An experience with Chalcogenide memristors, and implications on memory and computer applications

Escudero-López, Manuel ; Amat, Esteve; Rubio, Antonio; Pouyan, Peyman

DOI
10.1109/DCIS.2016.7845387

Publication date
2017

Document Version
Final published version

Published in
2016 Conference on Design of Circuits and Integrated Systems (DCIS)

An experience with Chalcogenide memristors, and implications on memory and computer applications

Manuel Escudero-López, Esteve Amat and Antonio Rubio
Electronics Engineering Department
Universitat Politècnica de Catalunya
Barcelona, Spain
manuel.escudero@upc.edu, esteve.amat@upc.edu, antonio.rubio@upc.edu

Peyman Pouyan
Computer Engineering Department
Delft University of Technology
Delft, The Netherlands
p.pouyan@tudelft.nl

***Abstract*—Memristors are considered a promising emerging device that may improve some specific applications, like memories, or make feasible new ones, mainly alternative computing architectures. However, it is not a mature technology and their characteristics can vary significantly depending on their structures. Also, variability and reliability might suppose an important issue in some applications. In this paper, a chalcogenide memristor is studied and their main parameters are extracted. Then, it's discused how their properties can affect two applications: a memory circuit and a digital computing alternative, the logic implication technique.**

I. Introduction

Memristor is a circuital element postulated by Leon Chua at 1971 [1]. Conceptually, it's a passive element that holds a nonlinear relationship between magnetic flux and electric charge. This fact implies that memristor resistance changes with the history of charge that has passed through the device at the past. Although theoretically demonstrated, little research activity related with memristors has been done until 2008, when Hewlett Packard connected their experimental nanodevices with the mathematical concept [2].

Memristive devices are a promising alternative because they are nonvolatile memory elements, storing data in form of resistance even if no charge is passing through it. It's BEOL compatible, and it can achieve higher density [3]. It's expected that memristive devices speed could match with the CMOS devices. Nonetheless, currently it isn't a mature technology and there is a need to improve some characteristics. For instance, variability and reliability are considered among the most important issues [4].

Memristor applications cover memories, digital computing applications and analog computing applications. Memristive device is nonvolatile, thus ideal to store data without loss. This feature and higher device density are two key properties for memory applications. Several techniques and circuit topologies to perform computing has been arised, each one with different and unique properties. Furthermore, their capacity to store data, and concretely process analog data, make them attractive to explore non Von Neumann computing paradigms, as memory and processing can be done with memristive devices in the same area. For instance, analog computing applications include neuromorphic computing might exploit these memristive device characteristics for their own benefits.

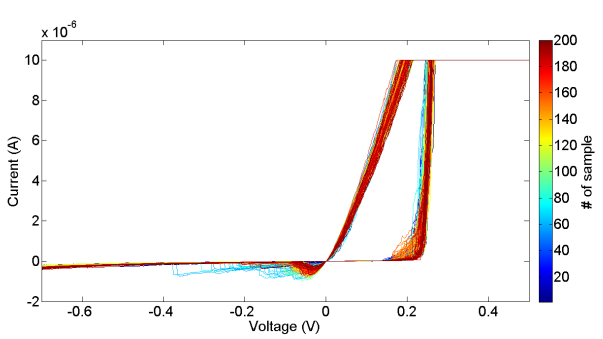
This paper is organized as follows. Section II presents the device studied and its memristive behavior and variability are evaluated. Section III and IV show a memory application and a digital computing application, respectively, and discuss about the feasibility of these applications using the parameters extracted from Section II. Finally, conclusions are summarized in Section V.

II. Device characteristics

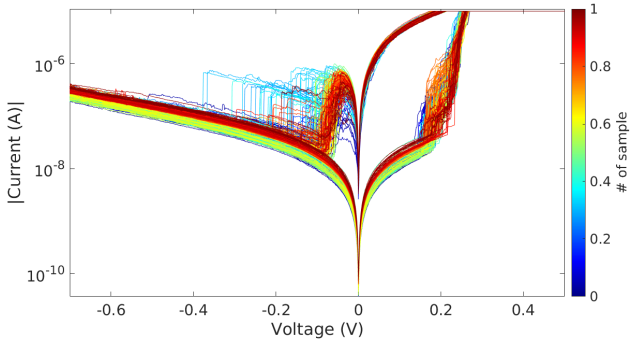
Memristors used in this study are manufactured by Knowm company [5]. The principal operation of these devices is based on the generation and movement of metal ions through a multilayer chalcogenide material stack when applying an electric field. Also, a phase change mechanism can appear if it is operated at high voltage ($>$ 1 V). However, this work region is omitted as is not relevant in this work. The metal layer is easily oxidizable, and located near one electrode. Applying a positive voltage, metal is oxidized and form ions. These ions drift to the lower potential electrode. They are reduced to their metallic form when they arrive at the electrode and form a conductive path between both electrodes, lowering the resistance of the device. Reversing the direction of the applied potential makes the conductive path to dissolve and the resistance of the device increases. The resistance is related in any time to the amount of metal located within the active layer [6]. This memristor corresponds to a bipolar one. Hence, applying a positive threshold voltage V_{close} the memristor goes to a lower resistance state (LRS) and applying a negative threshold voltage V_{open} its resistance drifts to a higher resistance state (HRS).

Before collecting any data, the memristor is fresh and needs to be electroformed. A 4156C Semiconductor Parameter Analyzer is used for this task and also for the rest of the

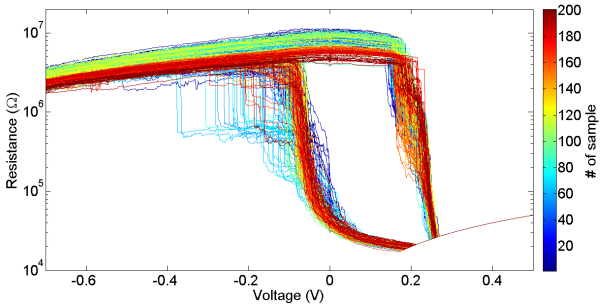
978-1-5090-4565-5/16$31.00



(a) Voltage vs. current plot



(b) Voltage vs. current (log scale) plot



(c) Voltage vs. resistance (log scale) plot

Fig. 1: Experimental data for 200 switching cycles of one memristor.

experimetal data shown in this paper. Electroforming is done applying a voltage sweep from 0 V to 1 V, limiting current up to 1 μA. Several switching cycles have been forced to memristors in order to study its behavior. Figure 1a shows about 190 cycles using a voltage sweep from -0.7 V to 0.5 V, with a resolution less than 2 mV. This voltage sweep is slow, each step is kept for 100 ms. The reason for this asymmetry in the voltage range is to be sure that the HRS and the LRS has been reached completely. Also note that current across the device is limited to 10 μA in order to prevent the destruction of the memristor when it changes from HRS to LRS. Curves show a hysteretic behavior, but note that below 0 V is difficult to perceive any trace of the LRS.

In order to understand better all collected data, two more graphics are provided. Figure 1b shows the current in logarithmic scale, curves now have a butterfly shape typical found in other memristive devices. This plot permits to observe better the negative part of the hysteretic cycle. In Figure 1c, each voltage-current points have been converted to resistance and plotted in a voltage vs resistance in logarithmic scale. Resistance is kept more or less invariant in the regions where memristor doesn't have to change its state. It's worth noting that the resistance switching event from HRS to LRS is abrupt while the switching from LRS to HRS is smoother.

Some parameters have been extracted from these graphics, concretely R_{LRS}, R_{HRS}, V_{CLOSE} and V_{OPEN}. To obtain R_{LRS} and R_{HRS} values, memristor state in form of current have been read at 50 mV, a voltage which memristor current can be monitorized without change the state. V_{CLOSE} and V_{OPEN} have been defined as the voltage where resistance change (in logarithmic form) is highest in the positive and negative half of the cycle, respectively. Mean values (X_μ) and the standard deviation (X_σ) have been calculated. Results obtained are $R_{LRS,\mu} = 29\text{k}\Omega$ and $R_{LRS,\sigma} = 2.656\text{k}\Omega$, $R_{HRS,\mu} = 7.388\text{M}\Omega$ and $R_{HRS,\sigma} = 1.742\text{M}\Omega$, $V_{CLOSE,\mu} = 0.2069$ V and $V_{CLOSE,\sigma} = 0.032$ V, $V_{OPEN,\mu} = -0.1176$ V and $V_{OPEN,\sigma} = -0.0806$ V.

Looking carefully, V_{CLOSE} and R_{LRS} show less variability than V_{OPEN} and R_{HRS}. V_{OPEN} is by far the worst parameter in terms of variability, as seen in 1b. As a final comment, this is only cycle-to-cycle variability and device-to-device variability should be added to have the full picture of these memristors variability.

III. A Memory Application

Memristive memory systems are commonly organized in a matrix-like structure called crossbar. The storage cell in the crossbar can be built with only one memristor device (1R cell), but due to the well-known problem of sneak-paths (related with the leakage paths in the unselected devices of the crossbar), what will degrade the output read signal in a specific selected cell and might induce error [7]. For this, usually, the bit storage cell is constructed by utilizing complementary devices. One globally accepted proposed alternative is to use

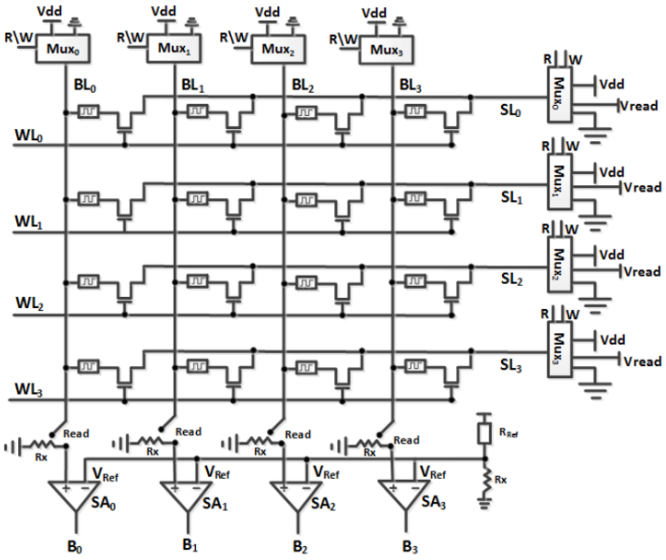


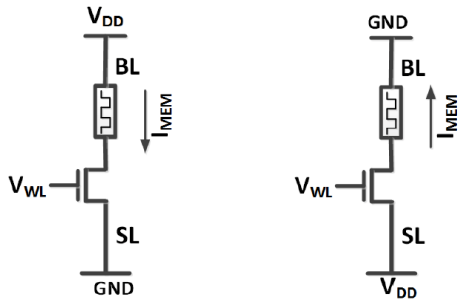
Fig. 2: An example of a 4x4 1T1R cell crossbar memory, including a row and column decoder and a read circuitry.

a selecting device such as a FET transistor, being the name of the bit-cell (1T1R) [8]. The 1T1R cell generally consists of an NMOS transistor and a resistive switching device (e.g. based on a resistive material such as the former shown memristor), then, in this structure the memristor current is correctly controlled through the crossbar. In this sense, each memristor is turned "on" or "off" based on the row address in the crossbar. Moreover, 1T1R is usually chosen for the memory implementation since it is CMOS compatible, to manufacture the cell in existing fabrication process. Additionally, this makes the crossbar completely sneak path-free, and improves the noise margin, and moreover it is widely used in other recent research works [9] [10] [11]. Regarding this, Figure 2 presents a memristive crossbar memory constructed with 1T1R as storage cells. Each 1T1R cell can be written and read by applying the appropriate signals through the bitline (BL), wordline (WL) and the select line (SL). Next, the write and read processes are exposed for a 1T1R cell inside the crossbar.

A. Writing cycle in memory

Writing '1' in the cell, also called SET process, consists on the state change of the respective memristor from HRS to LRS for the selected 1T1R cell. To perform a SET operation first the corresponding WL is activated in the crossbar, and next while the SL is grounded an appropriate voltage is applied at the BL (V_{DD}). In this sense, Figure 3a shows the corresponding voltages configuration to write '1' in a single 1T1R cell and the current path through the memristor device (I_{MEM}).

Changing the state of memristor from LRS to HRS, is called the RESET process, and it is equivalent to writing a '0' in the cell. For this process the WL is again activated, when a proper voltage (V_{DD}) is applied at the SL, and while the corresponding BL is grounded, the other BLs in the



(a) SET process. (b) RESET process.

Fig. 3: Writing schemes for a 1T1R cell.

block are all in high impedance mode. Figure 3b shows the corresponding voltages scheme to write '0' in a single 1T1R cell.

B. Reading cycle in memory

The read process in a memristive crossbar can be implemented by using two different methods; one is based on voltage, the other one is based on current. The former needs an extra resistor before the sense amplifier, as it is shown in Figure 2. In this approach, a Vread voltage (lower than the write voltage) is applied to the corresponding 1T1R cell, and the BL voltage (a voltage division of the read voltage though the memristor resistance and the resistor R_x), will be detected though the comparator giving the state of memristor and bit-cell. A reference cell, consisting of a reference resistance (R_{ref}), is used to generate a reference voltage to be compared in the comparator. A resistive device in LRS state will produce a higher voltage in comparison with one at HRS mode. Note that the read voltage (V_{read}) is much lower than the write voltage V_{DD} in order not to modify the state of memristor in the read operation. Moreover, regarding the read mode based on the current (Figure 4), a small current (I_{read}) is injected through the BL to the corresponding 1T1R cell and the memristor voltage is sensed and compared with a reference voltage, produced by a reference cell, through a sense amplifier. For instance, Figure 4 shows a 2x2 crossbar utilizing the read mode based on the current. The square boxes in this figure are multiplexers, which govern the appropriate voltage or current to be applied to the cells. Now then, the read mode based on voltage is chosen, the reason is controlling the applied voltage to the memristive cell is easier and also the model which we will later use for circuit simulation works better in voltage mode.

C. Impact of variability on the reliability of the memory

An analytical approach to evaluate the reliability can be described considering the cycle-to-cycle variability of the two resistance states (LRS and HRS) of the memristive devices as two normal distributions. From this assumption it can be derived the read probability error P_e, a variable that determines the likelihood of an incorrect read in the memristive memory

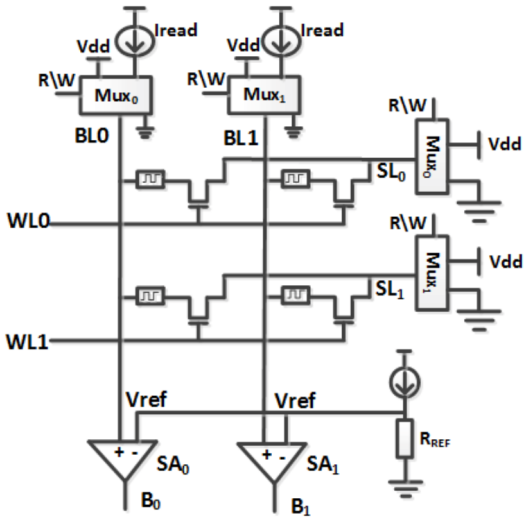


Fig. 4: A 2x2 crossbar with a read mode based on current instead of voltage.

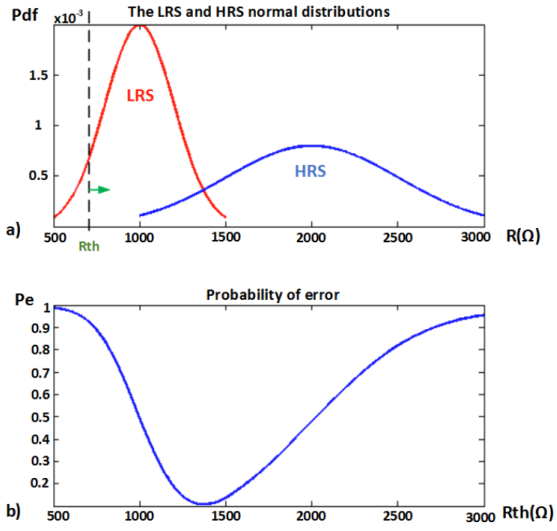


Fig. 5: Impact of variability on memories: (a) normal distributions of LRS and HRS and (b) probability of error depending on a reference resistor R_{th}.

cell in function of a reference resistance value (R_{th}), with which the reference resistance is compared. The variable is the key robustness factor of the memory. The P_e can be display as indicated in Figure 5, where also the two assumed normal distribution of the resistances is plotted (in the figure an arbitrary pair of normal distributions are plotted). The P_e graph is evaluated while considering a reference point R_{ref} and sweep it along the two distributions resistive domain corresponding to LRS and HRS.

As it can shown that the P_e is minimized for reference resistance located in the interval between the two resistances distribution. Using the data shown before for the chalcogenide

TABLE I: Logic Implication Truth Table

Case	p	q	$q' = p \rightarrow q$
1	0	0	1
2	0	1	1
3	1	0	0
4	1	1	1

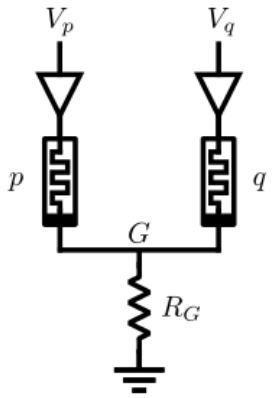


Fig. 6: Logic Implication gate circuit

memristor, the situation presented in Figure 5 is only possible if the continous use of the memristors makes HRS and LRS to get close. Hence, eventually errors when reading the cell can be a reality.

IV. A Computing Application

In this section, we have focused in digital computation and among this cathegory, the logic implication technique [12]. Among all computing techniques for memristors, logic implication is interesting because it exploits the store capability of the memristor and it's very scalable due to compatibility to memristor crossbar arrays. In addition, any function can be programmed using the same topology, only a sequence of voltage pulses have to be defined properly.

A. Logic Implication

Logic implication is a two-variable boolean function which truth table is shown in Table I and it can be interpreted as: "if p is 1, then output is equal to p; otherwise, output is always 1". A complete set of boolean functions can be produced using this operation and the false operation (e.g. put a variable to '0'). Note that each row of the truth table is labeled with a numbered case. Through the rest of the paper, they will be mentioned in this way. Figure 6 presents the circuit that performs logic implication. The circuit comprises two memristors p and q which store p and q variables.

Memristors codify p and q variables values as follows: a '1' is codified as a R_{LRS} state and a '0' is codified as a R_{HRS} state. Voltage drivers apply different voltages to write memristors, to perform logic implication operation or to read memristors state. Also, high impedance output may be useful when a memristor in a array is not used.

To succesfully perform logic implication operation, there are two aspects to check. First one is about initial conditions. Resistance switching in memristors will be triggered if voltage across them exceed V_{CLOSE} or V_{OPEN} at the instant of applying V_{COND} and V_{SET} pulses. Looking at Table I, this only must happen in case 1, where memristor q change from HRS to LRS. If for every truth table case the voltage across memristors p and q are analysed, a set of conditions (1), (2) and (3) are derived to guarantee the correct behavior of this circuit at $t = 0s$. For example, to derive (1), $V_q > V_{CLOSE}$ condition is used. When designing the logic implication gate circuit using these bounds, memristor p can't change in any case.

$$V_{SET} > V_{CLOSE} > V_{COND} \tag{1}$$
$$V_{SET} - V_{COND} < V_{CLOSE} \tag{2}$$
$$\frac{V_{COND} - V_{SET}}{2} > V_{OPEN} \tag{3}$$

Also, R_G bounds can be found using 1 and 2 conditions:

$$R_G < R_{OFF}\frac{V_{SET} - V_{CLOSE}}{2 \cdot V_{CLOSE} + V_{COND} - V_{SET}} \tag{4}$$
$$R_G > R_{ON}\frac{V_{SET} - V_{CLOSE}}{V_{COND} + V_{CLOSE} - V_{SET}} \tag{5}$$

Once initial conditions are verified, the evolution of resistance switching of memristor p in case 1 is observed. Memristor q will change its state depending on R_p and R_G and its final state is defined approximately by (6) [13]. In fact, it's demonstrated that is not possible to fully switch from HRS to LRS. Hence, there is an existence of a state drift in memristor q at the end of case 1. In addition, this state drift can impact in other cases from Table I when doing subsequent logic implication operations using affected memristors. To illustrate this phenomena, a simulation is run. This simulation computes the final resistance of memristor q for differents initial states of memristors p and q. Memristors models used are based on VTEAM model [14] and parameters are taken from the ones obtained in Section II, except for $R_{RHL} = 1\text{k}\Omega$ and $R_{LRS} = 1\text{M}\Omega$. Results are shown in Figure 7. For example, if case 1 is evaluated and R_p and R_q are equal to $1\text{M}\Omega$, memristor q should switch to R_{LRS} but instead is kept in an intermediate state. If this memristor is then used in other logic implication operation as a memristor p it can produce again an state drift and keep the output memristor in another intermediate state, as shown in the plot. This eventually can end in a wrong evaluation of a logic function.

$$R_{q,final} \simeq \frac{V_{CLOSE}}{V_{SET} - V_{CLOSE}} R_G \tag{6}$$

B. Variability

Variability also will be discussed in two parts, for initial conditions and for the state evolution of case 1. Of course, variability in V_{CLOSE} and V_{OPEN} will change initial conditions presented before. Now a range of possible V_{CLOSE} and V_{OPEN} values must be considered and V_{SET} and V_{CLOSE} bounds are checked for this whole range, shown in (7) - (10). This fact difficults the choice of the two design values.

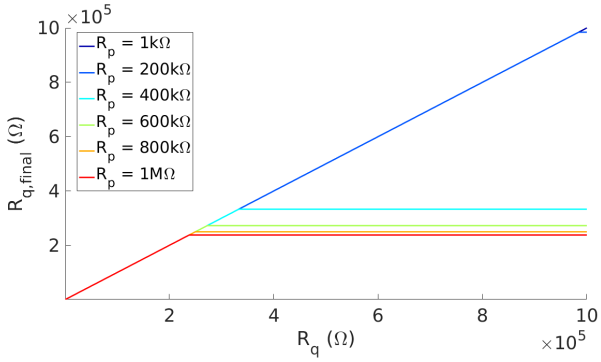


Fig. 7: Final state of memristor q after logic implication operation vs initial state for memristor p and q

$$V_{SET} > V_{CLOSE,max} \tag{7}$$
$$V_{CLOSE,min} > V_{COND} \tag{8}$$
$$V_{SET} - V_{COND} < V_{CLOSE,min} \tag{9}$$
$$\frac{V_{COND} - V_{SET}}{2} > V_{OPEN,max} \tag{10}$$

High device-to-device variability for HRS and LRS values can be an important issue for R_G. Even if V_{CLOSE} pulse comes before V_{COND}, q can switch to LRS in an undesirable manner. As for the state evolution in case 1 and further operations, variability could provoke easily larger drift state in memristor q, as state drift depends on R_p.

V. Conclusion

In this paper, a chalcogenide memristor device has been characterised in order to evaluate the memristive behavior and the cycle-to-cycle variability in short term. Next, these results have been used to discuss the impact of variability in a memory and a digital computing application. As shown, memory performance is primarily affected by memristor endurance and not for variability if enough HRS to LRS ratio is achieved. When explored logic implication technique, the analysis should the presence of a state drift phenomena fact that is a clear limitation for effective application, the cycle to cycle and process variabilities even increases the limitation worsening the problem.

References

[1] L. Chua, *Memristor - The Missing Circuit Element*, IEEE Trans. Circuit Theory, vol. CT-18, pp.507-519 Sept. 1971.

[2] D. B. Strukov, G. S. Snider, D. R. Stewart, S. Williams, *The missing memristor found*, Letters, Nature, vol. 453, pp. 80-84, May 2008.

[3] H.-S. Philip Wong, H.-Y. Lee, S. Y., Y.-S. Chen, et. al., *Metal-Oxide RRAM*, Proceedings of the IEEE, vol. 100, no. 6, pp. 1951-1970, June 2012.

[4] P. Pouyan, E. Amat, A. Rubio, *Reliability Challenges in Design of Memristive Memories*, 5th European Workshop on CMOS Variability (VARI), pp. 1-6, Sept. 2014.
[5] http://knowm.org
[6] http://knowm.org/product/bs-af-w-memristors/
[7] A. Mohammed, H. Aly, H. Fahmy, M. Mustafa, and K. Nabil, *Memristor-based memory: The sneak paths problem and solutions*, Microelectronics Journal, vol. 44, no. 2, pp. 176183, February 2013.
[8] I. Vourkas and G. Sirakoulis, *A Novel Design and Modeling Paradigm for Memristor-Based Crossbar Circuits*, IEEE Transactions on Nanotechnology, vol. 11, no. 6, pp. 11511159, September 2012.
[9] P. Huang, B. Chen, Y. Wang, F. Zhang, L. Shen, B. Weng, Y. Tang, G.-Q. Lo, and D.-L. Kwong, *Analytic model of endurance degradation and its practical applications for operation scheme optimization in metal oxide based RRAM*, in IEEE International Electron Devices Meeting (IEDM), 2013
[10] P. Pouyan, E. Amat, and A. Rubio, *Statistical Lifetime Analysis of Memristive Crossbar Matrix*, in International Conference on Design and Technology of Integrated Systems in Nanoscale Era (DTIS), 2015.
[11] D. Veksler, G. Bersuker, L. Vandelli, A. Padovani, L. Larcher, A. Muraviev, B. Chakrabarti, E. Vogel, D. Gilmer, and P. Kirsch, *Random telegraph noise (RTN) in scaled RRAM devices*, in IEEE International Reliability Physics Symposium (IRPS), 2013.
[12] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, R. S. Williams, *'Memristive' switches enable 'stateful' logic operations via material implication*, Letters, Nature, Vol.464, pp 873-876, Apr. 2010.
[13] X. Fang, Y. Tang, *Circuit analysis of the memristive stateful implication gate*, Electronic Letters, vol. 49, no. 20, pp.1282-1283, Sept. 2013.
[14] S. Kvatinsky, M. Ramadan, E. G. Friedman, A. Kolodny, *VTEAM: A General Model for Voltage-Controlled Memristors*, Circuits and Systems II: Express Briefs, vol. 62, no. 8, pp. 786-790, May 2015.