SCIENTIFIC REPORTS
natureresearch

OPEN

Stateful Three-Input Logic with Memristive Switches

A. Siemon[1,2], R. Drabinski[1,2], M. J. Schultis[3], X. Hu[3], E. Linn[1,2], A. Heittmann[4], R. Waser[1,2,4,6], D. Querlioz[5], S. Menzel[2,6] & J. S. Friedman[3,5]

Received: 30 November 2018
Accepted: 19 September 2019
Published online: 10 October 2019

Memristive switches are able to act as both storage and computing elements, which make them an excellent candidate for beyond-CMOS computing. In this paper, multi-input memristive switch logic is proposed, which enables the function X OR (Y NOR Z) to be performed in a single-step with three memristive switches. This ORNOR logic gate increases the capabilities of memristive switches, improving the overall system efficiency of a memristive switch-based computing architecture. Additionally, a computing system architecture and clocking scheme are proposed to further utilize memristive switching for computation. The system architecture is based on a design where multiple computational function blocks are interconnected and controlled by a master clock that synchronizes system data processing and transfer. The clocking steps to perform a full adder with the ORNOR gate are presented along with simulation results using a physics-based model. The full adder function block is integrated into the system architecture to realize a 64-bit full adder, which is also demonstrated through simulation.

The ever-increasing density of transistors in integrated circuits has spurred a revolution of engineering and technology over the last 50 years. With the increase in density quickly approaching its theoretical limits in silicon processes, continued innovation is required to catalyze additional advancements for computing into the next 50 years[1,2].

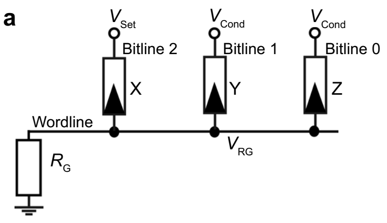
With standard CMOS approaching its theoretical limit for minimum feature size, the ability to produce circuits that operate at increasing frequencies is limited due to power dissipation[3]. Furthermore, von Neumann architectures have fundamental speed and power limitations resulting from the continual transfer of data between the processor and memory[4]. This so-called "von Neumann bottleneck" can be avoided if the information required computing an operation is already present within or near the processing unit[5–8]. In particular, if data is stored in the same location in which it is also being processed, a marked increase in calculation speed and decrease in energy dissipation may be achievable[9,10]. The research community is therefore searching for a device and related logic system that can both execute functions and store data in such a non-von Neumann computing architecture.

Memristive switches are particularly promising to aid in the advancement of beyond-CMOS computing[11–13]. Concisely, a memristive device can be switched by appropriate voltage stimuli between at least two different resistance states: a high resistive state (HRS) and a low resistive state (LRS)[14,15]. A very promising class of memristive switches are redox-based memristive switches based on the valence change mechanism (VCM) and the electrochemical metallization mechanism (ECM)[16]. These devices consist of a metal/insulator/metal structure. The motion of charged ions within the insulating layer is the origin of the memristive switching phenomenon. Thus, the switching operation is inherently bipolar. The device switches from the HRS to the LRS (SET operation) with one voltage polarity and back to the HRS (RESET operation) with the other polarity.

Memristive switches have been proposed as building blocks for beyond CMOS computing devices in von Neumann architectures due to their ultrahigh scalability[17–21]. Besides this, they can be exploited for *non*-von Neumann computing architectures. It has been demonstrated that memristive switches are able to compute all standard Boolean logic functions, and therefore are considered functionally complete[17,20,22–28]. Prominent examples are CRS-logic[29,30], MAGIC logic family[31] or IMPLY logic[17]. The CRS logic uses as inputs the applied voltages

[1]Institut für Werkstoffe der Elektrotechnik II (IWE II), RWTH Aachen University, Sommerfeldstr. 24, 52074, Aachen, Germany. [2]JARA-Fundamentals for Future Information Technology, Jülich, Germany. [3]Department of Electrical and Computer Engineering, The University of Texas at Dallas, Richardson, 75080, Texas, USA. [4]Peter Grünberg Institut 10 (PGI-10) Forschungszentrum Jülich GmbH, Jülich, Germany. [5]Centre de Nanosciences et de Nanotechnologies, CNRS, Univ. Paris-Sud, Université Paris-Saclay, Palaiseau, 91120, France. [6]Peter Grünberg Institut 7 (PGI-7) Forschungszentrum Jülich GmbH, Jülich, Germany. Correspondence and requests for materials should be addressed to S.M. (email: st.menzel@fz-juelich.de)

a



b Truth Table of the ORNOR Logic Gate

X Y Z	ORNOR (X, Y, Z)
0 0 0	1
0 0 1	0
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	1

c Simulation Parameter

V_{Set}	V_{Cond}	R_G	t_c
1.45 V	1.24 V	15 kΩ	1 μs

d

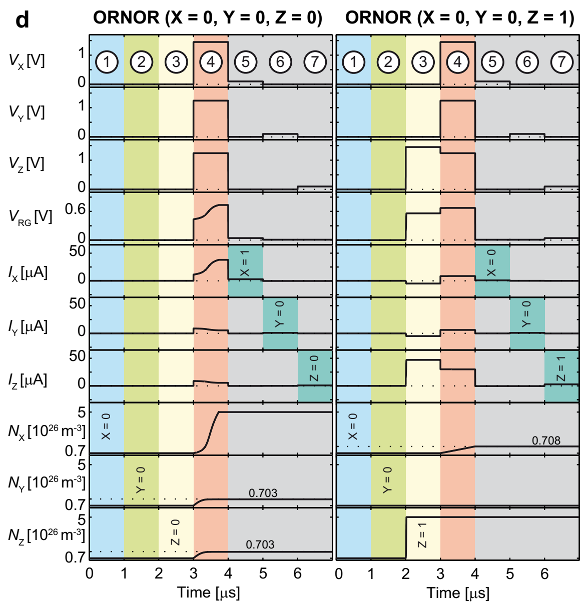


Figure 1. The ORNOR gate's (**a**) structure and (**b**) truth table. (**c**) Simulation parameter for the circuit of (**a**). (**d**) The simulation of the critical cases of the ORNOR gate are depicted. Here, the subscripts X, Y, Z are indicating the correlation of the applied voltages, currents or state variables to the devices X, Y and Z, respectively. In step 1–3 the initialization process of the ORNOR gate is shown by writing the three inputs to the devices X (blue), Y (light green) and Z (yellow). In step 4 (red) the ORNOR operation is executed, which is followed by three verifying read-out steps (dark green). If a high current is detected the read-out value is a 1, whereas a low current is a 0. Row 1/2/3: voltage applied to the Bitline 2/Bitline 1/Bitline 0. Row 4: potential at the wordline. Row 5/6/7: current at Bitline 2/Bitline 1/Bitline 0. Row 8/9/10: state variable of device X/Y/Z. The scale is changed for small state variable values.

and the resistance and as output the state representation. In contrast, the MAGIC and IMPLY logic families use only the device resistance as inputs and output and are thus stateful logic families.

To advance beyond functional completeness toward a broader logical computation structure, recent work demonstrated the capability of memristive switches to implement adder circuits based on the non-stateful CRS approach[26,27], and the stateful MAGIC[32] and IMPLY logic approaches[23,33,34]. Experimentally, an adder based on the CRS logic has been demonstrated using bipolar memristive devices[35]. The functionality of a MAGIC adder has been shown with organic unipolar switching devices[36]. Next to the demonstration of the IMPLY logic in the proposing publication[17] additional publications presented experimental studies for this approaches[37,38]. All of these adders require a certain number of devices and steps to perform a certain operation, e.g. a 1-bit addition. As memristive switches do not offer an unlimited endurance, reducing the number of steps required for the targeted operation will aid in the advancement of memristive switches as a promising computing technology.

In this paper, we extend the functionality of the IMPLY logic by implementing stateful logic with three devices simultaneously. Three devices are shown to execute the function $X + \overline{(Y + Z)}$, which will be called the ORNOR gate. We show that this function can aid in reducing the number of steps and improving efficiency in memristive computing. This concept is validated using a physics-based simulation model, which has been fitted to experimental data. This modeling approach enables us to identify possible limitations of the logic approach. The analysis of these limitations paves the way for deducing device and circuit requirements.

Results

ORNOR gate. The proposed ORNOR gate can be regarded as an extension of the IMPLY gate proposed by Borghetti *et al.*[17]. In an IMPLY gate, two memristive switches, P and Q are connected via a common node over a resistance to ground. Two different voltages V_{Set} and V_{Cond} are applied to Q and P, where V_{Cond} is not high enough to set P, but V_{Set} can set Q in the specific cycle time. By applying these voltages at the same time, the potential at the shared connection is rising depending on the states of the devices. Thus, the voltage drop over Q can be lowered depending on the state of P, so that Q does not switch (see also Supplementary Information). The **ORNOR** gate, in contrast, uses three memristive switches X, Y, and Z as depicted in Fig. 1a. The common line connecting the memristive devices with the resistance R_G is referred to as wordline in the following. The memristive devices X, Y and Z are contacted via the bitlines 2, 1 and 0, respectively. The conditional voltage V_{Cond} is now applied to two devices (Y and Z) and V_{Set} is applied to the third device (X). Only the device with V_{Set} applied (X) can change states from the HRS to the LRS, but now, two memristive switches (Y and Z) determine the final state of X. The voltage at the V_{RG} node is near to V_{Cond} relative to ground when either device Y or Z is in the LRS state. In this case, the voltage applied across device X is V_{Set} - V_{Cond} and therefore does not switch the binary state of X to the LRS. In the case where both Y and Z are in the HRS, the effective voltage at V_{RG} is nearly GND, as there is just a tiny current flow through resistor R_G. In this scenario, the voltage across device X is equal to V_{Set}, which is sufficiently large to change the state of X from the HRS to the LRS. The truth table of this circuit for different inputs (X, Y, and Z) is shown in Fig. 1b. This table can be simplified to the function $X' = X + \overline{(Y + Z)}$, which will be referred to as the ORNOR gate, as the function is stated as X **OR** (Y **NOR** Z). The output of the function is the state of X after the voltages are applied, written as X'.

To validate the function of the ORNOR gate circuit simulations are performed. For the memristive elements, we used a physics-based simulation model, which is described in detail in the Methods section. The model was fitted to experimental data of a $Pt/Ta_2O_5/Ta$ VCM device (see Supplementary Information) and it fulfills the six fundamental criteria required to model VCM devices[39,40], among which the nonlinear switching kinetics is the most important one. It means that the device will switch upon application of a non-zero voltage in a finite time. It depends, however, on the voltage magnitude how fast the switching will occur. Due to the involved physics, the switching time is a highly nonlinear function of the applied voltage[41]. Since the fastest switching occurs at higher voltages and the target voltages of this application are in the higher voltage range, the fit was chosen to be more accurate in this range.

The simulations of the two critical cases as described below are shown in Fig. 1d. To perform the ORNOR operation, the memristive devices are first initialized to the designated inputs (X blue, Y green and Z yellow), starting in an HRS (steps 1–3). To this end, zero volts are applied to the wordline and the desired inputs to the bitlines. Then, the ORNOR operation (red) is performed in step 4 and afterwards verified by the read-out steps 5–7. The last three rows in Fig. 1d present the state variables of the devices X, Y and Z. Tracking the state variable allows us to observe small state changes, which are hard to detect in the read-out current. Note that the scale is changed for small state variable values.

The critical case 1 ($X = 0$, $Y = 0$ and $Z = 0$) is the only one in which device X switches. Thus, it determines the minimum cycle time t_C. During operation an unwanted state drift occurs. The devices Y and Z, to which V_{Cond} is applied, show a small state drift. This state drift is independent of the cycle time since it stops as soon as device X becomes sufficiently low resistive and in turn the potential at V_{RG} is high enough. Consequently, the voltage drop over devices Y and Z decreases.

The critical case 2 is $X = 0$, $Y = 0$ and $Z = 1$ and has the same behavior as the case $X = 0$, $Y = 1$ and $Z = 0$. Here, a cycle time dependent state drift of device X is present, since V_{RG} is not increased sufficiently by the low ohmic connection of device Z to prevent this drift.

The case $X = 0$, $Y = 1$ and $Z = 1$ is not critical, since the potential V_{RG} is even closer to V_{Cond} as both devices Y and Z are low ohmic. Thus, the voltage drop over device X decreases and the switching process slows down.

The observed state drift is a direct consequence of the nonlinear switching kinetics of VCM devices, which are included in our device model. Due to a smaller voltage drop over the devices (here in a range of 0.8 V) during the operation the switching process starts in unselected cells[42]. As the switching process is slower in this regime according to the switching kinetics (Fig. 2b), only a small state drift is observed. To alleviate this problem a device with steeper kinetic could be chosen[43]. This effect imposes an additional circuit design constraint on the circuit parameter, i.e. the applied voltages, the timing and the series resistance. The used parameters listed in Fig. 1c were optimized to minimize the drifts and cycle time for performing the ORNOR function.

Computing system. In the proposed system architecture, arrays of memristive switches are dedicated to performing a specific function. The memristive switches have clock signals applied to their bitlines, as in Fig. 2a. These clocks encode the data for a specific function and drive the memristive switches to compute this function in a serial manner. These arrays are defined as function blocks. An example of one of these function blocks is shown on the left in Fig. 2a, where six memristive switches are set up to compose a full adder. This function block is one of many function blocks that ultimately comprise a computing system based on memristive switches for complex computations. Each of the function blocks repeatedly performs the defined function enabling pipelining[44]. One advantage to this system is that multiple function blocks can be driven by the same set of clocks, thereby performing parallel processing without much overhead in area. Data can be transferred from one function block to another function block for additional computation, or each function block could be reconfigured to perform a different function by changing the clock set. A master clock controls how the various function blocks are synchronized. As shown in Fig. 2a, there is a transistor that gates connections between the common node of one function block and the common node of another function block. Additionally, there are transistors that control the connection of the C0 and C1 memristive switches to this common node, which are specific to the full adder function. The additional transistors enable the transfer of data between function blocks, which use a common clock set for all full adder function blocks.

In this computing system architecture, a full adder circuit is realized. The N-bit full adder circuit proposed in this paper is optimized to exploit the ORNOR function, and requires $6 \bullet (N_{bit} + 1)$ memristive switches. Due to the doubling of the most significant bit to ensure a correct result for a two's complement addition, an extra full adder functional block (6 devices) is needed. The full adder circuit is realized here with a common transistor at the wordline (see Fig. 2b) instead of a resistor R_G as in the ORNOR gate (see Fig. 1a). This provides flexibility for using different functions on this block, as the conductance of the transistor can be tuned according to the performed functions (see also Supplementary Information). Figure 2c depicts a 2-bit adder circuit, which is composed of three full adder circuits with common transistor, as an example of a component of the 64-bit adder circuit. The 2-bit adder circuit includes parasitic line resistances R_P and parasitic capacitances C_P. Here, the parasitic capacitances C_p between the wordlines are not shown due to readability, but are considered later in the simulations.

To transfer the data from one functional block to another a COPY operation needs to be implemented. During this operation, the data transfer gate drive is set to a conductive state and thus it connects two functional blocks to transfer the data. By performing an IMP operation with two devices, one of each block, the data is transferred to the other block. To copy data of $C1_{WL1}$ to $C0_{WL2}$ in Fig. 2c, the voltage scheme highlighted in color needs to be applied. Since the applied voltages at the bitlines are always applied to all functional blocks, if they are sharing one

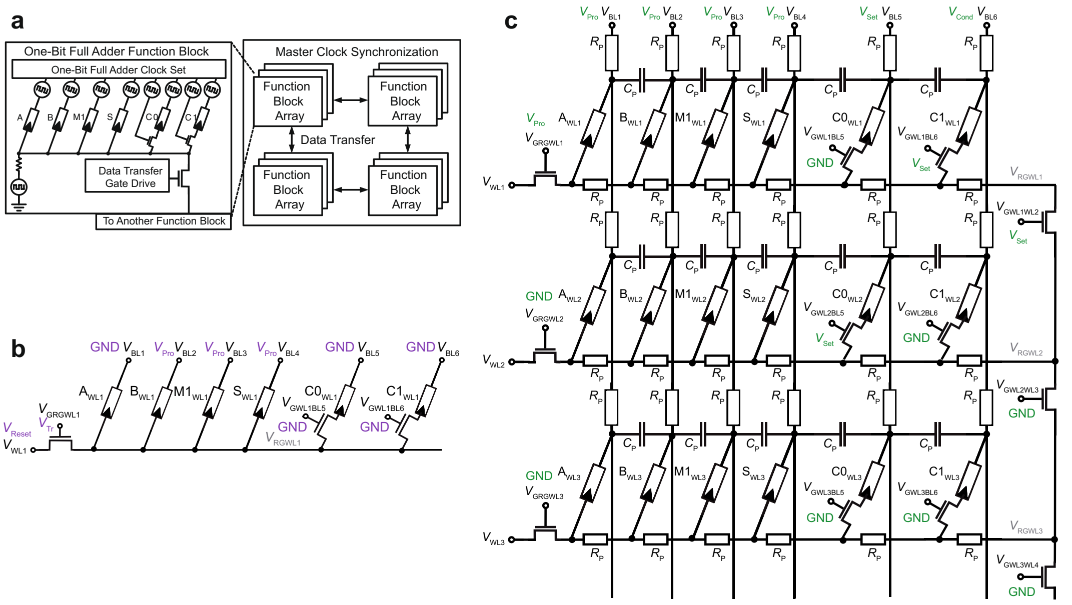


Figure 2. Computing architecture for a memristive computing system. (**a**) The computing system architecture utilizes a synchronized, interconnected array of function blocks to perform complex functions with improved efficiency. Each function block contains a set of memristive switches, which, in conjunction with a set of clock signals, perform a predefined function. For complex computations the data can be processed in multiple function blocks. The transfer between function blocks is done with the data transfer transistor. (**b**) An optimized full adder schematic with six total memristive switches. The signals, which need to be applied for a FALSE operation of device A_{WL1}, are displayed in violet. (**c**) Two-Bit adder circuit with parasitic elements. R_P and C_P are the parasitic resistors and the parasitic capacitances of the wordlines and bitlines. Here the parasitic capacitances between the wordlines are not depicted due to readability. The signals, which need to be applied for a COPY operation from device $C1_{WL1}$ to $C0_{WL2}$, are displayed in green.

clock set, selecting transistors must be added to the bitlines that are involved in the COPY operation. Otherwise, V_{Set} would be applied to two devices, which share the same bitline and the connected wordlines. In the same way, V_{Cond} is applied to two devices sharing another bitline. Thus, no COPY operation would be achieved. By adding the transistors to the circuit, only two devices on the active bitlines can be chosen to be connected to the wordlines by setting the selecting ($V_{GWL1BL6}$ and $V_{GWL2BL5}$) transistors to a conductive state, here by applying a high voltage V_{Tr} to the gates.

For implementing a functionally complete stateful logic system in this architecture, a FALSE operation is required, to switch the memristive devices to 0. When performing the function ORNOR(X, Y, Z) or IMP(P, Q), X and Q are the only memristive switches whose state can be changed. It can be observed from the truth tables that there is no set of inputs that causes the memristive switches X or Q (to which V_{Set} have been applied) to change from 1 to 0. Without the FALSE operation, all the memristive switches will eventually be changed to the 1 state, preventing further computation. Since the used memristive switches are bipolar switching devices, a voltage with the opposite polarity of the SET operation needs to be applied to reset the device to the HRS state. To this end, a RESET voltage V_{Reset} is applied to the wordline while the bitlines are set either to GND or to V_{Pro} in order to reset the device or keep its information. To reset device A in Fig. 2b, the voltage scheme illustrated in purple is applied to the respective terminals.

The full adder implementation proposed here takes 17 steps to perform a non-K2 one-bit addition, as described in Fig. 3. The step count assumes that setting the initial input values and the final readout are not part of the actual implementation of the function, which is consistent with previous research for a standardized analysis and comparison[23]. These steps are labelled as "–" in Fig. 3 and were required for the simulation. The functions are applied serially to perform the complete full adder function, where data for each step is encoded into the clocks that are applied to the memristive switches. The third column shows the operation of each step, where the memristive switches used in the operations are in parentheses. The outcome of the operation is shown in the column associated with each memristive switch. For example, in step 1, a FALSE operation is applied to the devices M1, S1, C0, and C1. Therefore, the output of each of these memristive switches is shown in their respective columns, where each device state is set to 0.

In general, A, B, and C0 are the memristive switches into which data is loaded, representing the standard A, B, and carry-in for a full adder. Before the execution of the function, data is initially loaded into A and B from another function block using a COPY function and a data transfer transistor. The carry-in is loaded into the C0 memristive switch in step 8. C1 contains the carry-out data of the function block array, and once the computation is complete, S contains the calculated sum. The M1 memristive switch is an additional supporting device. The additional transistors connected to the wordlines of C0 and C1 allow the carry-out of one stage to be shifted to the carry-in of the next stage in a multi-bit chain similar to how a shift register propagates data along a serial chain.

Step	Goal	Operation	Inputs		Functional	Sum	Carry-In	Carry-Out
			A	B	M1	S	C0	C1
0	Initial Input		1	1	X	X	X	X
1	Initialize/Reset M1, S, C0, C1	FALSE(M1, S,C0,C1)			0	0	0	0
-	Initialize A	SET(A)	A					
-	Initialize B	SET(B)		B				
2	Calculate A AND B	IMP (C0,A)					$\overline{A}$	
3		IMP (M1,B)			$\overline{B}$			
4		ORNOR(C1,C0,M1)						$A \cdot B$
5	Reset C0 and M1	FALSE(C0, M1)			0		0	
6	Calculate A XOR B	ORNOR(S, A, B)				$\overline{A+B}$		
7		ORNOR(M1, S, C1)			$A \oplus B$			
8	Input Carry-In	SET(C0)					$\overline{C_0}$	
9	Calculate Carry-Out	ORNOR(C1, C0, S)						C_{i+1}
10	Propagate Carry-Out	COPY($C0_{Next}$, C1)						
11	Reset A, B, S, C1	FALSE(A, B, S, C1)	0	0		0		0
12	Calculate C_{in} NOR(A XOR B)	IMP (A, C0)	C_i					
13		IMP(C1, M1)						$\overline{A \oplus B}$
14		ORNOR(B, A, M1)		$\overline{C_i + (A \oplus B)}$				
15	Reset A, M1	FALSE(A, M1)	0		0			
16	Calculate SUM	ORNOR(M1, C0, C1)			$C_i + (A \oplus B)$			
17		ORNOR(S, B, M1)				S		
--	Determine S	READ (S)						

Figure 3. Clocking scheme for a full adder. Each line corresponds to a step in the full adder, with the exception of simulation requirements for loading and transferring data (marked by–). Each step provides the operation that is performed, including the memristive switches that are part of the computation. The result of the operation is shown for each memristive switch, and if there is no value in a cell, it is assumed that the memristive switch maintains its previous state. SET operations are yellow, FALSE operations are violet, COPY operations are light green, IMP operations are green and ORNOR operations are dark green.

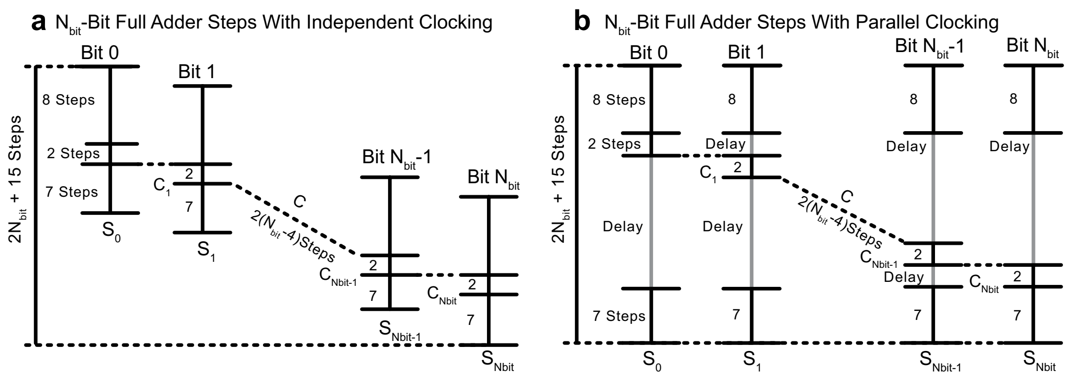


Figure 4. (**a**) Independent clocking and (**b**) parallel clocking scheme. (**a**) In the independent clocking scheme, each functional block has an independent set of clocks, which can be applied at any time to perform the function. In this way, each functional block is aligned so that there is no idle time, however, an increased number of clocks and drivers are required. Approach (**b**) uses the same set of clock signals to drive all memristive switches for the multi-bit addition. The first eight steps and last seven steps are driven across all function blocks simultaneously. This approach is able to compute the multi-bit full addition in the same number of steps while significantly reducing the number of clock signals required due to parallel control of multiple function blocks.

Although the memristive switches are defined as inputs or outputs to aid in the explanation of the full adder, every memristive switch can act both as an input or an output based on how they are used.

As shown in Fig. 3, with only two steps (9 and 10) of delay between each successive bit of a full-adder, there is increased parallel processing and therefore increased overall efficiency. Two different processing schemes can be realized. The addition is done bit by bit as illustrated in Fig. 4a. In this scheme, the individual bits are processed with an offset of two steps. This scheme, however, requires a unique clock for each memristive switch. In Fig. 4b, the use of the same clock set is coordinated such that the first eight steps and last seven steps are driven between all function blocks simultaneously. This reduces the overall number of drivers required for multi-bit addition.

Simulations. The proposed one-bit adder circuit with parallel clocking scheme is simulated using the model and the model parameters described in the Methods section, the circuit parameters given in Fig. 5a, and the clock scheme introduced in Fig. 3. Using the two's complement the addition of A = −1 and B = −1 is conducted. To secure a valid result the most significant bit is doubled. The applied voltages are depicted in Fig. 5b. Figure 5c shows the resulting terminal voltages, the change of the state variable of the individual bits during the computation and the currents on the wordlines WL1 and WL2. In the last step of the simulation, the result is read out by applying a small voltage to BL4 (connected to the S devices). For WL1 the detected wordline current is below 1 μA, resulting to a read 0, whereas the detected wordline current for WL2 is above 5 μA and thus interpreted as a 1. This means that the result of $(1)_2 + (1)_2 = (10)_2$, which verifies the functionality.

Next, the implementation and verification of a 64-Bit-Adder using the parallel clocking scheme is demonstrated. To ensure proper operation, the worst case in the matter of drift needs to be found first. Since the first eight steps and the last seven steps are executed in parallel, these steps do not differ from the one-bit adder. For multibit operations, however, the steps 9–10 are executed many times. In both steps, the lines BL5 and BL6 are active, but the select transistors only address the required two devices. As the active devices change each repetition in the carry propagation, no drift is expected in these devices. In step 9, BL4 is active, too. The devices connected to BL4 do not have a selector. Thus, a state drift is possible and this effect determines the maximum length of the addition. By means of simulation the worst case is found to be the operation B - A, with A = 0 and B = 0. Figure 6a shows the state variable transition of S_{WL65} for the worst-case operation. Here, a small drift is visible, but the calculated result is still valid. While the result is correct, the state variable of $M1_{WL65}$ is misbehaving as shown in Fig. 6b. Before the carry propagation phase begins, the device $M1_{WL65}$ should have switched to the LRS (N_{max}) but the state variable does not reach N_{max}. The reason can be found directly in the applied voltage as the potential on the bitline does not reach V_{Set} anymore and thus slows down the switching process. By using a shorter cycle time, the final value of the state will be even lower and eventually the carry would be interpreted as a 0, leading to a malfunction. To ensure proper operation, a cycle time of 250 ns is used here. Extending the cycle time even more to enable a complete switching of $M1_{WL65}$ to N_{max} would lead to state drift in other memristive devices.

Discussion

Previous approaches to calculating a 'stateful' full adder focused on solely IMP and FALSE operations, which resets the devices to the HRS[33], while others[23] extended this idea by utilizing the XOR operation in both serial and parallel optimized approaches. Table 1 lists the cycle steps and amount of memristive switches as stated in the original papers (if stated). As the adders in the referenced papers use varying methodologies, or count with or without input and output memristive switches, the given quantities have some ambiguity. This ambiguity becomes less important for a large number *N*. In this case, the proposed adder can reduce the needed steps by about 60%. Like in CMOS there is a tradeoff between area (here the amount of memristive switches) and time (here the steps). The amount of memristive switches can be reduced by reducing the parallelism of the algorithm. The number of steps and devices are also a hint towards the power consumption. If the array needs to be bigger (higher number of memristive switches), higher parasitic charging costs and a higher number of sneak paths need to be assumed. Moreover, if more steps are needed to achieve the functionality, more operations with higher voltages are executed on the array and thus the power consumption increases. More details on the energy consumptions are given in the Supplementary Information.

Whereas previous work on stateful memristor logic has proposed flexible functionality with enormous overhead control circuit costs[45], this proposed system architecture employs a parallel clocking scheme that trades functional flexibility for a drastic reduction in the overhead circuit footprint. There is a fundamental relationship between functional flexibility and overhead circuit cost, as functional flexibility requires additional control signals that must be generated by the control circuit. The total transistor count N_{TR} of the stateful memristor logic control circuit is given by[45]

$$N_{TR} = 28\log 2(S) + 2XS + 51X + 6S + TS - 2,$$

where *S* is the number of steps required performing a particular function, *X* is the number of memristive switches in the circuit, and *T* is the number of select transistors included for functional flexibility. As this overhead circuit cost is quite significant, the proposed system architecture minimizes the overhead circuit footprint by using each control signal to drive a large number of memristive switches and transistors in parallel (Fig. 4b). The decrease in the required number of steps resulting from the use of multi-input memristor logic, in concert with this parallel clocking of function blocks, therefore provide significant improvements to the efficiency of the control circuit and of the system as a whole.

The proposed computing system makes use of the ORNOR gate. The performance improvement relative to using only IMPLY gates is related to the fact that the ORNOR gate is a three input logic gate. The potential of using multi-input gates with three or more memristive devices have been described before[23–25,46], but the limitations of such circuits could not be addressed partly due to the lack of physics-based simulation models. To allow for multi-input gates, the resistor R_G needs to be chosen properly. It can be scaled with the number of inputs as proposed in literature[23,46]. In this case, the connection to ground becomes less resistive with each additional input, thus increasingly influencing the switching dynamics of the circuits. If the system is to enable functionality with a wide range in the number of inputs, additional complex periphery circuits must be added due to the scaling of R_G. A second option is to optimize R_G to enable proper functionality for the two- and three-input gates. Using the simulation model described in the Methods section, we investigated the functionality of multi-input gates for the two different choices of R_G.

Figure 7 depicts the simulation results of the slowest desired (red) and fastest erroneous (blue) switching times of gates with various number of inputs. For this study, gates of n-inputs were simulated, where the two-input gate

a

Simulation Parameter

V_{Set}	V_{Cond}	V_{Tr}	t_c	V_{Pro}
1.45 V	1.25 V	1.9 V	190 ns	0.435 V

b

c

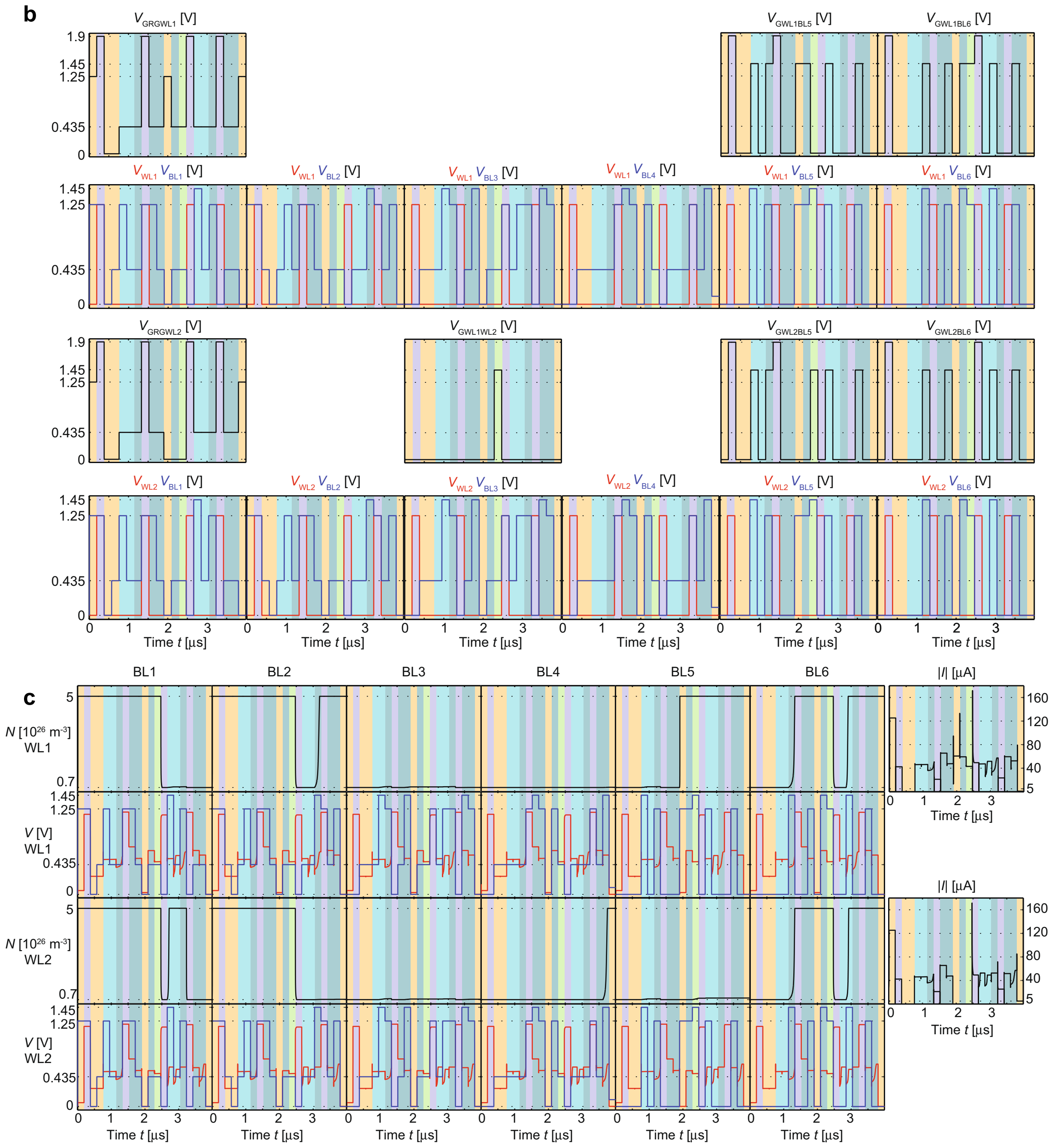


Figure 5. Simulation of a one-bit full adder. (**a**) Circuit parameter. (**b**) Applied voltages. SET processes are yellow, RESET operations are violet COPY operations are light green, IMP operations green and ORNOR operations dark green. Row 1/3: voltage applied to the load transistor and voltages applied to the selecting transistors of WL1/2. Additional in row 3: voltage transient applied to the data transfer gate, which can connect WL1 and WL2. Row 2/4: voltages applied to the bitlines (blue) and the wordlines (red) WL1/WL2. The circuit parameter voltages are marked for better readability. (**c**) State variables and terminal voltages of the one-bit adder. Row 1/3: transient of the state variable and wordline current of WL1/WL2. Row 2/4: wordline potential (red) of WL1/WL2 and voltages applied to the bitline (blue) of each device.

resembles an IMPLY gate and the three-input gate without scaling of R_G is the proposed ORNOR gate. Thus, an n-input gate includes n memristive switches. Here, n-1 of these switches are connected to V_{Cond} whereas always exactly one is connected to V_{Set}. If R_G is scaled, n-1 parallel R_Gs are assumed in the circuit. If R_G is not scaled,

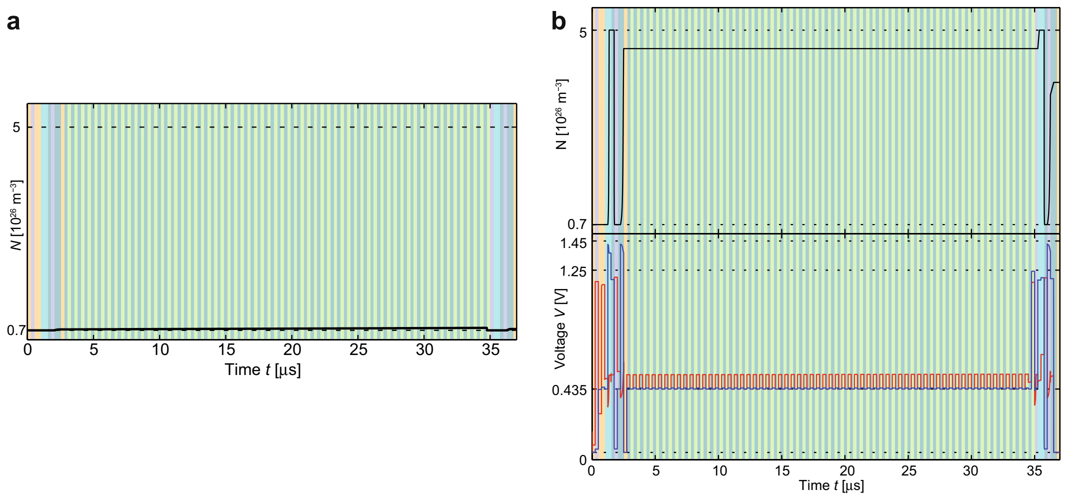


Figure 6. Simulation and comparison of the multibit adder. (**a**) State variable of S_{WL65} under the worst case conditions. The worst case is found to be A – B, where A is 0 and B is 0. The color scheme is adopted from Fig. 3. (**b**) State variable and applied voltage of $M1_{WL65}$ for the worst case simulation. The potential of the wordline is depicted in red and the potential of the connection point at the bitline is shown in blue.

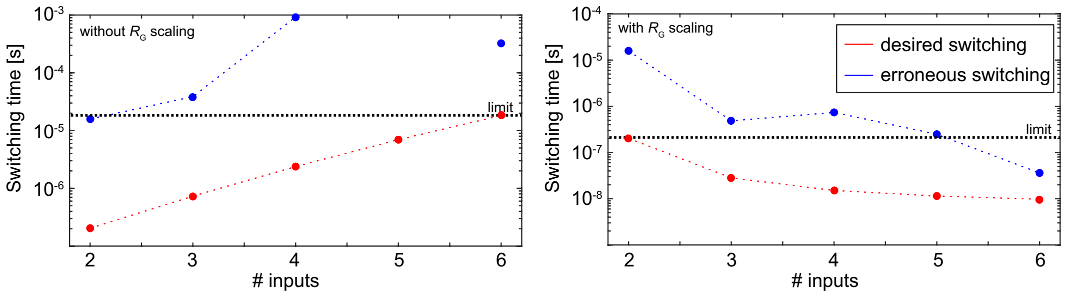


Figure 7. Simulation of the two multi-input approaches with (left) and without (right) R_G scaling. In both plots, two switching times of the device connected to V_{Set} are depicted. In the red case, all devices are in the HRS, and the proper desired functionality is that the device should switch (desired switching). The blue points depicts the switching time if one of the devices connected to V_{Cond} is in the LRS. In this case, the device should not switch (erroneous switching). The limit is chosen as the slowest of the desired switching processes; all erroneous switching processes must be slower than this limit.

Name	Memristive Switches	Steps	Name	Memristive Switches	Steps
Proposed	$6 (N_{bit} + 1)$	$2N_{bit} + 15$	MAGIC Conv. Area Optimized[32]	5	$15N_{bit}$
Lehtonen[33]	$3N_{bit} + 5$	$88N_{bit} + 48$	MAGIC Conv. Latency Optimized[32]	$11N_{bit} - 1$	$12N_{bit} + 1$
Kvatinsky serial[23]	$3N_{bit} + 3$	$29N_{bit}$	MAGIC Trans. I[32]	$22N_{bit} - 3$	$15N_{bit} + 1$
Kvatinsky parallel[23]	$9N_{bit}$	$5N_{bit} + 18$	MAGIC Trans. II[32]	$13N_{bit} - 3$	$10N_{bit} + 3$
Rohani[54]	$2N_{bit} + 3$	$22N_{bit}$			

Table 1. Comparison of different adder approaches.

no additional resistances are added to the circuit. The two worst cases for desired and erroneous switching are simulated for these circuits by applying V_{Set} and V_{Cond} as constant voltages. Then the switching time of the device connected to V_{Set} was analyzed, since it shows the fastest desired and the fastest erroneous switching. The slowest desired switching appears in the case that all devices are in the HRS. The fastest erroneous switching appears if only one device connected to V_{Cond} is in the LRS. The limit is set to the slowest desired switching process. All erroneous switching processes must be slower than this limit; hence, all desired switching process need to be completed before an erroneous switching process occurs. Therefore, in both cases the two-input device gates cannot be used in the same circuit with the same voltages and clock period as six-input circuits. This analysis also enables a rough estimation of how many operations can be conducted without additional refreshes. Depending on the minimum time interval between the slowest desired and fastest erroneous switching process, more or fewer sequential steps can operate on the same data without intermediate refresh cycles. This study also represents the stability of this logic approach against variability of the resistance states, since the connection to ground as well as the resistances to V_{Cond} are varied over multiple times. As it is depicted in Fig. 7, the scheme is still functioning for reasonable variations.

Figure 7 also depicts the strong influence of R_{G}, as the desired switching process without scaling of R_{G} becomes slower with a higher number of inputs, whereas this process gets faster with increasing number of inputs, if R_{G} is scaled. The results presented here are highly dependent on the device characteristic and the circuit parameters. The erroneous switching event is a consequence of the nonlinear switching kinetics of the memristive device. As the device will switch under non-zero voltage input in a finite amount of time, erroneous switching events cannot be avoided completely. Instead, the circuit parameters have to be chosen accordingly. These design constraints can be only deduced when using proper physics-based memristive device models as in this study. In this regard, the circuit parameters must be chosen in concert to ensure that the desired switching process is faster than the erroneous switching process. A desired switching speed can be chosen first, which enables determination of the minimum switching voltage with contemplation of the kinetic characteristic. The set voltage must be higher than this voltage, as there is also a voltage drop over R_{G}; but this V_{Set} must not be too high in order to prevent the device from switching faster in the erroneous switching cases. V_{Cond} must also be chosen carefully, as a too high value will cause drift in the devices connected to V_{Cond} in the desired switching case, while a too low value causes faster drift of the target device in the erroneous switching case. As can be seen in Fig. 7, a high R_{G} value causes a larger voltage drop and slower switching in all cases; a small R_{G} value speeds up all switching processes. The optimal circuit parameter can be found by maximizing the time window between the slowest desired and fastest erroneous switching processes.

Moreover, the nonlinear switching dynamics also have to be considered for the cells that do not take part actively in the functional operation. In arrays, a protection voltage V_{Pro} that is applied to the unselected devices is required[47]. Depending on the input cases and the states of the unselected devices, V_{Pro} has a huge impact on V_{RG} and so influences the speed of the operation and the unwanted state drift of the active device. Hence, V_{Pro} is also a parameter that needs to be optimized to achieve the best performance.

The 64-Bit-Adder simulation shows, that the desired state of $N_{\max}$ is not reached (Fig. 6b), but the resulting resistance of the device is nearly unchanged. Hence, the results are still valid. Here, the voltage levels V_{Set} and V_{Cond} as well as V_{Pro} are reduced compared to the optimal values, which are applied by drivers at the one end of the bl near to wl_0. Since parasitic elements of the wls and bls are included in the simulations, a voltage drop over the lines occurs, resulting in the reduction of applied voltages. Nevertheless, the logic scheme is still functioning, but it may be reasonable to find a better compromise of the circuit parameters for such a setup. Thus, the design is robust against moderate voltage deviations. Next to changing the circuit parameters, the lines could be widened to reduce the line resistance and thus the voltage drop. Moreover, the resistance levels of the memristive devices could be increased, thus less current is flowing over the bl and the line voltage drop is reduced. Due to the included parasitics, also sneak paths and programming disturbances are included in the simulation, but they do not show to have negative effects on the circuit in addition to the voltage drop.

Conclusions

Memristive switches enable a stateful beyond CMOS computing architecture. A novel extension to current computations with memristive switches is the three-input memristive switch logic gate, named the ORNOR function. A system architecture and clocking scheme have been proposed utilizing the ORNOR function, which enables the memristive switches to perform logic with fewer steps. In particular, a full adder was designed as an element of a multi-bit full adder function; the carry-in-to-carry-out delay was therefore minimized to optimize the overall number of steps required to perform the function. The solution shown here reduces the number of steps by up to 60%, providing a significant improvement in system efficiency. By using a physics-based simulation model, a couple of design constraints could be revealed. The major challenge is to choose the circuit parameters (voltages and cycle times) in a way that enables correct functionality. As memristive devices change their state under non-zero input in a finite time, devices that are not supposed to switch should see small voltage drops only for a limited amount of time. One consequence is that multi-input gates with a large difference in the number of inputs cannot be used with the same clocking scheme. The nonlinearity of the switching kinetics is not identical for all type of memristive devices. Thus, the circuit design parameters will differ when another type of memristive device is used.

Methods

Simulation model.

Since the physics of VCM devices is still not completely understood, finding an accurate model showing all aspects of memristive switches is an impossible task. There have been initial attempts to characterize the plethora of published ReRAM models and define needed features[39,40]: the most important one being the nonlinear switching kinetics. One model for VCM devices fulfilling these criteria is published by Fleck *et al.*[42]. Here, this model is adapted to model a Pt/TaO$_x$/Ta device. It is based on the movement of oxygen vacancies within a filamentary region and a concurrent resistance change. The corresponding equivalent circuit model is shown in the Supplementary Information (Fig. 1). In this model, the conductive oxygen-deficient filament is divided into two regions, the disc (light green) and the plug. The plug region is defined as the part of the filament located at the Ta electrode and has a constant high concentration of oxygen vacancies. The disc is located at the Pt electrode and has an oxygen vacancy concentration N_{disc} that varies between a minimum concentration $N_{\mathrm{disc,min}}$ and a maximum concentration $N_{\mathrm{disc,max}}$. As the resistance is altered by the change of N_{disc}, this quantity is considered as the state variable. The change of N_{disc} is defined as follows:

$$\frac{\mathrm{d}N_{\mathrm{disc}}}{\mathrm{d}t} = -\frac{1}{z_{\mathrm{Vo}}eAl_{\mathrm{disc}}}I_{\mathrm{ion}}, \tag{1}$$

where z_{Vo} is the charge of the oxygen vacancies, e is the elementary charge, A is the cross-section of the conducting filament, l_{disc} is the length of the disc region, and I_{ion} is the ionic current of oxygen vacancies defined at the interface between plug and disc. The ionic conduction can be modeled by a hopping conduction described by the Mott-Gurney law[48]:

$$I_{\mathrm{ion}} = 2Az_{\mathrm{Vo}}ec_{\mathrm{Vo}}a\nu_0 A\exp\left(\frac{-\Delta W_{\mathrm{A}}}{k_{\mathrm{B}}T}\right)\sinh\left(\frac{az_{\mathrm{Vo}}e}{2k_{\mathrm{B}}T}E\right) \tag{2}$$

Here, a is the hopping distance, ν_0 is the attempt frequency, ΔW_{A} is the barrier height for the ion hopping process, k_{B} is the Boltzmann constant, T is the local temperature, E is the electric field, which is considered to be the driving force for the hopping process, and c_{Vo} is the mean concentration of plug and disc. This means c_{Vo} is modeled by

$$c_{\mathrm{Vo}} = \frac{N_{\mathrm{plug}} + N_{\mathrm{disc}}}{2} \tag{3}$$

with N_{plug} being the oxygen vacancy concentration of the plug region. The electric field E is given by

$$E = \frac{V_{\mathrm{Schottky}} + V_{\mathrm{disc}} + V_{\mathrm{plug}}}{l_{\mathrm{cell}}}, \tag{4}$$

where V_{Schottky} is the voltage drop over the Schottky contact, V_{disc} is the voltage drop over the disc region, V_{plug} is the voltage drop over the plug, and l_{cell} is the oxide layer thickness. For positive voltages, only the thermionic emission is considered as a conduction mechanism of the Schottky contact and is modeled as[49]:

$$I_{\mathrm{Schottky},V>0\mathrm{V}} = AA^{*}T^{2}\exp\left(\frac{-e\phi_{\mathrm{Bn}}}{k_{\mathrm{B}}T}\right)\left(\exp\left(\frac{eV_{\mathrm{Schottky}}}{k_{\mathrm{B}}T}\right) - 1\right). \tag{5}$$

Here, A^* is the Richardson constant and $e\phi_{\mathrm{Bn}}$ is the effective Schottky barrier height, which is lowered by the image-force lowering effect. The effective Schottky barrier height can be described as follows[49]:

$$e\phi_{\mathrm{Bn}} = e\phi_{\mathrm{Bn0}} - e\sqrt[4]{\frac{e^{3}z_{\mathrm{Vo}}N_{\mathrm{disc}}\left(\phi_{\mathrm{Bn0}} - \phi_{\mathrm{n}} - V_{\mathrm{Schottky}}\right)}{8\pi^{2}\varepsilon_{\phi_B}^{3}}} \tag{6}$$

with $e\phi_{\mathrm{Bn0}}$ being the Schottky barrier height under zero bias, $e\phi_{\mathrm{n}}$ being the difference between the conduction band and the Fermi level, and $\varepsilon_{\phi\mathrm{n}}$ being the effective permittivity in the area of influence of the image-force lowering effect. The Schottky barrier transport mechanism is considered the thermionic-field emission for negative voltages. Thus, the current is calculated by[49]:

$$I_{\mathrm{Schottky},V<0\ \mathrm{V}} = -AA^{*}\frac{T_{\mathrm{S}}}{k_{\mathrm{B}}}\sqrt{\pi E_{00}e\left(-V_{\mathrm{Schottky}} + \frac{\phi_{\mathrm{Bn}}}{\cosh^{2}\left(\frac{E_{00}}{k_{\mathrm{B}}T}\right)}\right)}\exp\left(\frac{-e\phi_{\mathrm{Bn}}}{E_0}\right)\left(\exp\left(\frac{-eV_{\mathrm{Schottky}}}{\varepsilon'}\right) - 1\right) \tag{7}$$

with the parameters E_{00}, E_0 and ε':

$$E_{00} = \frac{eh}{4\pi}\sqrt{\frac{z_{\mathrm{Vo}}N_{\mathrm{disc}}}{m^{*}\varepsilon}}, \tag{8}$$

$$E_0 = E_{00}\coth\left(\frac{E_{00}}{k_{\mathrm{B}}T}\right) \tag{9}$$

and

$$\varepsilon' = \frac{E_{00}}{(E_{00}/k_{\mathrm{B}}T) - \tanh(E_{00}/k_{\mathrm{B}})}. \tag{10}$$

The contact resistance R_{contact} and the plug resistance R_{plug} are considered as constant resistances in the model. The contact resistance is supposed to result from the electrodes and the TaO_x/Ta interface, whereas the plug resistance depends on the geometry of the filament and the assumed oxygen vacancy concentration in the plug region N_{plug} and is set to:

Symbol		Value	Symbol		Value
l_{cell}	Insulator layer thickness	5 nm	R_{contact}	Contact resistance	1 kΩ
l_{disc}	Length of disc region	3 nm	$R_{\mathrm{th,eff}}$	Effective thermal resistance	$20.2\bullet10^{6}$ KW^{-1}
A	Filament area	140 nm^2	N_{plug}	Concentration of oxygen vacancies in the plug region	$5\bullet10^{26}$ m^{-3}
$A*$	Richardson constant	$11\bullet10^{5}$ AK^{-2} m^{-2}	$N_{\mathrm{disc,\,max}}$	Maximum concentration of oxygen vacancies in the disc region	$5\bullet10^{26}$ m^{-3}
ε	Permittivity	$21.5\bullet\varepsilon_0$	$N_{\mathrm{disc,min}}$	Minimum concentration of oxygen vacancies in the disc region	$0.7\bullet10^{26}$ m^{-3}
$\varepsilon_{\phi\mathrm{B}}$	Permittivity in the Schottky area	$11.6\bullet\varepsilon_0$	a	Hopping distance	0.5 nm
z_{Vo}	Charge of oxygen vacancies	2	T_0	Ambient temperature	293 K
$e\phi_{\mathrm{Bn0}}$	Schottky barrier height	0.36 eV	ΔW_{A}	Oxygen vacancy activation energy	0.855 eV
$e\phi_{\mathrm{n}}$	Difference between conduction band and Fermi level	0.1 eV	μ_{n}	Mobility of electrons	$13\bullet10^{-6}$ m^2 (Vs)$^{-1}$
ν_0	Attempt frequency	$1\bullet10^{13}$ Hz			

Table 2. Simulation parameter at $T = T_0$.

$$R_{\mathrm{plug}} = \frac{l_{\mathrm{plug}}}{ez_{\mathrm{Vo}}N_{\mathrm{plug}}\mu_{\mathrm{n}}A} \tag{11}$$

where μ_{n} is the mobility of the electrons and l_{plug} is the length of the plug region. In contrast to the plug and contact resistance, the disc resistance changes with the state variable N_{disc} and is calculated as follows:

$$R_{\mathrm{disc}} = \frac{l_{\mathrm{disc}}}{ez_{\mathrm{Vo}}N_{\mathrm{disc}}\mu_{\mathrm{n}}A}. \tag{12}$$

Filamentary VCM devices show strong nonlinear kinetics (cmp. Fig. 2b in the supplement). This feature can only be achieved if temperature acceleration is considered[41,50]. Thus, it is important to model the internal temperature:

$$T = V_{\mathrm{disc}}I_{\mathrm{disc}}\cdot R_{\mathrm{th,eff}} + T_0 \tag{13}$$

where $R_{\mathrm{th,eff}}$ is the effective thermal resistance of the disc region and T_0 is the ambient temperature.

Simulation parameters. For this work, the model is fitted to measured kinetic data of a TaO_x device for the region of applied voltages (0.5 V–1.3 V) (cmp. Fig. 2b)[51]. For small applied voltages the switching speed of the simulated device differs from the real device about some orders of magnitude. The applied voltages in this paper, however, are inside the fitted region. To estimate the values of the parasitic elements, Cu wires (bitlines/wordline) with a feature size of 40 nm and a height of 40 nm were considered, which are embedded in SiO_2 as the insulating material. Thus, for a line segment of 80 nm with a spacing of 40 nm and a height of 40 nm the coupling capacitance to the neighboring lines is calculated as $C_{\mathrm{P}} = 2.76\bullet10^{-18}$ F and the segment resistance is $R_{\mathrm{P}} = 0.86\ \Omega$. The transistors are modeled by a BSIM 4 model with the parameters of[52,53]. The remaining simulation parameters are listed in Table 2.

References

1. Frank, D. *et al.* Device scaling limits of Si MOSFETs and their application dependencies. *Proceedings of the IEEE* **89**, 259–288 (2001).
2. Williams, R. S. What's Next? [The end of Moore's law]. *Computing in Science & Engineering* **19**, 7–13 (2017).
3. ITRS, International Technology Roadmap for Semiconductors 2.0 - 2015 Edition (2015).
4. Wong, H.-S. P. & Salahuddin, S. Memory leads the way to better computing. *Nat. Nanotechnol.* **10**, 191–194 (2015).
5. Kautz, W. Cellular Logic-in-Memory Arrays. *IEEE Transactions on Computers C* **18**, 719–727 (1969).
6. Matsunaga, S. *et al.* Fabrication of a Nonvolatile Full Adder Based on Logic-in-Memory Architecture Using Magnetic Tunnel Junctions. *APEX* **1**, 091301 (2008).
7. Ferch, S., Linn, E., Waser, R. & Menzel, S. Simulation and Comparison of two Sequential Logic-in-Memory Approaches Using a Dynamic Electrochemical Metallization Cell Model. *Microelectron. J.* **45**, 1416–1428 (2014).
8. Gaillardon, P.-E. *et al.* The Programmable Logic-in-Memory (PLiM) Computer. *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, March 14–18, 2016*, 1–6 (2016).
9. Linn, E. Memristive Nano-Crossbar Arrays Enabling Novel Computing Paradigms. *IEEE International Symposium on Circuits and Systems (ISCAS), Melbourne VIC, Australia, 1–5 June 2014*, 2596–2599 (2014).
10. Le Gallo, M. *et al.* Mixed-precision in-memory computing. *Nature Electronics* **1**, 246–253 (2018).
11. Zidan, M. A., Strachan, J. P. & Lu, W. D. The future of electronics based on memristive systems. *Nat. Electron.* **1**, 22–29 (2018).
12. Ielmini, D. & Wong, H. P. In-memory computing with resistive switching devices. *Nature Electronics* **1**, 333–343 (2018).
13. Yang, J. J., Strukov, D. B. & Stewart, D. R. Memristive Devices for Computing. *Nat. Nanotechnol.* **8**, 13–24 (2013).
14. Yang, J. J. *et al.* Memristive switching mechanism for metal/oxide/metal nanodevices. *Nat. Nanotechnol.* **3**, 429–433 (2008).
15. Waser, R. & Aono, M. Nanoionics-based resistive switching memories. *Nat. Mater.* **6**, 833–840 (2007).
16. Waser, R., Dittmann, R., Staikov, G. & Szot, K. Redox-Based Resistive Switching Memories - Nanoionic Mechanisms, Prospects, and Challenges. *Adv. Mater.* **21**, 2632–2663 (2009).
17. Borghetti, J. *et al.* 'Memristive' switches enable 'stateful' logic operations via material implication. *Nature* **464**, 873–876 (2010).

18. Xie, L., Du Nguyen, H. A., Taouil, M., Hamdioui, S. & Bertels, K. Fast Boolean Logic Mapped on Memristor Crossbar. *IEEE ICCD* **2015**, 335–342 (2015).
19. Snider, G. Computing with hysteretic resistor crossbars. *Appl. Phys. A - Mater. Sci. Process.* **A80**, 1165–1172 (2005).
20. Kvatinsky, S. *et al*. MRL - Memristor Ratioed Logic. *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA), Turin, Italy*, *29–31 Aug. 2012*, 1–6 (2012).
21. Vourkas, I. & Sirakoulis, G. Ch. A Novel Design and Modeling Paradigm for Memristor-Based Crossbar Circuits. *IEEE Trans. Nanotechnol.* **11**, 1151–1159 (2012).
22. Shin, S., Kim, K. & Kang, S. M. Memristive XOR for resistive multiplier. *Electron. Lett.* **48**, 78–79 (2012).
23. Kvatinsky, S., Friedman, E. G., Kolodny, A. & Weiser, U. C. Memristor-Based Material Implication (IMPLY) Logic: Design Principles and Methodologies. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **22**, 2054–2066 (2014).
24. Balatti, S., Ambrogio, S. & Ielmini, D. Normally-off Logic Based on Resistive Switches-Part II: Logic Circuits. *IEEE Trans. Electron Devices* **62**, 1839–1847 (2015).
25. Marranghello, F. S., Callegaro, V., Martins, M. G. A., Reis, A. I. & Ribas, R. P. Factored Forms for Memristive Material Implication Stateful Logic. *IEEE J. Emerging Sel. Top. Circuits Syst.* **5**, 267–278 (2015).
26. Siemon, A., Menzel, S., Waser, R. & Linn, E. A Complementary Resistive Switch-based Crossbar Array Adder. *IEEE J. Emerging Sel. Top. Circuits Syst.* **5**, 64–74 (2015).
27. Siemon, A., Menzel, S., Chattopadhyay, A., Waser, R. & Linn, E. In-Memory Adder Functionality in 1S1R Arrays. *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, *Lisbon*, *Portugal*, *24–27 May 2015*, 1338–1341 (2015).
28. Adam, G. C., Hoskins, B. D., Prezioso, M. & Strukov, D. B. Optimized stateful material implication logic for threedimensional data manipulation. *Nano Research*, 1–10 (2016).
29. Rosezin, R., Linn, E., Kügeler, C., Bruchhaus, R. & Waser, R. Crossbar Logic Using Bipolar and Complementary Resistive Switches. *IEEE Electron Device Lett.* **32**, 710–712 (2011).
30. Linn, E., Rosezin, R., Tappertzhofen, S., Böttger, U. & Waser, R. Beyond von Neumann-logic operations in passive crossbar arrays alongside memory operations. *Nanotechnology* **23**, 305205 (2012).
31. Kvatinsky, S. *et al*. MAGIC-Memristor-Aided Logic. *IEEE Trans. Circuits Syst. II-Express Briefs* **61**, 895–899 (2014).
32. Talati, N., Gupta, S., Mane, P. & Kvatinsky, S. Logic Design Within Memristive Memories Using Memristor-Aided loGIC (MAGIC). *IEEE Trans. Nanotechnol.* **15**, 635–650 (2016).
33. Lehtonen, E. & Laiho, M. Stateful Implication Logic with Memristors. *2009 IEEE/ACM International Symposium on Nanoscale Architectures*, *San Francisco*, *CA*, *USA*, *July 30–31*, *2009*, 33–36 (2009).
34. Puglisi, F. M., Pacchioni, L., Zagni, N. & Pavan, P. Energy-Efficient Logic-in-Memory I-bit Full Adder Enabled by a Physics-Based RRAM Compact Model. *2018 48th European Solid-State Device Research Co* (2018).
35. Breuer, T. *et al*. A HfO2-Based Complementary Switching Crossbar Adder. *Adv. Electron. Mater.* **1**, 1500138 (2015).
36. Jang, B. C. *et al*. Zero-static-power nonvolatile logic-in-memory circuits for flexible electronics. *Nano Research* **10**, 2459–2470 (2017).
37. Cheng, L. *et al*. Reprogrammable logic in memristive crossbar for in-memory computing. *J. Phys. D Appl. Phys.* **50**, 505102/1-8 (2017).
38. Maestro-Izquierdo, M. *et al*. Experimental Time Evolution Study of theHfO2-Based IMPLY Gate Operation. *IEEE Trans. Electron Devices* **65**, 404–410 (2018).
39. Linn, E., Siemon, A., Waser, R. & Menzel, S. Applicability of Well-Established Memristive Models for Simulations of Resistive Switching Devices. *IEEE Transactions on Circuits and Systems - Part I: Regular Papers (TCAS-I)* **61**, 2402–2410 (2014).
40. Menzel, S., Siemon, A., Ascoli, A. & Tetzlaff, R. Requirements and Challenges for Modelling Redox-based Memristive Devices. *Proceedings of 2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, *27–30 May 2018*, *Florence*, *Italy* (2018).
41. Menzel, S., Salinga, M., Böttger, U. & Wimmer, M. Physics of the Switching Kinetics in Resistive Memories. *Adv. Funct. Mater.* **25**, 6306–6325 (2015).
42. Fleck, K. *et al*. Uniting Gradual and Abrupt SET Processes in Resistive Switching Oxides. *Phys. Rev. Applied* **6**, 064015 (2016).
43. Siemon, A., Wouters, D., Hamdioui, S. & Menzel, S. Memristive Device Modeling and Circuit Design Exploration for Computation-in-Memory. *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, *Sapporo*, *Japan*, *26-29 May*, *2019* (2019).
44. Kim, K., Shin, S. & Kang, S. M. Field Programmable Stateful Logic Array. IEEE Trans. *Comput-Aided Des. Integr. Circuits Sys* **30**, 1800–1813 (2011).
45. Hu, X. *et al*. Overhead Requirements for Stateful Memristor Logic. *TCAS I*, 1–11 (2018).
46. Shin, S., Kim, K. & Kang, S. M. Reconfigurable Stateful NOR Gate for Large-Scale Logic-Array Integrations. *IEEE Trans. Circuits Syst. II-Express Briefs* **58**, 442–446 (2011).
47. Zhu, X. *et al*. Performing Stateful Logic on Memristor Memory. *IEEE Transactions on Circuits and Systems Part II – Express Briefs* **60**, 682–686 (2013).
48. Mott, N. F. & Gurney, R. W. *Electronic Processes in Ionic Crystals* (Oxford at the Clarendon Press, 1950).
49. Sze, S. M. & Ng, K. K. *Physics of Semiconductor Devices* (Wiley, 2007).
50. Menzel, S. *et al*. Origin of the Ultra-nonlinear Switching Kinetics in Oxide-Based Resistive Switches. *Adv. Funct. Mater.* **21**, 4487–4492 (2011).
51. Havel, V. *et al*. Ultrafast Switching in Ta_2O_5*-based Resistive Memories*. *Silicon Nanoelectronics Worshop SNW 2016, Hawaii*, 82–83 (2016).
52. Zhao, W. & Cao, Y. New Generation of Predictive Technology Model for Sub-45 nm Early Design Exploration. *IEEE Trans. Electron Devices* **53**, 2816–2823 (2006).
53. Predictive Technology Model (PTM), Accessed on Oct. 05, [Online] Available, http://ptm.asu.edu/modelcard/2006/45nm_bulk.pm (2015)
54. Rohani, S. G. & TaheriNejad, N. An improved algorithm for IMPLY logic based memristive Full-adder. *2017 IEEE CCECE* (2017).

Acknowledgements
This work was supported in parts by the German Research Foundation (DFG) within the framework of SFB 917 Nanoswitches.

Author Contributions
A. Siemon performed the simulations, interpreted the data and wrote the manuscript. R. Drabinski co-performed the simulations. X. Hu, M. Schultis and A. Heittmann co-wrote the manuscript. E. Linn conceived the idea, initiated and supervised the research. R. Waser supervised the research. D. Querlioz initiated and supervised the research. S. Menzel co-wrote the manuscript, initiated and supervised the research. J. Friedmann conceived the idea, co-wrote the manuscript, initiated and supervised the research. All authors discussed the results and implications at all states and contributed to the improvement of the manuscript text.

Additional Information

Supplementary information accompanies this paper at https://doi.org/10.1038/s41598-019-51039-6.

Competing Interests: The authors declare no competing interests.