2016 ICSEE International Conference on the Science of Electrical Engineering

Design Methodology for Stateful Memristive Logic Gates

Nimrod Wald and Shahar Kvatinsky
Andrew and Erna Viterbi Department of Electrical Engineering
Technion – Israel Institute of Technology
Haifa, Israel, 3200003
Email: {nimrodw@tx, shahar@ee}.technion.ac.il

***Abstract* – Memristors are passive, two terminal, circuit elements with a resistance which depends on a state variable, and changes according to the voltage applied across the device. Alongside the natural use of memristors as memory, using these devices as building blocks for logic gates is widely researched. In this paper, a structured design methodology is presented to assist in the development of a class of such logic gates – 'stateful' logic, in which the memristor resistance is used as a logic value. The methodology is demonstrated by two examples, the first resulting in a recreation of the previously published MAGIC NOR gate, with the addition of NAND functionality on the same topology. In the second example, a novel gate is presented, realizing OR and XOR logic functions using non-polar memristors.**

Keywords – memristor, resistive switch, design methodology, MAGIC, stateful logic, in-memory computing, memory processing unit (MPU).

I. Introduction

Resistive switching devices, also referred to as memristors, RRAM or ReRAM, are an emerging technology postulated by Leon Chua in 1971 [1]. These devices are considered as attractive candidates for replacing current memory technologies. However, unique characteristics (*e.g.*, polarity, non-linearity) make memristors interesting devices for use in other domains, such as logic circuits, neuromorphic systems, and hardware security.

After the declaration by Hewlett Packard laboratories in 2008 on the connection between memristors and resistive switching properties in TiO_2 [2], the research in the field of memristor applications has been growing vigorously. One application, which is promising to affect the way computers are built, is performing logic operations using circuits with memristors. Existing methods of performing logic operations using memristors include *stateful logic* gates, where memristors are both the elements performing the logic operation and the ones storing the input/output values. All the input and output values of a stateful gate are therefore represented by resistance of a memristor. These gates include MAGIC [3], IMPLY [4, 5], and complementary resistive switching (CRS) logic [6]. Other types of gates (*i.e.*, not stateful) are gates where some or all of the input/output values are represented by quantities other than resistance (*e.g.*, by voltage). These gates include hybrid CMOS-memristor gates [7] and Akers logic [8]. These gates are only a fraction of the variety of memristive logic gates that have been developed, and there are many more that have not yet been thought of. To simplify the process of invention for new types of logic gates, there is a need to develop a systematic design methodology. In this paper, we present a methodology to meet this need when designing stateful memristive logic gates.

Previous work that deals with methodology for logic with memristors has mostly focused on methodology for designing a specific gate type (*i.e.*, optimizing parameters) [9, 10], a specific application [11], or using a specific device chosen for its unique properties [12]. Our methodology is wider and more general, and fits the design of any stateful logic gate.

The rest of the paper is organized as follows. In section II, a demonstration of the approach for logic gate design is provided by an overview of the design procedure for a MAGIC [3] gate. In section III, the proposed methodology is summarized in the form of a checklist that can be followed when designing a stateful logic gate. An example for using this methodology is provided in section IV with the design process for a novel two non-polar memristor gate. The paper is concluded, and the contribution and its limitations are discussed in section V.

II. Design OF A Memristor Aided Logic Gate

Recently, we have introduced in [3] a family of stateful memristive logic gates titled Memristor Aided logic (MAGIC), and specifically the MAGIC NOR gate. The design procedure for the MAGIC NOR gate is hereby presented to demonstrate the methodology used for inventing novel gates. Throughout this paper logical values '1' and '0' are represented, respectively, by the low (LRS, R_{ON}) and high (HRS, R_{OFF}) resistance states of the memristor. All memristor types are assumed to uphold the condition $R_{ON} \ll R_{OFF}$. The steps of the procedure that construct the invention of a MAGIC NOR gate are presented in the rest of this section as a demonstration of the proposed design methodology.

First, we choose the topology of the gate to consist of three bipolar memristors, all connected to a common node on the positive pole. The negative poles are connected to voltage sources. The gate schematic is shown in Figure 1 The input of the gate is chosen as the initial memristance values of two of the memristors, and the output is chosen as the memristance of the third memristor at the end of the operation. For symmetry, the voltage connected to the positive terminal of both input memristors is equal. Next, names are given to all meaningful values. Voltage names can be seen in Figure 1. Memristor resistance for OUT, IN_1 and IN_2 are, respectively. R_{OUT}, R_{IN1} and R_{IN2}.

The voltage in the common node, being the only relevant parameter in this circuit, is analyzed. This voltage is the result

978-1-5090-2152-9/16/$31.00 ©2016 IEEE

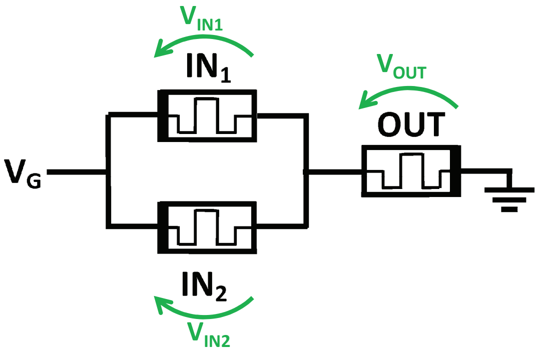

Figure 1. Schematic of a MAGIC NOR gate.

of a voltage divider between the input memristors and the output memristor and can be determined as

$$V_{OUT} = V_G \cdot \frac{R_{OUT}}{R_{OUT} + R_{IN1}||R_{IN2}}, \qquad (1a)$$

$$V_{IN1} = V_{IN2} = V_G \cdot \frac{R_{IN1}||R_{IN2}}{R_{OUT} + R_{IN1}||R_{IN2}}. \qquad (1b)$$

The initial state of the gate (*i.e.* before any switching has occurred) is then analyzed. The analysis is listed in Table I. Assuming V_G is positive, the constraints for desired gate operation are considered. To allow any switching in the output memristor with the given voltage and device polarity, the output memristor must be initialized to logic '1'. Another constraint to be enforced in this case is input stability, *i.e.*, the inputs should retain their values. To achieve this constraint, we demand that the maximal voltage across the input memristors shall not exceed the threshold voltage ($V_{th,on}$) for switching from *HRS* to *LRS*. Thus, we demand

$$V_G < |V_{th,on}|. \qquad (2)$$

The same process can be repeated for a choice of negative V_G, but we will not go into the details of it in this paper. Then, the gate dynamics are analyzed and all of the remaining parameters that affect gate operation are set. In this case, the dynamics are controlled by the value of V_{OUT} relative to $V_{th,off}$ which is the threshold voltage to be exceeded in order to switch OUT from logic '1' to '0'. To choose a proper value of V_G, while maintaining (2), the different possible values need to be analyzed. These values are divided to the following four regions:

$$0 < V_G < \frac{3}{2} V_{th,off}, \qquad (3a)$$

$$\frac{3}{2} V_{th,off} < V_G < 2V_{th,off}, \qquad (3b)$$

$$2V_{th,off} < V_G < \frac{R_{OFF}}{2R_{ON}} V_{th,off}\,, \qquad (3c)$$

$$\frac{R_{OFF}}{2R_{ON}} V_{th,off} < V_G. \qquad (3d)$$

The ranges in (3a) and (3d) cause no switching of the output, or switching regardless of the input values, respectively, and are therefore not suitable for the purpose of logic gate design. The range in (3b) causes switching only if both inputs have a value equal to logic '1', hence the gate performs a NAND operation. The range in (3c) causes switching if at least one of the inputs is logic '1', hence performing a NOR operation. Combining the constraint in (2) and the ranges in (3b) and (3c) we get the following voltages for NAND and NOR gates:

TABLE I. MAGIC NOR INITIAL TRUTH TABLE

Inputs		V_{OUT}		
R_{IN1}	R_{IN2}	General Form	For Initial R_{OUT}=R_{OFF}	For Initial R_{OUT}=R_{ON}
R_{OFF}	R_{OFF}	$V_G \frac{R_{OUT}}{R_{OUT} + \frac{1}{2} R_{OFF}}$	$\frac{2}{3} V_G$	$\sim \frac{2R_{ON}}{R_{OFF}} V_G$
R_{OFF}	R_{ON}	$\sim V_G \frac{R_{OUT}}{R_{OUT} + R_{ON}}$	$\sim V_G$	$\sim \frac{1}{2} V_G$
R_{ON}	R_{OFF}			
R_{ON}	R_{ON}	$V_G \frac{R_{OUT}}{R_{OUT} + \frac{1}{2} R_{ON}}$	$\sim V_G$	$\frac{2}{3} V_G$

$$\frac{3}{2} V_{th,off} < V_{G,NAND} < \min\left(2V_{th,off}, |V_{th,on}|\right), \qquad (4a)$$

$$2V_{th,off} < V_{G,NOR} < \min\left(\frac{R_{OFF}}{2R_{ON}} V_{th,off}, |V_{th,on}|\right). (4b)$$

While both results are valid and can be used, the option to use the proposed topology to design a NAND gate extends the MAGIC gate family beyond what we have proposed in [3].

III. Memristive Logic Gate Design Methodology

The aforementioned process is shaped into a structured set of steps, to be followed in order to efficiently invent and design a stateful memristive logic gate. The procedure is intended to facilitate a systematic choice of parameters and efficient estimation of the usability of a gate topology thought of by the inventor. The process consists of the following seven steps, the corresponding MAGIC steps are summarized in italics:

1. Define a basic gate topology – What type of elements are used (memristors, resistors, capacitors, *etc.*), how they are connected to each other and to the gate ports (*e.g.*, for voltage excitation) – *3 memristors, two of which connected in parallel.*
2. Define the inputs/outputs of the gate. All input values must be stable prior to execution and output values should be ready for reading when execution finishes. An output may run over an input value if needed, as in [4] – *Resistance values of $IN_{1,}$ IN_2 and OUT.*
3. Name all relevant circuit values that may change during execution (*e.g.*, voltage, current, memristance) – *Voltages V_{IN1}, V_{IN2}, V_{OUT} and V_G, Resistances R_{IN1}, R_{IN2}* and *R_{OUT}.*
4. Develop an expression to determine the momentary voltage/current on each of the memristors – *Equations (1a) and (1b).*
5. Make a truth table for the values of inputs and voltages at time *t*=0 for each of the circuit elements – *Table 1.*
6. Explore and understand the constraints in choosing the operating voltage/current and the initialization of output memristors (if they exist). For example, in the case of bipolar memristors, choosing the output memristor state

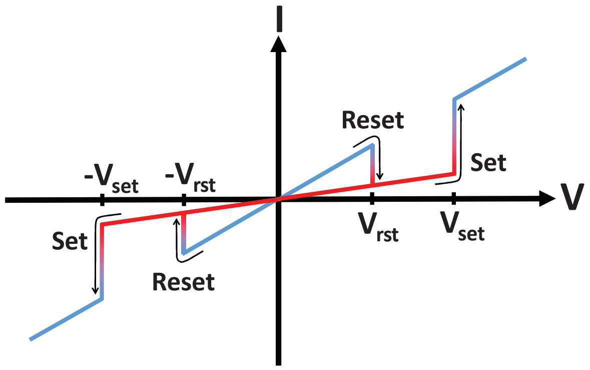

Figure 2. Non-polar memristor I-V curve.

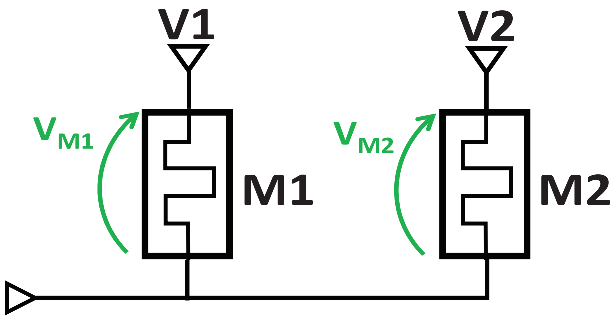

Figure 3. Non-polar gate topology.

TABLE II. NON-POLAR GATE INITIAL TRUTH TABLE

Inputs				Applied Voltage	
$M1_{init}$	$M2_{init}$	$R_{1,init}$	$R_{2,init}$	$V1$	$V2$
0	0	R_{OFF}	R_{OFF}	$\frac{V_{OP}}{2}$	$\frac{V_{OP}}{2}$
0	1	R_{OFF}	R_{ON}	$\sim V_{OP}$	~ 0
1	0	R_{ON}	R_{OFF}	~ 0	$\sim V_{OP}$
1	1	R_{ON}	R_{ON}	$\frac{V_{OP}}{2}$	$\frac{V_{OP}}{2}$

and applied voltage to allow the output memristor to change its state. Finally, after understanding the constraints, choose suitable values for operating voltage/current and output initialization value – *Initializing OUT to '1' and the constraint on V_G in (2).*

7. Examine the values that are still unconstrained, understand the dynamics of the circuit for different choices, and select the options that yield desired logic behavior – *Equations (4a) and (4b).*

Clauses 1 through 4 lay the groundwork for the new gate analysis. Clauses 5 and 6 verify that the chosen parameters do not infringe on constraints set by the circuit topology and device properties. And clause 7 is in fact the innovative leap in which a new logic gate with useful properties is created.

IV. DESIGN FOR A NOVEL TWO NON-POLAR MEMRISTOR LOGIC GATE

As an example for tightly following the proposed procedure for gate construction, we disclose the process behind the development of a novel, two non-polar memristor gate. In this gate, we use a device with non-polar characteristics, such as the one presented in [13]. These devices demonstrate a switching dynamics that depends only on the magnitude of the applied voltage, and independent on its polarity (*i.e.*, whether the voltage is positive or negative). Hence, applying a voltage greater than the reset voltage threshold (V_{rst}), in any polarity, switches the device to *HRS*. Further increasing the voltage, above the set voltage threshold (V_{set}), will switch the device to *LRS*. The switching characteristics of a generic non-polar memristor are shown in Figure 2. Following the proposed methodology, the gate is constructed as follows:

1. The chosen gate topology consists of two non-polar memristors connected in series, as shown in Figure 3. Note that this structure is compatible with a crossbar array topology.
2. The inputs are chosen as the initial states of both memristors. The output at this stage is still undetermined and can be the final state of either memristor.
3. The relevant naming is marked in Figure 3. The resistance of memristors *M1* and *M2* is denoted, respectively, as R_1 and R_2.
4. Expressions for the voltages on the memristors as a function of the applied voltage and momentary device resistance are given by

$$V_{M1} = (V1 - V2) \cdot \frac{R_1}{R_1 + R_2}, \tag{5a}$$

$$V_{M2} = (V2 - V1) \cdot \frac{R_2}{R_1 + R_2}. \tag{5b}$$

For simplicity, consider *V1* as ground and name *V2* as V_{OP}, (5a) and (5b) become

$$V_{M1} = -V_{OP} \cdot \frac{R_1}{R_1 + R_2}, \tag{6a}$$

$$V_{M2} = V_{OP} \cdot \frac{R_2}{R_1 + R_2}. \tag{6b}$$

5. The analysis for time t=0 is presented in Table II.
6. For the gate topology at hand there is only a single parameter to adjust (V_{OP}). This parameter is dealt with in the next clause. Therefore, there are no other constraints to be considered.
7. The memristor I-V curve is divided into sections $0<|Vrst|<|Vset|$ and the applied voltages on the memristors can be either 0, $V_{OP}/2$, or V_{OP}. There are three meaningful options for different values of V_{OP}:
   a. 0V→No change, $V_{OP}/2$→Reset, V_{OP}→Set
   b. 0V→No change, $V_{OP}/2$→No Change, V_{OP}→Reset
   c. 0V→No change, $V_{OP}/2$→No Change, V_{OP}→Set

   From Table II, one can conclude that option b will not lead to a switch in either of the memristors and is therefore uninteresting. For the remaining options, the final resistance of both memristors is identical. Hence, the output of the logic gate can be either of the

TABLE III. OPERATION RESULTS FOR NON-POLAR MEMRISTIVE GATE WITH DIFFERENT OPERATING VOLTAGES

Inputs		Initial state		Final state for option (a)		Final state for option (c)	
$R_{1,init}$	$R_{2,init}$	$V1$	$V2$	$R_{1,final}$	$R_{2,final}$	$R_{1,final}$	$R_{2,final}$
R_{OFF}	R_{OFF}	$\frac{V_{OP}}{2}$	$\frac{V_{OP}}{2}$	R_{OFF} (0)		R_{OFF} (0)	
R_{OFF}	R_{ON}	$\sim V_{OP}$	~ 0	R_{ON} (1)		R_{ON} (1)	
R_{ON}	R_{OFF}	~ 0	$\sim V_{OP}$	R_{ON} (1)		R_{ON} (1)	
R_{ON}	R_{ON}	$\frac{V_{OP}}{2}$	$\frac{V_{OP}}{2}$	R_{OFF} (0)		R_{ON} (1)	

memristors. Note that the operation of the gate is destructive.

Option (a) executes a XOR function, as can be seen in Table III. However, this gate is unstable since the final values for a XOR function are the basis for another "round" of calculation as long as the operation voltage is still applied. This will lead to a convergence of the gate to a value that is determined by the ON/OFF ratio of the memristors and by the *Vrst/Vset* quotient. If willing to allow for partial switching (asymptotically reaching full switching with the right device parameters) a XOR operation is still possible (verified by simulations). Realizing this function requires the I-V curve to maintain:

$$\left|\frac{V_{SET}}{V_{RST}}\right| > 2. \tag{7}$$

Option (c) executes an OR gate as listed in Table III. This function is stable and converges to the desired value for a wide range of parameter values. The condition needed to realize this gate is the complement condition of (7):

$$1 < \left|\frac{V_{SET}}{V_{RST}}\right| < 2. \tag{8}$$

The resulting gate can be used exclusively in one of the described options, depending on the physical properties of devices used. Some non-polar devices fulfil (7) [14-16] and therefore enable a XOR gate, while others fulfil (8) [17, 18] and can be used in OR gates. Some devices do not enable any of the two [18].

V. CONCLUSIONS

A design methodology for stateful memristive logic gates is presented in this paper. The methodology offers a simple procedure to be followed when designing novel circuits for executing logic functions using memristors. And is meant to guide stateful memristive gate developers toward an efficient and successful design process. By following this methodology, the design can be divided to a structured sequence of decisions. The range of possibilities for choosing gate parameters, inputs and outputs, even for basic gates, is usually too large to allow considering the benefits of all configurations simultaneously. By using the proposed procedure, a designer can rule out most of the combinations leading to unwanted behavior, as demonstrated in the two examples in sections II and IV.

Some initial decisions are still up to the designer to decide in the beginning of the design process. These include the topology of the gate and choice of inputs. The methodology may be extended in the future to support an analysis of multiple initial setups in parallel. Additionally, the proposed methodology is currently limited only to stateful logic gates. Future work may extend it for gates using other values (*e.g.*, node voltage) as inputs and outputs.

Finally, we do not discuss using the resulting gate design at the end of the process as a base for additional modifications, for example increasing the number of inputs for a given topology as done in [3]. We believe that designer who wishes to, can use the insight gained on the gate dynamics by following our methodology to extrapolate their design further.

REFERENCES

[1] L. O. Chua, “Memristor—The Missing Circuit Element,” *IEEE Transactions on Circuit Theory*, Vol. 18, No. 5, pp. 507–519, September 1971.

[2] 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, May 2008.

[3] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, “MAGIC - Memristor-Aided Logic,” *IEEE Transactions on. Circuits Syst. II Express Briefs*, Vol. 61, No. 11, pp. 895–899, November 2014.

[4] E. Lehtonen and M. Laiho, “Stateful Implication Logic with Memristors,” *Proceedings of the IEEE/ACM International Symposium on Nanoscale Architectures*, pp. 33–36, July 2009.

[5] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, “‘Memristive’ Switches Enable ‘Stateful’ Logic Operations via Material Implication.,” *Nature*, Vol. 464, No. 7290, pp. 873–876, April 2010.

[6] R. Rosezin, E. Linn, C. Kugeler, R. Bruchhaus, and R. Waser, “Crossbar Logic Using Bipolar and Complementary Resistive Switches,” *IEEE Electronic Device Letters*, Vol. 32, No. 6, pp. 710–712, April 2011.

[7] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. U. C. Weiser, and E. G. E. G. Friedman, “MRL - Memristor Ratioed Logic,” *Proceedings of the International*

Workshop on Cellular Nanoscale Networks and their Applications, pp. 1-6, August 2012.

[8] Y. Levy, J. Bruck, Y. Cassuto, E. G. Friedman, A. Kolodny, E. Yaakobi, and S. Kvatinsky, “Logic Operations in Memory Using a Memristive Akers Array,” *Microelectronics Jurnal*, Vol. 45, No. 11 pp. 1429–1437, November 2014.

[9] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, “Memristor-Based Material Implication (IMPLY) Logic: Design Principles and Methodologies,” *Proceedings of the IEEE Tranactions ob Very Large Scale Integration Systems,* Vol. 22, No. 10, pp. 2054–2066, October 2014.

[10] S. Kvatinsky, A. Kolodny, U. C. Weiser, and E. G. Friedman, “Memristor-Based IMPLY Logic Design Procedure,” *Proceedings of the IEEE 29th International Conference on Computer Design*, pp. 142–147, October 2011.

[11] I. Vourkas and G. Ch Sirakoulis, “Memristor-Based Combinational Circuits: A Design Methodology for Encoders/Decoders,” *Microelectronics Jurnal*, Vol. 45, No. 1, pp. 59–70, January 2014.

[12] G. Papandroulidakis, I. Vourkas, N. Vasileiadis, and G. C. Sirakoulis, “Boolean Logic Operations and Computing Circuits Based on Memristors,” *IEEE Transactions on Circuits and Systems II Express Briefs*, Vol. 61, No. 12, pp. 972–976, December 2014.

[13] M. Liu, Q. Liu, S. Long, and W. Guan, “Formation and Annihilation of Cu Conductive Filament in the Nonpolar Resistive Switching Cu/ZrO2:Cu/Pt ReRAM,” *Proceedings of the 2010 IEEE International Symposium on Circuits and Systems,* pp. 1–4, May 2010.

[14] W. I. Park *et al.*, “Self-Assembly-Induced Formation of High-Density Silicon Oxide Memristor Nanostructures on Graphene and Metal Electrodes,” *Nano Lett*ers, Vol. 12, No. 3, pp. 1235–1240, March 2012.

[15] H.-H. Huang, W.-C. Shih, and C.-H. Lai, “Nonpolar Resistive Switching in the Pt/MgO/Pt Nonvolatile Memory Device,” *Applied Physics Letters Jurnal*, Vol. 96, No. 19, pp. 1935, May 2010.

[16] W. Guan, S. Long, Q. Liu, M. Liu, and W. Wang, “Nonpolar Nonvolatile Resistive Switching in Cu Doped ZrO2,” *IEEE Electron Device Letters*, Vol. 29, No. 5, pp. 434–437, May 2008.

[17] W. Guan, M. Liu, S. Long, Q. Liu, and W. Wang, “On the Resistive Switching Mechanisms of Cu/ZrO2:Cu/Pt,” *Applied Physics Letters*, Vol. 93, No. 22, pp. 3506, December 2008.

[18] Y. Huang, Y. Luo, Z. Shen, G. Yuan, and H. Zeng, “Unipolar Resistive Switching of ZnO-Single-Wire Memristors,” *Nanoscale Research Letters*, Vol. 9, No 1, pp. 381-385, August 2014.