

BEhavioral Leakage and IntEr-cycle Variability Emulator model for ReRAMs (BELIEVER)

David Radakovits and Nima Taherinejad

arXiv:2103.04179v1 [cs.ET] 6 Mar 2021

Abstract—Emerging electronic devices are promising to drive the performance of computer systems to new heights, against the notable saturation in traditional transistor-based architectures. Among them, resistive RAM – or ReRAM – has attracted a lot of attention among scientists since its practical realization was reported in 2008 and numerous devices, circuits and systems, and also models have been described in the literature. However, behavioral models fail to reproduce device parameter variations and the drift of device state in the absence of a stimulus. This shortcoming substantially reduces the practical relevance of systems and circuits designed with existing models. The work at hand deals with the development of a behavioral model that integrates device parameter variation and state drift based on data collected from our measurements of real devices. As we show in this paper, BELIEVER model enables engineers to conduct more reliable and meaningful design and simulations of circuits and systems that use ReRAMs.

I. INTRODUCTION

Requirements for computer systems are in an ever increasing growth, especially today, where Big Data applications [1]–[4] on the one hand and the Internet of Things (IoT) on the other hand span a wide spectrum of requirements from high performance to low energy consumption and footprint. Interestingly both ends of the spectrum are still served by computer systems that are more or less based on the same architecture, namely the Von Neumann-architecture. A well-established technique to alleviate the Von Neumann-bottleneck is caching, where memory elements are located near the Central Processing Unit (CPU). In-Memory Computation (IMC) goes the other way round, i.e., installing elements in the memory, which are capable of doing logic operations, without the data leaving the memory [5]–[10].

The memristor is a basic element in electrical circuits, along with resistor, capacitor and inductor. Its theoretical behavior was first described in 1971 by Leon Chua [11] and in 2008 the first passive realization of a device that behaves like a memristor was reported [12]. Although memristive effects have been observed before without recognizing them as such [13], [14], this device was the first to intentionally behave like a memristor in terms of voltage-current relationship over time. The memristors that have been studied in this work, are of the Self Directed Channel (SDC) type [15], which are called resistance switching devices or ReRAMs [16]. The aforementioned memristor reported in [12] and many

D. Radakovits and N. TaheriNejad are with the TU Wien, 1040 Vienna, Austria

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

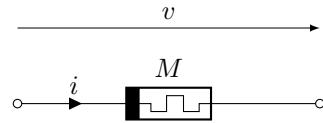


Fig. 1. Memristor symbol with reference direction for the current and voltage.

others [17]–[21] are also of the ReRAM type. Although in the community the term *memristor* is established as an umbrella term for different kinds of memristive devices apart from ReRAMs, e.g., Spin Transfer Torque (STT) or Phase Change Memory (PCM) [22], [23], whenever the term is used in this work, it refers to *resistance switching device*. The commonly used memristor symbol and its reference directions in terms of current and resistance change is shown in Figure 1.

An important step in hardware development is simulation. Models and modeling techniques such as Simulation Program With Integrated Circuit Emphasis (SPICE) have evolved in the last decades and make simulation and design of complex hardware possible throughout different levels of abstraction. While modeling of well-explored technologies such as Complementary Metal-Oxide Semiconductor (CMOS) has gained a level, in which engineers can rely on simulation results with very little probability of false predictions, this is not the case for emerging technologies such as memristors. As it was shown in [24], simulations of memristive circuits and systems are not backed up by practical implementations and measurements in a vast portion of cases. Figure 2, which was adopted from [24], illustrates, that only 12% of the top 30 Google Scholar search results on ‘memristor’, ‘ReRAM’, ‘resistance

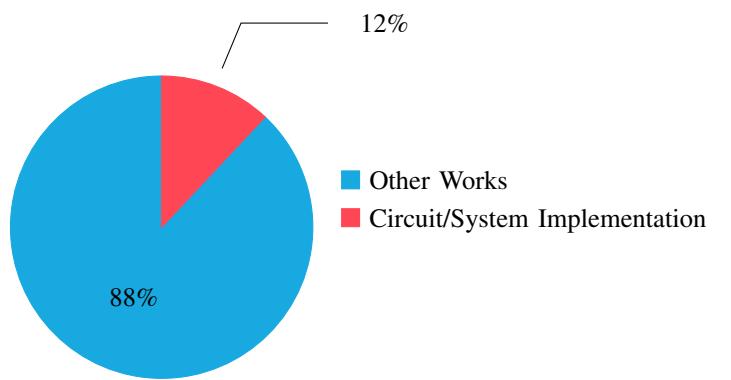


Fig. 2. Share of practically implemented and measured memristive circuits and systems from Google Scholar search results on memristors, ReRAM, resistance switch and other related keywords [24].

switch' and other keywords, actually report implementations and measurements of memristive circuits and systems.

This circumstance is quite alarming, since the available models do not reproduce cycle-to-cycle or device-to-device variations. Leakage, i.e., a change of state in absence of a stimulus, is not reproduced in models. While these mostly unwanted effects are well-explored in traditional technologies and their respective models (such as CMOS), the memristor community still needs to close this gap. To increase the simulation reliability of memristive circuits and systems, it is important to reproduce such effects in models.

Several memristor models are available in the literature. Generally they can be divided into two major groups: Those which model memristor behavior on a physical or electrochemical level, e.g., [25], and those which model macroscopic voltage-current behavior on a high level of abstraction, e.g., [26]–[28]. While the latter group naturally lacks exact reproduction of memristor internals, it has advantages, such as lower computational cost and a certain flexibility to be used among devices with similar functional internal behaviors (e.g., different resistance switching devices), given appropriate model fitting. Since there is no physical or chemical model available for the memristors investigated in this work and considering the mentioned advantages, the latter group was examined. However, to the best of the authors knowledge, none of the models reproduce device variability or leakage effects. Therefore, this work shall accomplish two major tasks: Fitting a memristor model, that is well received by the memristor community to SDC memristor devices built by KNOWM Inc., as well as incorporating variation and leakage effects into the mentioned model.

The designed experiments, the forming of the memristor under test and the outcome of the conducted experiments are clarified in Section II. The base model that was chosen to build the enhanced model upon, the model development process and the fitting of the enhanced model are explained in detail in Section III. The impact of the observed and modeled effects on popular memristive logics is showcased in Section V. The work at hand is concluded and an outlook to potential future works is given in Section VI.

II. EXPERIMENTS

The extraction of the necessary memristor parameters demands several different experiments to be conducted. The purpose, design and outcome of each experiment is explained in this chapter.

A. Forming

Before an SDC memristor can be used, it has to be *formed*. Forming is the process of initial creation of the conducting paths in the filament of the memristor. In order to form the device, as recommended by the manufacturer [15], a sinusoidal voltage with a frequency of 100 Hz is applied to the memristor. The amplitude is gradually increased until the typical hysteresis can be observed. Figure 3 shows the hysteresis of the memristor, which is the evolution of the voltage-current curve of the memristor, during forming. The color of the data

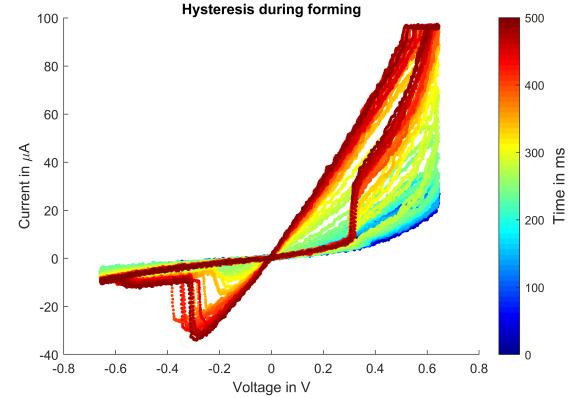


Fig. 3. Evolution of the voltage-current curve (hysteresis) of the memristor during forming. The color of the data points indicates the time, i.e., blue indicates early measurement points, red indicates late points.

points indicates the time, i.e., blue indicates early measurement points, red indicates late points. As it can be seen from the flattening of the curve in the upper right corner of the figure, the forming, as all other experiments, was conducted with a current limit set to approximately 100 μ A. The data sheet for the SDC memristors provided by the manufacturer [15] gives a maximum current of 1 mA. However, because Gomez et al. used 800 μ A with the same device in [29] and reported a very short life time for the devices, this large safety margin was chosen. Note that the current limit during forming has a large impact on memristor parameters, especially, but not exclusively on the High Resistance State (HRS) and Low Resistance State (LRS) resistance. As it can be seen in the memristor datasheet [15], HRS resistance, LRS resistance and also the span of the two decreases with increasing current limit during forming process.

B. Resistance Variation

The variation of the resistance that is measured after the SET(RESET) stimulus, i.e., the stimulus that is supposed to drive the memristor into LRS(HRS), is of great interest for the designers of a memristive circuit, since in most cases the circuit will need to be tuned to that parameter. In order to measure the cycle-to-cycle variation of the resistance after SET (R_{on}) and RESET (R_{off}), the stimulus in Figure 4 was applied 100 times to the memristor under test. As it can be seen from the orange graph in the figure, denoting the applied voltage, first a 1 ms wide 500 mV SET pulse was used to drive the memristor into LRS. The drop of the memristor's resistance can be seen in the increasing current in the blue graph. The resistance of the memristor after the SET pulse was determined during the 200 μ s wide 50 mV measurement pulse that was applied afterwards. After measuring R_{on} this way, a 1 ms wide -1 V RESET pulse was applied to the memristor in order to drive it to HRS and the resistance in HRS was measured with a 200 μ s wide 50 mV measurement pulse again. Theoretically the resistance of the memristor in LRS(HRS) could have been measured at the end of each SET(RESET) pulse. This would however hide a potential non-linear voltage-current relationship. If the memristor behaved

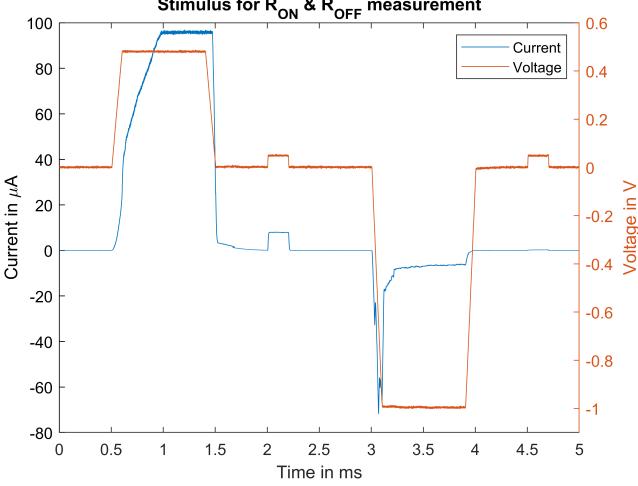


Fig. 4. SET and RESET stimulus for measurement of R_{on} and R_{off} variation. The orange graph denotes the voltage that was applied to the memristor and the blue graph denotes the memristor current.

like an ideal, i.e. linear, resistor while the state is constant, any given measurement voltage would result in the same resistance. Since it is not known, if the memristor does behave linearly, R_{on} and R_{off} have to be measured using the same voltage. The measurement voltage of 50mV was chosen upon the experience gained during forming: As it can be seen in Figure 3 the resistance starts to change significantly around 300 mV during SET and around -150 mV during RESET. Any voltage in between that range would theoretically qualify as a measurement voltage. Too small measurement voltages, however, lead to very low currents, which decrease the Signal to Noise Ratio (SNR) of the measurement. Note, that the plateau in the current during SET in Figure 4 does not indicate that the resistance of the memristor does not change anymore. The current saturation is caused by the current limit set to approximately $100 \mu\text{A}$. However, in Section II-C it is shown that, at the used amplitude, the resistance change saturates after approximately 1 ms, making the described stimulus suitable to drive the memristor into full LRS. During RESET the current saturates in the first half of the applied pulse, ensuring that the memristor is fully driven into HRS.

Figure 5 shows an example of the conducted experiment. As it can be seen in the figure, the resistance in both R_{on} and R_{off} varies around a mean value. The mean value of R_{on} (depicted in the figure in red) is $4.64 \text{ k}\Omega$, the mean value of R_{off} (depicted in the figure in blue) is $545.52 \text{ k}\Omega$. Figures 6 and 7 show the histogram of the recorded values of R_{off} and R_{on} , respectively. The standard deviation is $\sigma = 427.9 \Omega$ for R_{on} and $\sigma = 77.095 \text{ k}\Omega$ for R_{off} . A normal or gaussian distribution was chosen for both HRS and LRS for homogeneous and thus more convenient implementation of the variation model.

C. Resistance Change Dynamics

To determine the speed of state change during the SET process, an experiment was conducted repeatedly, that consisted of a series of eight SET pulses and a RESET pulse. Each SET and RESET pulse was followed by a measurement pulse

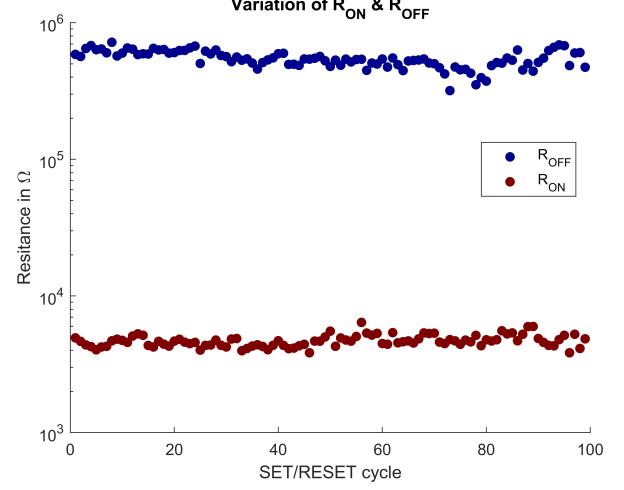


Fig. 5. Example of variation of R_{on} after SET and R_{off} after RESET stimulus. Blue dots denote the measured resistance after a RESET stimulus, red dots denote the measured resistance after the SET following the RESET stimulus.

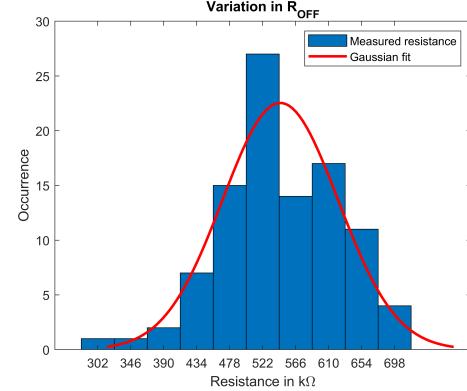


Fig. 6. Histogram of measured R_{off} values and fitted normal distribution: $\mu = 545.52 \text{ k}\Omega$, $\sigma = 77.065 \text{ k}\Omega$.

during which the resistance of the memristor at that instance was determined. The SET pulses measured $500 \text{ mV} \times 100 \mu\text{s}$, the measurement pulses measured $50 \text{ mV} \times 200 \mu\text{s}$ and the RESET pulse measured $-1 \text{ V} \times 2 \text{ ms}$, as it can be seen in Figure 8. Figure 9(a) shows the recorded resistance after each of the eight applied SET pulses for all 100 measurements, in which each line denotes the evolution of the resistance as it is influenced by the eight applied SET pulses. Figure 14 shows the resistance (in blue) after each SET pulse averaged over all 100 measurements. Using this data, the resistance change per applied stimulus of a given model can be fitted to the measured memristor behavior.

The RESET process dynamics could not be examined in the same fashion as the SET process, because it was not possible to find an appropriate stimulus for a similar, pulsed measurement. That is, no configuration of pulse length and height with the described setup could be found, that would switch the memristor only partially instead of full HRS. Hence, the aforementioned RESET pulse that was used to reset the memristor after the applied series of eight SET pulses was

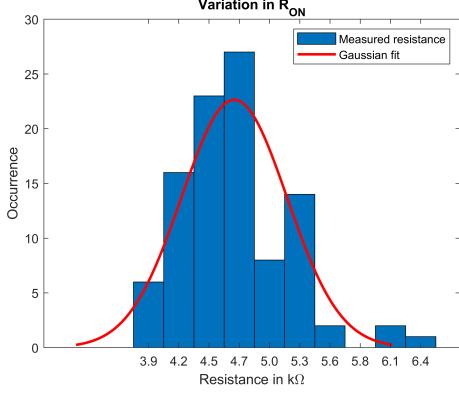
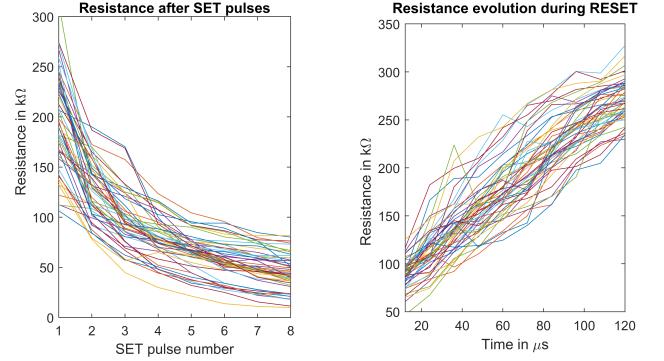


Fig. 7. Histogram of measured R_{on} values and fitted normal distribution: $\mu = 4.64 \text{ k}\Omega$, $\sigma = 427.9 \Omega$.



(a) Resistance after each applied SET pulse
(b) Evolution of resistance during RESET stimulus

Fig. 9. Evolution of resistance during SET and RESET.

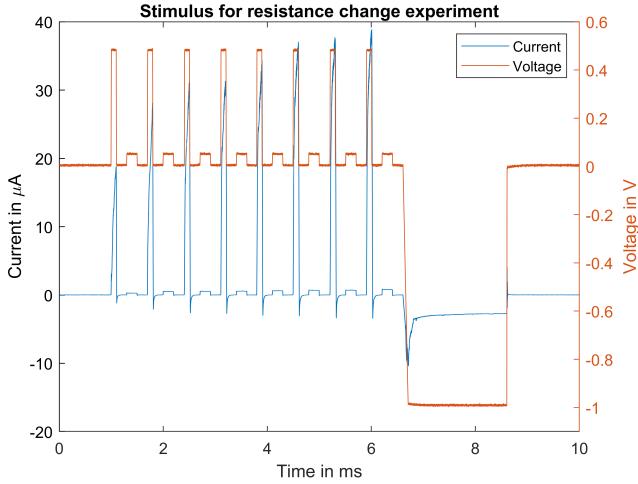


Fig. 8. Stimulus applied for the measurement of resistance change. Eight pairs of $500 \text{ mV} \times 100 \mu\text{s}$ SET pulse and $50 \text{ mV} \times 200 \mu\text{s}$ measurement pulse are followed by a $-1 \text{ V} \times 2 \text{ ms}$ RESET pulse.

used to study the RESET dynamics. Figure 9(b) shows the evolution of the resistance during each individual RESET pulse. Figure 16 shows the average evolution of resistance (in blue) during all applied RESET pulses. It should be noted that, since the resistance is determined at an applied voltage of -1 V , the RESET process is ongoing at that time, which may distort the measurement.

Moreover, the resistance is measured at a different voltage than for the SET process experiment, which may affect the comparability between the SET and RESET experiment due to non-linear voltage-resistance behavior.

D. Threshold Voltage

The threshold voltages of the memristor were extracted from the hystereses recorded during forming of the memristor. In order to do so, the threshold voltages were defined as the voltages, at which a major change in resistance, i.e., a change that is above a defined threshold, happens. This threshold was empirically found to be $\Delta I = 2 \text{ A/s} \approx 80 \text{ nA/sample}$. A MATLAB back-end for the LabVIEW environment was developed to automatically derive the threshold voltages from the

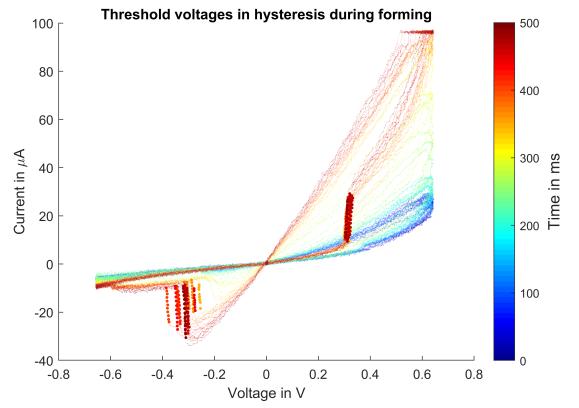


Fig. 10. Example of hysteresis recorded during forming (see Figure 3 and Section II-A) with highlighted areas (bold dots near -300 mV and $+300 \text{ mV}$), used for determining threshold voltages. The color of the dots determines time, i.e., dark blue denote early points, dark red denote late points.

recorded hystereses. Figure 10 shows an example of a recorded hysteresis (the same hysteresis as in Figure 3 was used as the example) with the highlighted areas of threshold effects, which have been identified by the software. The average threshold voltages of all recorded hystereses are $V_{TH+} = 370.2 \text{ mV}$ and $V_{TH-} = -373.8 \text{ mV}$. Fifty test cycles were used to extract threshold voltage values.

E. Leakage

Leakage refers to a change of the memristor's state represented by the measured resistance change at the absence of any stimulus, i.e., without any applied voltage to the memristor or any current going through the memristor. This property of a memristor is especially interesting for all kinds of memory and IMC applications, since it determines when and how the values might be lost if they are not refreshed.

To measure the leakage of the memristor, the memristor was driven to HRS. After that, a series of 100 measurement pulses ($50 \text{ mV} \times 200 \mu\text{s}$) was applied in one second intervals. Then, a SET pulse was applied to the memristor and the resistance was again determined with a series of measurement pulses. Figure 11 shows the recorded data points. The individual

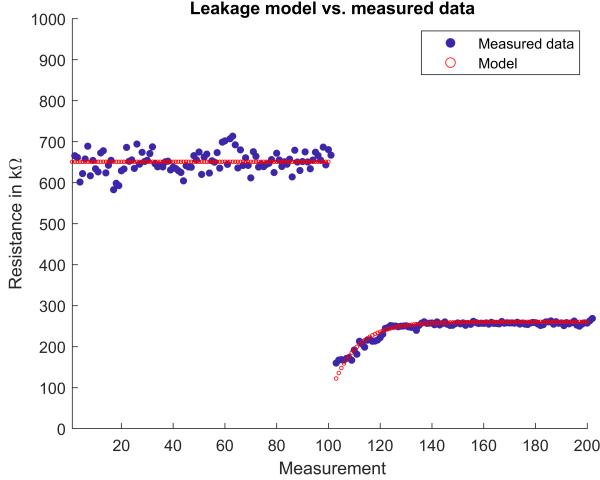


Fig. 11. Measured leakage effect versus model. Blue: Example of resistance recorded during leakage experiment. 100 measurements of the resistance were recorded after an applied RESET stimulus (measurements 1 to 100) and another 100 measurements were taken after application of a SET stimulus (measurements 101 to 200). The recorded points are 1 s apart. Red: Fitted model.

points were recorded in 1 s intervals. As it can be seen from the figure, there is no significant state drift after RESET. However, there is a state drift present after the SET pulse, which is stronger right after the application of the SET pulse and fades out in approximately 80-100 s. This experiment was repeated 50 times. During these tests no considerable state drift after RESET could be observed. The state drift after SET varies in terms of strength, i.e., initial drift rate and duration. However, the general shape that can be seen in Figure 11 stays present in all experiments. Additionally, an experiment was conducted in which a series of SET pulses was applied after a RESET pulse and the state drift was again recorded in 1 s intervals between the SET pulses. Figure 12 shows an example of the experiment, which shows, that state drift is stronger after the first SET pulse and becomes smaller for the consecutive SET pulses. This suggests, that the state drift after SET is dependent on the extent of the state change that preceded the drift.

Note that the drift occurs in negative direction, i.e., the resistance increases. Since the measurement pulses are $50 \text{ mV} \times 200 \mu\text{s}$, i.e., positive, any effect the measurement had on the resistance would drive it to lower values. This, apart from the measurement voltage being smaller than one fifth of the average measured threshold voltage, shows that the drift is not caused by the measurement itself, but is a separate effect. A possible explanation for this drift could be the collapsing of unstable portions of the conductive channel, a phenomenon observed in [30] too.

III. MODEL

A. Base Model Selection

As mentioned before, models that simulate the internal physical respectively electro-chemical effects of memristors, besides being computationally more expensive, are specially designed for a certain memristor type and even certain material pairings of which the modeled memristor is composed.

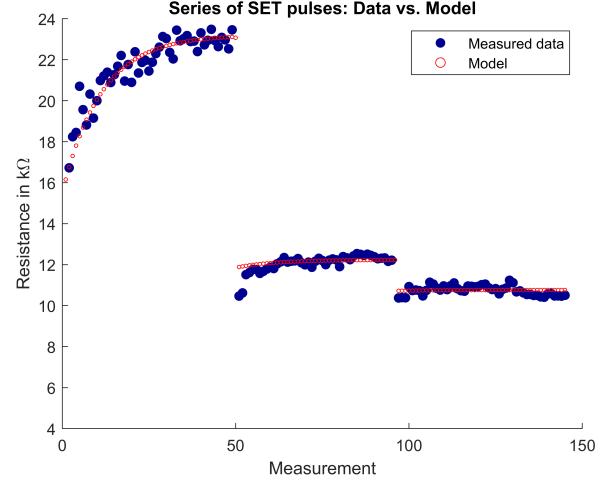


Fig. 12. Resistance during leakage experiment consisting of a series of SET pulses: Measurement versus model. Resistance was measured in 1 s time intervals.

Behavioral models on the other hand can be used for a family of similarly working memristors to a certain extend. A behavioral model for memristors that is both well-known and well-received by the community is the Voltage-controlled ThrEshold Adaptive Memristor (VTEAM) model [26]. It models the memristor as a voltage dependent resistance switch that has a continuous, bounded state variable. The change of the state variable depends on the applied voltage, making the model a so-called *voltage-controlled* model, which is reflected in the name of the model, in contrast to its twin brother, the ThrEshold Adaptive Memristor (TEAM) model [31], which is current-controlled. The VTEAM model was used as a starting point for the development of an enhanced memristor model because of the following reasons: (a) It is well-received by the community, (b) it is one of the most flexible behavioral models, (c) VTEAM can be implemented¹ in pure SPICE without any additional procedural code, for example, C or Verilog-A, (d) VTEAM was chosen over TEAM because in most circuit design situations it is more convenient to use voltage levels than current levels, (e) In VTEAM modeling of state change is independent from the modeling of voltage-current behavior.

As mentioned above, an advantage of VTEAM is, that the modeling of state change is in principle independent from the modeling of voltage-current behavior, as it can be seen from the Equations (1) and (3).

$$\frac{dw(t)}{dt} = \begin{cases} k_{\text{off}} \cdot \left(\frac{v(t)}{v_{\text{off}}} - 1 \right)^{\alpha_{\text{off}}} \cdot f_{\text{off}}(w) & , 0 < v_{\text{off}} < v \\ 0 & , v_{\text{on}} < v < v_{\text{off}} \\ k_{\text{on}} \cdot \left(\frac{v(t)}{v_{\text{on}}} - 1 \right)^{\alpha_{\text{on}}} \cdot f_{\text{on}}(w) & , v < v_{\text{on}} < 0 \end{cases} \quad (1)$$

$$f_{\text{off}}(w) = e^{-e^{\frac{w(t)-a_{\text{off}}}{w_c}}} \quad (2a)$$

$$f_{\text{on}}(w) = e^{-e^{\frac{-w(t)-a_{\text{on}}}{w_c}}} \quad (2b)$$

¹VTEAM was implemented in SPICE at the Institute of Computer Technology, TU Wien by Martin Jungwirth and David Radakovits and refined by Simon Laube [32].

TABLE I
VTEAM PARAMETERS

Parameter	Explanation
v_{off}	Positive threshold voltage.
v_{on}	Negative threshold voltage.
k_{off}	State change rate during SET.
k_{on}	State change rate during RESET.
α_{off}	Degree of (non-)linearity for SET.
α_{on}	Degree of (non-)linearity for RESET.
a_{off}	Window boundary for SET.
a_{on}	Window boundary for RESET.
w_c	Shaping parameter for window function.
w_{off}	Value of state variable at HRS.
w_{on}	Value of state variable at LRS.
R_{off}	Resistance at HRS.
R_{on}	Resistance at LRS.

$$v(t) = \left[R_{on} + \frac{R_{off} - R_{on}}{w_{off} - w_{on}} (w(t) - w_{on}) \right] \cdot i(t) \quad (3)$$

Table I lists all parameters for the VTEAM model. The parameters are briefly explained in the table. Since all parameters are constants during simulations with the VTEAM model – only the state $w(t)$ and the voltage, $v(t)$, and consequently, the current, $i(t)$, are variables – the model is not capable of simulating any variation in HRS and LRS resistance or state dynamics. In order to do so, this work introduces additions to this model in Sections III-B and III-C to enhance the model in terms of adding those capabilities.

B. Variation Model

Instead of setting the model parameters to constant values, the SPICE functions `.gauss()` and `.flat()` are used to model normal and uniform distribution of parameters, respectively. The functions expect a mean value and distribution interval (standard deviation in the case of normal distribution) and automatically pick respective parameters randomly using the chosen distribution. This enables the simulation of both cycle-to-cycle and device-to-device variations in memristors. This method is used to model the variations in HRS and LRS resistance, resistance change dynamics and threshold voltages.

C. Leakage Model

In order to simulate leakage, i.e., the drift of the memristor state in the absence of any stimulus, the model was extended by an additional influence on the state variable, $w(t)$. As it was shown in Section II-E, the drift depends on the extend of state change that happened during the SET process and fades out after a certain amount of time. Therefore, the state derivative, $\frac{dw(t)}{dt}$ in Equation (1), was extended by a term, that represents a decaying integral of state change. Equations (4a) and (4b) describe the enhanced model of state dynamics. The extension to the VTEAM model is given in blue. Equation (4b) in particular introduces a new term, which represents a capacity $\Theta(t)$, which integrates the state change and has a constant drain via the introduced model parameter τ_L . The introduced parameters θ_{off} and θ_{on} describe the influence of state change on the initial drift rate during SET and RESET process, respectively. Note that the model in principle is capable of modeling a state drift after RESET, however by setting $\theta_{on} = 0$

this feature is disabled. As it can be seen in Equation (4b), $\Theta(t)$ depends on the state change of the memristor that is calculated the same way as the VTEAM model, except for the additional factors θ_{off} and θ_{on} . The modeling of voltage-current behavior is unaltered and can be seen in Equation (3).

D. Fitting

This section explains the process of model fitting in four parts: Resistance in HRS and LRS, resistance change rate, threshold voltages and the observed leakage effect. The individual parameter values that were determined during the fitting process are collectively given in Section III-D5.

1) *HRS and LRS Resistance*: As it was shown before in Section II-B, the resistance in HRS and LRS can be modeled via a Gaussian or normal distribution with the parameters from Table IV.

Figure 13 shows an example of the measured HRS and LRS resistance versus the simulated variation.

2) *Resistance Change Dynamics*: The rate of resistance change is influenced by several parameters in the model for both SET and RESET process, respectively. These parameters are namely k_{off} , k_{on} , a_{off} , a_{on} and w_c . While k_{off} and k_{on} scale the state change linearly, a_{off} , a_{on} and w_c are used to influence the shape of the state change, i.e., to dampen the state change in proximity of state boundaries. Figure 14 shows the fitting of the shape parameters a_{off} , a_{on} and w_c and the average rate of state change during SET, which is $k_{off} = 780 \mu\text{m/s}$.

The histogram of the recorded resistance change is given in Figure 15. The resistance change shown in this figure was calculated as the difference between the resistance after the first of the eight applied SET pulses and the resistance after the third pulse, since in the fourth and fifth pulse, the change is not linear anymore. Since the distribution of resistance change, that was measured and shown in Figure 15, does not assimilate well-known distribution models, a different approach (compared to that of HRS and LRS resistance) was chosen. As it can be seen in the figure, 75 % of the measured values lie in the interval from $18.4 \text{ k}\Omega$ to $93 \text{ k}\Omega$.

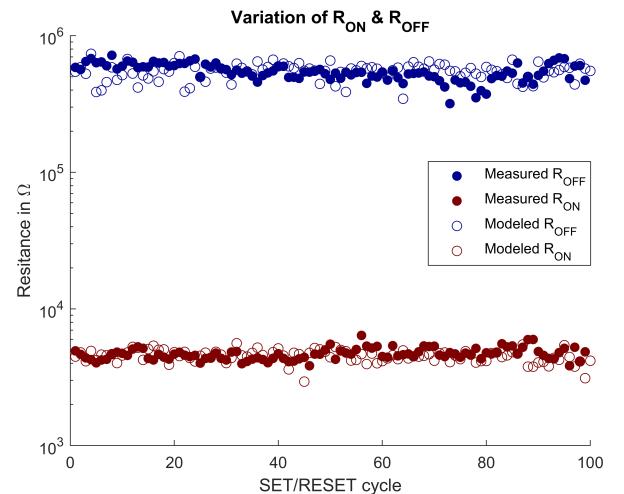


Fig. 13. Example of model vs. measured HRS and LRS resistances.

TABLE II
NEW MODEL EQUATIONS

$$\frac{dw(t)}{dt} = \begin{cases} k_{\text{off}} \cdot \left(\frac{v(t)}{v_{\text{off}}} - 1 \right)^{\alpha_{\text{off}}} \cdot f_{\text{off}}(w) & , 0 < v_{\text{off}} < v(t) \\ \Theta(t) & , v_{\text{on}} < v(t) < v_{\text{off}} \\ k_{\text{on}} \cdot \left(\frac{v(t)}{v_{\text{on}}} - 1 \right)^{\alpha_{\text{on}}} \cdot f_{\text{on}}(w) & , v(t) < v_{\text{on}} < 0 \end{cases} \quad (4a)$$

$$\frac{d\Theta(t)}{dt} = -\frac{\Theta(t)}{\tau_L} + \begin{cases} \theta_{\text{off}} \cdot k_{\text{off}} \cdot \left(\frac{v(t)}{v_{\text{off}}} - 1 \right)^{\alpha_{\text{off}}} \cdot f_{\text{off}}(w) & , 0 < v_{\text{off}} < v(t), \theta_{\text{off}} > 0 \\ 0 & , v_{\text{on}} < v(t) < v_{\text{off}} \\ \theta_{\text{on}} \cdot k_{\text{on}} \cdot \left(\frac{v(t)}{v_{\text{on}}} - 1 \right)^{\alpha_{\text{on}}} \cdot f_{\text{on}}(w) & , v(t) < v_{\text{on}} < 0, \theta_{\text{on}} < 0 \end{cases} \quad (4b)$$

with a steep decrease of measured values above and below this interval. Therefore, this interval was chosen for a uniform distribution of the resistance change variation during SET. With an average measured resistance change of $55.8 \text{ k}\Omega$ the variation calculates to $\pm 67\%$, which translates into a k_{off} variation of $\pm 522.6 \mu\text{m/s}$ (3σ).

Figure 16 shows the fitting of the average rate of state change during RESET, which matches $k_{\text{on}} = -4.67 \mu\text{m/s}$. The same shaping parameters are used for both SET and RESET process. The variation of the state change during RESET was determined in the same manner as described above for the SET process. Figure 17 shows the histogram of the measured data. More than 75 % of the measurements lie between $-65.9 \text{ k}\Omega$ and $-23.3 \text{ k}\Omega$. With an average measured resistance change of $-44.6 \text{ k}\Omega$ the variation calculates to $\pm 48\%$, which translates into a variation of k_{on} of $\pm 2.24 \mu\text{m/s}$ (3σ).

Table IV gives the statistical parameters for the resistance change variation model.

3) Threshold Voltage: The actual threshold behavior of the model is influenced not only by the v_{off} and v_{on} parameters, but also by the exponents α_{off} and α_{on} , which influence the sharpness of the threshold. The threshold voltages were fitted using the recorded hysteresses to mimic the observed behavior.

The values for the exponents α_{on} and α_{off} were taken from the original VTEAM publication [26]. The average threshold voltages, as reported in Section II-D, are $v_{\text{off}} = 370.2 \text{ mV}$ and $v_{\text{on}} = -373.8 \text{ mV}$. The largest deviation from the average was 30.57 % for v_{off} and 33.01 % for v_{on} . These values were used as a variation interval (3σ) for a variation model with uniform distribution, since the actual distribution could not be extracted based on the data set at hand. As explained in Section III-D2 this approach is suitable to give a worst case estimation of threshold voltage variation. Table IV gives the parameters for the distribution.

4) Leakage: The parameters that influence the modeling of the leakage effect are θ_{off} , θ_{on} and τ_L . As mentioned in Section II-E, no leakage effect was observed after RESET.

TABLE III
NEW PARAMETERS INTRODUCED TO THE MODEL.

Parameter	Explanation
θ_{off}	Influence of SET process state change on leakage
θ_{on}	Influence of RESET process state change on leakage
τ_L	Time constant of leakage fade-out

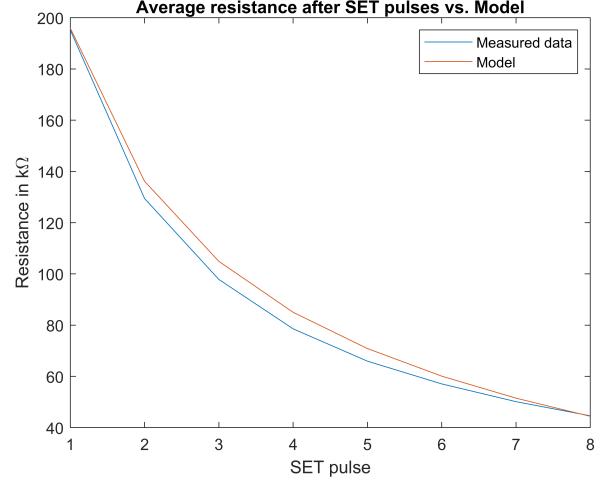


Fig. 14. Measured resistance change during SET process (average of 100 measurements) versus model.

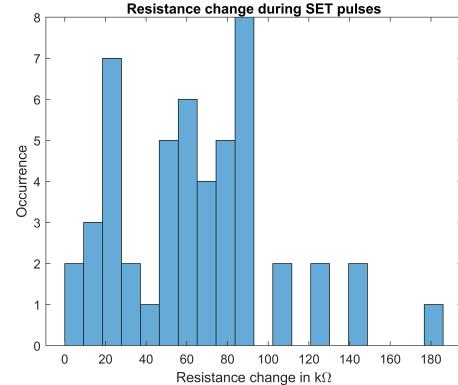


Fig. 15. Histogram of measured resistance change during SET, measured as difference between third and first measurement pulse. Note, that resistance change from HRS to LRS is considered positive in this figure.

Therefore, θ_{on} can be set to zero. θ_{off} and τ_L were set to mimic the observed behavior in the experiments. In order to do that, since the leakage model describes an exponential drift which at $t = 5\tau$ differs for less than 0.1 % from its value for $t \rightarrow \infty$, τ_L is set to a fifth of the observed duration of state change due to leakage. Figure 11 shows an example of the fitted model versus the data measured during a leakage measurement. Figure 12 shows another type of experiment, in which a series of SET

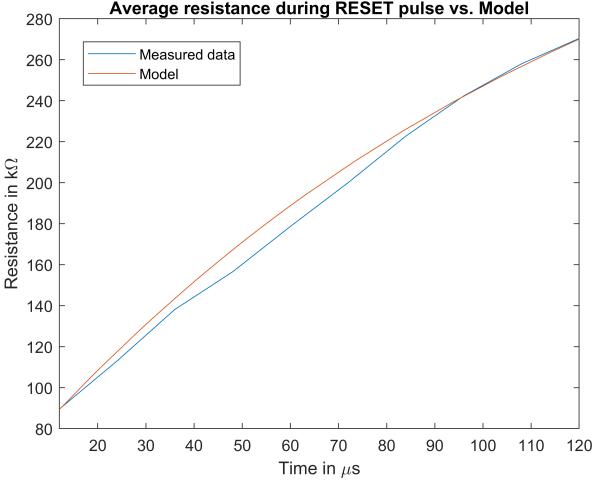


Fig. 16. Resistance measured during RESET pulse (average of 100 measurements) versus model.

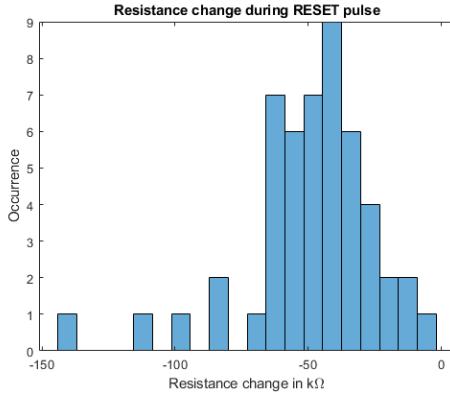


Fig. 17. Histogram of measured resistance change during RESET. Note, that resistance change from LRS to HRS is considered negative in this figure.

pulses was applied. As it can be seen from the figure, the model is capable of reproducing the observed behavior. The average values observed during all leakage experiments are $\theta_{\text{off}} = 0.0173 \text{ s}^{-1}$, $\tau_L = 10.3 \text{ s}$ and $\theta_{\text{on}} = 0$, since no leakage effect was observed after RESET.

5) *Summary:* The developed memristor model, that incorporates both the variation of parameters and the leakage effect, is based on the VTEAM model, as is given in Equations (1), (2) and (3). In the enhanced model, the memristor is also modeled as a linearly state-dependent, linear voltage-current relationship, cf. Equation (3).

The state dynamics are non-linearly dependent on the voltage applied to the memristor, cf. Equation (4a).

In this, f_{off} and f_{on} denote window functions, which are used to dampen the state change in the proximity of the state limits, cf. Equation (2).

Equation (4a) holds an additional term over the original VTEAM state equations, which is $\Theta(t)$. This term denotes the influence of leakage on the memristor state. $\Theta(t)$ is calculated as a decaying state drift, which is dependent on the previous change of memristor state and the current memristor state, cf. Equation (4b).

The modeling of parameter variations is achieved by a dynamic calculation of core model parameters, namely R_{ON} , R_{OFF} , k_{ON} , k_{OFF} , v_{ON} and v_{OFF} , using distribution functions. Table IV gives the final values for all model parameters. Note that the variation of the resistance change rates k_{on} and k_{off} and the variation of threshold voltages are a third of the values given in Sections III-D2 and III-D3, since both `.gauss()` and `.flat()` expect the standard deviation σ as a parameter, whereas in Sections III-D2 and III-D3 the overall variation interval (3σ) is given. The full SPICE model code is given in the appendix.

IV. DISCUSSION

The model presented in this work is capable of simulating the leakage effect, which is a novelty among behavioral models. The simulation inaccuracy of the leakage effect is $\leq 13.4\%$ for all points in all 20 tested cases. The average of absolute values of relative deviation of the model from the data is 1.1 %. This average deviation was calculated as

$$\Delta = \frac{1}{N} \sum_{i=1}^N \frac{|\tilde{x}_i - x_i|}{x_i} \quad (5)$$

where \tilde{x}_i denotes the modeled value and x_i denotes actual, measured value. It needs to be stated, that similar to all other parameters of the memristor, both the time constant of the leakage decay, as well as the initial leakage strength varied over the course of the investigation. However, the effect of those variations is small, i.e., the accuracy of the model was not affected dramatically and the variations were not included in the model to simplify the implementation.

The simulation inaccuracy of the state change rate is $\leq 8.2\%$ for all points and 4.6 % on average over the 100 tested cases. The average deviation was again calculated according to Equation (5).

The identification and fitting of threshold voltages is specially hard to manage. The optimal experiment to identify a memristors threshold voltage, e.g., the positive threshold voltage, would consist of the application of a pulsed stimulus, in which the pulses have constant duration but increasing

TABLE IV
VALUES FOR ALL MODEL PARAMETERS.

Parameter	Value		
Distribution	Mean	Standard deviation	Type
R_{off}	545.54 kΩ	77.095 kΩ	gaussian
R_{on}	4.92 kΩ	858.8 Ω	gaussian
v_{off}	370.2 mV	37.7 mV	uniform
v_{on}	-373.8 mV	41.1 mV	uniform
k_{off}	780 μm/s	174.2 μm/s	uniform
k_{on}	-4.67 μm/s	0.747 μm/s	uniform
α_{off}	3		
α_{on}	3		
a_{off}	1.3 nm		
a_{on}	1.8 nm		
w_c	980 pm		
w_{off}	3 nm		
w_{on}	0 nm		
θ_{off}	0.0173 s ⁻¹		
θ_{on}	0 s ⁻¹		
τ_L	10.3 s		

amplitude. The threshold voltage is defined as the voltage that causes the resistance change to exceed a certain defined value. Keep in mind, that the measurement of resistance would need to be carried out with measurement pulses in between the stimuli, to ensure suppression of any distortion due to non-linear voltage-resistance behavior. However, since the momentary resistance of a memristor depends on its past, the experiment would need to be repeated with only the stimulus that was just too small and the stimulus that caused a significant state change, to suppress the effect of the preceding stimuli. Since most probably the threshold voltage might vary during these measurements, a successful determination of the threshold voltage is still not guaranteed.

The model as it is presented and implemented in SPICE is capable of modeling the variation of the mentioned parameters in both cycle-to-cycle as well as device-to-device fashion. However, it is not possible to model parameter variations during simulation runs, as the parameters are newly chosen for each cycle but remain constant during a cycle. Simulation of device-to-device variations is achieved via separate selection of parameters for each individual device. The identification of device-to-device variations, however, is out of the scope of this work.

V. CASE STUDIES

Several memristive logics have been presented in the literature, some of the prominent ones being Material Implication (IMPLY) [33], Memristor-Aided Logic (MAGIC) [34], Fast and energy-efficient Logic in Memory (FELIX) [35] and Three Memristors Stateful Logic (TMSL) [36]. All these logics are *stateful*, i.e., the input and output logic values are represented by the resistance of a memristor and all of them function based on the application of certain, predefined voltages to calculate the desired state of the output memristor. In this chapter each of the four logics is used to design a logic gate under the specifications of the respective publication. In order to show the effect of parameter variations and leakage on the designed gates, each individual gate is designed with the nominal parameter values, i.e., the mean values, and without consideration of leakage, as it is done in the publications. Probabilities of correct output calculation are then determined through simulation of the memristive logic gates with the model that incorporates both parameter variations and leakage.

Note, that for all the investigated logics, there are certain degrees of freedom, e.g., execution time or the selection of certain voltages from an allowed interval, which result in different trade-offs for speed or different aspects of robustness. However, it is out of the scope of this work, to find an optimal gate design. Here, the design rules given in the publications are followed, i.e., an allowed parameter set is chosen and tested for functional correctness for all input combinations for each logic, and the impact of parameter variations and leakage is studied with the obtained gate designs in order to stress the importance of incorporating those non-idealities into memristor models. All investigations are conducted based on a logic threshold of 0.5, i.e., $0 \leq s < 0.5 \hat{=} '0'$ and $0.5 \leq s \leq 1 \hat{=} '1'$, where s denotes the state of the memristor,

was chosen as a mapping of memristor state to logical values. Overall correctness of logical operation was calculated as the overall probability of correctly processed outputs, i.e., $P_{\text{correct}} = \frac{1}{4}P_{00} + \frac{1}{4}P_{01} + \frac{1}{4}P_{10} + \frac{1}{4}P_{11}$, where P_{correct} is the overall correctness, and P_{00} to P_{11} are the probabilities of the individual input combinations to yield correct outputs.

Furthermore, the effect of leakage on the reliability of the gates under investigation was studied. In order to do so, a stable time was defined, which gives the time span after the logic operation within which the result of the logic operation remains unaltered by leakage, i.e., after the stable time the result becomes incorrect because the leakage caused the memristor state to cross the chosen logic threshold and a so-called bit flip occurs.

A. IMPLY

Material Implication Logic (IMPLY), as the name suggests, implements a material implication function. The q memristor, which holds both the second operand before, as well as the output after the logic operation, only changes its state in the case $0 \rightarrow 0$, where the stored bit goes from ' 0 ' to ' 1 '. That means that since in IMPLY the common mapping of ' 0 ' $\hat{=}$ HRS and ' 1 ' $\hat{=}$ LRS is used, the q memristor has to be turned on, i.e., SET, in Case 1, where both memristors are in HRS prior to the logic operation and must not be turned on, when only p is in LRS as in Case 3. Figure 18 shows the circuit of a basic IMPLY gate with $V_{\text{set}} > v_{\text{off}}$, $V_{\text{set}} > V_{\text{cond}}$ and $R_{\text{on}} < R_G < R_{\text{off}}$. Additional design constraints are given in [37]. Table V gives the logic parameters used for the simulation of IMPLY gates.

In IMPLY only Case 1, i.e., the input combination ' 00 ' is affected by parameter variations and leakage. Cases 2, 3 and 4 are correct in 100 % of the simulated cases. This is due to the inherent function of IMPLY, in which only in Case 1 of the truth table a switching occurs. Figure 21 shows the state histogram of the output memristor Q after logic operation. 86.4 % of the simulations resulted in the correct output, i.e., ' 1 '. Considering the correctness of input combinations ' 01 ', ' 10 ' and ' 11 ', which were always correct, as mentioned above, the overall probability of correct behavior calculates to 96.6 %. As for the parameter variation, also the leakage only affects Case 1 of the IMPLY gate truth table. Figure 22 gives the histogram of the time the output was stable during the simulations. The orange line in the figure gives the accumulated relative portion of all outputs being distorted by leakage. Using this figure, a designer can determine the maximum stable time for a certain probability of a correct output, i.e., that the output was not distorted by leakage. For example, if > 90 % of the calculated outputs shall remain correct, the stable time is $16.9 \mu\text{s}$, for > 99 %, the stable time is $1.98 \mu\text{s}$.

B. MAGIC

Memristor-Aided loGIC (MAGIC) implements a 2-bit NOR gate using three memristors. As opposed to IMPLY, a separate output memristor is used, as it can be seen in Figure 19. This output memristor is initialized to LRS, i.e., ' 1 ' prior to logic

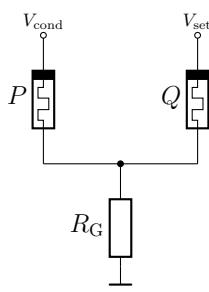


Fig. 18. Circuit of basic IMPLY gate. Q holds the result of $p \rightarrow q$ after the operation.



Fig. 19. Circuit of 2-bit MAGIC gate. in_1 and in_2 hold the inputs, out is initialized to LRS and holds the output after logic operation.

Fig. 20. Circuit of 2-bit FELIX OR-gate. in_1 and in_2 hold the inputs, out is initialized to HRS and holds the output after logic operation.

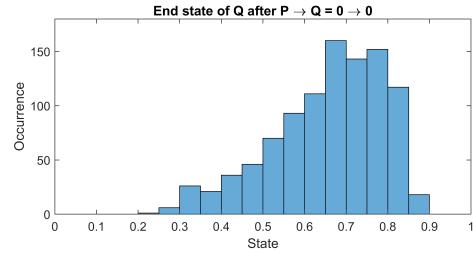


Fig. 21. IMPLY: State distribution of output memristor after logic operation for input '00'.

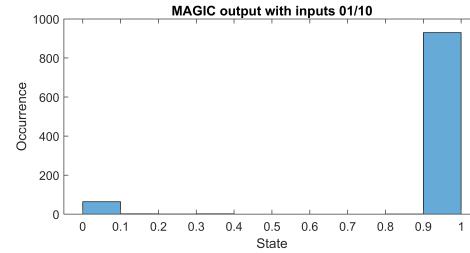


Fig. 23. MAGIC: State distribution of output memristor after logic operation for inputs '01' and '10'.

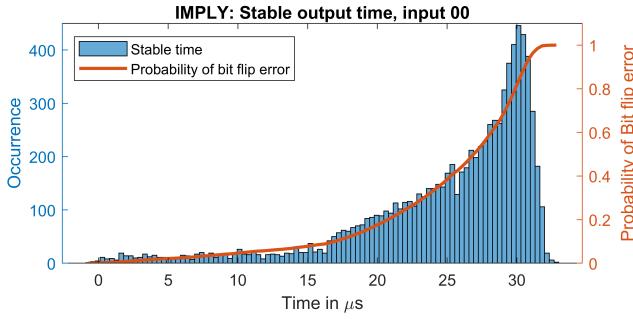


Fig. 22. IMPLY: Distribution of stable output time for input '00'. The orange line gives the accumulated portion of incorrect outputs over time.

operation. Table V gives the logic parameters used for the simulation of MAGIC gates.

Although the MAGIC gate was designed according to the constraints given in the original publication, i.e., Equation (1) in [34], the resulting NOR-gate does not work for the inputs '01' and '10', which should yield '0'. This is due to the fact, that the second constraint on applied voltage, i.e., Equation (2) in [34] cannot be fulfilled at the same time as Equation (1) with the parameter set measured for the memristors used in this work. This is manifested in a very low probability of 6.4 % correct output generation in cases '01' and '10', as it is shown in Figure 23. Simulations of cases '00' and '11' yielded probabilities of 95.2 % and 87.2 %, respectively.

MAGIC is not prone to distortion of outputs by leakage. This is caused by the inherent function of MAGIC which is a selective RESET of the output memristor, which is initialized

to LRS prior to logic operation. Since no leakage was observed after a RESET process, MAGIC NOR-gates are not influenced by this effect.

C. FELIX

Fast and Energy-efficient Logic in memory (FELIX) can be used to implement different logic functions using the same structure, such as NOR, NAND or the logical minority function, as it is shown in [35]. These functions are constructed very similar to MAGIC gates. However, FELIX can also be used to implement an OR function. Table V gives the logic parameters used for the simulation of FELIX gates.

Figure 24 shows the histogram of the simulated output states for the inputs '01' and '10'. The two cases do not need to be distinguished, since FELIX is symmetrical regarding the inputs. The gate gave correct outputs in 91.4 % of the simulated cases. Figure 25 shows the histogram of the simulated output states for the input '11'. The gate gave correct outputs in 94.8 % of the simulated cases. With 100 % correctness for the input '00' this calculates to an overall correctness of 94.4 %.

Leakage affects the output of a FELIX NOR-gate for the inputs '01', '10' and '11', where the cases '01' and '10' are again symmetrical and thus indistinguishable in simulation. Figures 26 and 27 give the histograms of the time the output was stable during the simulations for the inputs '01', '10' and '11', respectively. The orange line in the figures again gives the accumulated relative portion of all outputs being distorted by leakage. For the input '01' and '10', > 90 % of the calculated outputs remain correct until 14.1 μ s after the operation, for

> 99 % the stable time is $3.92 \mu\text{s}$. With the input ‘11’ for > 90 % the stable time is $12.3 \mu\text{s}$, for > 99 % it is $0.50 \mu\text{s}$.

D. TMSL

The Three Memristor Stateful Logic (TMSL) [36] uses a work resistor R_G , as in IMPLY, but uses a separate output memristor, as in MAGIC and FELIX. Table V gives the logic parameters for the simulation of TMSL gates.

Figure 29 shows the histogram of output memristor states observed during simulations for the input ‘11’. As it can be seen, 27.1 % of the results are incorrect. With 100 % correct results for the inputs ‘00’, ‘01’ and ‘10’, this calculates to an overall correctness of 93.2 %. Since in TMSL the output memristor is initialized to HRS and only SET if the input is ‘00’, this denotes the only case that is influenced by leakage. Figure 30 gives the histogram of the stable output time for this input. For > 90 % correct outputs the maximum stable time is $8.21 \mu\text{s}$, for > 99 % it is $5.95 \mu\text{s}$.

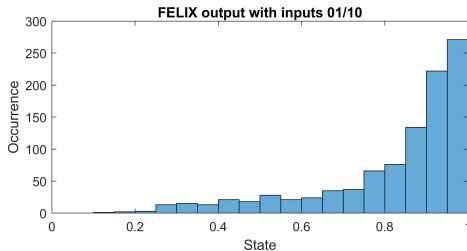


Fig. 24. FELIX: State distribution of output memristor after logic operation for inputs ‘01’ and ‘10’.

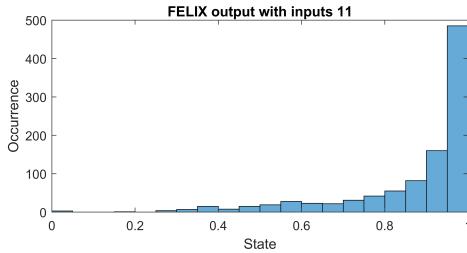


Fig. 25. FELIX: State distribution of output memristor after logic operation for input ‘11’.

TABLE V
DESIGN PARAMETERS FOR THE INVESTIGATED LOGICS. T DENOTES THE EXECUTION TIME OF ONE LOGIC OPERATION.

Logic	Parameter	Value
IMPLY	T	$50 \mu\text{s}$
	V_{set}	0.6 V
	V_{cond}	0.4 V
	R_G	$40 \text{ k}\Omega$
MAGIC & FELIX	T	10 ms
	V_0	1 V
TMSL	T	$100 \mu\text{s}$
	V_{set}	1 V
	V_{cond}	0.5 V
	R_G	$40 \text{ k}\Omega$

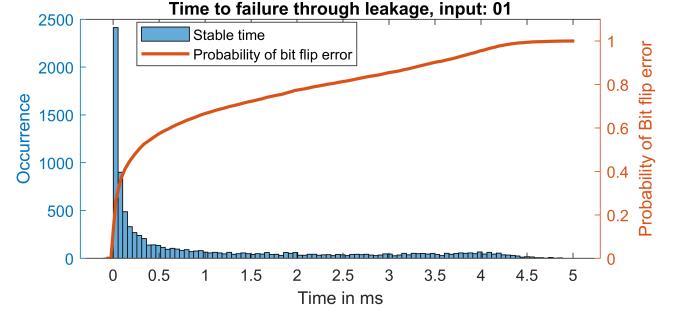


Fig. 26. FELIX: Distribution of stable output time for input ‘01’. The orange line gives the accumulated portion of incorrect outputs over time.

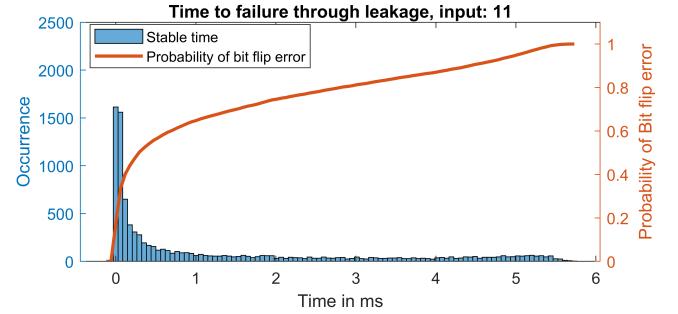


Fig. 27. FELIX: Distribution of stable output time for input ‘11’. The orange line gives the accumulated portion of incorrect outputs over time.

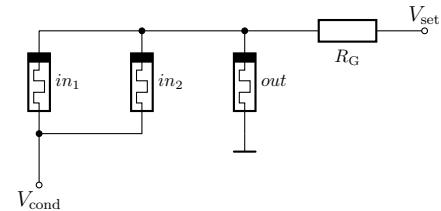


Fig. 28. Circuit of 2-bit TMSL NOR-gate. in_1 and in_2 hold the inputs, out is initialized to HRS and holds the output after logic operation.

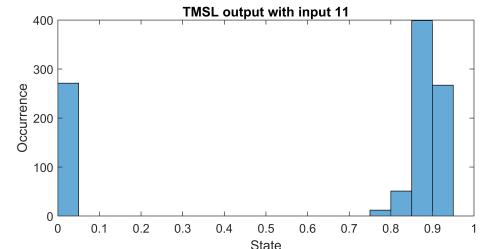


Fig. 29. TMSL: State distribution of output memristor after logic operation for logical inputs ‘11’, i.e. both input memristors are in HRS. Note, that the NAND operation should yield ‘0’ in this case, i.e., memristor state 1.

E. Summary

The presented case studies show, that all of the four most popular stateful memristive logics, i.e., IMPLY, MAGIC, FELIX and TMSL, are negatively influenced by parameter variations and the leakage effect. Even though the gates have been designed according to the given design rules. Table VI

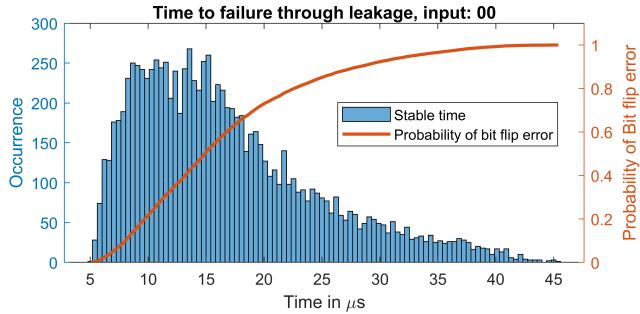


Fig. 30. TMSL: Distribution of stable output time for input ‘00’. The orange line gives the accumulated portion of incorrect outputs over time.

TABLE VI
PROBABILITY OF CORRECT OUTPUT FOR EACH OF THE INPUT COMBINATIONS ('00', '01', '10' AND '11') FOR THE INVESTIGATED LOGICS.

Logic	'00'	'01'	'10'	'11'	Overall
IMPLY	86.4 %	100 %	100 %	100 %	96.6 %
MAGIC	95.2 %	6.4 %	6.4 %	87.2 %	48.8 %
FELIX	100 %	91.4 %	91.4 %	94.8 %	94.4 %
TMSL	100 %	100 %	100 %	72.9 %	93.2 %

gives a summary of the observed performance in terms of output probabilities, Table VII shows the 90 %, 99 %, average and median output stable times for all investigated logics. Note that in Table VII, if more than one input case was influenced by leakage, the minimum times are given. t_{avg} and t_{med} denote the average and median stable time of all inputs affected by leakage, respectively. MAGIC is not listed in the table since in our technology it is structurally not prone to leakage, as explained in Section V-B.

VI. CONCLUSION & OUTLOOK

The work at hand presents an enhanced model for resistance switching type memristors or ReRAMs, which was fitted to KNOWM SDC memristors. For the first time in the field of behavioral ReRAM models, the so-called leakage effect, i.e., a drift of the memristor state in absence of any stimulus, is reproduced. The leakage effect was observed and modeled to shift the resistance of a memristor back to higher values after a SET process, i.e., after the memristor changed from higher resistance to lower resistance. This drift happens in an exponential manner over an average time span (5τ) of 51.5 s. Furthermore, the model supports simulation of parameter variations, which can be used to model both cycle-to-cycle as well as device-to-device variation. These variations of model parameters were observed to account for up to 23.3 % of

the nominal value. The model is completely implemented in SPICE without the need of additional, e.g., procedural, software or solver engines. The simulation inaccuracy of the leakage effect is 1.1 % on average and 13.4 % in the worst case over 20 tested cases. The simulation inaccuracy of resistance dynamics is 4.6 % on average and 8.2 % in the worst case over 100 tested cases. By incorporating the leakage effect and the statistical variation of model parameters, this model enables more significant, robust and reliable simulation of circuits and systems containing ReRAM devices.

Based on the presented model and fitting, a set of case studies was conducted, which revealed a negative impact of parameter variations and the leakage effect on four of the most popular stateful memristive logics, namely IMPLY, MAGIC, FELIX and TMSL. The work at hand provides a guideline on how to assess logic robustness and state stability times using the presented model. The probability of correct output generation was simulated to be as low as 48.8 % and the leakage effect was simulated to reduce the stable time of the output memristor state to as low as 0.5 μ s, if stable outputs with a probability ≥ 99 % are desired.

This work shows, that the evaluation of memristive circuits and systems based on nominal, i.e., static parameter sets for memristor models is insufficient. Parameter variations and the leakage effect need to be taken into account in order to assess the robustness, and in extreme cases even the function, of memristive circuits and systems. These effects proved to degrade the performance of the most popular memristive logics considerably and therefore cannot be neglected.

The presented model gives a versatile base to conduct further research in stochastic behavior of resistance switching devices and memristors in general. For example, the statistical properties of the leakage effect could be investigated and a thorough statistical analysis of memristive behavior around threshold voltages and currents could be conducted. Large scale statistical analysis could be performed to refine the modeling of parameters with uniform distribution and work-out more sophisticated distribution models. Future work could also investigate if there is any influence due to parameter variation of resistance dynamics on the variation of the leakage effect. To further increase the statistical relevance, large scale stochastic analysis of all discussed effects could be conducted. Such research requires elaborate and sophisticated designs regarding experimentation environments and measurement methods for which this work can act as a base in terms of instrumentation and methodology. Measured data availability in the field of memristive circuits and systems is very scarce in general and ongoing statistical examination of memristor behavior is necessary. This work strives to take one step forward in that direction.

MODEL SOURCE FILE

This model will be open-sourced and available to the public upon the acceptance of the paper.

REFERENCES

- [1] S. Madden, “From databases to big data,” *IEEE Internet Computing*, vol. 16, no. 3, pp. 4–6, 2012.

Logic	t_{90} in μ s	t_{99} in μ s	t_{avg} in μ s	t_{med} in μ s
IMPLY	16.9	1.98	25.04	27.08
FELIX	12.3	0.5	1170.5	311.5
TMSL	8.2	5.95	16.8	15.1

- [2] S. Sagiroglu and D. Sinanc, "Big data: A review," in *2013 international conference on collaboration technologies and systems (CTS)*. IEEE, 2013, pp. 42–47.
- [3] C. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Information sciences*, vol. 275, pp. 314–347, 2014.
- [4] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile networks and applications*, vol. 19, no. 2, pp. 171–209, 2014.
- [5] E. Linn, R. Rosezin, S. Tappertzhofen, U. Böttger, and R. Waser, "Beyond von Neumann—logic operations in passive crossbar arrays alongside memory operations," *Nanotechnology*, vol. 23, no. 30, p. 305205, July 2012.
- [6] S. Li *et al.*, "Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories," in *DAC2016*. IEEE, 2016, pp. 1–6.
- [7] R. B. Hur and S. Kvavitsky, "Memory processing unit for in-memory processing," in *IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, July 2016, pp. 171–172.
- [8] P. Gaillardon, L. Amarú, A. Siemon, E. Linn, R. Waser, A. Chattopadhyay, and G. De Micheli, "The programmable logic-in-memory (PLiM) computer," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2016, pp. 427–432.
- [9] G. Papandroulidakis, I. Vourkas, A. Abusleme, G. C. Sirakoulis, and A. Rubio, "Crossbar-based memristive logic-in-memory architecture," *IEEE Transactions on Nanotechnology*, vol. 16, no. 3, pp. 491–501, May 2017.
- [10] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang, J. P. Strachan, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, and Q. Xia, "In-memory computing with memristor arrays," in *IEEE International Memory Workshop (IMW)*, May 2018, pp. 1–4.
- [11] L. Chua, "Memristor—the missing circuit element," *IEEE Transactions on circuit theory*, vol. 18, no. 5, pp. 507–519, 1971.
- [12] D. B. Strukov *et al.*, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.
- [13] S. Menzel, A. Siemon, A. Ascoli, and R. Tetzlaff, "Requirements and challenges for modelling redox-based memristive devices," in *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, May 2018, pp. 1–5.
- [14] W. Rainer, D. Regina, S. Georgi, and S. Kristof, "Redox-based resistive switching memories - nanoionic mechanisms, prospects, and challenges," *Advanced Materials*, vol. 21, no. 25–26, pp. 2632–2663, 2009. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.200900375>
- [15] *Self Directed Channel Memristor*. KNOWM Inc., February 2020, https://knowm.org/downloads/Knowm_Memristors.pdf.
- [16] E. Gale, "TiO₂-based memristors and ReRAM: Materials, mechanisms and models (a review)," *Semiconductor Science and Technology*, vol. 29, no. 10, p. 104004, 2014.
- [17] A. J. Lohn, J. E. Stevens, P. R. Mickel, and M. J. Marinella, "Optimizing TaO_x memristor performance and consistency within the reactive sputtering "forbidden region"," *Applied Physics Letters*, vol. 103, no. 6, p. 063502, 2013.
- [18] B. Jin-shun and H. Zheng-sheng, "Study on Hf/HfO₂ bipolar resistive random-access-memory," *Journal of Functional Materials and Devices*, vol. 5, p. 002, 2014.
- [19] H. Jiang, L. Han, P. Lin, Z. Wang, M. H. Jang, Q. Wu, M. Barnell, J. J. Yang, H. L. Xin, and Q. Xia, "Sub-10 nm ta channel responsible for superior performance of a hfo 2 memristor," *Scientific reports*, vol. 6, p. 28525, 2016.
- [20] H. Abunahla, M. A. Jaoude, C. J. O'Kelly, Y. Halawani, M. Al-Qutayri, S. F. Al-Sarawi, and B. Mohammad, "Switching characteristics of microscale unipolar Pd/Hf/HfO₂/Pd memristors," *Microelectronic Engineering*, vol. 185, pp. 35–42, 2018.
- [21] X. Lian, M. Wang, M. Rao, P. Yan, J. J. Yang, and F. Miao, "Characteristics and transport mechanisms of triple switching regimes of TaO_x memristor," *Applied Physics Letters*, vol. 110, no. 17, p. 173504, 2017.
- [22] J. S. Meena, S. M. Sze, U. Chand, and T.-Y. Tseng, "Overview of emerging nonvolatile memory technologies," *Nanoscale research letters*, vol. 9, no. 1, p. 526, 2014.
- [23] R. Rizk, D. Rizk, A. Kumar, and M. Bayoumi, "Demystifying emerging nonvolatile memory technologies: understanding advantages, challenges, trends, and novel applications," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2019, pp. 1–5.
- [24] N. TaheriNejad and D. Radakovits, "From behavioral design of memristive circuits and systems to physical implementations," *IEEE Circuit and Systems (CAS) Magazine*, vol. 19, no. 4, pp. 6–18, Fourthquarter 2019.
- [25] Z. Jiang, S. Yu, Y. Wu, J. H. Engel, X. Guan, and H.-S. P. Wong, "Verilog-a compact model for oxide-based resistive random access memory (rram)," in *2014 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*. IEEE, 2014, pp. 41–44.
- [26] S. Kvavitsky, M. Ramadan, E. G. Friedman, and A. Kolodny, "VTEAM: A general model for voltage-controlled memristors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 8, pp. 786–790, August 2015.
- [27] C. Yakopcic, T. M. Taha, G. Subramanyam, R. E. Pino, and S. Rogers, "A memristor device model," *IEEE electron device letters*, vol. 32, no. 10, pp. 1436–1438, 2011.
- [28] D. Biolek, M. Di Ventra, and Y. V. Pershin, "Reliable spice simulations of memristors, memcapacitors and meminductors," *arXiv preprint arXiv:1307.2717*, 2013.
- [29] J. Gomez, I. Vourkas, and A. Abusleme, "Exploring memristor multi-level tuning dependencies on the applied pulse properties via a low cost instrumentation setup," *IEEE Access*, vol. 7, pp. 59413–59421, 2019.
- [30] I. Valov, "Redox-based resistive switching memories (rerams): Electrochemical systems at the atomic scale," *ChemElectroChem*, vol. 1, no. 1, pp. 26–36, 2014.
- [31] S. Kvavitsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Team: Threshold adaptive memristor model," *IEEE transactions on circuits and systems I: regular papers*, vol. 60, no. 1, pp. 211–221, 2012.
- [32] D. Radakovits, M. Jungwirth, S. M. Laube, and N. TaheriNejad, "Second (v2.0) LTSpice implementation of VTEAM," September 2019, <https://www.ict.tuwien.ac.at/staff/taherinejad/projects/memristor/files/vteam2.asc>, <https://www.ict.tuwien.ac.at/staff/taherinejad/projects/memristor/files/vteam2.asy>.
- [33] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, "'memristive' switches enable 'stateful' logic operations via material implication," *Nature*, vol. 464, no. 7290, p. 873, 2010.
- [34] S. Kvavitsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "MAGIC—memristor-aided logic," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, pp. 895–899, November 2014.
- [35] S. Gupta, M. Imani, and T. Rosing, "FELIX: Fast and energy-efficient logic in memory," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, November 2018, pp. 1–7.
- [36] P. Huang, J. Kang, Y. Zhao, S. Chen, R. Han, Z. Zhou, Z. Chen, W. Ma, M. Li, L. Liu *et al.*, "Reconfigurable nonvolatile logic operations in resistance switching crossbar array for large-scale circuits," *Advanced Materials*, vol. 28, no. 44, pp. 9758–9764, 2016.
- [37] S. Kvavitsky *et al.*, "Memristor-based imply logic design procedure," in *ICCD2011*. IEEE, 2011, pp. 142–147.



David Radakovits received his M.Sc. (2020) in Embedded Systems and B.Sc. (2017) in Electrical Engineering and Information Technology from Technische Universität Wien, Vienna University of Technology. His research interest includes the areas of memristor-based circuits and systems, hardware security, machine learning and embedded systems in which he published several articles in IEEE journals and conferences.



Nima TaheriNejad (S'08-M'15) received his Ph.D. degree in electrical and computer engineering from The University of British Columbia (UBC), Vancouver, Canada, in 2015. He is currently a "Universitätsassistent" at the TU Wien (formerly known as Vienna University of Technology), Vienna, Austria, where his areas of work include in-memory computing and self-awareness in resource-constrained cyber-physical and embedded systems, and healthcare. He has published two books and more than 60 peer-reviewed articles. Dr. Taherinejad has also served as a reviewer, an editor, an organizer, and the chair for various journals, conferences, and workshops. He has received several awards and scholarships from universities and conferences he has attended.