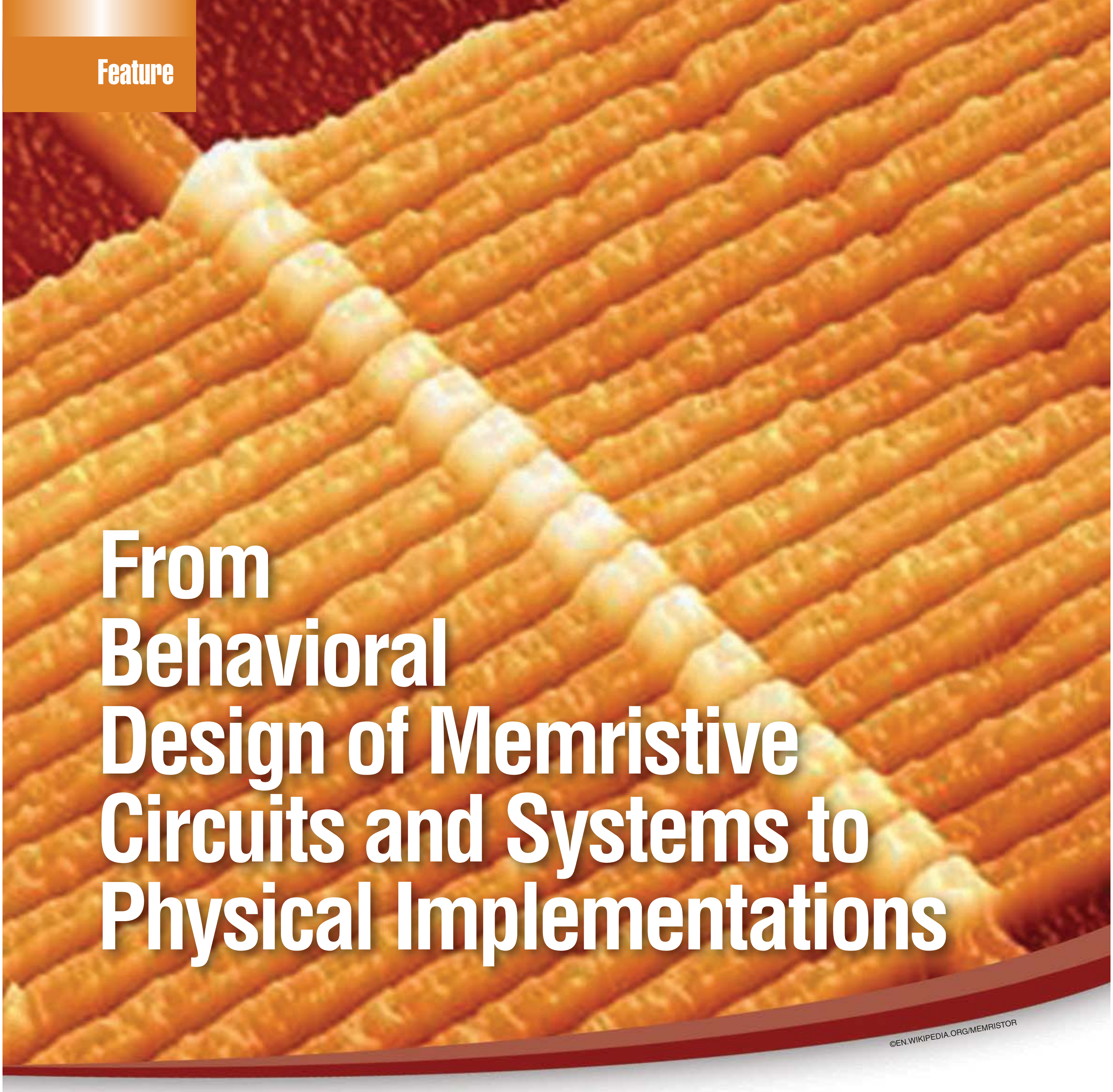

©EN.WIKIPEDIA.ORG/MEMRISTOR

From Behavioral Design of Memristive Circuits and Systems to Physical Implementations

Nima TaheriNejad and David Radakovits

Abstract

Since Hewlett Packard (HP) announced the passive fabrication of their memristors, various memristive technologies–as a promising emerging technology–have gained ever-increasing attention from the researchers. Although a natural application is using them as memory units, there have been several works in the literature showing their utilization in circuits and systems. While research on various aspects of memristive circuits and systems has been proliferating, the majority of these works are based on simulations at different levels of modeling abstraction. Simulation is a very helpful design tool, and there have been several efforts in modeling memristors; however, we contend that at this point these simulations represent the reality of the behavior of memristors, especially in a circuit or system set-up, only to a very limited extent. We show how this negatively affects the reproduction of designed circuits and systems in different simulation levels, and more importantly in a real-world set-up with physical implementation. Following that, we look into some considerations which can improve the reproducibility of the circuits and systems to be designed in the future. We conclude the paper by suggesting certain approaches to tackle these practical challenges at device level as well as circuit and system level.

Digital Object Identifier 10.1109/MCAS.2019.2945209
Date of current version: 19 November 2019

1531-636X/19©2019IEEE

I. Introduction

Reducing energy consumption is a crucial goal in the current circumstances of rapidly growing computational load. Mobile systems such as smartphones, embedded systems, wearable electronics, and Internet of Things (IoT) devices, which are often powered by batteries or rely on energy harvesting, require optimal utilization of the available energy [1]. A key factor in the energy budget of modern computing devices is memory [2]–[4]. In a modern chip, the number of transistors required to store data has a significant and increasing impact on the total transistor count [5], and consequently on the production cost. A promising solution for these problems is using memristors.

Although memristive behavior has been observed before [6], [7], a turning point for this type of basic circuit elements was when Hewlett Packard (HP) presented some of the (circuit level) applications of their passive solid-state Resistive Random Access Memory (ReRAM) devices with memristive characteristics in 2008 [8]. They advocated the memristors and their applications in the scientific community, especially that of circuits and systems. Thanks to their non-volatility, memristors could decrease the overall power consumption of the system dramatically [9]. Moreover, the relatively simple structure of memristors, allows compact implementations (device size of sub 10 nm × 10 nm [10] and 3 nm × 3 nm has been already reported [11], [12]) which can reduce their size up to one tenth of their Random Access Memory (RAM) counterparts [13]. It is worth noting that memristive behavior is not limited to ReRAMs. Other devices such as Phase Change Memory (PCM) and Spin Transfer Torque (STT) also show similar behaviors as described by Leon Chua in 1971 [14]. Since in many cases these devices face similar challenges, in this paper, we refer to them under the umbrella term of “memristors.” However, one should keep in mind that each of them has a different mechanism of operation which needs to be taken into account while working with them.

A natural candidate application for memristors has been in memory systems [9], [13], [15]–[22]. The possibility of integrating 1 TB of storage on a single chip [23] makes this technology a very attractive candidate for memory-intensive big-data applications [24]. Especially given that they can be integrated with Complementary Metal-Oxide Semiconductor (CMOS) technology with minimum changes. For examples, see MOSIS C5 CMOS [25], or CMOS Back End Of Line (BEOL) Memristor service [26], or the news on the planned offering of Taiwan Semiconductor Manufacturing Company (TSMC) in 2019 [27]. More importantly, they can be used for purposes other than secondary memory, code-storage, or similar conventional ways non-volatile memories are often used [24]. There have been several efforts in using memristors for implementing various logical functions [28]–[44], calculations [38], [45]–[49], and other applications [50], [51] such as learning [51]–[55] and even cancer detection [56]. Fig. 1 summarizes some of the major events in the memristive community since 2008 [57].

Logic circuit design is the key to the development of memristor-based computing systems. A noteworthy observation in this regard is that the majority of these logics are inspired by CMOS and in a certain way mimic behaviors of a CMOS circuit or replace parts of it. Among these logics CMOS/Memristor Threshold Logic (based on the Logic Threshold Gate (LTG)) [28], [37], Ratioed Logic [28], [30], and CMOS-like Logic [28] are among the most prominent ones. One of the major common properties of all of these logics is that they operate in the voltage domain (information and logical states are represented in the voltage of a node). This includes some more recent ones like Scouting Logic [44] too. Due to the voltage representation of the values, they often compete with their traditional and far more mature CMOS counterparts. However, memristive technology can be more successful if its native properties such as having memory are exploited. This idea has been used in IMPLY Logic [31], [35], [64] and Memristor-Aided Logic (MAGIC) [39], [43] which fundamentally operate in the memory domain. That is, the information and logical

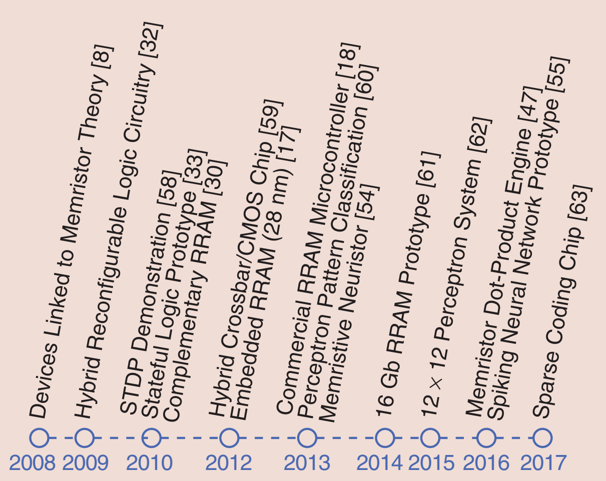
Devices Linked to Memristor Theory [8]
Hybrid Reconfigurable Logic Circuitry [32]
STDP Demonstration [58]
Stateful Logic Prototype [33]
Complementary RRAM [30]
Hybrid Crossbar/CMOS Chip [59]
Embedded RRAM (28 nm) [17]
Commercial RRAM Microcontroller [18]
Perceptron Pattern Classification [60]
Memristive Neuristor [54]
16 Gb RRAM Prototype [61]
12 × 12 Perceptron System [62]
Memristor Dot-Product Engine [47]
Spiking Neural Network Prototype [55]
Sparse Coding Chip [63]
2008
2009
2010
2012
2013
2014
2015
2016
2017

Figure 1. A summary of major events and milestones in memristive community according to [57].

Nima TaheriNejad and David Radakovits are with the Institute of Computer Technology, TU Wien, 1040 Vienna, Austria (email: nima.taherinejad@tuwien.ac.at, david.radakovits@tuwien.ac.at).

Among the top articles in the memristive circuit and system literature, less than 15% are verified by implementation and measurement.

states are represented as the memory states stored on the memristor. Therefore, no memory read or write is necessary before and after the logical operations. This is a substantial change compared to how CMOS circuits operate. Operating in memory domain is where CMOS has a significant disadvantage with respect to memristive technologies, since it requires a significantly larger area as well as power consumption to do a similar operation. Consequently, in-memory operation is the most propitious path for memristors as an emerging technology, especially considering the von-Neumann bottleneck which regards the constraints in data transmission between memory and processing units [57]. Few recent studies [42], [65]–[69] performed on in-memory computations have already shown promise.

The rest of this paper is organized as follows; In Section II, we first present our findings regarding the proportion (or from our point of view, in this case, disproportion) of physical implementation and simulation among memristor-based systems and circuits in the literature. Then, in Section III, we argue as to why this disproportion is very important in this field and should be ameliorated in the future. Next, in Section IV, we delve deeper into the shortcomings of models and some designs by presenting three examples. These experiments show how proper operation of memristive circuits cannot be taken for granted as it is often done in many memristive system and circuit designs. In Section V, we provide some suggestions as potential directions to take in order to tackle existing challenges. Finally, we draw our conclusions in Section VI.

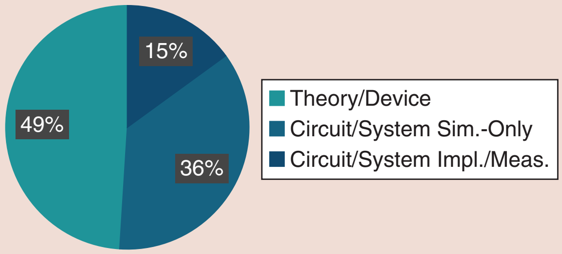

Figure 2. Distribution of "Top 30" literature works on "memristors" showing the share of circuit/system implementation/measurements (15%) vs. circuit/system simulations or theory/device level works (85%). We note that 52% of the search results were indeed circuit/system articles (considering both practical and simulation-only works).

II. Physical Implementation & Verification in the Literature

Studying the literature, to our surprise, there is a considerable shortage of physical implementation and verification among the myriads of circuits and systems proposed for memristive devices. To understand the dimensions of this shortage better, we conducted a survey in the literature. In this survey we searched five keywords, namely 'memristor', 'memristor circuit', 'memristor system', 'memristor device', 'memristive circuit', and 'memristive system', and picked the first thirty results (considering papers appearing in more than one search only once and dismissing it in subsequent appearances) and checked whether they were verified based on physical implementation or not. To make sure that our sample is not biased or limited to a certain community we chose "Google Scholar" as our search engine. "Google Scholar" searches a wide range of available materials including world-wide patents databases, research databases (such as IEEE, Elsevier, Springer, ResearchGate, ArXive, and others), university databases (for theses), and other published papers on the web (white papers published by companies or papers published by the authors on their personal website). Our results, visualized in Fig. 2, show that from the 142 works (38 of the 180 results were repetitive appearances), 51% were indeed circuit and system designs, however, only 30% of them (15.5% of all the search results) were based on physical implementation or measurements.

Knowing that some researchers prefer to use specific device names, we ran the same experiments with new keywords. Given the better reception of ReRAM among various types of memristive device, we chose the following keywords; 'RRAM', 'ReRAM', 'resistive RAM', 'resistive memory', 'resistance switch', 'resistance switching', 'resistive switch', 'resistive switching.' The result for this set of search shows that the majority of the papers using this keyword are at a device and model level, and not circuit or system design. From the 211 works, only 11% were circuit and system designs, of which 78% were physically implemented. This means that only 9% of the total number of works found using those additional keywords were implementations.

We also did a brief search with PCM, where out of the 30 results, 9 papers (30%) were circuit or system designs, 89% of those containing actual implementations, yielding 27% implementations in total. Surprisingly, none

> **Memristive technologies are young and on their way to maturity. Hence, the path to maturity is no less long for the models describing them.**

of these top 30 papers are concerned with modeling, even though the majority of them are device level works.

Considering the overall search combining the findings using all of above keywords, visualized in Fig. 3, the share of circuits and systems verified by physical implementation and measurements is as low as 12% (48 out of a total of 383 works)[1]. We contend that this disproportion between practical works and simulated designs is alarming and has negative effects. Our search shows some indications that in communities using PCM and ReRAM keywords, they are more aware of (and hence pay more attention to) the importance of practical implementation. However, the ratio of circuit and system works done in those communities seems to be lower.

We note that such disproportion between simulation and implementation exists in certain other communities such as digital CMOS circuit design too. However, it is important to note that in that case, such disproportion is justified by the maturity of the models and Computer-Aided Design (CAD) tools developed for CMOS technologies thanks to the heavy investment of major companies and consequently comprehensive and continuous effort of engineers and scientists in device and circuit development and modeling. Therefore, a successful simulation is, to a large extent, a good guarantee of a successful implementation for the majority of digital CMOS circuits. Despite the maturity of CMOS technology and respective models and CAD tools, a successful simulation is much less of a guarantee for a successful implementation in the case of analog CMOS circuits. Therefore, in the respective community, new designs, and concepts are often well-received and commonly spread, if and only if they are backed up by physical implementation and verification. In consequence, the proportion of high-quality publications verified by physical implementation is much higher compared to their digital counterpart.

III. The Pitfalls of Circuit & System Simulation

We contend, that in the memristor community the shortage of physical implementation and verification is an important problem due to three main reasons, which we discuss here.

A. Immaturity of Technology

In the case of CMOS transistors, the material and fabrication process is to a very large extent determined and clear. The exact technology and fabricating company determine further details. When it comes to memristors, in contrast, this can vary significantly in terms of material, fabrication, and operation dynamics. Some of the circuits and systems are designed for "memristors" without any considerations for specifics of operations of different device types. Even among a single type, for example, ReRAMs, although Titanium dioxide (T_iO_2) is one of the more well-known types, the literature contains many other materials and fabrication processes such as Tantalum Oxide (T_aO_x) [70], [71], Hafnium Oxide (H_fO_x) [72]–[74], amorphous silicon [75], carbon nanotubes [76], ferroelectric [77], SiNW [56], and silver-based ReRAMs [78], [79]. Given the sparse and ad-hoc approaches towards developing these memristors, they often remain in a pre-mature or maturing phase. Significantly less mature compared to any CMOS technology of the day. This negatively affects their characterization as well as their reproducibility. Some of the main challenges to be addressed at device level include device variability, cyclic variability, OFF/ON ratio, endurance, retention, and device speed.

Fig. 4 shows our measurement results performed on 8 ReRAMs of the same technology, fabrication round, and die packed in a single package[2]. In our measurements,

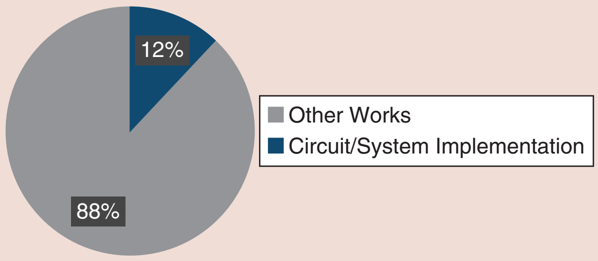

Figure 3. Distribution of "Top 30" literature works with all keywords (derived from memristor, ReRAM, and PCM) showing the share of circuit/system implementation/measurements (13%) from all articles found in our searches. Other works include theoretical papers, device level, or circuit and system designs which were not implemented (measured).

[1] The share of circuit and system simulations-only is 15% (57 out of 383 non-repetitive works).

[2] These memristors use metalization of chalcogenide material as the switching mechanism. We do not have permission to publish more specific details.

20 pulses of 800 mV amplitude and 25 μs width were used to drive the memristors to their ON state (or low resistance, shown in solid circles) and 20 pulses of –400 mV amplitude and 25 μs width were used to drive them to the OFF state (or high resistance, denoted by hollow circles). Each memristor was driven to the two opposite states and measured ten times for each state. We see that for the first 3–4 memristors (each column/group depicts the measurements of one memristor), shown in (shades of) blue, R_{on} and R_{off} values overlap in some measurements. This trend improves with a somewhat smaller or larger gap between the two values for other memristors, shown in different shades of green. Using memristors with overlapping R_{on} and R_{off} values is practically impossible. Therefore, either they should be identified beforehand and avoided, or other mechanisms should be devised to drive them into two distinct states. We note that the ratio of $R_{\text{off}}/R_{\text{on}}$ also varies from 57 to 1. This wide range of variation makes working even with "functional" memristors (that is, memristors without overlapping values of R_{on} and R_{off}) very difficult. Another difficulty is the variation in absolute values. The minimum R_{on} in this set of measurements is 17 kΩ, whereas the maximum was 1.17 MΩ, which shows a ×68 difference. The smallest observed R_{off} was 172 kΩ and the largest 1.59 MΩ, a ×9 difference, which is very large but significantly better than variations in R_{on}. It goes without saying that as the technology matures, we can hope to see more of the green memristors than the blue ones and more uniformity in fabrication.

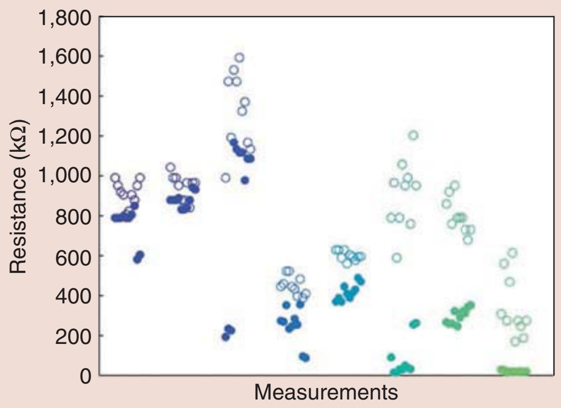

Figure 4. R_{on} and R_{off} measurement results for 8 ReRAMs in a single package. Each column represent measurements of one memristive device. Solid circles denote ON resistance, whereas hollow circles mark OFF resistances.

Table I.
Comparison of generic behaviors of four prominent models [6]. Where ○, ◐, and ● show unacceptable, acceptable, and good performance, respectively.

	I-V/I-t	Kinetics	CRS	Multi-Bit	State-Depe.	Fading	Flexibility
Strachan [87]	●	●	●	●	●	●	○
VTEAM [88]	○	○	●	◐	◐	◐	●
Stanford [89]	●	◐	●	●	●	●	●
RWTH [90]	●	●	●	●	●	●	●

B. Immaturity of Models

As mentioned, referring to memristors does not imply any specific technology, material or fabrication process. Even more specific terms such as ReRAM (or PCM) do not imply the same material or properties in that category. Various memristors have different properties and since they are often fabricated in research laboratories in small quantities, they are not available to other researchers and potential users for additional tests, experiments, or modeling. Consequently, the quality, verifiability, and scalability of respective models remain often very limited due to restricted experiments and modeling efforts possible at the research institute fabricating the memristor. A major issue, in this case, is that most models are developed at single device level under a few typical test scenarios (such as characterizing hysteresis loops, e.g. [80]–[83]). Although these models capture certain characteristics of the memristors, they rarely manage to sufficiently predict the behavior of the device under real application scenarios where the usage is substantially different from the tests. Moreover, the interaction between the devices and the environment is often neglected, leading to further inaccuracies when it comes to circuits involving more memristive devices, working in an uncontrolled environment.

We should bear in mind that even if the basic principles of operations of the memristor are known to us, some physical details, especially regarding the switching process, are yet rather unknown [6]. Based on this premise, very recently Menzel et al. [6] conducted an investigation of the quality of various prominent models on modeling the generic behavior of Redox-based memristive devices. This behavior includes voltage and current characteristics (I-V/I-t), non-linear switching kinetics, complementary resistive switching, multi-bit data storage, state-dependency, fading memory capability (asymptotic behavior), and model flexibility. Table I provides a summary of their study. We note that the performance is not measured against any real memristor, rather against generic behaviors of Redox based memristors (thus excludes any other types). Therefore,

Be aware of limitations! Abstractions for high-level simulations are very little reliable, so are the low-level simulations using only nominal values.

it does not show how good or bad they may model a specific physical memristor (of this or any other type). Additionally, this set of criteria does not provide a full insight to the behavior of a memristor.

As mentioned above, some models do consider the non-linearity and kinetics, which to some extent reflects the speed of the device. However, even for speed, more works such as extracting layout and wire parasitics and considering them in the models are missing. In addition, noise, variability, and temperature are also not reflected properly in the models. It is known that temperature can affect properties such as the mobility in the devices [57], which can, in turn, affect its behavior. This effect is stronger in some materials such as VO_2 and NbO_2, leading to the observation of negative differential resistance [57]. A property which has been used to create oscillation [84] and neurons [54], [84]. Although this behavior for these specific types has been modeled [85], [86], main-stream generic models do not consider any temperature effects.

C. The Nature of Designed Circuits

Memristive circuits such as neuromorphic systems [24], [53], [60], [91] or Threshold Logic [28], [37], and some other custom circuits [53], [92] treat memristors as a device with a continuous range of values. This is in nature similar to analog circuits rather than digital circuits and therefore, requires similar care and approach as in analog circuits. That is, new designs and concepts need to be backed up by physical implementation and verification in order to be reliable. On the other hand, in many of the digital memristive circuits, a highly non-linear behavior (sharp switching based on thresholds) is assumed and used. In reality the extent of this non-linearity is more limited and the analog continuous nature of memristors has a strong presence, making a notable difference in practice.

For example, in IMPLY Logic [31], [35], [64], the state change operation heavily depends on the voltage difference between V_{COND} and V_{SET}, which is significantly smaller than the power supply, providing a very weak non-binary drive for the change. On the other hand, it is assumed that if the voltage across a memristor is below the threshold, that memristor will not experience any changes. Whereas, in practice, that memristor can experience a state drift (we will show and discuss this more in Section IV-B). Although the input and output are considered as digital values, the dynamic of the operation is significantly closer to the traditional analog circuits than the digital circuits. Again, requiring considerations that are common in the analog domain but are often omitted in the digital domain.

IV. Reproducibility Challenge

Shortcomings discussed in Section II and Section III negatively affect reproducibility of designs in the real world and consequently, put a question mark on the extent of their practicality and usefulness. In some cases, these issues lead to inconsistencies and reproduction issues in simulations as well, which poses an even greater challenge to the community. We have tried to reproduce some of the existing works in the literature and faced certain problems which we briefly report here. We note that to answer questions such as “which logic is more reliable in practice?” a comprehensive set of studies is in order. In such a study one should test them in practice at the presence of practical challenges, and assess the difficulty to come up with solutions to resolve any problems they may face in practice. Only then a good perspective of the advantages or disadvantages of one logic or another can be discussed. Even though, we hope that this paper and particularly this section will ignite such discussions and plant the seed for such studies, such in-depth analysis and discussion is not in the scope of the present paper. Here we narrate our limited experiences to highlight the challenges of practical implementation and attract the attention of researchers and engineers to their importance while leaving comprehensive comparative studies for future works.

A. Memory Example

There are several papers in the literature on various aspects of using memristors as memories [9], [13], [15], [16], [19]–[22]. Although various models capture different characteristics of memristors, interestingly enough, so far and to the best of our knowledge, none of the existing ones model the leakage or state drift fully and properly. A factor that substantially affects the performance of memristors as memory. We bear in mind that there have been works on “history effect” [93], [94], however, that is a different concept. History effect concerns the eventuality of a steady state in memristors, which is independent of its initial state, after application of a certain input pattern. In studying history effect, the authors explicitly mention that they do not consider state changes in the absence of input [93], whereas the leakage or

Real devices show behaviors that may not be represented by current models, yet they can affect the function of the circuit and the system.

state drift discussed here regards the state changes in the absence of any input.

There are several suggestions regarding compensation of read-out mechanism, nonetheless, there are no concrete solutions addressing the leakage. To evaluate these effects we designed and implemented (shown in Fig. 5) a memory write and read circuit [95] and measured the maximum retention time of Knowm "BS-AF-W" memristors [96]. Our measurement showed 81 hours of retention time (488 reads, 10 minutes apart) for our memory system. In our experiments, the read operations were not compensated and hence could affect the retention time. However, in a different experiment with more frequent reads (every 1 second), we managed to have 6000 correct reads before a state change. Therefore, even though the 488 read operations contribute in reduction of retention time, the two experiments prove the presence of a different effect which we associate with the leakage. Since currently, no models for this effect exist, we could not simulate this effect. Similarly, no other memristive memory system design can be thoroughly verified in simulation (e.g., regarding its retention time). At least not using the models currently available to the public and academia. This indicates the need for developing new and more comprehensive models. More importantly, it highlights the importance of the physical implementation and verification of memristive systems. We note that there are memory products in the market, e.g., [97], which use memristors. Therefore, there is no doubt that there are and can be more memristive memory systems, for which similar characterization experiments are done. However, we could not find any similar reports accessible to the public, reporting those important characteristics we measured.

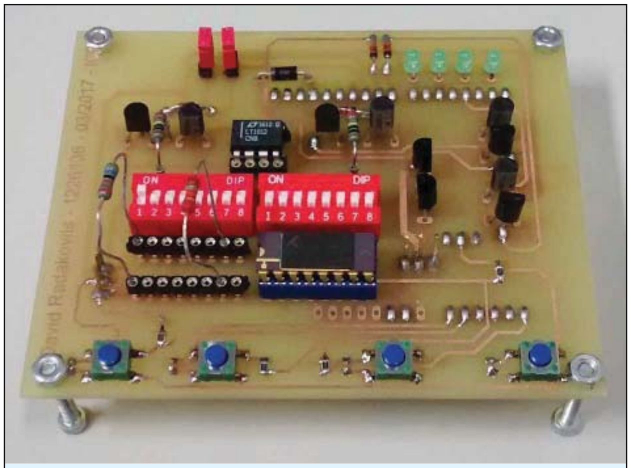

Figure 5. Our implementation of a complete memory system on PCB, populated with memristor chip and required circuits to read and write.

B. Logic Example—IMPLY

As one of the most prominent memristor-based logic design methods, IMPLY has been extensively used [31], [34], [38], [48], [49], [99]–[102]. There are also various works in the literature on its design and implementation as well, e.g. [34], [35], [38], [45]. This also includes closed-form formulas and determined boundaries regarding the value of various circuit elements necessary to implement an IMPLY [38], [45]. In the literature, however, often the circuit simulation of systems using IMPLY is skipped (see e.g., [45], [48], [99], [100]), given the assumption that the basic gate implementation as shown in [34] is functional. However, to verify our IMPLY based system published in [48], we tried to run the respective simulation in SPICE. The common assumption in $p \rightarrow q$ (p IMPLY q) is that p maintains its state while q changes its state to hold the result [31], [45], [48], [100]–[102]. Despite the fact that many of the designed systems work strictly based on this assumption, to our surprise, this does not seem to be always the case.

One of the problems with the closed-form calculations is that they assume a resistive switch with fixed resistance before and after crossing the threshold voltage. Whereas in reality, the memristors experience state drift on both sides of this threshold, which on itself affects the switching process. According to our simulations using four different models (namely, Biolek [98], Yakopcic [103], Joglekar [104] and TEAM [105]), in a single operation, or as shown in [34] in a few step operation, the parameters could be set such that p can be considered (in some cases only marginally) as keeping its previous states. However, the state drift in a sequence of operations leads to a potential loss of state for p and consequently false results (see Fig. 6). The only model in which we managed to simulate IMPLY with a small enough state drift that does not cause a loss of state is TEAM [105]. However, we notice that for doing so we had to set the model parameters arbitrarily and far from the characteristics of the real memristors we have at hand.

Taking all the above into account, the question of the practicality of a physical implementation of IMPLY in which p does not lose its state after a few operations remains an open question to us. However, this problem can be easily remedied. That is, to save the content of the p, whenever it is needed in the future. Despite the overhead, this approach seems to be practically more attainable.

C. Logic Example—MAGIC

Another prominent memristor-based logic is MAGIC [39], [43]. Several logic functions can be implemented using MAGIC, one of them being an n-input NOR, which naturally forms a NOT gate if only one input is used. Given its attractiveness, we have tried to implement it. This logic also faces several challenges in overcoming practical adversities of memristive circuit implementations.

Some of these challenges can be observed already in simulations. For example, our simulations of MAGIC NOR gates have shown that variation in memristor parameters, i.e., R_{on}, R_{off}, threshold current/voltage, and switching dynamics, cause the robustness to decrease dramatically. Simulations of these circuits were conducted using VTEAM [88]. Since VTEAM is available as a Verilog-A model only, we implemented it in LTspice. The implementation can be found in [106]. The used parameters for VTEAM can be found in Table II. Fig. 7 shows a sample result of our simulations, where the effect of the initial state on the performance can be seen, particularly timing. For example, as it can be seen in this figure, a 5% deviation from a 100% initial state leads to a double inversion time. Not taking this into account can lead to incomplete state changes and eventually false results in operations. It is appealing to think that by taking a very long operation time this problem should go away, however, we need to bear in mind that a longer operation time leads to a larger state drift in the input memristors. Therefore, even though the state change in the output memristor would be thus improved, a similar problem would be introduced to the input memristor which would bring us back to the same problem in further operations.

Moreover, as we showed in Section III-A, the mismatch between memristors, which can have a similar effect as to not fully ON or OFF initial states, could be more than an order of magnitude larger than what is shown here. Therefore, finding pairs or groups of memristor with similar enough properties to implement the gate is challenging too. Even harder is their inter-operation. Whereas we successfully implemented MAGIC NOT gates (i.e., gates involving two memristors), due to parameter variations existing in the memristors available to us, it was barely possible to implement 2-input NOR gates in MAGIC. It is important to notice that our implementation of this gate did not prove to be particularly reliable and repeatable.

We would like to remark, that this does not undermine the value and the promise of this design. The question for us (and we believe the rest of the community) is

Table II.
Parameter values used in VTEAM.

Parameter	v_{off}	v_{on}	α_{off}	α_{on}	R_{off}	R_{on}
Value	0.7 V	–10 mV	3	3	1 MΩ	10 kΩ
k_{on}	k_{off}	w_{off}	w_{on}	w_C	a_{off}	a_{on}
–0.5 nm/s	1 cm/s	3 nm	0 nm	107 pm	3 nm	0 nm

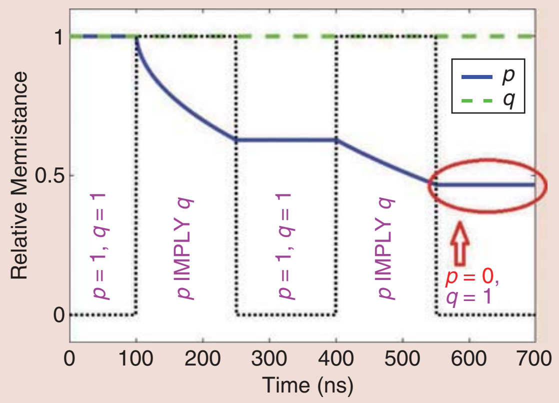

Figure 6. Simulation of two consecutive IMPLY operations without refreshing state of memristors (this example uses Biolek model [98]). Memristor p (blue solid line) looses its state during the operations (black dotted line going to "1" shows when IMPLY operation was performed), while q (green dashed line) keeps its state as expected.

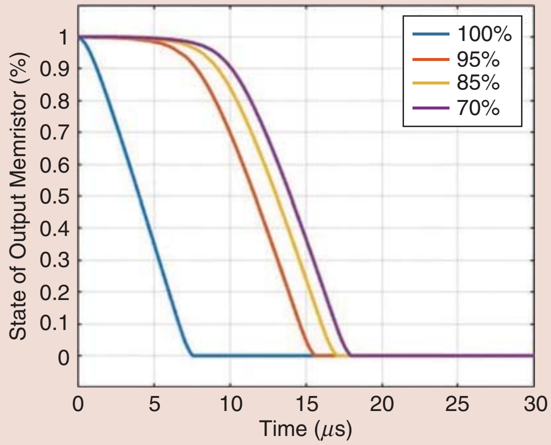

Figure 7. Dependability of the result of a MAGIC "NOT" operation on the initial state of the input memristor. The NOT operation should yield $in = 1 \rightarrow out = 0$, but if the state of $in = 0.95$ it already consumes twice the time to reach $out = 0$, compared to the fully $in = 1$ state. After the original operation time of 8 μs, the $in = 0.95$ yields $out = 0.89$ (which is a false result).

how a more reliable and repeatable implementation of MAGIC can be made possible? Be it a solution at device level (e.g., less variation could help), circuit level (e.g., could any extra circuit help?), or system level. For example, at system level, the state drifts of input memristors due to longer operation time (which improves the state change in output memristor) can be compensated by refreshing the state of the input memristors (as we suggested for IMPLY too). Similar to the IMPLY example above, practical issues may be therefore overcome with certain considerations which affect the system at higher (e.g., at algorithm) levels too.

V. Moving Forward

In previous sections, we discussed major challenges that memristive circuits and systems face in practice. We would like to emphasize that this is not meant to undermine the practicality of building memristive circuits and systems. It has been repeatedly proven that it is possible to build memristive circuits which work in practice, for example, see [107] and [108], among many others. Our intention here is to raise awareness about these challenges and by considering these issues, empower engineers and designers to design circuits and systems which have a shorter path to practical implementations. Other than taking the challenges of the current state into account, there are certain steps that we, as a community, could take to alleviate these adversities and reduce the existing challenges. In this section, we summarize some of these potential steps, which provide formidable research questions and challenges. Addressing them could have a positive effect on the design and implementation of memristive circuits and systems. Some of the challenges to be addressed at the device level are:

Device Variability As we saw in Section A, particularly the example of Fig. 4, the variation between devices, even within the same package, is so large that the R_{off} of a memristor can be smaller than the R_{on} of another in the same package. This makes it impossible to consider even an arbitrary range of these values, within which both memristors can be considered either ON or OFF. Therefore, the biggest hurdle of practical implementations is this extremely large variation between device characteristics. Reducing the size of memristors seems to be a key solution to this problem. Material research also could lead to improvements [109]–[111].

OFF/ON Ratio Even in a single memristor, it is important to have a minimum of $R_{\text{off}}/R_{\text{on}}$ so that the two different states can be distinguished. As evident in our example, Fig. 4, this ratio undergoes a large variation too. Values equal to or smaller than 1, which speak of practically indistinguishable states, are the major problem. In addition, certain applications require much larger ratios. This issue, even though has a different effect in practice, solution-wise, follows the previous issue closely. Solutions which could alleviate the problem of device variability could help in improving the OFF/ON ratio as well.

Cyclic Variability Variations during the lifetime of the memristor is a well-known phenomenon as well [112]. This challenge might be significantly more difficult than the device and ratio variation to address at the device level. Considering it at circuit and system level may be a somewhat easier approach. However, in the presence of the previous two issues, this challenge has little priority. Moreover, in many cases, its pattern is hardly distinguishable. For example, we could not see a particular pattern of cyclic variations in our measurements, however, once this pattern is distinguished and modeled, circuit and system designers could better consider it in their designs.

Endurance Currently, the lifetime of memristors is not very high in all technologies. In some cases, it could be as low as 10,000 cycles [112], [113]. That is certainly a limitation which could affect the wide-spread use of them, particularly in applications such as in-memory logic and computations which come with frequent changes of state. Therefore, device level research is needed to improve this aspect too. However, this problem seems to be of a secondary priority compared to device variability and OFF/ON ratio.

Retention This problem mainly concerns memory applications. However, given that in future architectures under investigation (which try to use in-memory computation), memristors are going to act both as memory and computation unit, retention becomes an important aspect for them as well. At this stage, this is also a secondary concern, but very important for the wide-spread use of these devices in consumer electronics. Material and device research seems to hold the answer to this question.

Device Speed In the literature a large range (from sub-nanosecond to microseconds) of device speeds can be observed [96], [114], [115]. This affects the performance and power consumption of the systems using memristive circuits and systems. Improving the speed of state changes at device level can make memristive circuits and systems more competitive in the CMOS-dominated market and hence improve its reception by the industry and users.

Sneak Path is a well-known issue in the literature [116], [117] and there have been efforts in reducing the effect of sneak path [118], [119]. For example, in [118] the authors use a system with buffer amplifiers to reduce the number of memristors which would normally be affected by the sneak path. Thus, they alleviate the problem, even though it does not go completely away. In addition, the complexity of the system is increased

While real devices are hard to access, awareness of potential challenges and thorough low-level simulations are crucial for a reliable design.

and issues such as amplifier noise are introduced. A simple solution is adding a transistor to switch a memristor in or out of the crossbar. The main problem with this structure, also called 1T1M, is the additional transistor gate wire which makes the crossbar less compact compared to a cell structure that requires only a memristor. The required transistors can have an even larger effect in the total area of the crossbar. Device level solutions such as larger OFF/ON ratios also could alleviate the sneak path problem. In that case, 1T1M could change from a functional necessity to a luxurious addition for applications which require higher reliability and can accept the additional costs of 1T1M structures.

At the circuit and system level, the following steps can help the design and implementation process;

Models Memristor models still have a long way to completion. Modeling temperature effects, device variations (particularly the variations in the absolute value of R_{on} and R_{off}), threshold variation, cyclic variations, leakage (retention time), and endurance are some of the practical effects which, to the best of our knowledge, are not reflected in any of the existing models. Modeling these effects — especially in one integrated model— can enable more realistic simulations, particularly corner simulations.

Parasitics More often than not, memristor models are developed in a laboratory environment and are based on on-die measurements. They do not consider any of the parasitics which can be formed due to the wires and connections as well as the layout of the circuits. Creating and using models for these effects can help in the design process and lead to more realistic simulations which better represent implementations.

Functional Simulations Using better models which reflect the reality better, in terms of values and variations and include leakage, extracted parasitics, and unideal initial states (memristors that enter an operation without having reached their full ON or OFF states in previous operations), the circuits can be thoroughly tested to see whether they are functional under all those circumstances or not. If not, the range of functional operations, as well as more problematic issues, can be identified. The former allows selecting suitable applications or technologies and the latter helps in devising solutions to address the relevant functional issues.

Corner Simulation Evaluating a circuit in different corners, such as the ones mentioned above, help in predicting the chances of prototypes being functional, or selection of the technology to fabricate the designed circuit, as well as the universality of the design. That is, how much of variation in those parameters the design can tolerate before showing functional problems? Consequently, this helps to find suitable technologies since technologies which have a variation within those bounds can be used to implement that design.

Design and Test Awareness Once designers are aware of the challenges of practical implementations, as we discussed in this paper, they can design their circuits such that they can overcome or better tolerate these adversities. Whether these solutions be at circuit level, or at system level (e.g., the one we proposed here for the practical problems of the IMPLY gate). It is also important to test the circuits and systems against them, both in simulations and in practice.

Integration Many of working implementations are based on Integrated Circuit (IC) solutions [107], [108]. That is, the memristors and the CMOS circuits are on the same die or in the same package. This seems to be a possible solution for more reliable implementations. This facility is hardly available to the public but it seems to have a considerable effect since most published practical implementations are ICs. With the announcement of TSMC [27] regarding their new fabrication rounds which include memristive devices, this could change and we hope to see more practical implementations.

VI. Conclusions

Our experiments above show critical issues to which the community needs to pay more attention to create more effective and realistic memristive circuits and systems. First, is the necessity of developing improved, more comprehensive and more realistic memristor models which represent the behavior of real-world memristors better. In particular, factors such as device, threshold and cyclic variations, as well as temperature effects, leakage (retention time) and endurance. This point was presented in the example of the memory system we have developed. There, we showed factors such as retention time or realistic refresh cycles cannot be simulated since parameters such as leakage or device variation are not fully or properly represented in the existing models. Completion of models presents a longer-term challenge since the memristive technology itself is evolving. This process of evolution requires renewing respective models to better represent the physical behavior of newer technologies.

Second is the necessity of running full functional and corner circuit-level simulations for designed systems. Current circuit models have room for improvement and they do not provide a comprehensive insight as to what may happen in the physical world. Nevertheless, they do provide an understanding of certain potential problems in the designed systems or points where a deeper investigation is necessary. An example of this point is the IMPLY operation where, as presented here, most of the existing models show that with realistic model configurations, it is most likely that consecutive operations lead to loss of information on the *p* memristor. However, many system designs do not consider this potential loss of information which can have a significant (and potentially negative) effect on the operation of these systems. Therefore, certain considerations (such as refreshing the value of *p* in this example) are in order, to ensure the proper operation of higher level systems using IMPLY.

Lastly, we contend that many of the current memristive circuits, such as IMPLY for example, although having digital inputs and outputs, operate in a manner that can be identified better with analog CMOS circuit operations, rather than their digital counterpart. Therefore, to reliably verify their operation, and to realistically characterize them, they need to be physically implemented and tested. A matter that we believe is relatively overlooked in the community and it deserves more attention. These practical implementations show us the way forward to improve both devices and memristive circuits and systems.

Nima TaheriNejad (S'08-M'15) received his B.Sc. and M.Sc. degrees in electrical and electronic engineering from the Babol Nushirvani University of Technology, Babol, Iran, and the Iran University of Science and Technology, Tehran, Iran, in 2007 and 2009, respectively; and the Ph.D. degree in electrical and computer engineering from The University of British Columbia, Vancouver, Canada, in 2015.

He is currently a "Universitätsassistant" at the TU Wien (formerly known also as Vienna University of Technology), Vienna, Austria, where his areas of work include self-awareness in resource-constrained cyber-physical systems, embedded systems, systems on chip, memristor-based circuit and systems, health-care, and robotics. He has published two books and more than forty peer reviewed articles. He has also served as a reviewer, editor, organizer, and chair for various journals, conferences, and workshops. Dr. TaheriNejad has received several awards and scholarships from universities and conferences he has attended.

David Radakovits received his B.Sc. degree in electrical engineering and information technology from TU Wien, Vienna, Austria, where he is currently a graduate student. His B.Sc. thesis was on "Binary storage on Memristors" and he continues to work on memristive circuits and systems for his M.Sc. degree in Embedded Systems.

His main research interests are memristive circuits and systems, embedded systems and systems-on-chip, and hardware security. He has two published papers and two papers under review at IEEE journals and conferences on different aspects of memristive circuits and systems.

References

[1] R. Iyer and E. Ozer, "Visual IoT: Architectural challenges and opportunities; toward a self-learning and energy-neutral IoT," *IEEE Micro*, vol. 36, no. 6, pp. 45–49, Nov. 2016.

[2] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. USENIX Annu. Tech. Conf.*, vol. 14. Boston, MA, 2010, p. 21.

[3] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan, "Full-system power analysis and modeling for server environments," in *Proc. IEEE Int. Symp. Computer Architecture*, 2006.

[4] A. Mahesri and V. Vardhan, "Power consumption breakdown on a modern laptop," in *Proc. Int. Workshop on Power-Aware Computer Systems*. Berlin, Germany: Springer-Verlag, 2004, pp. 165–180.

[5] *International Technology Roadmap for Semiconductors – System Drivers*, 2011th ed., ITRS Technology Working Group, 2011.

[6] S. Menzel, A. Siemon, A. Ascoli, and R. Tetzlaff, "Requirements and challenges for modelling redox-based memristive devices," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, May 2018, pp. 1–5.

[7] W. Rainer, D. Regina, S. Georgi, and S. Kristof, "Redox-based resistive switching memories nanoionic mechanisms, prospects, and challenges," *Adv. Mater.*, vol. 21, no. 2526, pp. 2632–2663, 2009. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.200900375

[8] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.

[9] B. Mohammad, D. Homouz, and H. Elgabra, "Robust hybrid memristor-CMOS memory: Modeling and design," *IEEE Trans. VLSI Syst.*, vol. 21, no. 11, pp. 2069–2079, Nov. 2013.

[10] B. Govoreanu et al., "10×10nm^2 Hf/HfOxcrossbar resistive RAM with excellent performance, reliability and low-energy operation," in *Proc. Int. Electron Devices Meeting*, Dec. 2011, pp. 31.6.1–31.6.4.

[11] A. Sinha, M. S. Kulkarni, and C. Teuscher, "Evolving nanoscale associative memories with memristors," in *Proc. 11th IEEE Int. Conf. Nanotechnology*, Aug. 2011, pp. 860–864.

[12] R. S. Williams, "Finding the missing memristor," 2010. Accessed on: Nov. 6, 2017. [Online]. Available: https://www.youtube.com/watch?v=bKGhvKyjgLY

[13] H. Kim, M. P. Sah, C. Yang, and L. O. Chua, "Memristor-based multilevel memory," in *Proc. 12th Int. Workshop Cellular Nanoscale Networks and Their Applications (CNNA)*, Feb. 2010, pp. 1–6.

[14] L. Chua, "Memristor – The missing circuit element," *IEEE Trans. Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971.

[15] D. Niu, Y. Chen, and Y. Xie, "Low-power dual-element memristor based memory design," in *Proc. ACM/IEEE Int. Symp. Low-Power Electronics and Design (ISLPED)*, Aug. 2010, pp. 25–30.

[16] Y. Ho, G. Huang, and P. Li, "Dynamical properties and design analysis for nonvolatile memristor memories," *IEEE Trans. Circuits and Systems I, Regular Papers*, vol. 58, no. 4, pp. 724–736, Apr. 2011.

[17] W. C. Shen et al., "High-K metal gate contact RRAM (CRRAM) in pure 28nm CMOS logic process," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, 2012, pp. 31–36.

[18] G. Hilson, "Imec, Panasonic Push progress on ReRAM," *eeTimes*, 2015. [Online]. Available: https://www.eetimes.com/document.asp?doc_id=1327307

[19] V. Baghel and S. Akashe, “Low power memristor based 7T SRAM using MTCMOS technique,” in Proc. *5th Int. Conf. Advanced Computing Communication Technologies (ACCT)*, Feb. 2015, pp. 222–226.

[20] M. Zangeneh and A. Joshi, “Design and optimization of nonvolatile multibit 1T1R resistive RAM,” *IEEE Trans. VLSI Syst.*, vol. 22, no. 8, pp. 1815–1828, Aug. 2014.

[21] N. Taherinejad, P. D. S. Manoj, and A. Jantsch, “Memristors’ potential for multi-bit storage and pattern learning,” in *Proc. IEEE European Modelling Symp. (EMS)*, Oct. 2015, pp. 450–455.

[22] N. Taherinejad, S. M. P. D., M. Rathmair, and A. Jantsch, “Fully digital write-in scheme for multi-bit memristive storage,” in *Proc. 13th Int. Conf. Electrical Engineering, Computing Science and Automatic Control (CCE)*, Sept. 2016, pp. 1–6.

[23] J. J. Yang, D. B. Strukov, and D. R. Stewart, “Memristive devices for computing,” *Nature Nanotechnol.*, vol. 8, pp. 13–24, 2013.

[24] S. Hamdioui et al., “Memristor for computing: Myth or reality?” in *Proc. Design, Automation Test in Europe Conf. Exhibition (DATE)*, Mar. 2017, pp. 722–731.

[25] “Neuro-bit,” Bio Inspired Technologies LLC, 2017. [Online]. Available: http://www.bioinspired.net/home.html

[26] Knowm, “Knowm Inc.,” 2017. [Online]. Available: https://knowm.org

[27] P. Clarke, “TSMC to offer embedded ReRAM in 2019,” *eeNews*, 2017. [Online]. Available: http://www.eenewsanalog.com/news/report-tsmc-offer-embedded-reram-2019

[28] I. Vourkas and G. Sirakoulis, *Memristor-Based Nanoelectronic Computing Circuits and Architectures: Foreword by Leon Chua* (Emergence, Complexity and Computation). Berlin, Germany: Springer-Verlag, 2015.

[29] S. Muroga, *Threshold Logic and Its Applications*. New York: Wiley, 1971.

[30] E. Linn, R. Rosezin, C. Kügeler, and R. Waser, “Complementary resistive switches for passive nanocrossbar memories,” *Nature Mater.*, vol. 9, no. 5, pp. 403, 2010.

[31] E. Lehtonen and M. Laiho, “Stateful implication logic with memristors,” in *Proc. 2009 IEEE/ACM Int. Symp. Nanoscale Architectures*. IEEE Computer Society, 2009, pp. 33–36.

[32] J. Borghetti et al., “A hybrid nanomemristor/transistor logic circuit capable of self-programming,” *Proc. Nat. Acad. Sci.*, vol. 106, no. 6, pp. 1699–1703, 2009.

[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 implications,” *Nature*, vol. 464, pp. 873–876, 2010.

[34] S. Kvatinsky, A. Kolodny, U. C. Weiser, and E. G. Friedman, “Memristor-based imply logic design procedure,” in *Proc. IEEE 29th Int. Conf. Computer Design (ICCD)*, 2011, pp. 142–147.

[35] E. Linn, R. Rosezin, S. Tappertzhofen, R. Waser, et al. “Beyond von Neumann logic operations in passive crossbar arrays alongside memory operations,” *Nanotechnology*, vol. 23, no. 30, pp. 305,205, 2012.

[36] L. Gao, F. Alibart, and D. B. Strukov, “Programmable CMOS/memristor threshold logic,” *IEEE Trans. Nanotechnol.*, vol. 12, no. 2, pp. 115–119, 2013.

[37] S. Kvatinsky et al. “Memristor-based material implication (IMPLY) logic: Design principles and methodologies,” *IEEE Trans. VLSI Syst.*, vol. 22, no. 10, pp. 2054–2066, 2014.

[38] S. Kvatinsky et al., “MAGIC—Memristor-aided logic,” *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 61, no. 11, pp. 895–899, Nov. 2014.

[39] G. Snider, “Computing with hysteretic resistor crossbars,” *Appl. Phys. A*, vol. 80, no. 6, pp. 1165–1172, 2005.

[40] A. Siemon, S. Menzel, R. Waser, and E. Linn, “A complementary resistive switch-based crossbar array adder,” *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 5, no. 1, pp. 64–74, 2015.

[41] S. Li, C. Xu, Q. Zou, J. Zhao, Y. Lu, and Y. Xie, “Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories,” in *Proc. 53nd ACM/EDAC/IEEE Design Automation Conf. (DAC)*, June 2016, pp. 1–6.

[42] N. Talati, S. Gupta, P. Mane, and S. Kvatinsky, “Logic design within memristive memories using memristor-aided logic (MAGIC),” *IEEE Trans. Nanotechnol.*, vol. 15, no. 4, pp. 635–650, July 2016.

[43] L. Xie et al., “Scouting logic: A novel memristor-based logic design for resistive computing,” in *Proc. IEEE Computer Society Annu. Symp. VLSI (ISVLSI)*, July 2017, pp. 176–181.

[44] M. Teimoory, A. Amirsoleimani, J. Shamsi, A. Ahmadi, S. Alirezaee, and M. Ahmadi, “Optimized implementation of memristor-based full adder by material implication logic,” in *Proc. 21st IEEE Int. Conf. Electronics, Circuits and Systems (ICECS)*, 2014, pp. 562–565.

[45] X. Wang et al., “Memristor-based XOR gate for full adder,” in *Proc. 35th Chinese Control Conf. (CCC)*, 2016, pp. 5847–5851.

[46] M. Hu et al., “Dot-product engine for neuromorphic computing: programming 1T1M crossbar to accelerate matrix-vector multiplication,” in *Proc. 53rd Annu. Design Automation Conf.* ACM, 2016, p. 19.

[47] S. G. Rohani and N. TaheriNejad, “An improved algorithm for imply logic based memristive full-adder,” in *Proc. IEEE 30th Canadian Conf. Electrical and Computer Engineering (CCECE)*, Apr. 2017, pp. 1–4.

[48] N. TaheriNejad, T. Delaroche, D. Radakovits, and S. Mirabbasi, “A semi-serial topology for compact and fast IMPLY-based memristive full adders,” in *Proc. IEEE New Circuits and Systems Symp. (NewCAS)*, 2019, pp. 1–5.

[49] P. Mazumder, S.-M. Kang, and R. Waser, “Memristors: Devices, models, and applications,” *Proc. IEEE*, vol. 100, no. 6, pp. 1911–1919, 2012.

[50] Y. Pershin and M. Di Ventra, “Neuromorphic, digital, and quantum computation with memory circuit elements,” *Proc. IEEE*, vol. 100, no. 6, pp. 2071–2080, June 2012.

[51] Y. V. Pershin, S. L. Fontaine, and M. D. Ventra, “Memristive model of amoeba learning,” *Phys. Rev. E*, vol. 80, pp. 1–6, 2009.

[52] A. Thomas, “Memristor-based neural networks,” *J. Phys. D, Appl. Phys.*, vol. 46, no. 9, p. 093,001, 2013.

[53] M. D. Pickett, G. Medeiros-Ribeiro, and R. S. Williams, “A scalable neuristor built with Mott memristors,” *Nature Mater.*, vol. 12, no. 2, p. 114, 2013.

[54] V. Milo et al., “Demonstration of hybrid CMOS/RRAM neural networks with spike time/rate-dependent plasticity,” in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*. IEEE, 2016, pp. 16–18.

[55] A. Zaher, P. Hfliger, F. Puppo, G. D. Micheli, and S. Carrara, “Novel readout circuit for memristive biosensors in cancer detection,” in *IEEE Biomedical Circuits and Systems Conf. (BioCAS) Proc.*, Oct. 2014, pp. 448–451.

[56] M. A. Zidan, J. P. Strachan, and W. D. Lu, “The future of electronics based on memristive systems,” *Nature Electron.*, vol. 1, pp. 22–29, 2018.

[57] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, “Nanoscale memristor device as synapse in neuromorphic systems,” *Nano Lett.*, vol. 10, no. 4, pp. 1297–1301, 2010.

[58] K.-H. Kim et al., “A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications,” *Nano Lett.*, vol. 12, no. 1, pp. 389–395, Jan. 2012.

[59] F. Alibart, E. Zamanidoost, and D. B. Strukov, “Pattern classification by memristive crossbar circuits using ex situ and in situ training,” *Nature Commun.*, vol. 4, 2013.

[60] R. Fackenthal et al., “19.7 A 16Gb ReRAM with 200MB/s write and 1GB/s read in 27nm technology,” in *Proc. IEEE Int. Solid-State Circuits Conf. Dig. Technical Papers (ISSCC)*, 2014, pp. 338–339.

[61] M. Prezioso, F. Merrikh-Bayat, B. Hoskins, G. Adam, K. K. Likharev, and D. B. Strukov, “Training and operation of an integrated neuromorphic network based on metal-oxide memristors,” *Nature*, vol. 521, no. 7550, p. 61, 2015.

[62] P. M. Sheridan, F. Cai, C. Du, W. Ma, Z. Zhang, and W. D. Lu, “Sparse coding with memristor networks,” *Nature Nanotechnol.*, vol. 12, no. 8, p. 784, 2017.

[63] Q. Chen, X. Wang, H. Wan, and R. Yang, “A logic circuit design for perfecting memristor-based material implication,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 2, pp. 279–284, 2017.

[64] R. B. Hur and S. Kvatinsky, “Memory processing unit for in-memory processing,” in *Proc. IEEE/ACM Int. Symp. Nanoscale Architectures (NANOARCH)*, July 2016, pp. 171–172.

[65] R. B. Hur and S. Kvatinsky, “Memristive memory processing unit (MPU) controller for in-memory processing,” in *Proc. IEEE Int. Conf. Science Electrical Engineering (ICSEE)*, Nov. 2016, pp. 1–5.

[66] P.-E. Gaillardon et al., “The programmable logic-in-memory (PLiM) computer,” in *Proc. IEEE Design, Automation & Test in Europe Conf. & Exhibition (DATE*), 2016, pp. 427–432.

[67] G. Papandroulidakis, I. Vourkas, A. Abusleme, G. C. Sirakoulis, and A. Rubio, “Crossbar-based memristive logic-in-memory architecture,” *IEEE Trans. Nanotechnol.*, vol. 16, no. 3, pp. 491–501, May 2017.

[68] R. B. Hur, N. Wald, N. Talati, and S. Kvatinsky, “SIMPLE MAGIC: Synthesis and in-memory mapping of logic execution for memristor-aided logic,” in *Proc. 36th Int. Conf. Computer-Aided Design*. Piscataway, NJ: IEEE Press, 2017, pp. 225–232.

[69] A. J. Lohn, J. E. Stevens, P. R. Mickel, and M. J. Marinella, "Optimizing TaOx memristor performance and consistency within the reactive sputtering forbidden region," *Appl. Phys. Lett.*, vol. 103, no. 6, p. 063,502, 2013.
[70] X. Lian, M. Wang, M. Rao, P. Yan, J. J. Yang, and F. Miao, "Characteristics and transport mechanisms of triple switching regimes of TaOx memristor," *Appl. Phys. Lett.*, vol. 110, no. 17, p. 173,504, 2017.
[71] H. Abunahla et al., "Switching characteristics of microscale unipolar Pd/Hf/HfO_2/Pd memristors," *Microelectron. Eng.*, vol. 185, pp. 35–42, 2018.
[72] B. Jin-shun and H. Zheng-sheng, "Study on Hf/HfO_2 bipolar resistive random-access-memory," *J. Functional Mater. Devices*, vol. 5, p. 002, 2014.
[73] H. Jiang et al., *Sci. Rep.*, vol. 6, p. 28,525, 2016.
[74] S. H. Jo, K.-H. Kim, and W. Lu, "Programmable resistance switching in nanoscale two-terminal devices," *Nano Lett.*, vol. 9, no. 1, pp. 496–500, 2009.
[75] A. Radoi, M. Dragoman, and D. Dragoman, "Memristor device based on carbon nanotubes decorated with gold nanoislands," *Appl. Phys. Lett.*, vol. 99, no. 9, pp. 093,102, 2011. doi: 10.1063/1.3633352.
[76] A. Chanthbouala, V. Garcia, R. O. Cherifi, K. Bouzehouance, S. Fusil, X. Moya, and M. Bibes, "A ferroelectric memristor," *Nature Materials*, vol. 11, pp. 860–864, 2012.
[77] T. D. Dongale et al., "Development of Ag/ZnO/FTO thin film memristor using aqueous chemical route," *Mater. Sci. Semicond. Process*. vol. 40, no. Suppl. C, pp. 523–526, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1369800115300408
[78] T. D. Dongale et al., "Development of Ag/WO_3/ITO thin film memristor using spray pyrolysis method," *Electron. Mater. Lett.*, vol. 11, no. 6, pp. 944–948, Nov. 2015. doi: 10.1007/s13391-015-4180-4
[79] A. Ascoli, R. Tetzlaff, F. Corinto, and M. Gilli, "PSpice switch-based versatile memristor model," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, 2013, pp. 205–208.
[80] D. Wheeler et al., "Fabrication and characterization of tungsten-oxide-based memristors for neuromorphic circuits," in *Proc. IEEE 14th Int. Workshop Cellular Nanoscale Networks and their Applications (CNNA)*, 2014, pp. 1–2.
[81] K. Miller, K. S. Nalwa, A. Bergerud, N. M. Neihart, and S. Chaudhary, "Memristive behavior in thin anodic titania," *IEEE Electron Device Lett.*, vol. 31, no. 7, pp. 737–739, 2010.
[82] O. Kavehei, K. Cho, S. Lee, S.-J. Kim, S. Al-Sarawi, D. Abbott, and K. Eshraghian, "Fabrication and modeling of Ag/TiO/ITO memristor," in *Proc. IEEE 54th Int. Midwest Symp. Circuits and Systems (MWSCAS)*, 2011, pp. 1–4.
[83] L. Gao, P.-Y. Chen, and S. Yu, "NbOx based oscillation neuron for neuromorphic computing," *Appl. Phys. Lett.*, vol. 111, no. 10, p. 103,503, 2017. [Online].
[84] C. Funck et al., "Multidimensional simulation of threshold switching in NbO2 based on an electric field triggered thermal runaway model," *Adv. Electron. Mater.* vol. 2, no. 7, p. 1,600,169, [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/aelm.201600169
[85] G. A. Gibson et al., "An accurate locally active memristor model for s-type negative differential resistance in NbOx," *Appl. Phys. Lett.*, vol. 108, no. 2, p. 023,505, 2016. [Online].
[86] J. P. Strachan et al., "State dynamics and modeling of tantalum oxide memristors," *IEEE Trans. on Electron Devices*, vol. 60, no. 7, pp. 2194–2202, July 2013.
[87] S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny, "VTEAM: A general model for voltage-controlled memristors," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 62, no. 8, pp. 786–790, 2015.
[88] Z. Jiang et al., "A compact model for metal-oxide resistive random access memory with experiment verification," *IEEE Trans. Electron. Devices*, vol. 63, no. 5, pp. 1884–1892, May 2016.
[89] K. Fleck, C. La Torre, N. Aslam, S. Hoffmann-Eifert, U. Böttger, and S. Menzel, "Uniting gradual and abrupt set processes in resistive switching oxides," *Phys. Rev. Appl.*, vol. 6, p. 064,015, Dec. 2016. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevApplied.6.064015
[90] A. Serb, J. Bill, A. Khiat, R. Berdan, R. Legenstein, and T. Prodromakis, "Unsupervised learning in probabilistic neural networks with multi-state metal-oxide memristive synapses," *Nature Commun.*, vol. 4, 2016.
[91] Y. Pershin and M. D. Ventra, "Solving mazes memristors: A massively parallel approach," vol. 84, p. 046,703, Oct. 2011.
[92] A. Ascoli, R. Tetzlaff, L. O. Chua, J. P. Strachan, and R. S. Williams, "History erase effect in a non-volatile memristor," *IEEE Trans. Circuits Syst. I, Regular Papers*, vol. 63, no. 3, pp. 389–400, Mar. 2016.
[93] S. Ghedira, F. O. Rziga, K. Mbarek, and K. Besbes, "Dynamical resistive switching of a generic memristor model: Analysis and simulation," in *Proc. Int. Conf. Engineering MIS (ICEMIS)*, May 2017, pp. 1–5.
[94] D. Radakovits and N. TaheriNejad, "Implementation and characterization of a memristive memory system," in *Proc. IEEE 32nd Canadian Conf. Electrical and Computer Engineering (CCECE)*, May 2019, pp. 1–5.
[95] Knowm, "Knowm BS-AF-W memristor datasheet," 2017. [Online]. Available: https://knowm.org/wp-content/uploads/DM8-16DIP-BS-AF-W.pdf
[96] Crossbar. (2017). *ReRAM.* [Online]. Available: https://www.crossbar-inc.com/en/
[97] D. Biolek, M. Di Ventra, and Y. V. Pershin, "Reliable spice simulations of memristors, memcapacitors and meminductors," arXiv Preprint, arXiv:1307.2717, 2013.
[98] B. K'A and E. E. Swartzlander, "Memristor-based arithmetic," in *Conf. Rec. 44th Asilomar Conf. Sig., Sys. and Comp.*, 2010.
[99] S. Haghiri, A. Nemati, S. Feizi, A. Amirsoleimani, A. Ahmadi, and M. Ahmadi, "A memristor based binary multiplier," in *Proc. IEEE 30th Canadian Conf. Electrical and Computer Engineering (CCECE),* Apr. 2017, pp. 1–4.
[100] A. Karimi and A. Rezai, "Novel design for a memristor-based full adder using a new imply logic approach," *J. Computational Electronics*, vol. 17, no. 3, pp. 1303–1314, Sept. 2018. https://doi.org/10.1007/s10825-018-1198-5
[101] S. G. Rohani, N. TaheriNejad, and D. Radakovits, "A semi-parallel full-adder in imply logic," *Submitted to The IEEE Trans. on VLSI Systems*, June 2018.
[102] C. Yakopcic, T. M. Taha, G. Subramanyam, R. E. Pino, and S. Rogers, "A memristor device model," *IEEE Electron Device Lett.*, vol. 32, no. 10, pp. 1436–1438, 2011.
[103] Y. N. Joglekar and S. J. Wolf, "The elusive memristor: Properties of basic electrical circuits," *Eur. J. Phys.*, vol. 30, no. 4, p. 661, 2009.
[104] 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, 2013.
[105] Mar. 2018. [Online]. Available: https://www.ict.tuwien.ac.at/staff/taherinejad/projects/memristor/files/VTEAM.sub
[106] S. H. Jo, K.-H. Kim, and W. Lu, "High-density crossbar arrays based on a Si memristive system," *Nano letters*, vol. 9, no. 2, pp. 870–874, 2009.
[107] M. Hu, H. Li, Y. Chen, X. Wang, and R. E. Pino, "Geometry variations analysis of TiO_2 thin-film and spintronic memristors," in *Proc. IEEE 16th Asia and South Pacific Design Automation Conf. (ASP-DAC)*, 2011, pp. 25–30.
[108] B. Jiao et al., "Resisitive switching variability study on 1T1R AlOx/WOx-based RRAM array," in *Proc. IEEE Int. Conf. Electron Devices and Solid-State Circuits (EDSSC)*, 2013, pp. 1–2.
[109] S. Yu, X. Guan, and H.-S. P. Wong, "On the stochastic nature of resistive switching in metal oxide RRAM: Physical modeling, Monte Carlo simulation, and experimental characterization," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, 2011, p. 17-3.
[110] D. Niu, Y. Chen, C. Xu, and Y. Xie, "Impact of process variations on emerging memristor," in *Proc. 47th Design Automation Conf.* ACM, 2010, pp. 877–882.
[111] P. Pouyan, E. Amat, and A. Rubio, "Statistical lifetime analysis of memristive crossbar matrix," in *Proc. IEEE 10th Int. Conf. Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, 2015, pp. 1–6.
[112] A. C. Torrezan, J. P. Strachan, G. Medeiros-Ribeiro, and R. S. Williams, "Sub-nanosecond switching of a tantalum oxide memristor," *Nanotechnology*, vol. 22, no. 48, p. 485,203, 2011.
[113] H. Lee et al., "Evidence and solution of over-RESET problem for HfOx based resistive memory with sub-ns switching speed and high endurance," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, 2010, pp. 19–17.
[114] M. A. Zidan, H. A. H. Fahmy, M. M. Hussain, and K. N. Salama, "Memristor-based memory: The sneak paths problem and solutions," *Microelectron. J.*, vol. 44, no. 2, pp. 176–183, 2013.
[115] Y. Cassuto, S. Kvatinsky, and E. Yaakobi, "Sneak-path constraints in memristor crossbar arrays," in *Proc. IEEE Int. Symp. Information Theory Proc. (ISIT)*, 2013, pp. 156–160.
[116] R. Berdan, A. Serb, A. Khiat, A. Regoutz, C. Papavassiliou, and T. Prodromakis, "A μ-controller-based system for interfacing selectorless RRAM crossbar arrays," *IEEE Trans. Electron. Devices*, vol. 62, no. 7, pp. 2190–2196, 2015.
[117] J. K. Eshraghian, K.-R. Cho, H. H. Iu, T. Fernando, N. Iannella, S.-M. Kang, and K. Eshraghian, "Maximization of crossbar array memory using fundamental memristor theory," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 64, no. 12, pp. 1402–1406, 2017.