2012 International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)

# A CAD Framework for the Characterization and Use of Memristor Models

Fernando García-Redondo, Marisa López-Vallejo, Pablo Ituero, Carlos López Barrio

*Abstract*— **In the recent years the missing fourth component, the memristor, was successfully synthesized. However, the mathematical complexity and variety of the models behind this component, in addition to the existence of convergence problems in the simulations, make the design of memristor-based applications long and difficult. In this work we present a memristor model characterization framework which supports the automated generation of subcircuit files. The proposed environment allows the designer to choose and parameterize the memristor model that best suits for a given application. The framework carries out characterizing simulations in order to study the possible non-convergence problems, solving the dependence on the simulation conditions and guaranteeing the functionality and performance of the design. Additionally, the occurrence of undesirable effects related to PVT variations is also taken into account. By performing a Monte Carlo or a corner analysis, the designer is aware of the safety margins which assure the correct device operation.**

## I. Introduction

In 1971 Leon Chua postulated the existence of a fourth component that would complete the set of classical circuit elements consisting of resistors, capacitors and inductors [1]. The *memristor* device relates the flow and the charge by the memristor representative magnitude, the *memristance*.

After decades of study of many memristance behaviors, in 2008 the Hewlett Packard Laboratories presented a physical device that followed the equations of the memristor [2], changing its internal resistance according to the current that flows through the component. Following HP memristor fabrication numerous research groups have been able to manufacturate similar devices [3]–[5].

Due to its reduced size, memristors can be used to replace some CMOS structures, decreasing the used area and entailing a lower power consumption. Both area and power consumption are critical factors in electronics nowadays, thus, the memristor is gathering much attention.

Important efforts have been made on providing different memristor models that allow their study and simulation. There are models designed for dynamic system analysis and numerical computer environments [6] which perform the analysis of a single device, leaving the complete circuit out of study. On the other hand, the implementation of memristors in circuit simulators [7], [8] leaves in the simulator hands some of the memristor main characteristics such as the fast-switching conditions at the beginning of the simulation. Additionally, the high non-linearity of the memristor demands that, apart from the inherited characteristics of the model, we should take into account several parameters pertaining to the simulator that can affect the computation. Therefore, a complete analysis and simulation of memristive devices is needed.

In this paper we present the development of the first automated tool for memristor model characterization and subcircuit generation, Memristor Application Framework, MAF; whose main contributions are:

- Implementation of the most important and referenced models in literature [2], [8], [10]–[14].
- Characterization of the memristor response to a given input, analysis of the simulations convergence problems and study of the device data storage speed.
- Given a circuit with voltage feeding and timing constrains, the framework aids the designer to choose the memristor model and parameters that fulfill the system constraints.
- Automated generation of subcircuit and data files. This allows the interaction with the desired circuit simulator and external mathematical software, acting as a quick validation and time saving tool.
- Capability of handling PVT variations, studying scenarios which affect the memristor behavior.

The paper is structured as follows. First, the proposed approach is overviewed paying special attention to the designer needs. The design methodology is presented in section III, followed by the library of implemented memristor models. Section IV includes a complete description of the framework, focusing on the two main modules: *Device Characterization* and the *Multi-level Characterization*. Afterwards, the implemented techniques that handle process variations are explained. Finally, section V presents a use example and draws some concluding remarks.

## II. Approach Overview

From the point of view of a memristor-based application designer, a complete set of support tools would be desirable. The wishlist for this framework is the following:

1) As a novel technology, the immaturity of memristor models can lead to non-convergent simulations, or designs that do not meet timing or power specifications. Support to deal with these problems must be provided.

2) The variety and complexity of the models makes it a hard decision to choose the most suitable one for a given application. The framework should be able to work with as many of those different models as possible.
3) There is a wide set of parameters that constrain the configuration of the desired memristor model. It would be desirable to analyze the impact of all these parameters.
4) In the case of using the non-linear capabilities of a particular memristor, the comprehension of its dynamic behavior under some defined stimuli is required.
5) Supposing that the design implies multilevel storage [9], the high accuracy required at the magnitude margins such as voltage, time or memristance, makes the design process arduous and laborious. Automated computation of these boundary levels is highly desirable.
6) The valid range of the model variables considering PVT variations should be guaranteed.
7) Finally, there should be interaction between the support framework and the final circuit simulator software or the numerical computer environment used, simplifying the design process and saving time.

Our proposed support framework and design methodology have been developed to solve these demands.

## III. Design Methodology

Figure 1 illustrates the proposed design methodology. In the first stage, external to the framework, the constraints that the memristors must satisfy, like the state change speed at some supply voltage, are specified. This step corresponds to stage 1 in the diagram. The designer should select the memristor to be analyzed from the memristor model library (stage number 2 in the figure), set up the instance and parameterize the input voltage and simulation parameters. As soon as the scenario is set, the designer is able to perform the device characterization (stage 3 in the diagram), accomplishing a transient simulation which follows the device behavior equations. If this characterization does not converge, the analysis conditions should be refined. Otherwise, if the analysis converges, the designer will be able to study the memristor response to the selected input using the included graphical representation tools or handling the exported data results into external software such as GnuPlot or Matlab. As a result, the designer is allowed to study the device voltage, current, memristance and the model state variables in a particular scenario.

If the result of the simulation is satisfactory, the designer can proceed to perform the level characterization, number 4 in the diagram. In this module the designer can evaluate the evolution of the different states of the memristor under different voltage stimuli. This way the framework characterizes (if required) the multi-level storage of data into the memristor. The study of the time required to perform a state change under an input voltage is especially important. Furthermore, the designer can analyze the memristor behavior by carrying out a full Monte Carlo or a corner transient analysis. One of the main contributions of the software is that it allows varying as many parameters as desired. Once again, if the simulation converges, the designer

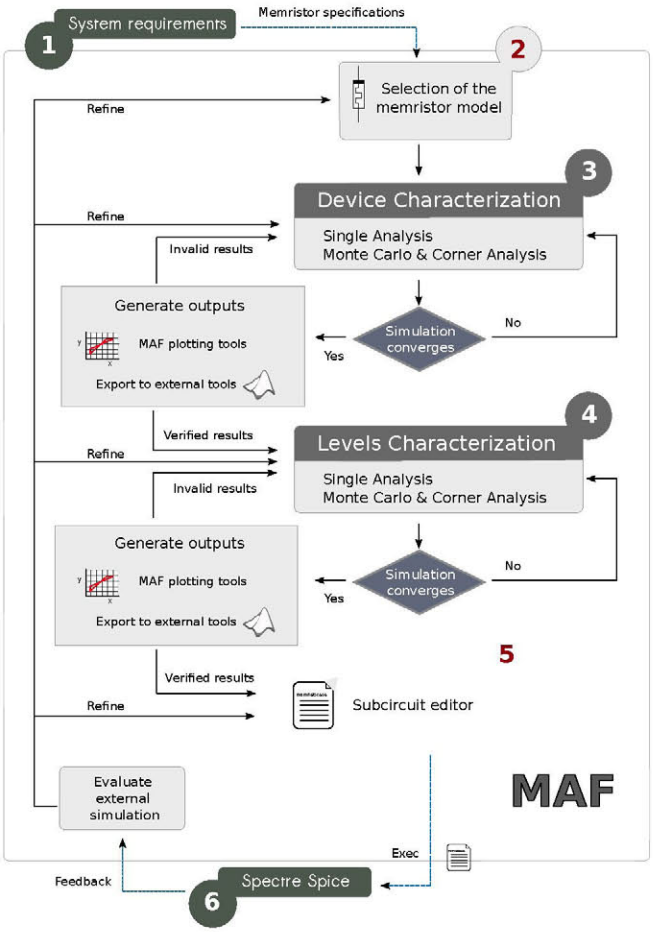


Fig. 1. Design Methodology for memristor-based applications.

will be able to examine the results. The designer can now evaluate if the memristor is fast enough, or if some parameter variation disrupts the operation of the device.

When the circuit requirements are satisfied, a Spice / Spectre sub-circuit netlist of the memristor is generated, allowing the designer to customize the circuit netlist and execute the simulation in the external circuit simulation (stages 5 and 6).

## IV. Framework Description

In this section, the main characteristics of the MAF are presented.

### A. Library of Memristor Models

A key point for the framework is the set of models included in the library. The most referenced models in literature have been included in the proposed framework.

- *$Rm(q)$ and Linear ion drift [2].* Generic charge controlled memristor whose memristance depends on the charge in a polynomial way. The Linear ion drift model [2] can be expressed through these polynomial equations.
- *Non-linear ion drift using different window functions [8], [10], [11].* Most of the memristor based circuit applications are developed using this model.

TABLE I
MEMRISTOR MODELS SUMMARY.

| Model | Accuracy | Computational load | Convergence problems |
|---|---|---|---|
| $Rm(q)$ **and Linear** ion drift [2] | Low | Low | Non-existent & Very few |
| **Non-linear ion drift [8], [10], [11]** | Low | Low | Non-existent & Very few |
| **Simmons tunneling barrier [12]** | Highest | Highest | Numerous |
| **Yakopcic [13]** | Medium & High | Medium & Moderate | Possibles |
| **TEAM [14]** | Medium & High | Medium & Moderate | Possibles |

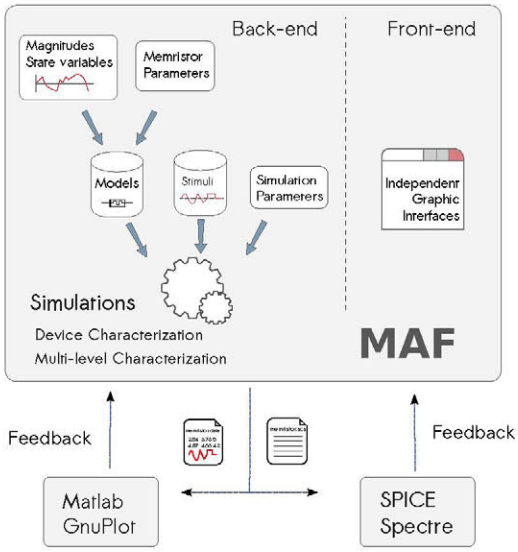


Fig. 2. MAF structure and integration among platforms.

- *Simmons tunneling barrier model [12]* is the most complete memristor model, built as a Metal/Insulator/Metal (MIM) junction, combined with a series resistance.
- *Yakopcic's model [13]* proposes an exponential drift model which combines performance and accuracy.
- *TEAM model [14].* The latest model appeared in the literature derives from the Simmons tunneling model.

Table I summarizes the main simulation characteristics of the models included in the MAF library, regarding accuracy, computational load and possible convergence problems.

### B. Framework Main Modules

Figure 2 presents an overview of the MAF. All the modules have been programmed to act as independent functional units which interact exchanging data models and scenarios. The figure shows the relations between circuit simulators, numerical computing software and our tools.

Inputs to the framework are the memristor models, the input stimuli and the parameters which specify the simulation conditions. As shown, our software interacts and exports data and netlists to circuit simulators and numerical computation software. Additionally, the outputs provided by these external programs are fed-back to our framework, allowing the designer to change the simulation parameters until the expected and validated results are reached.

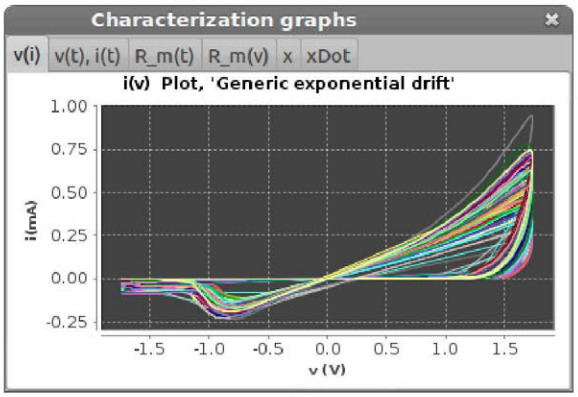


Fig. 3. Device characterization Monte Carlo simulation, $i(v)$ curve tab.

It is likely that new memristor devices, and their related models, will be presented in the future. The inclusion of these memristors in the models library is a key point in the extensibility of the framework. To deal with this problem, the MAF engine has been carefully designed as a modular system to allow the scalability and extensibility of its functionalities. Figure 2 also represents the different software modules.

In the next sections we describe the three main modules:

*1) Device Characterization Module:* Some memristor models require smaller time steps during the simulation in order to obtain valid results [15]. The computation of some variables needs time-integration, so to avoid computation overflow we must guarantee that the time step is small enough to converge. This module provides the environment to study the behavior of the models included in the database once excited by different patterns of voltage, guarantying the viability of the memristor behavior and the simulation convergence. The module performs a comprehensive analysis of the scenario chosen by the designer using the memristor model equations. As a result, the designer is allowed to study the device electrical magnitudes and the model state variables in the simulated scenario. Figure 3 displays the graphical representation of the illustrative $i(v)$ curve at the end of a sine-excited memristor characterization.

*2) Multi-level Characterization Module:* Given the highly non-linear flux dependency of the memristor, the designer needs high accuracy in the flux value and related parameters. Thus, to perform the multilevel encoding and storage of each specific value in the memristor, the user has to accurately know the amplitude and duration of this input signal. This pulse length will depend on the memristor speed.

For this reason, one of the main issues when designing memristive applications is to determine the device speed given a supply voltage. The *Multi-level Characterization Module* provides, for a configurable device, a comprehensive study of the relationship memristance-flow when storing a fixed number of values ??in a single memristor.

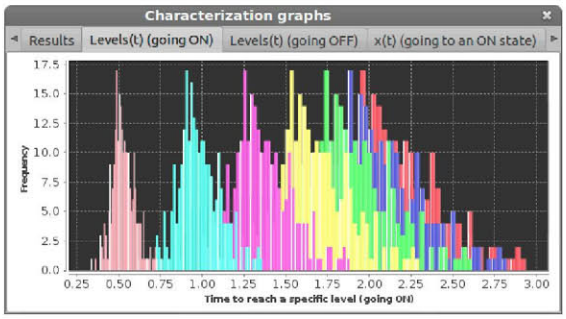


Fig. 4. Multi-level characterization including Monte Carlo simulation. Time-to reach specific levels histogram, in an scenario with parameter variations.

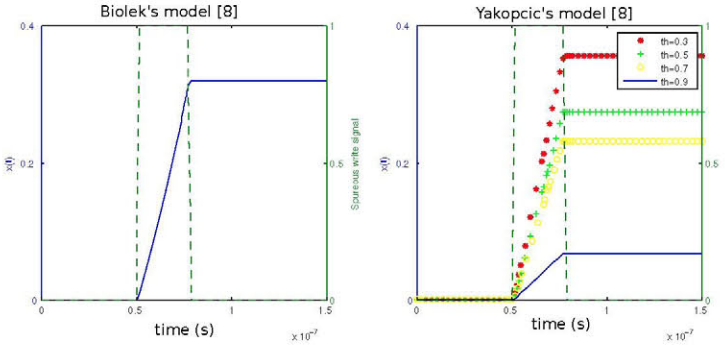


Fig. 5. State variable evolution in adjacent RRAM cells.

The designer can study the evolution of the model state variable and the memristance over time. Also, the tool allows calculating the pulse lengths required to store the desired value.

*3) Subcircuit Editor Module:* This module provides a mechanism to communicate with external circuit simulators, loading the customized parameters brought from simulated memristor and creating the appropriate subcircuit. In that stage, the user is able to modify the parameters, include additional components or change the subcircuit structure.

### C. PVT Variations

PVT variations can destabilize the memristor behavior. As an added value, the main MAF modules provide the functionality to perform the study of the dynamics of a specific memristor in presence of parameters variations. Therefore, instead of a single scenario, several simulations are executed in parallel, performing a deeper analysis.

Considering PVT variations, two analysis techniques have been included: *Corner analysis* and *Monte Carlo* analysis. Both of them play a key role in the circuit validation. Voltage, temperature (depending on the memristor model) and model inherent parameter variations are considered. Figure 4 displays the Monte Carlo simulation results for an 8-level memristor operation. The designer is able to acquire mean values and deviations, while the histogram helps in the selection of the safety margins.

## V. Results and Conclusions

Several examples have been simulated to validate the proposed framework and methodology. For instance, the MAF has been helpful in the study of the threshold behavior and its impact on the performance of crossbar Resistive RAMs with different memristor models. Figure 5 shows the non-desired modification of the previously stored data in contiguous cells to the one which is going to be written, and its relation with the threshold presence. The example displays the Cadence Spectre® output of the simulation of a subcircuit generated in MAF, where the adjacent cell state variable $x(t)$, directly related to the stored value, is unwittingly modified.

The presented framework stands as a support tool that complements and helps in the usual circuit design work flow. Given the relevance of the memristor in the future of electronics, the developed framework is expected to be a useful tool to the research community, saving time and easing the design flow.

With our environment, the designer is able to carry out a design space exploration of the response of the main memristor models in the desired scenario, using the results in a complete circuit simulation. Moreover, the developed Monte Carlo and corner analysis simulations check whether the devices work when considering manufacturing tolerances for the components, temperature range and voltage fluctuations. The results obtained from the execution of analysis and simulations provide valuable information for the design and development of memristor-based circuits.

Finally, the modularity of the framework allows the inclusion of plug-ins and model extensions in the future, ensuring and expanding its usefulness.

## References


[1] L. O. Chua, "Memristor, The Missing Circuit Element," *IEEE Transactions on Circuit Theory*, vol. 18, 1971.

[2] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found." *Nature*, vol. 453, no. 7191, May 2008.

[3] J. Borghetti, et al."Memristive switches enable 'stateful' logic operations via material implication," *Nature*, vol. 464, no. 7290, 2010.

[4] O.Jeong-Hoon et al. "An access-transistor-free resistive random access memory (RRAM) using a GST/TiO2 stack and its novel access mechanism," in *EDSSC 2010 IEEE*, 2010.

[5] N. Gergel-Hackett, B. Hamadani et al."A Flexible Solution-Processed Memristor," *IEEE Electron Device Letters*, vol. 30, no. 7, Jul. 2009.

[6] K. Zaplatilek, "Memristor modeling in MATLAB ® &Simulink ®," in *Proc. of the 5th European Computing Conference*, 2011, pp. 62–67.

[7] A. Rak and G. Cserey, "Macromodeling of the Memristor in SPICE," *IEEE TCAD* , vol. 29, no. 4, Apr. 2010.

[8] Z. Biolek, D. Biolek, and V. Biolková, "SPICE Model of Memristor with Nonlinear Dopant Drift," *Radioengineering*, vol. 18, no. 2, 2009.

[9] H. Kim, et al. "Memristor-based multilevel memory," *12th International Workshop on Cellular Nanoscale Networks and their Applications*, 2010.

[10] Y. N. Joglekar et al. "The elusive memristor: properties of basic electrical circuits," *European Journal of Physics*, vol. 30, no. 4, Jul. 2009.

[11] T. Prodromakis et al. "A Versatile Memristor Model With Nonlinear Dopant Kinetics," *IEEE Trans. on Electron Devices*, vol. 58, no. 9, 2011.

[12] M. D. Pickett, et al. "Switching dynamics in titanium dioxide memristive devices," *Journal of Applied Physics*, vol. 106, no. 7, 2009.

[13] C. Yakopcic et al. "A Memristor Device Model," *IEEE Electron Device Letters*, vol. 32, no. 10, 2011.

[14] S. Kvatinsky, et al. "TEAM: ThrEshold Adaptive Memristor Model," *IEEE Trans. on Circuits and Systems I: Regular Papers*, 2012.

[15] H. Abdalla and M. D. Pickett, "SPICE Modeling of Memristors," in *IEEE International Symposium on Circuits and Systems (ISCAS), 2011.*