

Logic with Unipolar Memristors – Circuits and Design Methodology

Nimrod Wald^(✉), Elad Amrani, Avishay Drori,
and Shahar Kvatinsky^(✉)

Faculty of Electrical Engineering, Technion – Israel Institute of Technology,
3200003 Haifa, Israel
nimrodw@tx.technion.ac.il, shahar@ee.technion.ac.il

Abstract. Memristors are a general name for a set of emerging resistive switching technologies. These two terminal devices are characterized by a varying resistance, which is controlled by the voltage or current applied to them. The resistance state of a memristor is nonvolatile, and as such makes memristors attractive candidates for use as novel memory elements. Apart from their use for memory applications, the use of memristors in logic circuits is widely researched. A class of logic circuits named ‘stateful logic’, where the logic state of the inputs and outputs is stored in the form of resistance, is a promising approach for carrying out logic computations within memory. This chapter discusses the use of non-polar memristors, a type of memristors whose resistance depends only on the magnitude of the voltage across its terminals, for performing stateful logic operations. A design methodology is presented to allow structured development of stateful logic gates, and backed by a demonstration of the design process of OR and XOR gates using non-polar memristors.

Keywords: Memristor · Unipolar memristors · Resistive switch · Logic design · Design methodology · Stateful logic · In-memory computing · mMPU

1 Introduction

Memristor is a general term for a family of emerging technologies [1, 2], including metal oxide thin film resistive switches (RRAM or ReRAM) [3], spin torque transfer magneto-resistive RAM (STT-MRAM) [4] and phase change memory (PCM) [5]. The electrical properties of memristors were formulated in 1971 by Leon Chua [6] in an effort to achieve a symmetric relation between the known electric quantities of voltage, current, electric charge and magnetic flux. The research of memristors has been dormant from that time, until in 2008 researchers at Hewlett Packard (HP) laboratories have linked the known phenomenon of resistive switching to memristors [7]. Since then, research of memristors is being performed in the fields of memory, neuromorphic circuits [8], hardware security [9, 10] and logic [11]. Memristors are characterized by an intrinsic state variable, which determines the device resistance (sometimes called memristance), varying from a low resistance state (LRS , R_{ON}) to a high resistance state (HRS , R_{OFF}). The state variable represents the physical switching mechanism

(e.g. filament forming state in RRAM devices), and changes its value according to the current or voltage applied to the device.

Increasing power dissipation due to leakage in transistors as they are being shrunk is motivation for use of novel non-volatile devices for performing logic operations. Furthermore, the fact that processor performance increase greatly outpaces that of memories, causes a bottleneck named ‘*the memory wall*’, meaning that most energy and latency of computations is spent on moving data between the CPU and memory [12]. Using memristors, natural candidates for replacing conventional memory technologies, as logic elements could solve this problem by performing the logic operations within the memory, eliminating much of the need for fetching data. The combination of data storage and processing in a single element enables the design of memristive memory processing unit (mMPU) [13, 14]. Many methods for performing logic operations using memristors have been previously proposed, including memristor ratioed logic (MRL) [15], Akers logic arrays [16], complementary resistive switching (CRS) [17], implication logic (IMPLY) [18], and memristor-aided logic (MAGIC) [19]. The latter two utilize the state of memristors as the logic value of both inputs and output. This method is known as ‘*stateful logic*’ and is especially suited for performing logic within memory arrays [20, 21].

This chapter discusses the implementation of logic circuits using a more uncommon type of memristors, namely unipolar (or non-polar) memristors. The characteristics of these memristors are covered in Sect. 2, and an example for the use of such devices for logic design is presented in Sect. 3. A design methodology for developing stateful memristive logic gates with any type of memristors is described in Sect. 4, followed by another example of a unipolar memristive logic gate design in Sect. 5, pursuing the proposed methodology. All simulations are conducted using an internally developed VerilogA model for unipolar memristors, based on [22]. The chapter is concluded in Sect. 6.

2 Unipolar Memristors

The majority of research in the field of memristive logic concentrates on the use of bipolar memristors. These devices have a state variable that changes its value according to both the magnitude and polarity of the voltage. Thus, applying a positive voltage higher than a certain threshold V_{RESET} increases the resistance of the device up to HRS, and applying a negative voltage exceeding a negative threshold voltage V_{SET} lowers the resistance down to LRS. This work deals with the use of a different memristor, the unipolar memristor, which differs from bipolar memristors in the fact that only the magnitude of the voltage across the device determines the change in the resistance. Thus, applying a voltage higher than $|V_{RST}|$ across the device in any direction increases the resistance. Applying a voltage higher than a different threshold (e.g., $|V_{SET}| > |V_{RESET}|$) causes the resistance to drop. Once a device is switched to LRS, a compliance current limitation is usually necessary to avoid excess current that damages the device. Resistive switching technologies that result in unipolar switching behavior include PCM and some of RRAM technologies with thermochemical mechanism [23–27]. Examples for I-V curves of both bipolar and unipolar memristors are shown in Fig. 1.

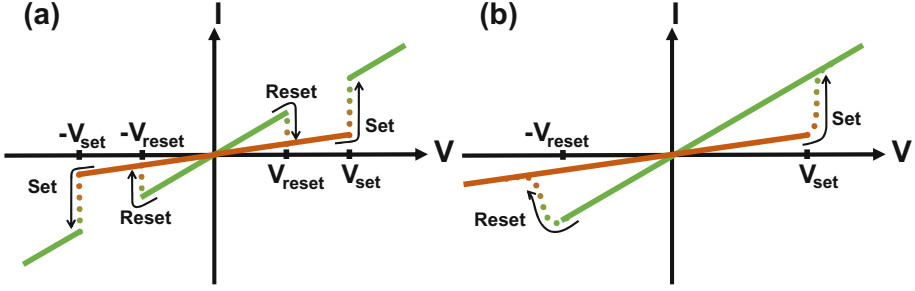


Fig. 1. I-V curves for unipolar (a) and bipolar (b) memristors. The regions in which the device is in LRS are in green, the ones in HRS are in orange, and the dotted lines are transitions between the two. (Color figure online)

We define the logic values stored in a memristor in the following manner, HRS is denoted as logical ‘0’ and LRS as logical ‘1’. The use of unipolar memristors for logic gates opens the possibility of performing computation within memristive arrays of types previously not considered for use as logic. Furthermore, the use of unipolar memristors allows designing simpler controllers and voltage sources due to the fact that only a single voltage polarity is required for switching back and forth.

3 A Unipolar Memristive Logic Gate Example

In this section, a concept to design logic gates with unipolar memristors is presented [28]. The operation mechanism is first presented, followed by examples of OR and NOT gates.

3.1 Operation Principle

The basic mechanism of the proposed logic technique is a voltage divider between two resistive elements: a memristor and a resistor for a NOT gate or two memristors for an OR gate. The proposed circuits are based on connecting two resistive elements in series and applying a voltage bias. The ratio of voltages on the two elements complies with the ratio of their resistance, *i.e.*, the states are distinguished using a bias voltage. The first step of operation is translating resistance to resistive states. The applied voltage for distinction is called the *preset voltage*.

After state distinction has been achieved, a higher voltage is applied to the circuit, adding higher applied voltage across both elements, regardless of their states. The voltage in this step is predetermined to a value that promotes switching if necessary for proper execution, thus this voltage is called the *evaluation voltage*. The operation is therefore comprised of two execution steps: *preset* and *switching*.

One obstacle to operate properly arises from the fact that every change in resistance immediately changes the voltages, hence, possibly changing the distinction between states. This phenomenon may lead to an incorrect result. Therefore, maintaining the initial voltage distinction for a sufficient time is required to reach the desired resistance

(HRS or LRS). One possible solution is to incorporate capacitors in the circuit in parallel with each resistive device. The capacitors add delay to the system due to the need to charge/discharge them during operation. Thus, we call them *suspension capacitors*. In addition to prolonging the validity of voltage values in the switching stage, suspension capacitors also delay the preset stage and in the case of the NOT gate, are actually mandatory for proper operation. Furthermore, the transition from preset to switching stages cannot be instantaneous. Hence, the intermediate evaluation stage is abstractly depicted as a transitive state and three stages are used to execute the operation as illustrated in Fig. 2.

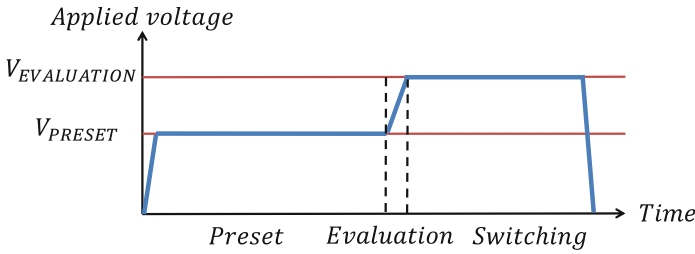


Fig. 2. The sequence of the applied voltage for the three stages of a general logic operation. The preset voltage distinguishes between logical states and charges the suspension capacitors. The evaluation stage converts the preceding voltages to the required voltages for switching.

(a) Preset Stage

In the preset stage, a voltage V_{PRESET} is applied to the circuit to charge the capacitors and initialize the voltage division between the resistive devices. The applied voltage is sufficiently high to distinguish between resistive states, but lower than the threshold voltage, thus does not change the state of the memristors. After sufficient time, approximately no current passes through the capacitors and their voltages are consistent with the voltage divider.

(b) Evaluation Stage

The evaluation stage starts immediately after the preset stage. A voltage pulse $V_{EVALUATION}$ is applied to the circuit. The purpose of this stage is to increase the voltage on both resistive elements abruptly. The final voltage in this stage depends on the final voltage of the preset stage, hence correlates with the resistance of the circuit elements. However, the voltage increase $V_{EVALUATION} - V_{PRESET}$ is fixed for all scenarios. The exact increase in voltage after the voltage jump is determined by the capacitance ratio (charge sharing).

(c) Switching Stage

In the switching stage, $V_{EVALUATION}$ is still applied for sufficient time to allow switching of the memristors. The key is to choose proper pulse length and voltage magnitude to switch the memristors according to the desired logical functionality.

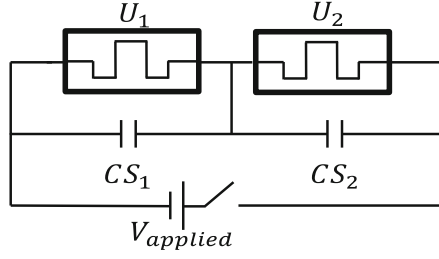


Fig. 3. Schematic of an OR gate. The input memristors U_1, U_2 are overwritten with the output.

3.2 OR Gate

A two-input OR gate consists of two unipolar memristors U_1 and U_2 connected in series. A suspension capacitor is connected in parallel to each memristor, as shown in Fig. 3. The initial logical state of the memristors is the input of the gate and after execution both memristors have the same logical state, which serves as the output of the gate.

Assume $V_{SET} > V_{RESET}$, for proper behavior of the gate certain conditions need to be fulfilled. First, when both inputs are identical (*i.e.*, both are logical ‘1’ or ‘0’) there is no memristor switching. Second, when the inputs are different, the HRS memristor (in logical ‘0’) has to switch to LRS since the desired output is logical ‘1’. Assuming that the voltage on the HRS memristor equals V_{PRESET} in the preset stage and $V_{PRESET} + \frac{1}{2}(V_{EVALUATION} - V_{PRESET})$ in the evaluation stage; the constraints on the voltages are therefore

$$V_{PRESET} < 2V_{RESET}, \quad (1a)$$

$$2V_{SET} - V_{PRESET} < V_{EVALUATION} < 2V_{RESET}. \quad (1b)$$

Figure 4 shows simulation results of an OR gate for the case where the inputs are different and U_2 switches for proper result. Note that when U_1 is logical ‘0’ and U_2 is logical ‘1’, the operation is destructive, *i.e.*, the value of the inputs is overwritten.

3.3 NOT Gate

The NOT gate consists of a single unipolar memristor connected in series with a reference resistor. The memristor acts as both input and output of the NOT gate. For proper operation both the memristor and the resistor have a suspension capacitor connected to them in parallel as shown in Fig. 5. Without the suspension capacitors, $V_{EVALUATION}$ must be absurdly high to allow switching the memristor in the case of RESET operations. The resistance of the reference resistor is between LRS and HRS. This value ensures that the voltage at the end of the preset stage across a HRS (LRS) memristor is high (low), as illustrated in Fig. 6. A reasonable choice is $R_{REF} = \sqrt{R_{OFF}R_{ON}}$. For proper operation, the conditions on the applied voltage are

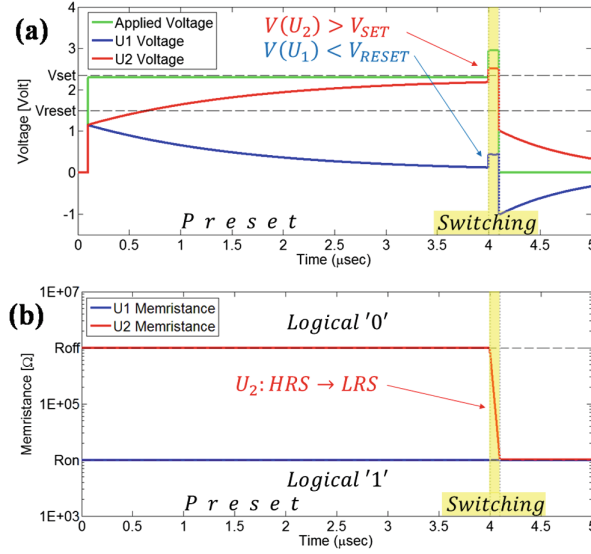


Fig. 4. OR gate simulation results. U_1 and U_2 are initialized to, respectively, LRS (logical ‘1’) and HRS (logical ‘0’). (a) Voltages across the memristors during the operation, and (b) their resistance. In the first 4 μs the system is in the preset stage, and the capacitors are charged/discharged to distinctive voltages. In the switching stage, U_2 voltage is higher than V_{set} for sufficient time and its logical value is switched to logical ‘1’ as desired.

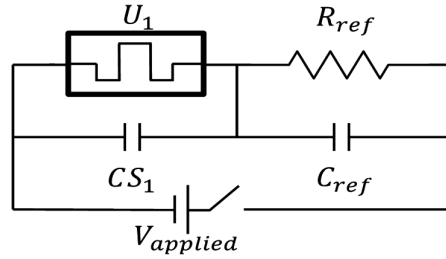


Fig. 5. Schematic of a NOT gate. A resistor is used as a reference to determine the state of the memristor

$$V_{PRESET} < \min \left\{ \sqrt{\frac{R_{OFF}}{R_{ON}}} V_{RESET}, V_{SET} \right\}, \quad (2a)$$

$$\frac{1}{\gamma} \max \{ V_{SET}, V_{RESET} + V_{PRESET} \} < V_{EVALUATION}, \quad (2b)$$

$$V_{EVALUATION} < \frac{1}{\gamma} (V_{SET} + V_{PRESET}), \quad (2c)$$

where $\gamma \triangleq \frac{C_{REF}}{C_{REF} + C_{S1}}$.

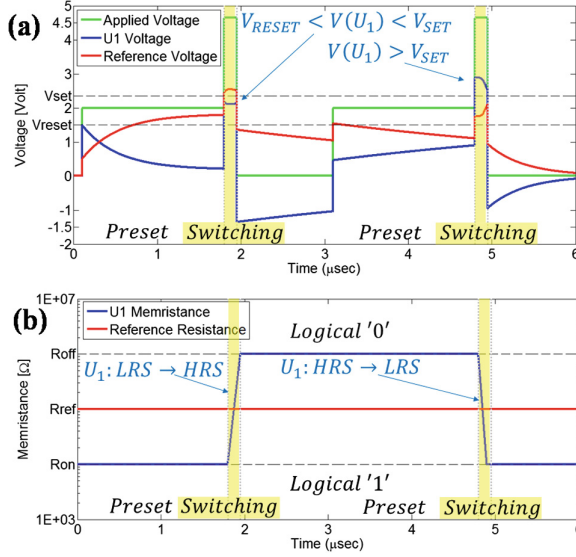


Fig. 6. NOT gate simulation results. (a) Voltages and (b) resistance during two consecutive memristor switching. In the first 3 μ s, U_1 switches from *LRS* \rightarrow *HRS*. In the second NOT operation U_1 switches back to *LRS*.

3.4 Timing Considerations

One of the critical points for proper behavior of the proposed logic technique is to apply the right voltage for a sufficient time during the switching stage. In this section, the timing constraints in the switching stage are explored. Assume $\tau_{SET}(\tau_{RESET})$ is a minimal transition time from HRS (LRS) to LRS (HRS) [29]. For successful switching, the duration of the switching stage must be greater than the minimal required switching time. The minimum condition on the length of the stage is therefore

$$T_{pulse} > \max\{\tau_{set}, \tau_{reset}\} = T_{pulse,min}. \quad (3)$$

At the beginning of the switching stage, each memristor is biased with a voltage which promotes switching (if necessary). The validity of the specified voltage level is maintained for a short period of time, due to the use of suspension capacitors, but will eventually become invalid. If the switching stage is not terminated in time, a memristor might reach a voltage range which promotes the opposite transition, *i.e.*, reverse switching. The maximal length of the switching stage is determined according to the

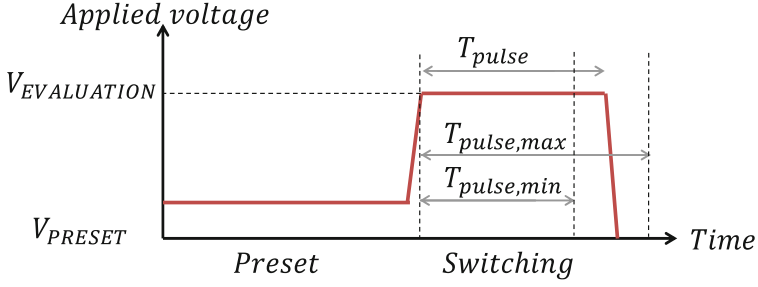


Fig. 7. Applied voltage duration in the switching stage. T_{pulse} satisfies (3) to reach the desired resistance and also meets (4) to avoid reverse switching.

transient analysis of voltages in the circuit, and might be different for SET and RESET operations. For this purpose it is possible to define T_{SET} (T_{RESET}) as the approximate period of time in which the conditions for a SET (RESET) operation are met. It is important to understand that while τ_{set} and τ_{reset} are properties of the memristor, T_{SET} and T_{RESET} are determined by the selection of the different circuit parameters, namely V_{PRESET} , $V_{EVALUATION}$, R_{REF} , C_{REF} , C_S , and T_{PRESET} . Hence, the maximum condition on the length of the switching step is

$$T_{pulse} < \min\{T_{SET}, T_{RESET}\} = T_{pulse,max}. \quad (4)$$

To comply with both minimum and maximum conditions, both (3) and (4) must apply, as illustrated in Fig. 7. The parameters V_{PRESET} , $V_{EVALUATION}$, R_{ref} , and the switching capacitors can be chosen to support (3) and (4). Different circuit parameters, however, may lead to a reduction in performance. For example, larger capacitors ease the maximum condition, but slow the preset stage and increase power consumption.

3.5 Evaluation and Comparison

We evaluate the proposed circuits in terms of speed, power, and area, and compare them to previously proposed memristive logic families that are suitable for bipolar memristors. Evaluation is conducted using the model mentioned in Sect. 1 based on [22]. All simulations are conducted in Cadence Virtuoso environment, and using device parameter values of $R_{ON} = 10 \text{ k}\Omega$, $R_{OFF} = 1 \text{ M}\Omega$, $V_{SET} = 2.5 \text{ V}$ and $V_{RESET} = 1.5 \text{ V}$. In terms of speed, the need for a long preset stage is a disadvantage of the proposed mechanism. To accelerate the preset stage, higher voltages can be used in the cost of higher power consumption. Our simulations show that for a memristor with switching time τ , the delay time of the presented basic logic gates (OR\NOT) is approximately $10 \cdot \tau$.

The basic cell that would be incorporated into a crossbar array consists of a memristor and a capacitor. Suspension capacitors increase the area of the memory cell; the exact area of the capacitor depends on the switching time of the memristor. For example, memristors with switching time of 1 ns require suspension capacitors with

Table 1. Latency and area of different functions using OR, NOT and COPY

<i>Operation</i>	# <i>OR</i>	# <i>NOT</i>	# <i>COPY</i>	# <i>Backup</i>	<i>Latency</i>
<i>NOT</i>	0	1	0	0	1
<i>OR</i>	1	0	0	0	1
<i>NAND</i>	1	2	0	0	3
<i>NOR</i>	1	1	0	0	2
<i>AND</i>	1	3	0	0	4
<i>XOR</i>	3	4	2	2	9
<i>ADD 1 bit</i>	4	7	2	2	13
<i>ADD N bit</i>	$11N - 7$	$14N - 7$	$9N - 7$	5	$34N - 21$

capacitance of approximately 0.8 pF. The usage of suspension capacitors clearly impacts power consumption. Furthermore, the use of several computing phases (preset-switching) requires a clock that contributes to the power consumption and needs to be considered as well.

Some bipolar logic techniques for computation within memory are IMPLY [18] and MAGIC [19]. IMPLY and MAGIC are stateful logic techniques, similar in nature to the proposed technique. In both techniques, logical state is represented by resistance and the computation consists of multi-stage voltage application. Similarly to our proposed unipolar technique, in IMPLY the input data is overwritten with the output result. For devices with switching time of τ , the switching times of IMPLY and MAGIC are 3.15τ and 1.3τ respectively. To compare the fundamentals of the performance and area of the different techniques, we have evaluated a test case of an N -bit adder. Recent unipolar and bipolar memristor technology exhibit switching times in the order of 1 ns–10 ns and device area of $4F^2$ [30], making IMPLY and LOGIC comparable to each other and to the proposed logic.

Assume the operation is incorporated in a crossbar that is optimized for area, *e.g.*, only a single operation can be performed at a clock cycle and backup devices can be discarded after usage. The latency and number of backup memristors needed for different logical operations are listed in Table 1. A single bit addition can be performed in 13 cycles. An N bit addition can be performed in $34N - 21$ cycles. A comparison of this result with existing bipolar logic families is presented in Table 2. Note that due to the requirement of a long preset stage, logic execution for the proposed logic is slower. Given the capacitance and memristor resistance used in simulations, the preset stage is in the order of 100 ns. Thus, the operating frequency of the proposed method is probably lower than the bipolar methods, possibly reducing performance.

Table 2. Latency and area of N -bit adder with different memristor-based logic methods, optimized for minimum area

Method of Execution	Latency (# Cycles)	Latency (τ)	Area (# Memristors)
IMPLY (Serial) [10]	29N	$91N \cdot \tau$	2
MAGIC [17]	15N	$19N \cdot \tau$	5
Unipolar (This work)	34N-21	$(340N-210)\tau$	5

4 Methodology for Stateful Memristive Logic Design

One of the most important things when designing novel stateful memristive logic gates is proper selection of the circuit parameters, *i.e.* voltages, currents, resistances, etc. The space of possibilities for choosing these is usually too large to explore, forcing the designer to rely on heuristics. Recently, we have proposed a set of steps to form a structured methodology for the design of stateful memristor-based logic gates [31]. This methodology improves efficiency when inventing new stateful memristive logic gates, and allows a systematic choice of circuit parameters. The design process consists of seven steps, as detailed next. The methodology treats voltage across a memristor as the value that determines its dynamic behavior (switching). While this methodology assume voltage-controlled memristors [32], the same methodology can be adapted with small adjustments for current-controlled memristors. The steps of the design methodology are:

1. *Definition of gate topology* – Decide what are the elements being used (memristors, resistors, capacitors, *etc.*), and how are they connected to each other and to the ports of the logic gate (*e.g.*, connecting the gate to external voltage/current sources).
2. *Definition of gate inputs/outputs* – Decide which memristor values are used as input variables and which as output. All the inputs must have their updated values prior to execution. The output values should be written to the output memristor before execution finishes. An output may run over an input value if needed, as in the OR gate in the previous section and in [18]. When several options exist, this step may be postponed until after step 6 to make a decision relying on a better understanding of the circuit dynamics.
3. *Naming of relevant circuit parameters* that may change their value during execution (*e.g.*, voltage, current, memristance).
4. *Developing an expression for the momentary voltage/current* on each of the memristors in the circuit.
5. *Constructing a truth table of initial voltages* - For each combination of input values, determine what are the voltages across each circuit element at time $t = 0$ (*i.e.*, before any change is observed).
6. *Exploring constraints for choosing the operating voltage/current and the initialization of output memristors* (if they exist). For example, when using bipolar

memristors, the initial state of the output memristor and the applied voltage must be carefully chosen to allow a change of the state.

7. *Examining the unconstrained values and understanding the circuit dynamics* - To allow the proper ranges for each unconstrained value that may produce different behaviors. Once the behavior of the memristors for all parameter ranges is known, select the options that yield the desired logic functionality.

Clauses 1 through 4 are basic groundwork for the gate analysis. Clauses 5 and 6 put restrictions on the chosen parameters so they do not infringe on constraints set by the circuit topology and device properties. Clause 7 requires the most in-depth analysis and should result in parameter selection leading to a new logic gate with useful properties. We demonstrate this design methodology in the next section for unipolar memristors.

5 Design Procedure for a Novel Unipolar Memristor Based Logic Gate

The methodology presented in the previous section is demonstrated for developing another logic gate using unipolar memristors. The steps followed in the development of the gate are presented next.

1. The gate comprises of two unipolar memristors connected in series. The structure, shown in Fig. 8, is compatible for use within a crossbar array.
2. The inputs of the gate are represented by the resistances of the two memristors before the logic function is executed. The output is not selected at this point and will be dealt after step 6. Note that either memristor can be selected as an output after execution since the circuit is symmetrical, and that the lack of a dedicated output memristor makes the gate undoubtedly destructive to at least one of the inputs.
3. The memristors are named $M1$ and $M2$, and their resistance, voltage drops and applied voltages to connected terminals are respectively denoted R_1 , V_{M1} , V_1 and R_2 , V_{M2} , V_2 . These notations are shown in Fig. 8.
4. The expressions of the momentary voltages as functions of the applied voltages to the gate terminals are given by

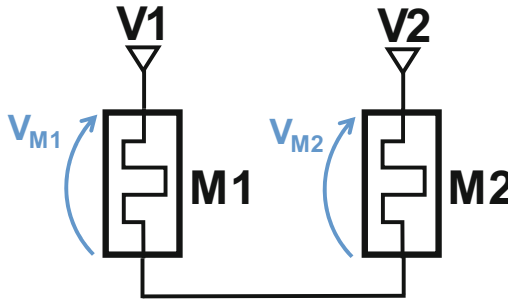


Fig. 8. Gate topology of the analyzed two unipolar logic gate.

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

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

To simplify (5a) and (5b), $V1$ is set as ground and $V2$ is named V_{OP} . Thus, the simplified expressions are

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

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

5. A truth table for the applied voltage on each device prior to logic execution is presented in Table 3.
6. The chosen topology involves only a single parameter (V_{OP}), whose value will be set in the next clause. Due to the fact that the memristors are unipolar and connected in a symmetric manner, there are no constraints on the polarity of the voltage.
7. Examining the I-V curve shown in Fig. 1, we see that $0 < |V_{reset}| < |V_{set}|$. The initial truth table demonstrates that any single memristor within the gate has either 0, $V_{OP}/2$, or V_{OP} applied across it. Considering all of the above, three meaningful options for selecting the value of V_{OP} are present:
 - (a) $0V \rightarrow$ No change, $2V_{OP}/2 \rightarrow$ Reset, $V_{OP} \rightarrow$ Set.
 - (b) $0V \rightarrow$ No change, $V_{OP}/2 \rightarrow$ No change, $V_{OP} \rightarrow$ Reset.
 - (c) $0V \rightarrow$ No change, $V_{OP}/2 \rightarrow$ No change, $V_{OP} \rightarrow$ Set.

Option (b) does not lead to switching of any of the memristors. As is apparent in Table 4, both remaining options lead to an identical state in both memristors at the end of the computation. Hence, we are free to choose the output of the gate to be either of the memristors, affirming the conclusion of step 2.

Table 3. Truth table for memristor voltages before any change in device state

Input Values		Input Resistance		Applied Voltage	
$M1_{init}$	$M2_{init}$	$R_{1,init}$	$R_{2,init}$	V_{M1}	V_{M2}
0	0	R_{OFF}	R_{OFF}	$V_{OP}/2$	$V_{OP}/2$
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}	$V_{OP}/2$	$V_{OP}/2$

Table 4. Analysis of gate operation for the two relevant operating voltage selections

Input Resistance		Initial Applied Voltage		Option (a) Final state		Option (c) Final state	
$R_{1,init}$	$R_{2,init}$	V_{M1}	V_{M2}	$R_{1,final}$	$R_{2,final}$	$R_{1,final}$	$R_{2,final}$
R_{OFF}	R_{OFF}	$V_{OP}/2$	$V_{OP}/2$	$R_{OFF} (0)$		$R_{OFF} (0)$	
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}	$V_{OP}/2$	$V_{OP}/2$	$R_{OFF} (0)$		$R_{ON} (1)$	

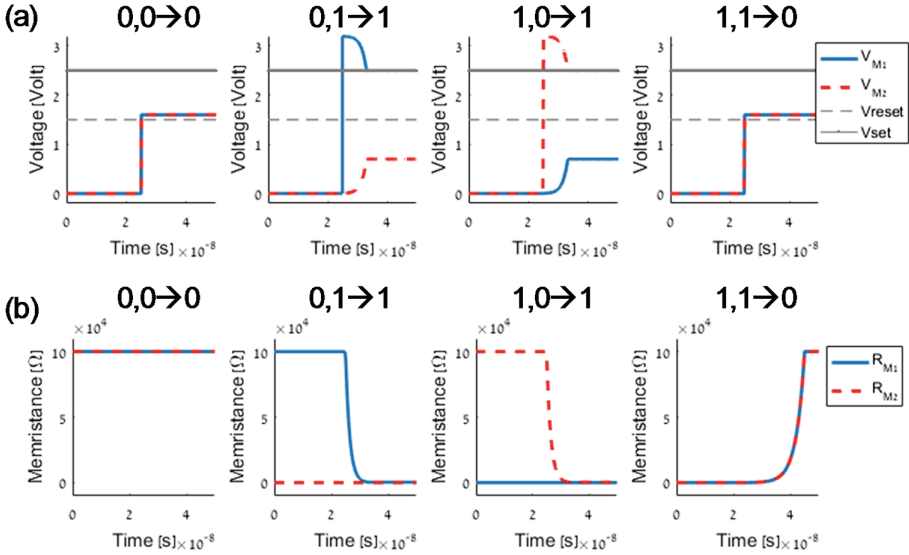


Fig. 9. (a) Applied voltage and (b) memristance for an XOR gate. The memristor is characterized by $R_{ON} = 100 \, \Omega$ and $R_{OFF} = 100 \, \text{k}\Omega$. The circuit parameters are $V_{RESET} = 1.5 \, \text{V}$, $V_{SET} = 2.5 \, \text{V}$, $V_{OP} = 3.2 \, \text{V}$.

Option (a) results in an XOR gate. However, this gate is unstable since the output values for an XOR function are, theoretically, initial values for another round of computation, resulting in a constant output equal to R_{OFF} . Using the model discussed in Sect. 1, our results show convergence of the output at a resistance of approximately R_{ON} . The exact value depends on R_{OFF}/R_{ON} and V_{set}/V_{reset} , as shown in Fig. 9. Thus, executing an XOR operation is possible if we allow partial switching, although the noise margin of the gate is relatively low (asymptotically reaching a full switching with a proper selection of parameters, improving the noise margin).

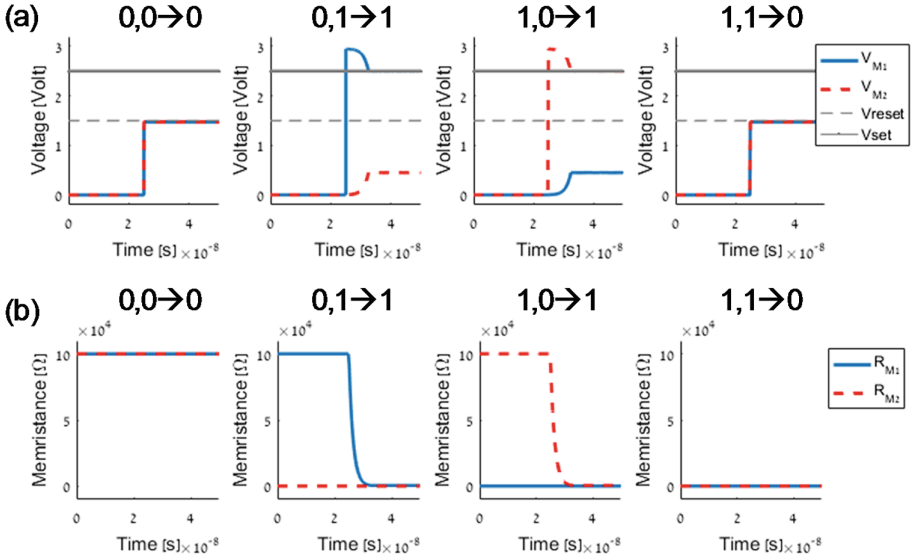


Fig. 10. (a) Applied voltage and (b) memristance of an OR gate. Memristor and circuit parameters are identical to an XOR gate, except $V_{OP} = 2.95$ V

Option (c) results in an OR gate. This gate is stable and, with a wide range of parameter values, correctly converges to the desired output with no noise margin issues. Simulation results of this gate are shown in Fig. 10. For proper operation of this OR gate the threshold voltages of the memristors are required to uphold

$$1 < \left| \frac{V_{SET}}{V_{RST}} \right| < 2. \quad (7)$$

Some physical unipolar devices exhibit (7) [33, 34], while other devices exhibit a higher ratio ($2V_{RESET} < V_{SET}$) [35–37], enabling only XOR operations, or do not fulfill any of these conditions (*i.e.*, uphold $V_{SET} < V_{RESET}$) [34] and therefore are not suitable for use with the proposed topology.

Contrary to the gate described in Sect. 3, these gates do not contain any capacitors, nor do they rely on retaining previous voltage divider values. For these reasons, there are no timing constraints on gate operation, apart from the obvious necessity to apply V_{OP} for a time sufficient for achieving full swing in the device states (τ). This time depends on properties of the used device and may vary substantially between different types of devices.

The gate described in this section outperforms the gates from Sect. 3 in several aspects. First, the topology does not include the use of capacitors or resistors, which is area efficient, allows implementing gates within a pure memristive crossbar, and eliminates the need to use two different input voltages to perform the logic function. Second, the topology allows, with a proper selection of devices and parameters, to use the same gate for two different logic functions by changing only the operating voltage.

6 Conclusions

Combining data storage and processing is appealing since it can solve some of the critical issues in modern computing, such as limited memory bandwidth and power consumption. Both unipolar and bipolar memristors enable the execution of logic operations within memory using different methods. Since it is still unclear whether unipolar or bipolar mechanisms will become dominant for data storage, both phenomena are of interest. In this chapter, we focus on unipolar mechanism and propose logic techniques for these devices using NOT, XOR and two types of OR gates. The proposed techniques can be naturally integrated within memristive crossbar memory. The proposed technique can fit different unipolar technologies such as Phase Change Memory, 3D-Xpoint, RRAM, and Thermochemical Resistive Memory.

We present how a design methodology helps in the invention of new logic gates that can be executed within memristive memories to form memristive memory processing units (mMPPU). The methodology is demonstrated by designing XOR and OR gates. This procedure is formed from a series of simple steps, and meant to facilitate a successful choice of circuit parameters and an overall efficient design process.

References

1. Chua, L.: If it's pinched it's a memristor. *Semicond. Sci. Technol.* **29**(10), 104001 (2014)
2. Chua, L.: Resistance switching memories are memristors. *Appl. Phys. A* **102**(4), 765–783 (2011)
3. Wong, H.S.P., et al.: Metal–oxide RRAM. *Proc. IEEE* **100**(6), 1951–1970 (2012)
4. Diao, Z., et al.: Spin-transfer torque switching in magnetic tunnel junctions and spin-transfer torque random access memory. *J. Phys. Condens. Matter* **19**(16), 165209 (2007)
5. Wong, H.S.P., et al.: Phase change memory. *Proc. IEEE* **98**(12), 2201–2227 (2010)
6. Chua, L.: Memristor-the missing circuit element. *IEEE Trans. Circ. Theory* **18**(5), 507–519 (1971)
7. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. *Nature* **453**(7191), 80–83 (2008)
8. Jo, S.H., Chang, T., Ebong, I., Bhadviya, B.B., Mazumder, P., Lu, W.: Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* **10**(4), 1297–1301 (2010)
9. Rose, G.S., McDonald, N., Yan, L.-K., Wysocki, B.: A write-time based memristive PUF for hardware security applications. In: *Proceedings of the International Conference on Computer-Aided Design*, pp. 830–833 (2013)
10. Rajendran, J., Rose, G.S., Karri, R., Potkonjak, M.: Nano-PPUF: a memristor-based security primitive. In: *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, pp. 84–87 (2012)
11. Borghetti, J., Snider, G.S., Kuekes, P.J., Yang, J.J., Stewart, D.R., Williams, R.S.: 'Memristive' switches enable 'stateful' logic operations via material implication. *Nature* **464**(7290), 873–876 (2010)
12. McKee, S.A.: Reflections on the memory wall. In: *Proceedings of the First Conference on Computing Frontiers on Computing frontiers*, p. 162 (2004)
13. Ben Hur, R., Kvatinsky, S.: Memory processing unit for in-memory processing. In: *Proceedings of the International Symposium on Nanoscale Architectures*, p. 208 (2016)

14. Ben Hur, R., Kvatinsky, S.: Memristive memory processing unit (MPU) controller for in-memory processing. In: Proceedings of the IEEE International Conference on the Science of Electrical Engineering (ICSEE) (2016)
15. Kvatinsky, S., Wald, N., Satat, G., Kolodny, A., Weiser, U.C., Friedman, E.G.: MRL — memristor ratioed logic. In: Proceedings of the International Workshop on Cellular Nanoscale Networks and Their Applications, pp. 1–6 (2012)
16. Levy, Y., et al.: Logic operations in memory using a memristive Akers array. *Microelectron. J.* **45**, 1429–1437 (2014)
17. Rosezin, R., Linn, E., Kugeler, C., Bruchhaus, R., Waser, R.: Crossbar logic using bipolar and complementary resistive switches. *IEEE Electron Device Lett.* **32**(6), 710–712 (2011)
18. Kvatinsky, S., Satat, G., Wald, N., Friedman, E.G., Kolodny, A., Weiser, U.C.: Memristor-based material implication (IMPLY) logic: design principles and methodologies. *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **22**(10), 2054–2066 (2014)
19. Kvatinsky, S., et al.: MAGIC - memristor-aided logic. *IEEE Trans. Circuits Syst. II Express Briefs* **61**(11), 895–899 (2014)
20. Ben Hur, R., Talati, N., Kvatinsky, S.: Algorithmic considerations in memristive memory processing units (MPU). In: Proceedings of the International Workshop on Cellular Nanoscale Networks and their Applications (2016)
21. Talati, N., Gupta, S., Mane, P., Kvatinsky, S.: Logic design within memristive memories using memristor-aided logic (MAGIC). *IEEE Trans. Nanotechnol.* **15**(4), 635–650 (2016)
22. Su Kim, Y., Min, K.-S.: Behavioral Current-voltage model with intermediate states for unipolar resistive memories. *J. Semiconductor Technol. Sci.* **13**(6), 539–545 (2013)
23. Ielmini, D., Bruchhaus, R., Waser, R.: Thermochemical resistive switching: materials, mechanisms, and Scaling projections. *Phase Transit.* **84**(7), 570–602 (2011)
24. Long, S., Cagli, C., Ielmini, D., Liu, M., Sune, J.: Cell-based models for the switching statistics of RRAM. In: Proceedings of the Annual Non-Volatile Memory Technology Symposium, pp. 1–5 (2011)
25. Tran, X.A., et al.: High performance unipolar AlO_y/HfO_x/Ni based RRAM compatible with Si diodes for 3D application. In: Symposium on VLSI Technology - Digest of Technical Papers, pp. 44–45 (2011)
26. Pirovano, A., Lacaita, A.L., Benvenuti, A., Pellizzer, F., Bez, R.: Electronic switching in phase-change memories. *IEEE Trans. Electron. Devices* **51**(3), 452–459 (2004)
27. Redaelli, A., Pirovano, A., Pellizzer, F., Lacaita, A.L., Ielmini, D., Bez, R.: Electronic switching effect and phase-change transition in chalcogenide materials. *IEEE Electron Device Lett.* **25**(10), 684–686 (2004)
28. Amrani, E., Drori, A., Kvatinsky, S.: Logic design with unipolar memristors. In: Proceedings of the International Conference on Very Large Scale Integration (VLSI-SoC), pp. 1–5 (2016)
29. Waser, R., Dittmann, R., Staikov, G., Szot, K.: Redox-based resistive switching memories - nanoionic mechanisms, prospects, and challenges. *Adv. Mater.* **21**(25–26), 2632–2663 (2009)
30. Yang, J.J., Strukov, D.B., Stewart, D.R.: Memristive devices for computing. *Nat. Nanotechnol.* **8**(1), 13–24 (2012)
31. Wald, N., Kvatinsky, S.: Design methodology for stateful memristive logic gates. In: Proceedings of the ICSEE International Conference on the Science of Electrical Engineering (ICSEE) (2016)
32. Kvatinsky, S., Ramadan, M., Friedman, E.G., Kolodny, A.: VTEAM: a general model for voltage-controlled memristors. *IEEE Trans. Circ. Syst. II Express Briefs* **62**(8), 786–790 (2015)
33. Guan, W., Liu, M., Long, S., Liu, Q., Wang, W.: On the resistive switching mechanisms of Cu/ZrO₂:Cu/Pt. *Appl. Phys. Lett.* **93**(22), 223506 (2008)

34. Huang, Y., Luo, Y., Shen, Z., Yuan, G., Zeng, H.: Unipolar resistive switching of ZnO-single-wire memristors. *Nanoscale Res. Lett.* **9**(1), 381 (2014)
35. Park, W.I., et al.: Self-assembly-induced formation of high-density silicon oxide memristor nanostructures on graphene and metal electrodes. *Nano Lett.* **12**(3), 1235–1240 (2012)
36. Huang, H.H., Shih, W.C., Lai, C.H.: Nonpolar resistive switching in the Pt/MgO/Pt nonvolatile memory device. *Appl. Phys. Lett.* **96**(19), 193505 (2010)
37. Guan, W., Long, S., Liu, Q., Liu, M., Wang, W.: Nonpolar nonvolatile resistive switching in Cu doped ZrO₂. *IEEE Electron Device Lett.* **29**(5), 434–437 (2008)

VLSI-SoC: System-on-Chip in the Nanoscale Era -

Design, Verification and Reliability

24th IFIP WG 10.5/IEEE International Conference on

Very Large Scale Integration, VLSI-SoC 2016, Tallinn,

Estonia, September 26-28, 2016, Revised Selected

Papers

Hollstein, Th.; Raik, J.; Kostin, S.; Tšertov, A.; O'Connor,

I.; Reis, R. (Eds.)

2017, XIV, 233 p. 129 illus., Hardcover

ISBN: 978-3-319-67103-1