
date of current version December 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2954753

RRAM Device Models: A Comparative Analysis With Experimental Validation

BASMA HAJRI[1], HASSEN AZIZA[2], MOHAMMAD M. MANSOUR[1], (Senior Member, IEEE), AND ALI CHEHAB[1], (Senior Member, IEEE)
[1]Electrical Engineering Department, American University of Beirut, Beirut 1107 2020, Lebanon
[2]IM2NP Laboratory, Aix-Marseille University, 13013 Marseille, France

Corresponding author: Basma Hajri (bh24@aub.edu.lb)

ABSTRACT Resistive Random Access Memories (RRAM) have recently shown outstanding characteristics such as high-scalability, high-speed, high-density, and low-energy operation. A simple and accurate model is crucial for rapid design and verification when using RRAM devices at the circuit level. The appropriate model selection gives insight into the behavior of RRAM as well as the efficient use of its unique properties. This work intends to guide the circuit designers in selecting the most appropriate RRAM model for their applications. We introduce a complete set of evaluation criteria for memristor models: type of model, type of switching, genericity, complexity, compatibility with actual physical switching mechanisms, linearity, symmetry, voltage/current control, hard set/soft reset, support electroforming, support for high programming signal frequencies, existence of a threshold, voltage level, timing dependence, temperature dependence and variability. This study compares the main existing RRAM models and summarizes the results in a table showing the main features and limitations of each model. Through extensive simulations and comparisons with experimental data, we provide an analysis and a validation of the reviewed models within the same simulation environment, ranging from individual elementary cells to large memory arrays. Furthermore, we provide a single and unique Verilog-A code integrating all the compared models.

INDEX TERMS Resistive random access memory (RRAM), memristor models, Verilog-A, model comparison, models assessment, simulation, experimental validation.

I. INTRODUCTION

The end of lithographic scaling of conventional memory technologies such as SRAM, DRAM, and NAND flash has been an eminent call for the past few years, with many touting the emergence of new memory technologies including spin-torque-transfer random access memory (STT-RAM), phase-change memory (PCM), and resistive random access memory (RRAM). Recently, RRAM devices received considerable attention given their fast programming and high scalability. In its primitive form, a resistive memory element relies on a Metal/Insulator/Metal (MIM) stack acting as a resistive switch. This concept also matches the core behavior of the so-called memristor devices discovered by Chua [1].

A critical requirement for using RRAMs in circuits is a predictive model for the device behavior that can guide the circuit designers in their different applications. An appropriate choice of the model will not only lead to a better understanding of the memory cell's behavior but also results in a better exploitation of its unique properties in novel systems and architectures combining data storage and data processing in the same physical location such as neuromorphic applications, memory computing, etc.

The motivation of this work is to provide designers with a guide to select the most appropriate RRAM model for their applications. Multiple reviews on RRAM device modeling have appeared in the literature [2]–[4]; however, to the best of the authors' knowledge, this paper is the first work that presents a complete set of evaluation criteria and an experimentally validated comparative analysis for RRAM models. In this study, we provide a comparative analysis of several popular RRAM models tested within the same simulation environment. We also introduce a unique implementation of all the models in Verilog-A. The different RRAM models

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

used in this work include the Linear Ion Drift Model [5], Non-Linear Ion Drift Model [6], Simmons Tunnel Barrier Model [7], TEAM Model [8], VTEAM model [9], Stanford model [10], SPICE model [11], and IM2NP model [12].

The manuscript is organized as follows. Section II gives an overview of RRAM technology. Section III reviews various previously published RRAM device models and novel modeling techniques. Section IV presents the simulation results of the various models at the cell level in different configurations. Section V is a comparative analysis of the different models with experimental validation. Section VI includes the validation and assessment of the models at the circuit level. Concluding remarks are in Section VII.

II. RRAM TECHNOLOGY

In RRAM the data is stored as two (or multiple) resistance states of the resistive switching device. After an initial electroforming process, the RRAM device resistance state can switch from low (ON or LRS state) to high (OFF or HRS state) and vice versa. The switching process is mainly due to the formation and dissolution of the Conductive Filament (CF) [13] as oxygen vacancies are created and ions are redistributed under the influence of the local electric field and temperature distribution. The electroforming process corresponds to the first switching of the RRAM device from a virgin state (very high resistance) to a conductive state (low resistance) by applying a high voltage. In the case of bipolar switching, bipolar voltage sweeps are required to switch the memory element, as shown in Figure 1.

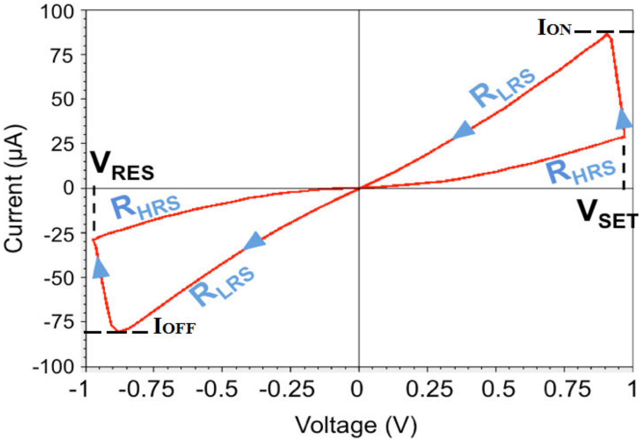


FIGURE 1. **A typical I-V curve of bipolar switching RRAM cell.**

The resistance state switching occurs by applying a specific voltage to the structure (i.e., V_{SET} and V_{RESET}). Based on the I-V characteristic shown in Figure 1, four RRAM critical reliability parameters can be considered: V_{SET}, V_{RESET}, R_{OFF}, and R_{ON}. From a design point of view, these parameters are critical since V_{SET} and V_{RESET} are the programming thresholds and R_{ON}/R_{OFF} ratio guarantees the memory function.

Several other parameters play an important role at the design level, such as the maximum current during switching (namely I_{OFF} and I_{ON}). To allow for low power and reliable SET and RESET operations, Burr *et al.* introduced a 1T/1R memory cell (one MOS transistor in series with one resistor) [14]. In this configuration, the MOS transistor compliance allows control of the maximum available current during transitions. In terms of performance, the programming speed, which is the time required to SET (T_{SET}) or RESET (T_{RESET}) the resistive device, is one of the most critical parameters [15]. Table 1 summarizes the main RRAM cell parameters.

TABLE 1. **RRAM cell parameters.**

Parameter	Definition
R_{ON}	SET state resistance (LRS)
R_{OFF}	RESET state resistance (HRS)
I_{ON}	Maximum current during SET process
I_{OFF}	Maximum current during RESET process
V_{SET}	Minimum voltage needed to turn ON the cell
V_{RESET}	Minimum voltage needed to turn OFF the cell
T_{SET}	Minimum time needed to turn ON the cell
T_{RESET}	Minimum time needed to turn OFF the cell

Although RRAM-based devices have shown promising properties, some challenges remain, among which the device variability (or reproducibility) is the main one. Therefore, both design and modeling community are giving increasing attention to the impact of variability on the RRAM cell parameters [16]. Another important marker of RRAM is the variation of SET/RESET/FORMING thresholds versus temperature [17].

Modeling and accurate characterization of the SET/RESET mechanisms remain a significant challenge [18], [19]. Many details are still under discussion, such as the origin of the nonlinear switching kinetics [20]. In memory devices relying on a resistance change, complex physical mechanisms are responsible for the reversible switching of the electrical conductivity between high and low resistance states. The resistivity change is generally attributed to the formation and dissolution of conductive paths between metallic electrodes. Various mechanisms may explain the resistance change (oxygen vacancy migration, oxidation-reduction processes, thermal diffusion, etc.). RRAM models exploit the various resistance change mechanisms to evaluate the impact of external stimulus on the cell parameters during programming operations. Moreover, RRAM models' complexity should not be high to allow an implementation into electrical simulators and assess cell performances at the circuit level.

III. GENERAL OVERVIEW OF RRAM MODELS

A. LINEAR ION DRIFT MODEL

Developed by R.S Williams at HP labs [5], this model was the first physical model of memristor. This model, assumes an

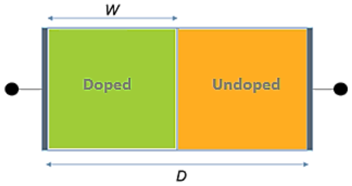


FIGURE 2. Linear ion drift model.

average ion mobility, uniform field and ohmic conductance for the memristor device.

As shown in Figure 2, in the memristive element (width D) there are two regions: doped region (width w) with positive oxygen ions (commonly TiO2) that has a limited resistance and then higher conductivity; and a second region that is undoped.

B. NON-LINEAR ION DRIFT MODEL

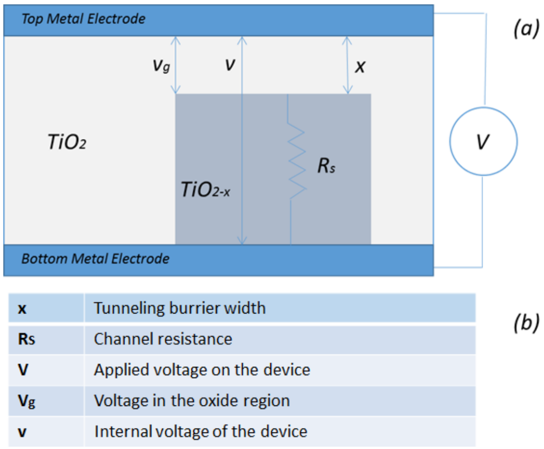
After the fabrication of memristive devices, Strukov and Williams [6] demonstrated that they exhibit high non-linearity behavior. Therefore, the HP laboratory developed the first nonlinear ion drift model. The non-linearity of the device assumed in this model is voltage-dependent (nonlinear dependence between the internal state derivative and the voltage). The state variable w is a standardized parameter varying from 0 to 1. By setting the appropriate values of the model parameters, this model accurately describes both the static electric conduction as well as the dynamic switching behaviors and fits the experimental data well. Logic gates are the primary application of this model.

C. SIMMONS TUNNEL BARRIER MODEL

Pickett *et al.* proposed a complex memristor physical model with higher accuracy in [7]. As shown in Figure 3, it represents the memristor as an electron tunnel barrier in series with a resistor. Nonlinear and asymmetric switching of the memristor is also assumed. The Simmons tunnel barrier width [21] is considered as the state variable x, which is the oxide region width. Therefore, the oxygen vacancy drift velocity can be deduced from the state variable derivative.

D. THRESHOLD ADAPTIVE MEMRISTOR MODEL (TEAM)

The ThrEshold Adaptive Memristor model [8] has been developed to overcome the complexity and limited computational efficiency of the Simmons tunnel barrier model. This model preserves the same physics principle, but using simpler mathematical equations. This model developed two main contributions (1) the state variable does not change unless the applied current exceeds a certain threshold. (2) The equations relating the current to the internal state derivative use polynomials rather than exponentials. This model is simple, general, and computationally efficient.



x	Tunneling burrier width
Rs	Channel resistance
V	Applied voltage on the device
Vg	Voltage in the oxide region
v	Internal voltage of the device

FIGURE 3. Representation of the Simmons tunnel barrier model.

E. VOLTAGE THRESHOLD ADAPTIVE MEMRISTOR MODEL (VTEAM)

VTEAM model [9] is a modified version of the TEAM model, where a threshold voltage is used rather than a threshold current. This model is appropriate for certain logic and memory applications.

F. STANFORD MODEL

This SPICE-compatible compact model characterizes the Metal-Oxide RRAM bipolar switching behavior [10].

Jiang *et al.* abbreviated the ions/vacancies migration complicated process into the progress of a unique primary filament that preserves the main switching physics. The dominant variable determining the device resistance is g the tunneling gap size that represents the distance from the filament tip to the opposite electrode, as shown in Figure 4. An exponential relation between the tunneling gap distance and the current conduction is assumed. The tunneling gap distance is deduced after calculating the gap progress taking into account the local temperature (Joule heating), the electric field, and oxygen ion migration (temperature-enhanced). Furthermore, this model includes temperature-dependent and stochastic filament movement ($\delta\ g$).

G. SPICE MODEL

The SPICE model [11] assumes a memristance controlled by a voltage source. The memristive system considered in this

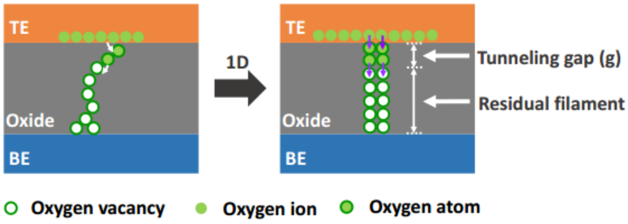


FIGURE 4. Stanford-PKU RRAM model illustration.

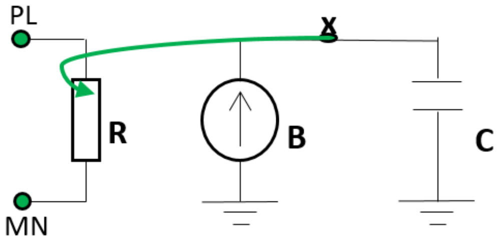


FIGURE 5. **SPICE model representation.**

model is a subcircuit, shown in Figure 5, including a resistor *R*, a current source *B,* and capacitor *C*. The capacitor voltage (Vx between the capacitor and the current source) defines the resistance of *R*.

H. IM2NP RRAM MODEL

The IM2NP Eldo model is a compact model that describes well the SET and RESET operations in bipolar resistive switching in Oxide-based memory devises. The model takes into account the conductive filament (CF) electric field-induced creation and dissolution as well, by including in a unique equation of electrochemical reaction and thermal mechanism. This model is also the first that includes the electroforming process of RRAM devices. The model has been calibrated on dynamic and quasi-static experimental data [12]. Figure 6 gives a representation of the model and its main parameters.

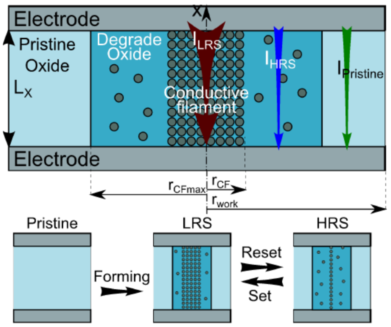


FIGURE 6. **IM2NP RRAM model illustration.**

I. OTHER MODELING TECHNIQUES

A different modeling technique proposed in [22] includes truncated-cone shaped filaments which are known to be close to the real conductive filament (CF) geometry and a detailed thermal approach, where two temperatures are considered to describe the rupture process at the CF's narrowest part and also the main CF body's electrical conductivity variations.

Another interesting advanced 3D physical modeling technique to predict correctly the switching phenomena has been developed recently in [23] focusing on the promising silicon-rich silica (SiOx) RRAMs. This technique provides new insight on RRAM physics; however, it is not included in this work since it does not explicitly provide the modeling equations.

Additionally, a SPICE model developed in [24] exhibits a voltage threshold. This model appears to match well the characterization data of different memristors.

New SPICE models with simpler analytical approximations have been developed to overcome the complexity of the memristor physical processes implementation. Bayat *et al.* proposed a TiO2 memristor SPICE model based on Simmons Tunnel Barrier Model, but with improved accuracy and lower numerical simulation cost [25]. However, in this model, only mathematical approximations of measured characteristics are used with no qualitative insight.

Alternatively, several behavioral models (e.g., those by Biolek *et al.* [26], [27]) simplify the physical memristor mechanism to some useful abstractions fitted to the experimentally observed behaviors to maximize the size of the memristive networks.

IV. MODELS SIMULATION RESULTS

For a fair comparison between the several models listed in Section III, we created an identical Cadence simulation environment. Moreover, to help the design community selecting the most suitable model for their applications without wasting time reading each model publication separately and writing the corresponding code, a Verilog-A code of all the listed models is presented in Appendix-A. In this implementation, a "num_model" parameter should be set to target a specific RRAM model.

The TEAM, VTEAM, Simmons Tunnel Barrier Model, Linear Ion Drift Model, and Non-Linear Ion Drift models are implemented in Verilog-A within the same code [28], [29]. Stanford, IM2NP and SPICE models developed initially in Verilog, Eldo, and Pspice respectively are added.

A. SIMULATION SETUP

The design is implemented using ST-Microelectronics HCMOS9 (130nm) technology under a supply voltage of VDD = 1.8. In transient simulations, we apply to the Top/Bottom electrodes a pulse wave of 1MHz frequency, 1.8V peak-to-peak, 100ns period, and 1ns rise/fall time.

In the 1T1R configuration, we selected an NMOS transistor since it provides more current for a given *L* and *W* than a PMOS device.

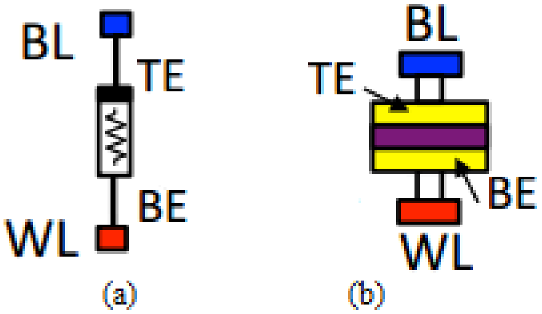
B. 1R CONFIGURATION

A single RRAM device (Figure 7) is simulated using each model. The transient I-V characteristics of the different simulated models using this configuration have been published in our previous work [30]; and the results are explained in Section V.

C. 1T1R CONFIGURATION

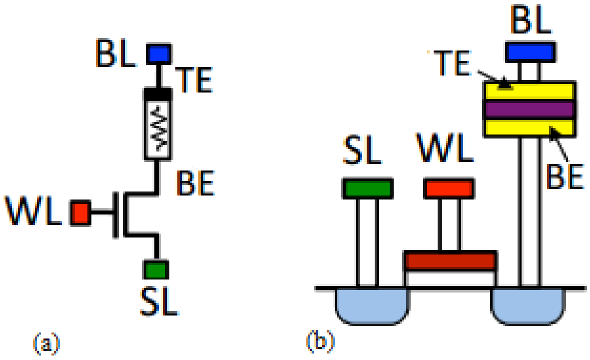
As shown in Figure 8, the 1T1R structure is composed of one memristor (RRAM cell) and one transistor.

TE the Top electrode of the memristor is connected to the Bit Line (BL) of the RRAM and the bottom electrode BE is



(a) (b)

FIGURE 7. (a) 1R RRAM cell; (b) cell cross-section view.



(a) (b)

FIGURE 8. (a) 1T1R RRAM cell (b) cell cross-section view.

connected to the transistor drain. WL is the word line, and SL is the source line of the RRAM memory.

The transistor is used to access the selected cell and isolate it from unselected ones. Another primary reason for using the transistor is to limit the write current (set the compliance). However, the choice of the transistor dimensions is fundamental since it determines the area of the cell and consequently, the density of cells and the possible final device applications.

The major issue of the 1T1R topology is that a single device used as a switch cannot pass well both high and low voltages: NMOS devices are good at passing low voltages, while PMOS devices high voltages.

We decided to keep the transistor width and length to their minimum values: W=240nm and L=180nm, and to connect the transistor gate to 1.8V or 0V; the compliance current is hence around 100uA. Figure 9 shows the transient I-V characteristics of the different simulated models.

V. MODEL COMPARISON

We performed the simulations of all the studied models under the same Cadence environment and the same initial conditions; then, we compared the extracted parameters to the experimental data [31]. The experimental reference is an RRAM device of 10 nm thickness that consists of TiN/Ti/HfO_2/TiN structure with a 3-nm-thick HfO_2 layer; tested under input voltage between [−1.8V, 1.8V] at very high frequency. We selected [31] as a reference since it matches best our simulation conditions and settings. Table 2 presents the extracted parameters of all the simulated models at 1T1R configuration. For easier comparison, Figure 10 compares the measured parameters (V_{SET}, V_{RESET}, T_{SET}, and T_{RESET}) to the experimental data [31] shown in red.

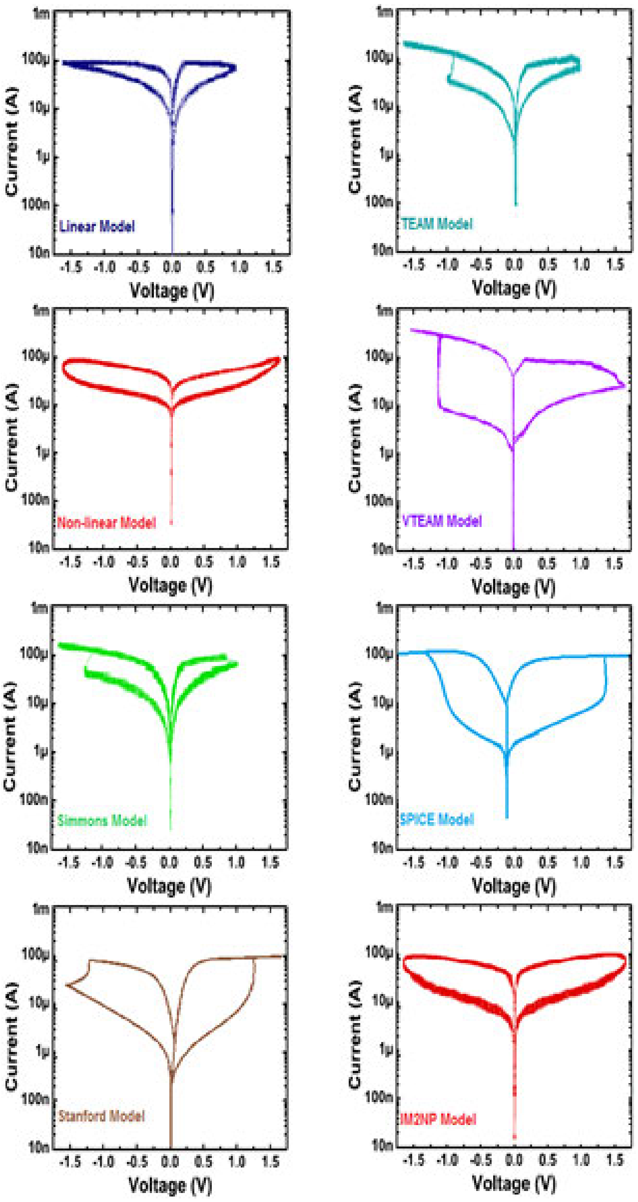


FIGURE 9. Models I-V characteristics in 1T1R configuration.

According to the simulation results presented in Table 2, we performed a ranking of the models based on the number of parameters that match the experimental data [31]. The results are shown in Figure 11. The weight of each parameter is determined regarding its importance and impact at a design level; in decreasing order V_{SET}/V_{RESET}, the resistance ratio (R_{OFF}/R_{ON}) and T_{SET}/T_{RESET}.

A. COMPARISON METRICS

Many review papers and comparative studies in the literature tried to classify the RRAM models but with no clear

TABLE 2. 1T1R cell all models extracted parameters compared to experimental measurements [31].

Model	*Parameters*					
	SET			*RESET*		
	R_{ON} *(KΩ)*	V_{SET} *(V)*	T_{SET} *(ns)*	R_{OFF} *(KΩ)*	V_{RES} *(V)*	T_{RES} *(ns)*
Linear [5]	0.05	1.78	4.2	1	1.78	5.5
Nonlinear [6]	79	1.59	8.36	399	1.69	8.75
Simmons [7]	0.01	1.59	8	0.71	0.19	0.001
TEAM [8]	0.01	1.78	0.59	10	0.17	0.003
VTEAM [9]	0.01	0.19	0.48	10	0.87	0.002
Stanford [10]	3.8	1.87	3.01	357	1.05	0.487
SPICE [11]	0.048	0.95	4.57	1.01	0.95	4.79
IM2NP [12]	5	1.79	9.12	803	1.75	17.56
Exp. data [31]	**4.82**	**1.08**	**10**	**1000**	**1**	**10**

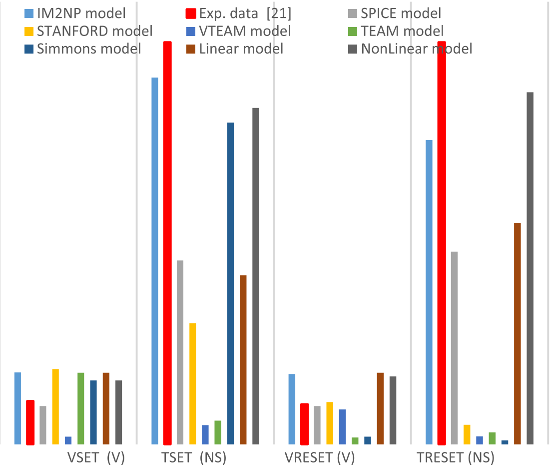


FIGURE 10. Comparison between the experimental data [31] (RED columns) and the parameters of each simulated model V_{SET}, V_{RESET}, T_{SET}, and T_{RESET}.

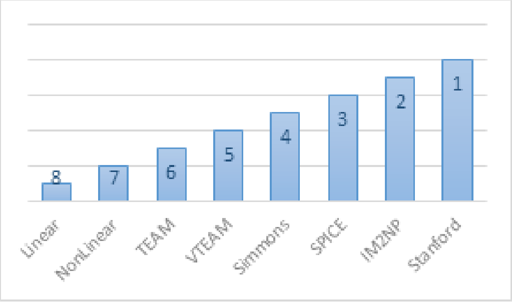


FIGURE 11. Models ranking based on the number of parameters that match the experimental data [31].

evaluation criteria. The study suggested in [32] is the first work to introduce three main evaluation criteria of the models: plausibility of the I-V characteristics, nonlinearity of the switching kinetics, and the correct prediction of two anti-serially connected devices behavior. However, these metrics are still incomplete to compare the different models thoroughly. In the purpose of performing a fair comparison between the several studied models, we introduce the following complete set of metrics:

- **Type of the model:** whether it is a compact, analytical, or physics-based model.
- **Efficient use in RRAM arrays:** checks if the model can be used for RRAM (1t1R and crossbar) arrays.
- **Type of switching** (unipolar or bipolar): checked by applying first a unique positive voltage then a sequence of positive and negative voltage pulses.
- **Genericity**: this characteristic allows the model to be adapted to different memristor technologies.
- **Complexity:** A model is considered complex if the equations use hyperbolic sine and exponents rather than polynomials [33]. This metric is determined from the model equations and double-checked by measuring the simulation runtime of the different models.
- **Compatibility with the actual physical switching mechanism** (creation and destruction of conductive filaments): checked from the equations of the model.
- **Non-Linearity:** it should be in the model equations and reflected in the I-V characteristic. The origin of this nonlinearity has been attributed to the nonlinear drift of the ionic defects accelerated by Joule heating [34].
- **Symmetry** of the modeled SET/RESET processes. This feature appears in the simulated I-V characteristic of the model.
- **Voltage-controlled** or **current controlled**
- **Hard set/soft reset**, presented in the literature for actual memristive devices as the ratio between reset time and set time [35], a high ratio means a hard set and a soft reset. This metric is checked from the I-V characteristic and time parameters measured in Table 2.
- **Electroforming**: A voltage higher than the regular operation should applied to the device to construct an initial filament between top and bottom electrodes. An explicit electroforming equation should be provided in the model description.
- **Support for high programming signal frequencies**: A model should support a wide range of working frequencies to make possible the simulation of novel fabricated devices that present very fast switching times [36]. For each model, a frequency sweep is performed to check whether it supports high frequencies or not.
- **Existence of a threshold**: physical memristor devices demonstrate a threshold voltage where hysteresis only appears when the voltage across the memristor exceeds the threshold [37].
- **Pulse-programming Voltage dependence**: A simulation is performed using a sweep on the amplitude of the applied pulsed voltage to confirm this dependence.

- **Pulse-programming Timing dependence**: The same simulation is performed using a square signal by varying the pulse width to check if the duration of the applied voltage affects the model parameters.
- **Temperature dependence**: A temperature sweep is applied to check if the temperature affects the model parameters.
- **Variability dependence** [38]. It includes Device-to-device variability and Cycle-to-cycle variability. A simulation is performed by changing a variability parameter dx (Appendix A) to check this dependence.
- **Random Telegraph Noise (RTN)** [39] is another important source of variability in RRAM devices. The RTN fluctuations are not included in the studied models; however, Puglisi *et al.* [39] developed a Verilog-A physics-based compact model of the RTN that can be easily plugged in any existing RRAM device models.
- **Retention/endurance:** the endurance characteristic is measured by performing a series of consecutive set/read/reset/read cycles [40]. The retention measures the stability of HRS and LRS over time under a given temperature. Several models have been proposed to explain retention losses [41]–[43]. The compared models in this work do not include the retention feature; however, an updated version of the Stanford model proposed in [44] includes a resistance retention failure mechanism modeled in Verilog-A.

Linear [5] and nonlinear [6] models are intuitive and straightforward, based on the same assumption of two resistors in series, yet they present the lowest accuracy compared to other models [32]. Besides, these models are not generic. The Simmons tunnel barrier model [7] is known to be an accurate physical model [45] however, compared to the first two models; it is not generic and fits best for only specific memristor devices based on TiO_2 stacks. That's why the error between our experimental reference (HfOx based) [31] and this model is high. This model is also complicated, as the relationship between current and voltage is not explicit and computationally inefficient.

Moreover, the Linear Ion Drift Model [5], the non-Linear Ion Drift Model [6] and the Simmons Tunnel Barrier Model [7] do not contain any threshold, which means that their resistance varies for any voltage or current value.

The TEAM [8] and VTEAM [9] models are simpler versions of the Simmons model, generic, more computationally efficient and include threshold current and threshold voltage respectively; however, they are not physical models.

The SPICE model [10] is simple; it fits all the experimental parameters [31] except the current since the current-voltage relationship is not physics-based. The model does not match the actual memristive behavior, and its state variable has no physical explanation.

The Stanford [10] and IM2NP [12] are two physics-based compact models for bipolar RRAM memristive devices; they are the only models to take into consideration temperature effects, timing effects and variability observed in many actual RRAM cells.

For easier use and understanding, we summarized the comparison results in Appendix A.

B. THERMAL EFFECT ANALYSIS

The Stanford model [10] and the IM2NP model [12] as almost all the models, which include the thermal effect [22], [23], [26], [46], [47], are based on the filament dissolution model proposed in [48], [49]. In this model, the conductive filament rupture, or dissolution occurs under the effect of a significant change in temperature based on the fundamental concept of Joule heating [50]. During the RESET process and by increasing the applied voltage, the temperature steadily rises until a value called the critical temperature. Above this critical temperature, the conductive filament dissolution/rupture takes place at a fast rate inducing a High Resistance State (HRS) of the device.

During the SET process, the temperature rises due to the increase in the CF radius.

In the Stanford model [10], the applied voltage, as shown in Figure 12, directly affects the temperature change in the CF radius. A temperature peak is observed at each SET and RESET sequence.

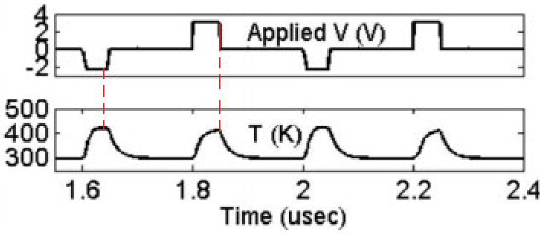


FIGURE 12. Stanford model simulated temperature evolution as a function of the applied voltage [51].

For the IM2NP model, both V_{SET} and V_{RESET} are not much affected by the ambient temperature (almost 50mV variation in the given range) as shown in Figure 13. However, the electroforming voltage is highly activated by the ambient temperature.

C. ACCURACY

The accuracy of the three models, that best match the experimental results, is evaluated by comparison of the simulation results and the measured data on silicon at cell level, provided by ST-Microelectronics [31]. The mean error between the simulated I-V curves of each model and measured data is given in Table 3.

The smallest error observed is between the Stanford model and the experimental data. Figure 14 shows the simulated I-V curves of three selected models along with the experimental I-V curve [31].

In conclusion, only two physics-based models (Stanford and IM2NP) fulfill most of the essential evaluation criteria with reasonable accuracy.

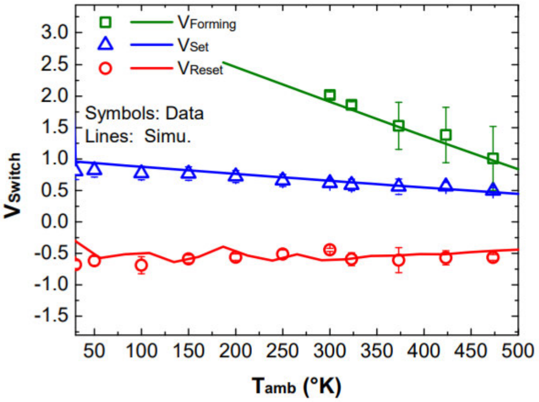


FIGURE 13. IM2NP model experimental and simulation results of switching voltage as a function of temperature [52].

TABLE 3. Mean error.

MODEL	MEAN ERROR (%)
STANFORD- EXPERIMENTAL DATA	15.01
IM2NP- EXPERIMENTAL DATA	24.4
SPICE- EXPERIMENTAL DATA	29.1

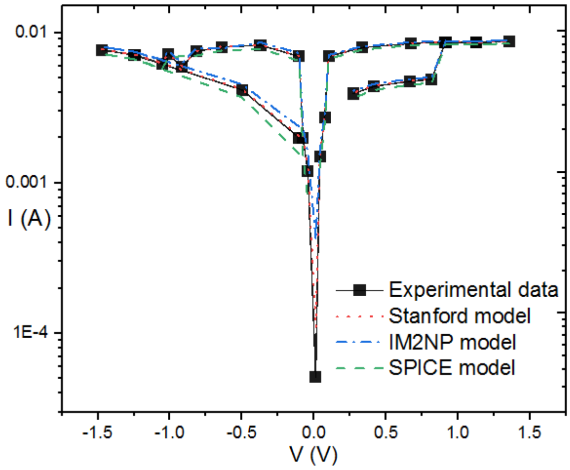


FIGURE 14. I-V characteristics of the models that best match the experimental data [31].

VI. MODEL VALIDATION AND ASSESSMENT AT THE CIRCUIT LEVEL

A. 1-K BIT RRAM ARRAY SIMULATION

In the previous sections, we have exhaustively evaluated the dynamic behavior of each model at a single RRAM cell level. Nevertheless, to validate the model comparison at the circuit level, the models' functionality has been evaluated by simulating numerous RRAM cells connected in a 1T1R memory array. Using the physics-based models for simulations of large memristor arrays remains challenging because of the enormous computing resources required. For example, to simulate a 400 Mb memristor array, it may take about a year [53] using a complex physics-based model with a SPICE simulator. For our analysis, we use a 1K-bit RRAM array to keep a reasonable simulation time and get significant results at the same time. The complete simulated memory array is shown in Figure 15. It consists of, a Bit Line decoder, a Source Line decoder and a Word Line decoder connected to the different 1T1R cells.

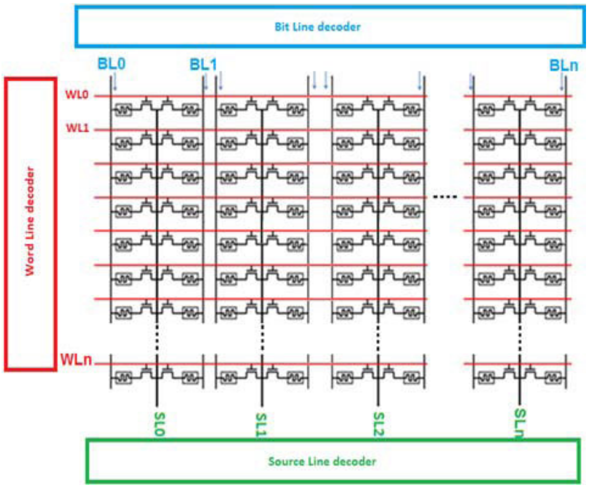


FIGURE 15. Schematic of the simulated 1T1R memory array (n=1K) with decoders.

First, all the RRAM cells are RESET (set to high resistance state). Then, two cycles are required to perform the programming of the memory array. First, all memory cells are set (logical "1"), then the memory array is reset (logical "0"). Simulation results are consistent with those of single-cell operation and confirm the data provided in Appendix A, though not included here for brevity.

Some models of the simulated models (linear, non-linear, TEAM, and VTEAM models) presented convergence problems and are not capable of large-scale circuit simulation. A solution for the encountered problems of each model is presented in [54].

An additional critical concern when using the RRAM model for simulations of large arrays is the huge computing resources required. Table 4 gives the simulation run time and the computation memory usage for the models that best match the experimental data: Stanford, IM2NP, and SPICE models.

B. VARIABILITY

According to section IV, Stanford and IM2NP models are the only models that can be used to simulate variability in RRAM cells. Appendix B includes the equations implementing the variability of both models.

TABLE 4. Computation time and memory usage for 1k-bit array.

Model	Simulation run time (s)	Computation memory (MB)
Stanford [10]	46.47	538
SPICE [11]	41.16	535
IM2NP [12]	80.67	542

TABLE 5. Comparative Analysis of the Main Models.

Model Metric	Model [5] (Linear)	Model [6] (Nonlinear)	Model [7] (Simmons)	Model [8] (TEAM)	Model [9] (VTEAM)	Model [10] (Stanford)	Model [11] (SPICE)	Model [12] (IM2NP)
Type of the model	Ideal physics	Physics-based	Physics-based	Physics-based	Physics-based	Physics-based	Analytical	Physics-based
Efficient use in RRAM arrays	✗	✗	✗	✗	✓	✓	✗	✓
Bipolar switching	✓	✓	✓	✓	✓	✓	✓	✓
Low complexity	✓	✗	✗	✓	✓	✗	✓	✓
Matching the actual memristive behavior	✗	✗	✓	✓	✓	✓	✗	✓
Genericity	✗	✗	✗	✓	✓	✓ (1)	✓	✓ (1)
Non-Linearity	✗	✓	✓	✓	✓	✓	✓	✓
Symmetric	✓	✗	✗	✗	✗	✗	✓ (2)	✗
Voltage-controlled	✗ (3)	✓	✗ (3)	✗ (3)	✓	✓	✓	✓
Hard set	✗	✗	✗	✓	✓	✓	✓	✓
Soft reset (4)	✗	✗	✗	✓	✓	✗	✓	✓
Electro Forming	✗	✗	✗	✗	✗	✗	✗	✓
Support of high frequencies (5)	✗	✗	✓	✓	✓	✓	✓	✓
Threshold	✗	✗	✗	✓	✓	✓	✓	✓
Pulse-programming Voltage dependence	✗	✗	✗	✗	✓	✓	✓	✓
Pulse-programming Timing dependence (6)	✗	✗	✗	✗	✗	✓	✗	✓
Temperature dependence	✗	✗	✗	✗	✗	✓	✗	✓
Variability dependence	✗	✗	✗	✗	✗	✓	✗	✓
Retention	✗	✗	✗	✗	✗	✓ (7)	✗	✗

(1) Covers all oxide based RRAM devices (2) fitting parameter alpha is used to change from symmetric to asymmetric
(3) Current controlled (4) RESET is slower than SET operation (shape of I-V curve slope) (5) starting 1MHz, (6) Not the frequency of Vin but pulse width,
(7) This feature has been included in the updated version of the model [44].

In this section, we provide an example of application where we studied the two types of RRAM variability of both models:

- Device-to-device variability describes the RRAM cell behavior consistency inside the memory array.
- Cycle-to-cycle variability measures the RRAM cell stability over different cycles.

A B1500 semiconductor parameter analyzer is used for measurements. Quasi-static experimental measurements are performed to extract the memory cell main characteristics (i.e., V_{SET}, V_{RESET}, etc.) by applying a 200ms triangular pulses across the 1T1R cell.

Figure 16 shows the cumulative distributions of R_{ON} (LRS) and R_{OFF} (HRS) in different RRAM cells within the 1T1R memory array using the Stanford model and the IM2NP model compared to the experimental data [31]. The excellent agreement between the experimental data and the simulation data proves that both models capture very well the randomness of the resistance levels during the SET and RESET processes of different RRAM cells.

Figure 17 depicts the simulated I-V curves of Stanford model for 10 SET cycles compared to the experimental data. The variation is mainly due to the random generation of oxygen vacancy (V_0). Figure 18 shows the simulated I-V curves of the IM2NP model for 10 SET cycles. A wider range of variation is observed compared to the Stanford model, and that matches better the experimental results.

TABLE 6. Verilog-A Implementation for all Memristive Models.

```
// VerilogA for ALL models
`include "disciplines.h"
`include "constants.h"

// Unit for w is meter

nature distance
  access = Metr;
  units = "m";
  abstol = 0.01n;
endnature
discipline Distance
  potential distance;
enddiscipline

module Memristor (t, b, w_s);
  input t;  //TOP electrode
  input b;  //BOTTOM electrode
  output w_s;
  electrical t, m, b, gnd;
  ground gnd;
  Distance w_s;

   parameter real num_model = 4;
// num_model=0: “Linear Ion Drift Model”
// num_model=1: “Simmons Tunnel Barrier Model”
// num_model=2: “Nonlinear Ion Drift Model”
// num_model=3: “Team Model”
// num_model=4: “Stanford Model”
// num_model=5: “Vteam Model”
// num_model=6: “Spice Model”
// num_model=7: “Im2np Model”

  parameter real wind_typ = 2;
// define the window type:
// wind_typ = 0: No Window
// wind_typ = 1: Jogelkar
// wind_typ = 2: Biolek
// wind_typ = 3: Prodromakis
// wind_typ = 4: Kvatinsky window ONLY for Team model
// wind_typ = 5: Kvatinsky2window ONLY for Vteam
model

   parameter real dt=20e-12;
// At least dt < T(period)/10^3

   parameter real in_st=0.5;
// Initial state should be between [0, 1]

/////////////“Linear Ion Drift” ///////////////

    parameter real R_hrs = 1000;
    parameter real R_lrs = 50;
    parameter real D = 3e-9;
    parameter real w_multip= 1e9;
    parameter real uv = 1e-15;    // transformation
factor (m)
    parameter real coef = 2; // Wind_func_coef
    parameter real J = 1.5;
    parameter real p_wind_NS = 1e-18;
    parameter real V_threshold=0;

    // Definition of local variables
    real sign_multp;
    real stp_multp;
    real w;
    real fst_it;
    real w_prev;
    real R;
    real dwdt;
```

```
  ////////// “Simmons Tunnel Barrier” ///////////

    parameter real c_n = 40e-6;
    parameter real bx = 500e-6;
    parameter real i_n = -1e-6;
    parameter real x_c = 107e-11;
    parameter real i_f = 1e-6;
    parameter real a_f = 1.2e-9;
    parameter real c_f = 3.5e-6;
    parameter real a_n = 2e-9;
    // loc_var
   real dydt;
   real x_prev;
   real x;

////////////“Nonlinear Ion Drift” ////////////
    parameter real N     = 14;
    parameter real c     = 0.01;
    parameter real alpha = 2;
    parameter real q     = 13;
    parameter real g     = 4;
    parameter real a     = 4;
    parameter real beta  = 9;

 analog function integer stp;
     real arg; input arg;
     stp = (arg >= 0 ? 1 : 0 );
 endfunction

 analog function integer sign;
     real arg; input arg;
     sign = (arg >= 0 ? 1 : -1 );
 endfunction
////////////////“TEAM” ////////////////////

    parameter real K_n=-10;
    parameter real K_f=5e-4;
    parameter real Alpha_n=3;
    parameter real Alpha_f=1;
    parameter real v_n=-0.2;
    parameter real v_f=0.02;
    parameter real IV_rel=0;
// If zero V=IR (linear), if one V=I*exp{..}
(nonlinear)
    real lambda;
    parameter real x_n=0;
    parameter real x_f=3e-09;
///////////// “Stanford” //////////////
    parameter real aa = 0.25e-9 ;       // unit: m
distance between adjacent oxygen vacancy
    parameter real f = 1e13 ;          // unit: HZ
vibration frequency of oxygen atom in VO
    parameter real Ea = 0.7 ;          // unit: ev
average active energy of VO
    parameter real Eh = 1.12 ;          // unit: ev
hopping barrier of oxygen ion (O2-)
    parameter real Ei = 0.82 ;          // unit: ev
energy barrier between the electrode and oxide
    parameter real r = 1.5 ;           // enhancement
factor of external voltage
    parameter real alphah = 0.75e-9 ;  // unit: m
enhancement factor in lower Ea & Eh
    parameter real alphaa = 0.75e-9 ;   // unit: m
enhancement factor in lower Ea & Eh
    parameter real Z = 1 ;
    parameter real XT = 0.4e-9 ;       // unit: m
    parameter real VT = 0.4 ;          // unit: V

    parameter real L0 = 3e-9 ;         // unit: m
L0 is defined as the initial fixed length of the RRAM
switching layer
```

TABLE 6. ***(Continued)*** **Verilog-A Implementation for all Memristive Models.**

```
    parameter real x0 = 3e-9 ;          // unit: m
x0 is defined as the initial length of gap region
during both SET/RESET (for SET: x0=L0)
    parameter real w0 = 0.5e-9 ;        // unit: m
initial CF width
    parameter real WCF = 5e-9 ;         // unit: m
fixed width of the RRAM switching layer
    parameter real weff = 0.5e-9 ;      // unit: m
effective CF extending width
    parameter real I0 = 1e13 ;          // unit: A/m^2
hopping current density in the gap region

    parameter real rou = 1.9635e-5 ;    // unit: ohm*m
resistivity of the formed conductive filament (CF)
    parameter real pi = 3.1415926 ;
    parameter real Rth = 5e5 ;          // unit: K/W
effective thermal resistance
    parameter real Kb = 8.61733e-5 ;    // unit: ev/K
    parameter real T0 = 300 ;           // unit: K
ambient temperature
///////////////////////////////////////////////////
    parameter real switch = 1  ;           // switch =
1: variation-included ; switch = 0: no variations
    parameter real deltaGap = 4e-5 ;      // gap
distance variation amplitude
    parameter real deltaCF     = 1e-4 ;   // filament
radius variation amplitude
    parameter real crit_x = 0.5e-9 ;      // decided
by measured variation amplitude
    parameter real user_seed = 0 ;        // specified
user seed for random number generator
// local variables
    real Temp ;

/////// CF evolution speed ///////
    real dxr1 ;
    real dxr2 ;
    real dxs ;
    real dx ;
    real dw ;
/////// initial setup of CF geometry ///////
    real xs;
    real ws ;
    real RCF ;
/////// intermediate electrical parameters///////
    real Vg ;
    real I1 ;
/////////////////////////////////
    integer rand_seed_set  ;
    integer rand_seed_reset ;
    real cf_random_ddt ;
    real gap_random_ddt ;

///////////// "Spice" //////////////
parameter real res = 1T;
parameter real cap = 1;
parameter real Rn=1K ;
parameter real Rf=10K ;
parameter real Rinit=5K ;
parameter real betas=1E13 ;
parameter real Vtp=0.9 ;
parameter real Vtm=0.9 ;
parameter real nu2=0.1 ;
parameter real nu1=0.0001;

///////////// "IM2NP" //////////////
    parameter real rcfmax = 20e-9;
    parameter real Scell = 1e-12; // unit:1um x 1um
    parameter real AHRS = 5e-9; // unit: A/ (V2)

    parameter real Lx = 5e-9;   // oxide thickness,
unit: m
    parameter real Eb = 2 ;           // unit: ev
```

```
    parameter real EaForm = 2.7 ;           // unit:
ev  average active energy of VO during forming
    parameter real Vform = 1 ;
    parameter real alphaRED = 0.7 ;   // unit: m
the SET transfer coefficient (ranging between 0 and
1)
    parameter real alphaOX = 0.7 ;   // unit: m
the RESET transfer coefficient (ranging between 0 and
1)
    parameter real alphaHRS = 2 ;   //
    parameter real sigmaCF = 5e6 ;   // unit: m.S
    parameter real sigmaOX = 50 ;   // unit: m.S
    parameter real tauRedOX = 1e-5 ;   // unit: S
nominal redox rate.
    parameter real tauForm = 1e-21 ;   // unit: S
    parameter real meox = 0.1 ;           // me
    parameter real Kth = 2 ;           // thermal
conductivity unit: W/(K.m)


    parameter real Tamb = 300 ;           // unit: K
ambient temperature
/////// intermediate electrical parameters ///////
    real Ai ;
    real Bi ;
    real Fi ;
    real ICF ;
    real IOX ;
    real T ;
    real rcf ; //Conductive Filament radius between 0
and rcfmax
    real drcf ;
    real tRED ;
    real tOX ;
    real Iox ; // conductive species current
    real Icf ; // oxide conduction current

/////////////////// MAIN Function///////////////
 analog begin
    if(fst_it==0) begin
        w_prev=in_st*D;
        x_prev=in_st*D;
    end
/////////////"Linear Ion Drift" /////////////

if (num_model==0) begin
    dwdt =(uv*R_lrs/D)*I(t,b);
     //w updated only when threshold permits
       if(abs(I(t,b)) < V_threshold/R)
begin
            w = w_prev;
        dwdt=0;
       end
// Without Wind
    if ((wind_typ==0)|| (wind_typ==4))
begin
        w=dwdt*dt+w_prev;
    end // Without Window

// W_Jogelkar
     if (wind_typ==1)
begin

        if (sign(I(t,b)) == 1)
begin
            sign_multp=0;
            if(w<p_wind_NS)
begin
            sign_multp=1;
            end
        end
        if (sign(I(t,b))==-1)
begin
```

TABLE 6. ***(Continued)*** **Verilog-A Implementation for all Memristive Models.**

```
                sign_multp=0;
                if(w>(D-p_wind_NS)) begin
                    sign_multp=-1;
                end
        end
w= (1 - pow(pow(2*w/D-1,2), coef))*dwdt*dt
+w_prev+sign_multp*p_wind_NS;
    end //

// W_Biolek
    if (wind_typ==2) begin
        if (stp(-I(t,b))==1)
begin
            stp_multp=1;
        end
        if (stp(-I(t,b))==0)
begin
                stp_multp=0;
                end
w= (1 - pow(pow(w/D-stp_multp,2), coef))*dwdt*dt
+w_prev;
    end

  // W_Prodromakis
    if (wind_typ==3) begin

        if (sign(I(t,b))==1)
begin
            sign_multp=0;
            if(w < p_wind_NS)
begin
            sign_multp=1;
            end
        end
        if (sign(I(t,b))==-1) begin
                sign_multp=0;
                if(w>(D-p_wind_NS)) begin
                    sign_multp=-1;
                    end
        end
w= (1-pow(pow(w/D-0.5,2)+0.75,coef))*dwdt*dt*J
+w_prev+sign_multp*p_wind_NS;

    end
    if (w >= D) begin
        w = D;
                dwdt=0;
    end
    if (w <= 0) begin
        w = 0;
                dwdt=0;
    end
       // updated outputs

       w_prev=w;
       R=R_lrs*w/D+R_hrs*(1-w/D);
       Metr(w_s) <+ w_multip*w;
 V(t,b) <+ (w/D*R_lrs +R_hrs*I(t,b)*(1-(w/D)));
       fst_it=1;
end

///////// “Simmons Tunnel Barrier” ///////////

if (num_model==1)
begin
    if (sign (I(t,b))== 1)
begin
dydt = c_f*sinh(I(t,b)/i_f)*exp(-exp((x_prev-
a_f)/x_c-abs(I(t,b)/bx))-x_prev/x_c);
    end
    if (sign (I(t,b))== -1)
begin
```

```
dydt =c_n*sinh(I(t,b)/i_n)*exp(-exp((a_n-x_prev)/x_c-
abs(I(t,b)/bx))-x_prev/x_c);
    end
        x = x_prev+dydt*dt;
     if (x >= D)
begin
            x = D;
            dydt = 0;
     end
     if (x<=0) begin
            x=0;
            dydt=0;
     end
       // updated outputs
       R=R_lrs*(1-x/D)+R_hrs*x/D;
       x_prev=x;
       Metr(w_s) <+ x/D;
       V(t,b) <+ (R_lrs*(1-x/D)+R_hrs*x/D)*I(t,b);
       fst_it=1;
end

//////////// “Nonlinear Ion Drift” ////////////
if (num_model==2)
begin
    if (fst_it==0) begin
        w_prev=in_st;
    end
    dwdt = a*pow(V(t,b),q);

// Without Wind
    if ((wind_typ==0) || (wind_typ==4)) begin
        w=w_prev+ dwdt*dt;
        end


  // W_Jogelkar
  if (wind_typ==1) begin
    if (sign (I(t,b)) == 1) begin
        sign_multp = 0;
        if(w < p_wind_NS) begin
            sign_multp=1;
        end
    end
    if (sign (I(t,b)) == -1)
        begin
        sign_multp = 0;
        if(w > (D-p_wind_NS))
        begin
            sign_multp = -1;
        end
    end
w = w_prev+ (1 - pow(pow((2*w_prev-1),2), coef))
*dt*dwdt+sign_multp*p_wind_NS;
    end
// W_Biolek
    if (wind_typ==2) begin
        if (stp(-V(t,b))==1) begin
            stp_multp=1;
        end
        if (stp(-V(t,b))==0) begin
                stp_multp=0;
            end
w = w_prev+ (1-pow(pow((w_prev-
stp_multp),2),coef))*dt*dwdt;

end

// W_Prodromakis
    if (wind_typ==3) begin

    if (sign(I(t,b))==1)
       begin
        sign_multp=0;
```

TABLE 6. ***(Continued)*** **Verilog-A Implementation for all Memristive Models.**

```
        if(w<p_wind_NS)
        begin
            sign_multp=1;
        end
    end
    if (sign(I(t,b))==-1) begin
        sign_multp=0;
        if(w>(D-p_wind_NS)) begin
            sign_multp=-1;
        end
    end
w = w_prev+ (1- pow((pow((w_prev-0.5),2)+0.75),
coef))*dt*dwdt*J+sign_multp*p_wind_NS;
     end
    if (w>=1) begin
        w=1;
        dwdt=0;
    end
    if (w <= 0) begin
        w = 0;
        dwdt = 0;
    end

        if(abs(V(t,b))<V_threshold) begin
            w=w_prev;
        end
//w updated when threshold exceeded

       w_prev=w;
           Metr(w_s) <+ w;
           I(t,b) <+ pow(w,N) *sinh(alpha*V(t,b))
*beta + (exp(g*V(t,b))-1)*c;
           fst_it=1;
end

//////////////////// “TEAM” //////////////////
if (num_model==3) begin
    if (I(t,b) >= i_f) begin
        dydt = pow((I(t,b)/i_f-1),Alpha_f)*K_f;
    end
    if (I(t,b) <= i_n) begin
        dydt = pow((I(t,b)/i_n-1),Alpha_n)*K_n;
    end
        if ((i_n<I(t,b)) && (I(t,b)<i_f)) begin
    dydt=0;
    end

    if (wind_typ==0) begin
// Without Wind
    x=x_prev+dt*dydt;
    end
    // W_Jogelkar
if (wind_typ==1)
begin
    if (sign(I(t,b))==1) begin
        sign_multp=0;
        if(x<p_wind_NS) begin
            sign_multp=1;
        end
    end
    if (sign(I(t,b))==-1) begin
        sign_multp=0;
        if(x>(D-p_wind_NS)) begin
            sign_multp=-1;
        end
    end

x= x_prev + (1 - pow(pow((2*x_prev/D-
1),2),coef))*dt*dydt+sign_multp*p_wind_NS;
           end
// W_Biolek
    if (wind_typ==2) begin
         if (stp(-I(t,b))==1) begin
```

```
            stp_multp=1;
        end
        if (stp(-I(t,b))==0) begin
            stp_multp=0;
            end
x= x_prev+ (1 - pow(pow((x_prev/D-
stp_multp),2),coef))*dt*dydt;
    end
// W_Prodromakis
    if (wind_typ==3)
begin
    if (sign(I(t,b))==1)
begin
        sign_multp=0;
        if(x<p_wind_NS) begin
            sign_multp=1;
        end
    end
    if (sign(I(t,b))==-1) begin
        sign_multp=0;
        if(x>(D-p_wind_NS)) begin
            sign_multp=-1;
        end
    end
x= x_prev + (1 - pow((pow((x_prev/D-0.5),2)+0.75),
coef))*dt*dydt*J+sign_multp*p_wind_NS;
    end
//W_Kvatinsky
    if (wind_typ==4) begin
            if (I(t,b) >= 0) begin
x= x_prev + exp(-exp((x_prev-a_f)/x_c))*dt*dydt;
           end
       if (I(t,b) < 0) begin
x= x_prev + exp(-exp((a_n-x_prev)/x_c))*dt*dydt;
       end
        end
    if (x >= D) begin
    dydt = 0;
    x = D;
    end
    if (x <= 0) begin
    dydt = 0;
    x = 0;
    end
       lambda = ln(R_hrs/R_lrs);
     // updated  outputs
        x_prev=x;
        Metr(w_s) <+ x/D;
     if (IV_rel==1) begin
        V(t,b) <+ R_lrs*I(t,b)*exp(lambda*(x-
x_n)/(x_f-x_n));
     end
     else if (IV_rel==0) begin
        V(t,b) <+ (R_hrs*x/D+R_lrs*(1-x/D))*I(t,b);
     end
    fst_it=1;
end
//////////////////// “Stanford” //////////////

if (num_model==4) begin
    begin
        Temp = (T0 + abs(I(t,b) * V(t,b)) * Rth) ;
        I1 = I0*pi*(WCF*WCF/4-ws*ws/4)*exp(-
L0/XT)*sinh(V(t,b)/VT) ;
        RCF = rou*(L0-xs)/(pi*ws*ws/4) ;
        Vg = V(t,b) - (I(t,b)-I1)*RCF ;
        if ( V(t,b) > 0 )
        begin
            if ( xs > 0 )
            begin

                dxs = -aa*f*exp(-(Ea-
Vg*alphaa*Z/x)/(Kb*Temp)) ;
```

TABLE 6. ***(Continued)*** **Verilog-A Implementation for all Memristive Models.**

```
                if ( abs(dxs) <= 1e2 )
                begin
                    dx = dxs ;
                end
                else
                begin
                    dx = -1e2 ;
                end
            end
            else if ( xs <= 0 )
            begin
                dx = 0 ;
            end
            if ( ws < WCF )
            begin
                dw = (xs>0)*0 +
(xs<=0)*(weff+pow(weff,2)/(2*ws))*f*exp(-(Ea-
V(t,b)*alphaa*Z/L0)/(Kb*Temp)) ;
                if ( dw > 1e2 )
                begin
                    dw = 1e2 ;
                end
            end
            else if ( ws >= WCF)
                dw = 0 ;
        end
        else if ( V(t,b) < 0 )
        begin
            dw = 0 ;
            if ( xs <= 0 )
            begin
          dxr1 = aa*f*exp(-(Ei+r*Z*Vg)/(Kb*Temp)) ;
          dx = dxr1 ;
            end
            else if ( xs > 0 )
            begin
                if ( xs < L0 )
                begin
dxr1 = aa*f*exp(-(Ei+r*Z*Vg)/(Kb*Temp)) ;
dxr2 = aa*f*exp(-Eh/(Kb*Temp))*sinh(alphah*Z*-
1*Vg/(xs*Kb*Temp)) ;
dx = (dxr1<dxr2)*dxr1 + (dxr2<=dxr1)*dxr2 ;
                end
                else if ( xs >= L0 )
                begin
dx = 0 ;
                end
            end
        end
        else if ( V(t,b) == 0 )
        begin
            dx = 0 ;
            dw = 0 ;
        end
/////// electrical behavior transmission///////
        if ( V(t,b) == 0 )
        begin
            I(t,b) <+ 0 ;
        end
        else if ( V(t,b) > 0 )
        begin
            if ( xs > 0 )
            begin
                I(t,b) <+ I1 + I0*pi*(ws*ws/4)*exp(-
xs/XT)*sinh(Vg/VT) ;
            end
            else if ( xs <= 0 )
            begin

                I(t,b) <+ I1 +
V(t,b)/(rou*L0/(pi*ws*ws/4)) ;
            end
        end
```

```
        else if ( V(t,b) < 0 )
        begin
I(t,b) <+ I1 + I0*pi*(ws*ws/4)*exp(-
xs/XT)*sinh(Vg/VT) ;
        end
/////// DIES of CF geometry///////
        if ( switch == 1 )
        begin
            if ( V(t,b) > 0 )
            begin
                if ( dw == 0 )
                begin
                    cf_random_ddt = 0 ;
                end
                else
                begin
rand_seed_set = $random + user_seed ;
cf_random_ddt = $rdist_normal(rand_seed_set,0,1) *
deltaCF ;
                end
            end
            else if ( V(t,b) < 0 )
            begin
                if ( xs < crit_x )
                begin
                    gap_random_ddt = 0 ;
                end
                else if ( xs >= crit_x )
                begin
rand_seed_reset = $random + user_seed ;
                    gap_random_ddt =
$rdist_normal(rand_seed_reset,0,1) * deltaGap ;
                end
            end
            else
            begin
                gap_random_ddt = 0 ;
                cf_random_ddt = 0 ;
            end
        end
        else
        begin
            gap_random_ddt = 0 ;
            cf_random_ddt = 0 ;
        end
        xs = idt(dx+gap_random_ddt,x0) ;
        ws = idt(dw+cf_random_ddt,w0) ;
        if ( xs < 0 )
        begin
            xs = 0 ;
        end
        if ( xs > L0 )
        begin
            xs = L0 ;
        end
        if ( ws > WCF )
        begin
            ws = WCF ;
        end
 Metr(w_s) <+ L0-xs;
    end
//////////////////// "VTEAM" ////////////////
if (num_model==5) begin
    if (V(t,b) >= v_f) begin
        dydt =K_f*pow((V(t,b)/v_f-1),Alpha_f);
    end

    if (V(t,b) <= v_n) begin
        dydt =K_n*pow((V(t,b)/v_n-1),Alpha_n);
    end
        if ((v_n<V(t,b)) && (V(t,b)<v_f)) begin
    dydt=0;
```

TABLE 6. ***(Continued)*** **Verilog-A Implementation for all Memristive Models.**

```
 end

// Without Wind
    if (wind_typ==0) begin
    x=x_prev+dt*dydt;
    end

    // W_Jogelkar
if (wind_typ==1) begin
    if (sign(V(t,b))==1) begin
        sign_multp=0;
        if(x<p_wind_NS) begin
            sign_multp=1;
        end
    end
    if (sign(V(t,b))==-1) begin
        sign_multp=0;
        if(x>(D-p_wind_NS)) begin
            sign_multp=-1;
        end
    end
x= x_prev+ (1- pow(pow((2*x_prev/D-
1),2),coef))*dt*dydt +sign_multp*p_wind_NS;
    end

// W_Biolek
    if (wind_typ==2) begin
         if (stp(-V(t,b))==1) begin
            stp_multp=1;
        end
        if (stp(-V(t,b))==0) begin
            stp_multp=0;
            end
x= x_prev+ (1- pow(pow((x_prev/D-
stp_multp),2),coef))*dt*dydt;
    end

// W_Prodromakis
    if (wind_typ==3) begin


    if (sign(V(t,b))==1) begin
        sign_multp=0;
        if(x<p_wind_NS) begin
            sign_multp=1;
        end
    end
    if (sign(V(t,b))==-1) begin
        sign_multp=0;
        if(x>(D-p_wind_NS)) begin
            sign_multp=-1;
        end
    end
x= x_prev+ (1-pow((pow((x_prev/D-0.5),2)+0.75),
coef))*dt*dydt*J+sign_multp*p_wind_NS;
    end

// W_VTEAM Kvatinsky

    if (wind_typ==5) begin
           if (V(t,b) >= 0) begin
x= x_prev+ exp(-exp((x_prev-a_f)/x_c))*dt*dydt;
           end
       if (V(t,b) < 0) begin
x= x_prev+ exp(-exp((a_n-x_prev)/x_c))*dt*dydt;
       end
        end
    if (x>=D) begin
    dydt=0;
    x=D;
    end
    if (x<=0) begin
    dydt=0;
    x=0;
    end
       lambda = ln(R_hrs/R_lrs);
     // updated outputs
        x_prev=x;
        Metr(w_s) <+ x/D;

     if (IV_rel==1) begin
        V(t,b) <+ R_lrs*I(t,b)*exp(lambda*(x-
x_n)/(x_f-x_n));
     end
     else if (IV_rel==0) begin
        V(t,b) <+ (R_hrs*x/D+R_lrs*(1-x/D))*I(t,b);
     end
    fst_it=1;
end

///////////////////// “Spice” /////////////////
if (num_model==6) begin

V(m,gnd) <+ res*I(m,gnd);
I(m,gnd) <+ cap*ddt(V(m,gnd));

I(m,gnd) <+ (beta*(V(t,b)-Vtp)/(exp(-(V(t,b)-
Vtp)/nu1)+1)+beta*(V(t,b)+Vtm)/(exp(-(-V(t,b)-
Vtm)/nu1)+1))*(1/(exp(-(beta*(V(t,b)-Vtp)/(exp(-
(V(t,b)-Vtp)/nu1)+1)+beta*(V(t,b)+Vtm)/(exp(-(-
V(t,b)-Vtm)/nu1)+1))/nu1)+1))*(1/(exp(-(Rf-
V(m))/nu2)+1))+(1/(exp((beta*(V(t,b)-Vtp)/(exp(-
(V(t,b)-Vtp)/nu1)+1)+beta*(V(t,b)+Vtm)/(exp(-(-
V(t,b)-Vtm)/nu1)+1))/nu1)+1))*(1/(exp(-(V(m)-
Rn)/nu2)+1));
end
    end

///////////////////// “IM2NP” /////////////////

if (num_model==7) begin
T = Tamb +
(V(t,b)*V(t,b)/8*Kth)*((rcf*rcf/rcfmax*rcfmax)*(sigma
CF-sigmaOX)+sigmaOX);
if ( V(t,b) == 0 ) begin
I(t,b) <+ 0;
end
else if ( V(t,b) > 0 )
            begin
tRED  = 1/(tauRedOX*exp(-(Ea-
q*alphaRED*V(t,b))/Kb*T));
            end
else if ( V(t,b) < 0 )
            begin
tOX = 1/(tauRedOX*exp(-(Ea+q*alphaOX*V(t,b))/Kb*T));
            end
drcf= ((rcfmax-rcf)/tRED)-(rcf/tOX);
//  drcf = idt(rcf,rcfmax) ;

Icf= (V(t,b)/Lx)*(rcf*rcf*pi*(sigmaCF-sigmaOX)+
rcfmax*rcfmax*pi*sigmaOX);
Iox= AHRS*Scell*pow((V(t,b)/Lx), alphaHRS);
if ( V(t,b)  > 0 ) begin
I(t,b) <+ -Iox + Icf;
end

else if ( V(t,b)  < 0 ) begin
I(t,b) <+ Iox - Icf;
end

    end

         end   // end analog

endmodule
```

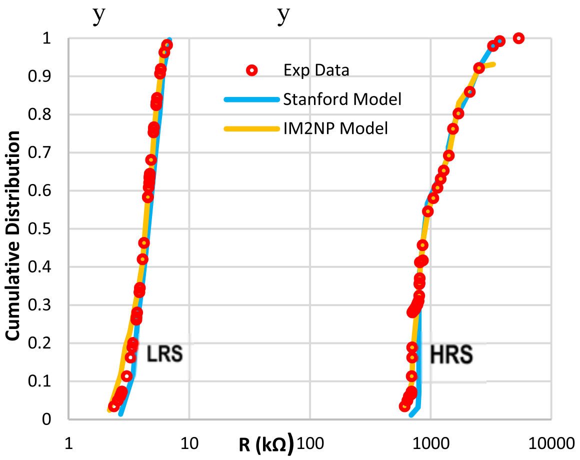


FIGURE 16. **Cumulative distribution of R_{ON} (LRS) and R_{OFF} (HRS) of Stanford model (blue), IM2NP model (yellow) compared to the experimental data (red).**

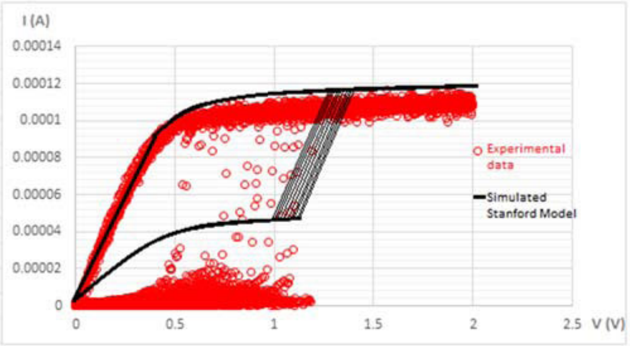


FIGURE 17. **Single-cell Stanford model variability for 10 SET cycles.**

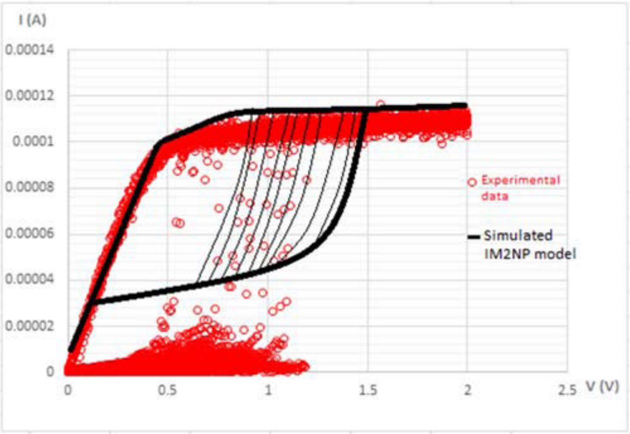


FIGURE 18. **Single-cell IM2NP model variability for 10 SET cycles.**

By tuning the model fitting parameters, the I-V curves of the two models can match better the experimental data as proposed in [10] and [12]. However, the fitting procedure is not performed here since the objective of this section is to validate the existence of variability in two models and show the difference between them.

VII. CONCLUSION

In this paper, we surveyed the major existing RRAM device models. We simulated the models within the same environment and under the same conditions to fulfill the modeling community requirement for a unified discussion on the various RRAM models. This work is the first in the modeling community literature; it presents a complete set of evaluation criteria and an experimentally validated comparative analysis for all widely accepted RRAM models. We summarized the evaluation results as a quick reference table, which represents a tool for the designers to select the model that best matches their applications. Furthermore, this comparative analysis is beneficial to the modeling community since it highlights the limitations of a given model (Appendix A); thus, it points out the areas of improvement.

For all the models discussed in this study, we provide an implementation in the same Verilog-A code (Appendix B), for easier access and assessment by the designers. The user can specify the model number and compare the performance of the different models at the same time. Finally, we validated our comparative analysis at the circuit level using a 1K-bit 1T1R RRAM array, and the results are consistent with the previously published ones at the cell level.

APPENDIXES

APPENDIX A

See Table 5.

APPENDIX B

See Table 6.

REFERENCES

[1] L. Chua, “Memristor-The missing circuit element,” *IEEE Trans. Circuit Theory*, vol. CT-18, no. 5, pp. 507–519, Sep. 1971.

[2] A. Ascoli, F. Corinto, V. Senger, and R. Tetzlaff, “Memristor model comparison,” *IEEE Circuits Syst. Mag.*, vol. 13, no. 2, pp. 89–105, 2nd Quart., 2013.

[3] D. Panda, P. P. Sahu, and T. Y. Tseng, “A collective study on modeling and simulation of resistive random access memory,” *Nanosc. Res. Lett.*, vol. 13, no. 1, pp. 8, Jan. 2018.

[4] M. Lanza *et al.*, “Recommended methods to study resistive switching devices,” *Adv. Electron. Mater.*, vol. 5, no. 1, Jan. 2019, Art. no. 1800143.

[5] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, “The missing memristor found,” *Nature*, vol. 453, pp. 80–83, May 2008.

[6] D. B. Strukov and R. S. Williams, “Exponential ionic drift: Fast switching and low volatility of thin-film memristors,” *Appl. Phys. A, Solids Surf.*, vol. 94, no. 3, pp. 515–519, 2009.

[7] M. D. Pickett, D. B. Strukov, J. L. Borghetti, J. J. Yang, G. S. Snider, D. R. Stewart, and R. S. Williams, “Switching dynamics in titanium dioxide memristive devices,” *J. Appl. Phys.*, vol. 106, no. 7, 2009, Art. no. 074508.

[8] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, “TEAM: ThrEshold adaptive memristor model,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 1, pp. 211–221, Jan. 2013.

[9] S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny, “VTEAM: A general model for voltage-controlled memristors,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 8, pp. 786–790, Aug. 2015.

[10] Z. Jiang and H.-S. P. Wong, (Oct. 2014). *Resistive-Switching Random Access Memory (RRAM) Verilog-A Model*. [Online]. Available: https://nanohub.org/publications/19/1

[11] Y. V. Pershin and M. Di Ventra, “SPICE model of memristive devices with threshold,” *Radioengineering*, vol. 22, no. 2, pp. 485–489, 2013.

[12] M. Bocquet, H. Aziza, W. Zhao, Y. Zhang, S. Onkaraiah, C. Müller, M. Reyboz, D. Deleruyelle, F. Clermidy, and J.-M. Portal, "Compact modeling solutions for oxide-based resistive switching memories (OxRAM)," *J. Low Power Electron. Appl.*, vol. 4, no. 1, pp. 1–14, 2014.

[13] A. Mehonic, S. Cueff, M. Wojdak, S. Hudziak, O. Jambois, C. Labbé, B. Garrido, R. Rizk, and A. J. Kenyon, "Resistive switching in silicon suboxide films," *J. Appl. Phys.*, vol. 111, no. 7, Apr. 2012, Art. no. 074507.

[14] G. W. Burr, B. N. Kurdi, J. C. Scott, C. H. Lam, K. Gopalakrishnan, and R. S. Shenoy, "Overview of candidate device technologies for storage-class memory," *IBM J. Res. Develop.*, vol. 52, nos. 4–5, pp. 449–464, Jul. 2008.

[15] M.-J. Lee, C. B. Lee, D. Lee, S. R. Lee, M. Chang, J. H. Hur, Y.-B. Kim, C.-J. Kim, D. H. Seo, S. Seo, U.-I. Chung, I.-K. Yoo, and K. Kim, "A fast, high-endurance and scalable non-volatile memory device made from asymmetric Ta_2O_{5-x}/TaO_{2-x} bilayer structures," *Nature Mater.*, vol. 10, no. 8, p. 625–630, 2011.

[16] R. Degraeve, A. Fantini, S. Clima, B. Govoreanu, L. Goux, Y. Y. Chen, D. J. Wouters, P. Roussel, G. S. Kar, G. Pourtois, S. Cosemans, J. A. Kittl, G. Groeseneken, M. Jurczak, and L. Altimime, "Dynamic 'hour glass' model for SET and RESET in HfO2 RRAM," in *Proc. Symp. VLSI Technol. (VLSIT)*, Jun. 2012, pp. 75–76.

[17] M.-F. Chang, "Role of field and temperature in triggering ON/OFF switching mechanisms in HF/HFO2 resistive RAM cells," *IEEE J. Solid-State Circuits*, vol. 49, no. 4, pp. 908–916, 2014.

[18] F. Nardi, S. Larentis, S. Balatti, D. C. Gilmer, and D. Ielmini, "Resistive switching by voltage-driven ion migration in bipolar RRAM—Part I: Experimental study," *IEEE Trans. Electron Devices*, vol. 59, no. 9, pp. 2461–2467, Sep. 2012.

[19] S. Larentis, F. Nardi, S. Balatti, D. C. Gilmer, and D. Ielmini, "Resistive switching by voltage-driven ion migration in bipolar RRAM—Part II: Modeling," *IEEE Trans. Electron Devices*, vol. 59, no. 9, pp. 2468–2475, Sep. 2012.

[20] Y. Nishi, S. Menzel, K. Fleck, U. Böttger, and R. Waser, "Origin of the SET kinetics of the resistive switching in tantalum oxide thin films," *IEEE Trans. Electron Devices*, vol. 35, no. 2, pp. 259–261, Feb. 2014.

[21] J. G. Simmons, "Electric tunnel effect between dissimilar electrodes separated by a thin insulating film," *J. Appl. Phys.*, vol. 34, no. 9, pp. 2581–2590, Sep. 1963, doi: 10.1063/1.1729774.

[22] G. González-Cordero, J. B. Roldan, F. Jiménez-Molinos, J. Suñé, S. Long, and M. Liu, "A new compact model for bipolar RRAMs based on truncated-cone conductive filaments—A Verilog-A approach," *Semicond. Sci. Technol.*, vol. 31, no. 11, 2016, Art. no. 115013.

[23] T. Sadi, A. Mehonic, L. Montesi, M. Buckwell, A. Kenyon, and A. Asenov, "Investigation of resistance switching in SiO_x RRAM cells using a 3D multi-scale kinetic Monte Carlo simulator," *J. Phys., Condens. Matter*, vol. 30, no. 8, 2018, Art. no. 084005.

[24] C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, "Generalized memristive device SPICE model and its application in circuit design," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 32, no. 8, pp. 1201–1214, Aug. 2013.

[25] F. M. Bayat, B. Hoskins, and D. B. Strukov, "Phenomenological modeling of memristive devices," *Appl. Phys. A, Solids Surf.*, vol. 118, no. 3, pp. 779–786, Mar. 2014. [Online]. Available: http://arxiv.org/abs/1406.4219

[26] D. Biolek, M. Di Ventra, and Y. V. Pershin, "Reliable SPICE simulations of memristors, memcapacitors and meminductors," *Radioengineering*, vol. 22, no. 4, pp. 945–968, 2013.

[27] D. Biolek, Z. Kolka, V. Biolková, Z. Biolek, M. Potrebić, and D. Tošić, "Modeling and simulation of large memristive networks," *Int. J. Circuit Theory Appl.*, vol. 46, no. 1, pp. 50–65, Jan. 2018.

[28] S. Kvatinsky, K. Talisveyberg, D. Fliter, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Verilog-A for memristor models," *CCIT Tech. Rep.*, vol. 801, pp. 1–6, Dec. 2011.

[29] D. Fliter, K. Talisveyberg, and M. Ramadan, "Manual for using memristor models," EE Dept., Inst. Technol., Haifa, Tech. Rep., Nov. 2014.

[30] B. Hajri, M. M. Mansour, A. Chehab, and H. Aziza, "Oxide-based RRAM models for circuit designers: A comparative analysis," in *Proc. 12th Int. Conf. Design Technol. Integr. Syst. Nanosc. Era (DTIS)*, Apr. 2017, pp. 1–6.

[31] A. Grossi, E. Nowak, C. Zambelli, C. Pellissier, S. Bernasconi, G. Cibrario, K. El Hajjam, R. Crochemore, J. F. Nodin, P. Olivo, and L. Perniola, "Fundamental variability limits of filament-based RRAM," in *IEDM Tech. Dig.*, Dec. 2016, pp. 4.7.1–4.7.4.

[32] E. Linn, A. Siemon, R. Waser, and S. Menzel, "Applicability of well-established memristive models for simulations of resistive switching devices," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 8, pp. 2402–2410, Aug. 2014.

[33] E. Defez, J. Sastre, J. Ibáñez, and P. A. Ruiz, "Computing hyperbolic matrix functions using orthogonal matrix polynomials," in *Progress in Industrial Mathematics at ECMI*, 1st ed. Cham, Switzerland: Springer, 2014, pp. 403–407.

[34] S. Menzel, M. Waters, A. Marchewka, U. Böttger, R. Dittmann, and R. Waser, "Origin of the ultra-nonlinear switching kinetics in oxide-based resistive switches," *Adv. Funct. Mater.*, vol. 21, no. 23, pp. 4487–4492, 2011.

[35] D. B. Strukov, J. L. Borghetti, and R. S. Williams, "Coupled ionic and electronic transport model of thin-film semiconductor memristive behavior," *Small*, vol. 5, no. 9, pp. 1058–1063, 2009.

[36] I. Vourkas and G. Sirakoulis, "Memristor modeling," in *Memristor-Based Nanoelectronic Computing Circuits and Architectures*, vol. 19, 1st ed. Springer, 2016, pp. 9–28.

[37] Y. V. Pershin and M. Di Ventra, "Memory effects in complex materials and nanoscale systems," *Adv. Phys.*, vol. 60, no. 2, pp. 145–227, 2011.

[38] P. Pouyan, E. Amat, S. Hamdioui, and A. Rubio, "RRAM variability and its mitigation schemes," in *Proc. 26th Int. Workshop Power Timing Modelling, Optim. Simulation (PATMOS)*, Sep. 2016, pp. 141–146.

[39] F. M. Puglisi, N. Zagni, L. Larcher, and P. Pavan, "Random telegraph noise in resistive random access memories: Compact modeling and advanced circuit design," *IEEE Trans. Electron Devices*, vol. 65, no. 7, pp. 2964–2972, Jul. 2018, doi: 10.1109/TED.2018.2833208.

[40] C. Y. Chen, L. Goux, A. Fantini, A. Redolfi, S. Clima, R. Degraeve, Y. Y. Chen, G. Groeseneken, and M. Jurczak, "Understanding the impact of programming pulses and electrode materials on the endurance properties of scaled Ta2O5 RRAM cells," in *IEDM Tech. Dig.*, San Francisco, CA, USA, Dec. 2014, pp. 14.2.1–14.2.4.

[41] S. Choi, J. Lee, S. Kim, and W. D. Lu, "Retention failure analysis of metal-oxide based resistive memory," *Appl. Phys. Lett.*, vol. 105, no. 11, 2014, Art. no. 113510.

[42] S. Yu, Y. Y. Chen, X. Guan, H.-S. P. Wong, and J. A. Kittl, "A Monte Carlo study of the low resistance state retention of HfO_x based resistive switching memory," *Appl. Phys. Lett.*, vol. 100, no. 4, 2012, Art. no. 043507.

[43] C. Y. Chen, A. Fantini, L. Goux, R. Degraeve, S. Clima, A. Redolfi, G. Groeseneken, and M. Jurczak, "Programming-conditions solutions towards suppression of retention tails of scaled oxide-based RRAM," in *IEDM Tech. Dig.*, Dec. 2015, pp. 10.6.1–10.6.4.

[44] R. Liu, D. Mahalanabis, H. J. Barnaby, and S. Yu, "Investigation of single-bit and multiple-bit upsets in oxide RRAM-based 1T1R and crossbar memory arrays," *IEEE Trans. Nucl. Sci.*, vol. 62, no. 5, pp. 2294–2301, Oct. 2015.

[45] A. G. Radwan and M. E. Fouda, *On the Mathematical Modeling of Memristor, Memcapacitor, and Meminductor*, vol. 26. New York, NY, USA: Springer, 2015, pp. 18–19.

[46] P. Huang, X. Y. Liu, B. Chen, H. T. Li, Y. J. Wang, Y. X. Deng, K. L. Wei, L. Zeng, B. Gao, G. Du, X. Zhang, and J. F. Kang, "A physics-based compact model of metal-oxide-based RRAM DC and AC operations," *IEEE Trans. Electron Devices*, vol. 60, no. 12, pp. 4090–4097, Dec. 2013.

[47] F. Jiménez-Molinos, M. A. Villena, J. B. Roldán, and A. M. Roldán, "A SPICE compact model for unipolar RRAM reset process analysis," *IEEE Trans. Electron Devices*, vol. 62, no. 3, pp. 955–962, Mar. 2015.

[48] U. Russo, D. Ielmini, C. Cagli, and A. L. Lacaita, "Filament conduction and reset mechanism in NiO-based resistive-switching memory (RRAM) devices," *IEEE Trans. Electron Devices*, vol. 56, no. 2, pp. 186–192, Feb. 2009.

[49] U. Russo, D. Ielmini, C. Cagli, and A. L. Lacaita, "Self-accelerated thermal dissolution model for reset programming in unipolar resistive-switching memory (RRAM) devices," *IEEE Trans. Electron Devices*, vol. 56, no. 2, pp. 193–200, Feb. 2009.

[50] M. Uenuma, Y. Ishikawa, and Y. Uraoka, "Joule heating effect in nonpolar and bipolar resistive random access memory," *Appl. Phys. Lett.*, vol. 107, no. 7, 2015, Art. no. 073503.

[51] X. Guan, S. Yu, and H.-S. P. Wong, "A SPICE compact model of metal oxide resistive switching memory with variations," *IEEE Electron Device Lett.*, vol. 33, no. 10, pp. 1405–1407, Oct. 2012.

[52] M. Bocquet, D. Deleruyelle, H. Aziza, C. Müller, J.-M. Portal, T. Cabout, and E. Jalaguier, "Robust compact model for bipolar oxide-based resistive switching memories," *IEEE Trans. Electron Devices*, vol. 61, no. 3, pp. 674–681, Mar. 2014.

[53] Z. Jiang, P. Huang, L. Zhao, S. Kvatinsky, S. Yu, X. Liu, J. Kang, Y. Nishi, and H.-S. P. Wong, "Performance prediction of large-scale 1S1R resistive memory array using machine learning," in *Proc. IEEE Int. Memory Workshop (IMW)*, May 2015, pp. 1–4.

[54] T. Wang and J. Roychowdhury, "Well-posed models of memristive devices," May 2016, *arXiv:1605.04897*. [Online]. Available: https://arxiv.org/abs/1605.04897

BASMA HAJRI received the M.S. degree in electrical engineering from the Institut National Polytechnique de Grenoble (INPG), Grenoble, France, in 2006, and the Research master's degree in nanotechnologies from University Joseph Fourier, Grenoble, in 2006. She is currently pursuing the Ph.D. degree in electrical engineering in joint supervision with Aix-Marseille University and the American University of Beirut. From 2006 to 2007, she worked with the Design Kit Model Team, Research and Development Unit, Freescale, Grenoble. From 2007 to 2010, she worked with the Standard Cell Layout Team, Research and Development Unit, ARM, Grenoble. Since 2010, she has been an Instructor with the Electrical Engineering Department, American University of Beirut. She is also the coauthor of two conference papers, one journal, and one submitted U.S. patent.

HASSEN AZIZA received the B.S. and M.S. degrees in electrical engineering and the Ph.D. degree from the University of Marseille, France, in 2002. He is currently an Associate Professor with the IM2NP Laboratory, Institute of Materials, Microelectronics and Nanosciences in Provence, Aix-Marseille University. He is the (co)author of more than 100 articles in international conferences and journals. He is also a (co)inventor of four patents. His research fields include design, test, and reliability of conventional non-volatile memories (Flash and EEPROM), as well as emerging memories (resistive RAM).

MOHAMMAD M. MANSOUR (S'97–M'03–SM'08) received the B.E. degree (Hons.) and the M.E. degree in computer and communications engineering from the American University of Beirut (AUB), Beirut, Lebanon, in 1996 and 1998, respectively, and the M.S. degree in mathematics and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign (UIUC), Champaign, IL, USA, in 2002 and 2003, respectively. He was a Research Assistant with the Department of Electrical and Computer Engineering, AUB, in 1997, and a Teaching Assistant, in 1996. He was with the Wireless Research Group, National Semiconductor Corporation, San Francisco, CA, USA, in 2000. He was a Research Assistant with the Coordinated Science Laboratory, UIUC, from 1998 to 2003. He was on research leave with Qualcomm Flarion Technologies, Bridgewater, NJ, USA, from 2006 to 2008, where he was involved in modem design and implementation for 3GPPLTE, 3GPP2-UMB, and peer-to-peer wireless networking physical layer SoC architecture and algorithm development. He was a Visiting Researcher with Broadcom, Sunnyvale, CA, USA, from 2012 to 2014, where he was involved in the physical layer SoC architecture and algorithm development for LTE-advanced baseband receivers. He was a Visiting Researcher with Qualcomm, San Jose, CA, USA, in 2016, where he focused on baseband receiver architectures for the IEEE 802.11ax standard. He joined as a Faculty Member with the Department of Electrical and Computer Engineering, AUB, in 2003, where he is currently a Professor. He has seven issued U.S. patents. His research interests are in the area of energy efficient and high-performance VLSI circuits, architectures, algorithms, and systems for computing, communications, and signal processing. He is also a member of the Design and Implementation of Signal Processing Systems (DISPS) Technical Committee Advisory Board of the IEEE Signal Processing Society. He has served as a member for the DISPS Technical Committee, from 2006 to 2013. He has served as the Technical Co-Chair for the IEEE Workshop on Signal Processing Systems, in 2011, and a member of the Technical Program Committee of various international conferences and workshops. He was a recipient of the Phi Kappa Phi Honor Society Award twice, in 2000 and 2001, and the Hewlett Foundation Fellowship Award, in 2006. He has served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, from 2008 to 2013, the IEEE SIGNAL PROCESSING LETTERS, from 2012 to 2016, and the IEEE TRANSACTIONS ON VLSI SYSTEMS, from 2011 to 2016.

ALI CHEHAB (S'98–M'02–SM'06) received the bachelor's degree in electrical engineering from the American University of Beirut (AUB), in 1987, the master's degree in electrical engineering from Syracuse University, in 1989, and the Ph.D. degree in electronics and communication engineering from the University of North Carolina at Charlotte, in 2002. From 1989 to 1998, he was a Lecturer with the Electronics and Communication Engineering Department, AUB, where he rejoined as an Assistant Professor, in 2002, and became a Full Professor, in 2014. He has over 200 publications. He teaches courses in programming, electronics, digital systems design, computer organization, cryptography, and digital systems testing. His research interests include cryptography, wireless communications security, cloud computing security, multimedia security, trust in distributed computing, low energy VLSI design, and VLSI testing. He is a Senior Member of ACM. He received the AUB Teaching Excellence Award, in 2007.

• • •