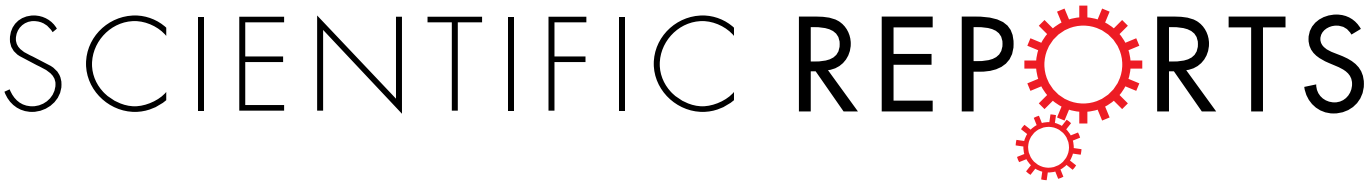


OPEN

Multi-valued and Fuzzy Logic Realization using TaOx Memristive Devices

Received: 20 September 2017
Accepted: 8 December 2017
Published online: 08 January 2018

Debjyoti Bhattacharjee[1], Wonjoo Kim[2], Anupam Chattopadhyay[1,3], Rainer Waser[2,4] & Vikas Rana[2]

Among emerging non-volatile storage technologies, redox-based resistive switching Random Access Memory (ReRAM) is a prominent one. The realization of Boolean logic functionalities using ReRAM adds an extra edge to this technology. Recently, 7-state ReRAM devices were used to realize ternary arithmetic circuits, which opens up the computing space beyond traditional binary values. In this manuscript, we report realization of multi-valued and fuzzy logic operators with a representative application using ReRAM devices. Multi-valued logic (MVL), such as Łukasiewicz logic generalizes Boolean logic by allowing more than two truth values. MVL also permits operations on fuzzy sets, where, in contrast to standard crisp logic, an element is permitted to have a degree of membership to a given set. Fuzzy operations generally model human reasoning better than Boolean logic operations, which is predominant in current computing technologies. When the available information for the modelling of a system is imprecise and incomplete, fuzzy logic provides an excellent framework for the system design. Practical applications of fuzzy logic include, industrial control systems, robotics, and in general, design of expert systems through knowledge-based reasoning. Our experimental results show, for the first time, that it is possible to model fuzzy logic natively using multi-state memristive devices.

Claude Shannon, in his landmark work[1], demonstrated that the two-valued logic system developed by George Boole[2], can be mimicked through operations of an electrical circuit. This resulted in widespread adoption of two-valued switching algebra or Boolean algebra. For every digital device in the modern world, Boolean algebra is used to perform the underlying computation. Boolean logic uses two truth values, *true* and *false*, even though in real life, we often require more than two truth values for describing an event. For example, we describe a day as 'sunny', 'partly clouded' or 'clouded', which means that the element weather cannot be discretely classified into the set 'sunny' or 'clouded' To capture such phenomena and their underlying logical process, *multi-valued* logic system was introduced.

In 300 BC, Aristotle proposed the *principle of non-contradiction* which ruled out simultaneous existence of two contradictory propositions[3]. This principle, also known as the *law of excluded middle*, is one of the classical laws of thought. In modern times, Jan Łukasiewicz introduced a third truth value, thereby first formally studying the field of multi-valued logic (MVL) in 1920[4]. In the following year, Emil Post published a system of functionally complete MVL algebra with additional truth degrees ($n \geq 2$), 'chained Post algebra'[5]. Later on, finite-valued Łukasiewicz logic was rigorously axiomatized and extended to arbitrarily many truth values[6,7]. Applications of MVL can be found in linguistics[8], circuit simulation and manufacturing testing of digital circuits[9].

An important offshoot of MVL that permits *inference under vagueness* and allows real-valued member elements is Fuzzy logic. The term *fuzzy logic* was introduced by Lotfi A. Zadeh in context of fuzzy set theory[10]. In contrast to the classical logic systems that adheres to a set of elements with *crisp* truth values, fuzzy logic operates on fuzzy sets. In a fuzzy set, elements of the set can have a degree of membership. Operators from a MVL, e.g., Łukasiewicz logic, can be applied to the fuzzy set, akin to how Boolean logic operators are applied to the crisp set. In fuzzy logic, a linguistic model is built from a set of IF–THEN rules which describe the control model. Mamdani Warren demonstrated that fuzzy logic could be used for developing operational automated control

[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore. [2]Peter Grünberg Institut 7, Forschungszentrum Jülich GmbH, 52425, Jülich, Germany. [3]School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, Singapore. [4]Institut für Werkstoffe der Elektrotechnik II, RWTH Aachen University, 52074, Aachen, Germany. Correspondence and requests for materials should be addressed to V.R. (email: v.rana@fz-juelich.de)

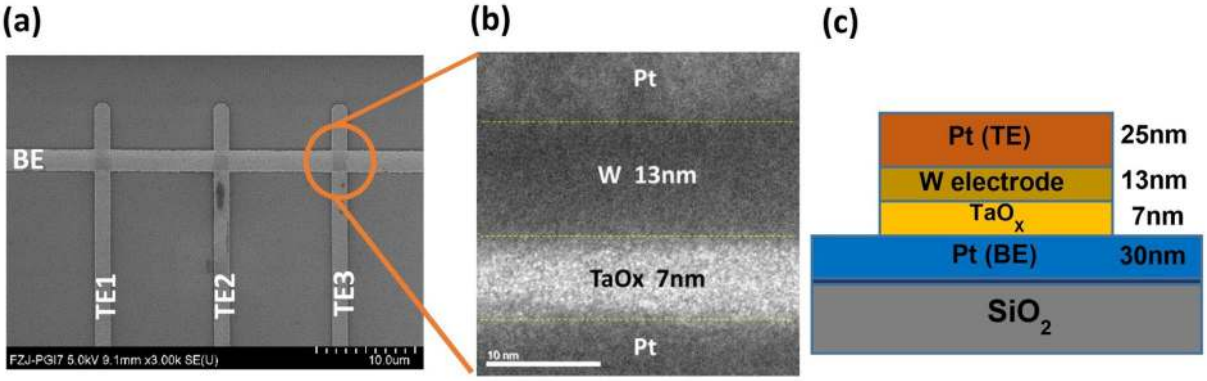


Figure 1. Resistive switching device structures. (**a**) Scanning electron microscopy image of 1×3 array, with $2\,\mu m \times 2\,\mu m$ device. (**b**) Tunnelling electron microscopy image of a single device cross-section with 7 *nm*-thick TaO_x switching layer and 13*nm*-thick tungsten ohmic electrode. (**c**) Schematic diagram of the single resistive device.

systems[11] and clinical practice decision support systems[12]. The most well known application of fuzzy logic based control system was deployed in Sendai Subway 1000 series subway trains in Japan for speed control[13]. The fuzzy controller based train had a higher relative smoothness of the starts and stops when compared to other trains, and was 10% more energy efficient than human-controlled accelerated trains. Further applications of fuzzy logic include expert systems[14], robotics[15] and diverse sub-domains of machine intelligence[16].

Despite these wide ranging applications of MVL, the present computing technology is heavily based on Boolean logic. There have been prior studies on porting MVL to digital and analog circuits with promising results. MVL has been demonstrated to improve energy-efficiency by reducing the switching activity of VLSI interconnects[17,18]. MVL based arithmetic circuits are simpler and more efficient over corresponding Boolean logic based implementations[19–21]. However, a limiting factor of MVL realization has been the inherent representation of information in binary format in semiconductor devices, thereby forcing a designer to switch between logic formats, which was clearly an inefficient solution. In this manuscript, we leverage the multi-state memristive devices which can inherently operate in the multi-valued domain. The multi-state memristive devices used in this experiment are Redox-based resisitve switches (ReRAMs), which are considered as one of the most promising emerging non-volatile memory technologies[22–24]. Implementation using passive crossbar configuration enables ultra-dense $4F^2$ integration. Recently, TaO_x based ReRAMs draw significant attention due to excellent performance in term of high endurance ($>10^{12}$)[25], long retention (10 years)[26], multi-level switching capability (3-bit)[27] and fast read/write speed of below 200 *ps*[28]. Besides the memory applications, ReRAM based passive crossbar arrays offer the implementation of memory-intensive computing paradigms, i.e. the logic operations are directly processed in the memory and arithmetic tasks. This merges the boundaries between memory and arithmetic logic units and eases the von-Neumann-bottleneck for computation[29]. Furthermore, memristive crossbar arrays can enable the multi-parallel search algorithms for pattern recognition tasks, widely required for neuromorphic applications[30].

The current paper reports the first implementation of Łukasiewicz logic using the ReRAM-based memristive devices. We do not impose any theoretical limit on the number of states for the memristive devices[31] and hence this work can be used for realizing any application that uses finite-valued Łukasiewicz logic family L_n. Before going to the implementation details, we briefly review the basics of Łukasiewicz logic. Formally, a finitely-valued Łukasiewicz logic family L_n can be defined over the following truth values.

$$L_n = \left\{0, \frac{1}{n-1}, \frac{2}{n-1}, \ \ldots, \ \frac{n-2}{n-1}, \ 1\right\} \tag{1}$$

The designated truth value is 1. Implication (IMP$\rightarrow$) and negation (NEG$\neg$) are the two operators used in Łukasiewicz logic, defined through the following functions.

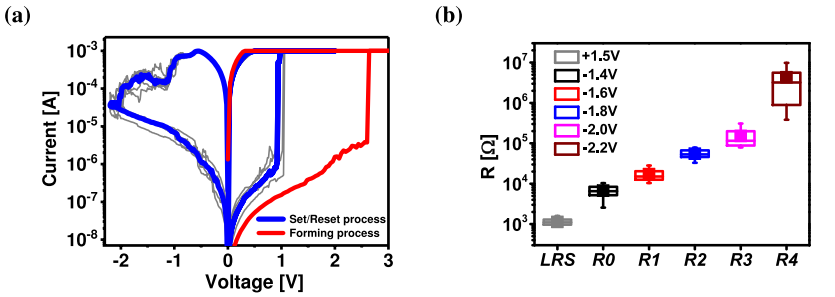
$$\neg u = 1 - u \tag{2}$$

$$u \rightarrow v = min\{1, 1 - u + v\}\ u, v \in L_n \tag{3}$$

In this paper, we also present a detailed case study to realize fuzzy logic control using Łukasiewicz logic and the implementation of a fuzzy logic controller using multi-state ReRAM crossbar arrays.

Results

Device Properties. In this work, $2\,\mu m \times 2\,\mu m$ *Pt*/*W*/TaO_x/*Pt* cross-point bipolar memristive devices arranged in word structure have been fabricated. Figure 1 shows the scanning electron microscope and transmission electron microscopy image of the devices used in this experiment. The ReRAM device stack of 25 *nm Pt*/13 *nm W*/7 *nm* TaO_x/30 *nm Pt* is depicted in Fig. 1(b). Figure 1(c) shows the schematic of a single device along with the details of the stacked layers. Figure 2(a) shows the typical *I*–*V* characteristics of the ReRAM device with



States	Δ [V]	R [Ω]
LRS	DC	1.1k
R0	-1.4	6.5k
R1	-1.6	15.0k
R2	-1.8	53.1k
R3	-2.0	115.5k
R4	-2.2	3.2M

Figure 2. Device characteristics. (**a**) Current–Voltage (I–V) characteristics and electroforming curve of TaO_x–based ReRAM device. (**b**) Resistance distribution based on median, obtained by pulse duration of 200 ns and amplitude in the range of $-1.4V$ to $-2.2V$ ($0.2V$ steps) enable highly accurate resistive state control. (**c**) Mean value of final resistance is estimated based on measurements of 4 devices for 10 cycles per state.

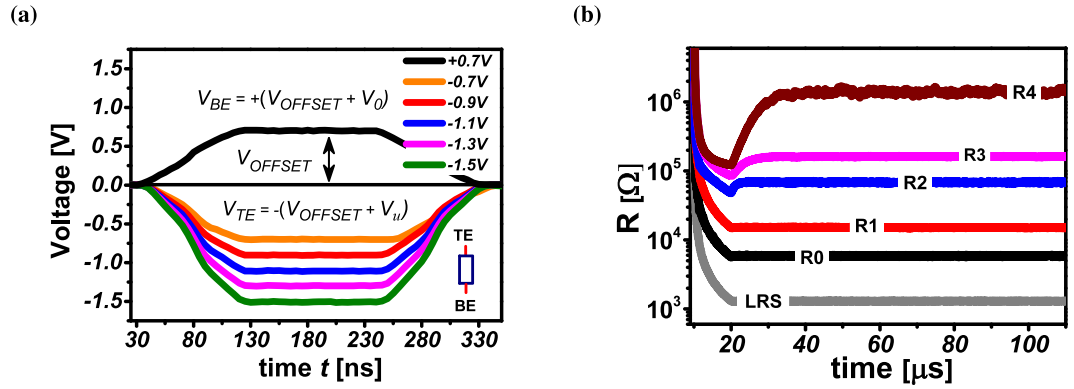


Figure 3. Primary logic operation. The logic operands u and v are applied to top (TE) and bottom (BE) electrode, respectively. Operand voltages V_u and V_v range from $0V$ to $0.8V$ in steps of $0.2V$. Equal stepping of operand voltages is enabled using an OFFSET voltage $V_{OFFSET} = 0.7V$. (**a**) Keeping the $V_{BE} = 0.7V$ constant, V_{TE} is varied from $-0.7V$ to $-1.5V$ in steps of $-0.2V$ i.e. $V_v = 0V$ and $V_u = 0, 0.2, \ldots, 0.8V$. (**b**) The corresponding resistance levels $R0, \ldots, R4$ states are programmed to the device. The actual resistive states are read by means of a 120 μs long voltage pulse with $0.1V$ (V_{READ}) amplitude.

set current compliance of 1.0 mA, along with the electroforming curve. After the electroforming process, the device was toggled to high resistance state by applying the reset voltage. The maximum applied voltage $|V_{stop}|$ during RESET process defines the final resistive state of the device. This feature is also used in pulse mode operation, and can thus be used in memory and logic operations for controlling the multi-level states. To enable highly reproducible RESET operation, we have always applied a DC SET operation before each pulsed RESET operation (200 ns). Note that a nanosecond pulsed SET operations are also feasible, but have not been applied in this work. Figure 2(b) shows a very tight resistance distribution of low resistance state (LRS) and six multi-level resistive states. This confirms the excellent switching properties of these devices. For the multi-level programming, we have split the total applied amplitude across the bottom electrode and the top electrode. A fixed positive amplitude ($+0.7V$) is assigned at the bottom electrode while a varying negative amplitude ($-0.7V$ to $-1.5V$) is applied to the top electrode. Under this configuration, the total applied amplitude across the cell varies from $-1.4V$ to $-2.2V$ for the given pulse width of 200 ns. For each V_{stop} voltage, the cell is toggled to a different high resistance state (HRS). The final resistance state has been read by the means of a 120 μs pulse with amplitude $V_{READ} = 0.1V$. Figure 2(c) shows the mean value of each resistive state from R0 to R4.

For Łukasiewicz logic family L_3 (or three-valued Łukasiewicz logic family), three logic values are used. The three resistance states (R0, R1, R2) of the multi-level device correspond to logic values (0, 0.5, 1) respectively in the Łukasiewicz logic family L_3. Additionally, the intermediate results in L_3 can be (1.5 or 2) which require two additional logic values. The resistance states (R3, R4) represent the intermediate logic values (1.5, 2) respectively. Corresponding to the logic values $u = \{0, 0.5, 1, 1.5, 2\}$, the operand voltage V_u is $\{0, 0.2, 0.4, 0.6, 0.8\}V$ respectively. The multi-valued operands say u and v can be applied as operand voltages to the top electrode (TE) and the bottom electrode (BE). Note that, u and v are always from the multi-valed logic set L_3 $\{0, 0.5, 1\}$. A predefined OFFSET voltage V_{OFFSET} is used for each pulse to allow equidistant voltage stepping. The voltage applied to the TE is $V_{TE} = -(V_{OFFSET} + V_u)$. Depending on the operation being realized, the actual voltage applied to the BE is $V_{BE} = V_{OFFSET} \pm V_v$. The effective potential difference across the device is $V_{eff} = V_{TE} - V_{BE}$. If $V_{BE} = V_{OFFSET} + V_v$, the resulting resistance state of the device is R_{u+v}. Otherwise if $V_{BE} = V_{OFFSET} - V_v$, the resulting resistance state of the device is R_{u-v}. Fig. 3 demonstrates the multi-level operation of the device.

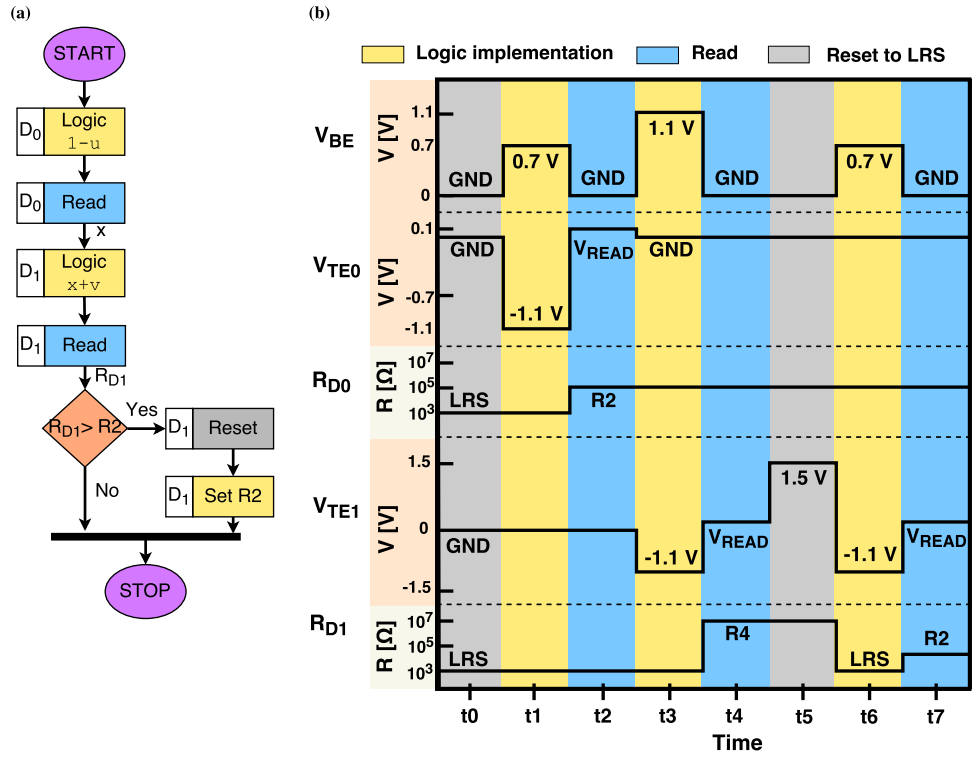


Figure 4. Implication implementation. (**a**) Flowchart for the implication ($u \rightarrow v = min(1, 1-u+v)$) computation. First, the negation of operand u, i.e. $1-u$, is computed. Next, the logic operation is conducted to compute $1-u+v$ in device D_1. The device state R is read out from the D_1. To compute $min(1, 1+u-v)$, we check whether $R > R2$ or not. If $R > R2$, then the device D_1 is reset to the LRS and set to resistive state $R2$. (**b**) Implication computation $u \rightarrow v$ for $u = 0$ and $v = 1$. V_{BE} is the voltage sequence applied to the BE of the devices whereas V_{TE0} and V_{TE1} is the voltage applied to the TE of device D_0 and D_1 respectively. The transition of resistive state for device D_0 and D_1 is shown as R_{D0} and R_{D1} respectively. After the computation, the final resistive state of device D_1 is the result of $u \rightarrow v$, which in this case is equal to $R2$, corresponding to logic 1.

Proposed Łukasiewicz Logic Implementation. The developed implementation strategy for realization of the Łukasiewicz logic operates and stores the multi-valued results directly in the ReRAM devices. The computed results are available as the resistive states of the device and can be read out. Before any operation, the devices are initialized to LRS. The realization of the NEG¬ and IMP→ operator for L_3 is explained below.

NEG¬ operator. The negation operator works on a single operand. For computing $\neg u$, a constant voltage $-(V_{OFFSET} + V_1)$ is applied to the TE while the $V_{OFFSET} - V_u$ is applied to the BE. The negated operand is stored in the ReRAM device as a corresponding resistive state.

IMP→ operator. The implication operator works on two operands. The flowchart for performing the implication operation $u \rightarrow v$ is shown in Fig. 4(a) and the steps are described below.

Step 1: In the beginning, all the devices are initialized to the LRS.
Step 2: To compute $\neg u$ in device D_0, $-(V_{OFFSET} + V_1)$ is applied to the TE and $V_{OFFSET} - V_u$ to the BE.
Step 3: The resistive state of the device D_0 is read out by means of $V_{READ} = 0.1V$.
Step 4: In the second device D_1, the negated sum of offset voltage V_{OFFSET} and the voltage corresponding to read out value $V_{\neg u}$ is applied to the TE i.e $V_{TE} = -(V_{OFFSET} + V_{\neg u})$ whereas at the BE, $V_{BE} = V_{OFFSET} + V_v$ is applied.
Step 5: In the following step, resistive state R of the D_1 is read out. If the resistive state $R < R2$, then the operation is complete. Otherwise, device D_1 is toggled to the LRS and the set operation is used to change the device state to $R2$ i.e., $V_{TE} = -1.1V$ and $V_{BE} = 0.7V$ is applied.

Figure 4(b) demonstrates the computation of $0 \rightarrow 1$ using the proposed method in terms of applied operation voltages and corresponding states. The overall operation requires seven steps. Initially, the device D_0 and D_1 are in LRS state, shown as R_{D0} and R_{D1} respectively. In cycle $t1$, $1-u$ is computed by applying $0.7V$ to the BE and $1.1-V$ to the TE of device D_0, shown as V_{BE} and V_{TE0} respectively. In cycle $t2$, the resistive state of D_0 is read out, which is $R2$. In the next cycle, $1-u+v$ is computed by applying $1.1V$ and -1.1 to the BE and TE of device D_1. In cycle $t4$, the current resistive state(R_{D1}) of device D_1 is read out. Since $R_{D1} = R4 > R2$, device D_1 is reset to the LRS and

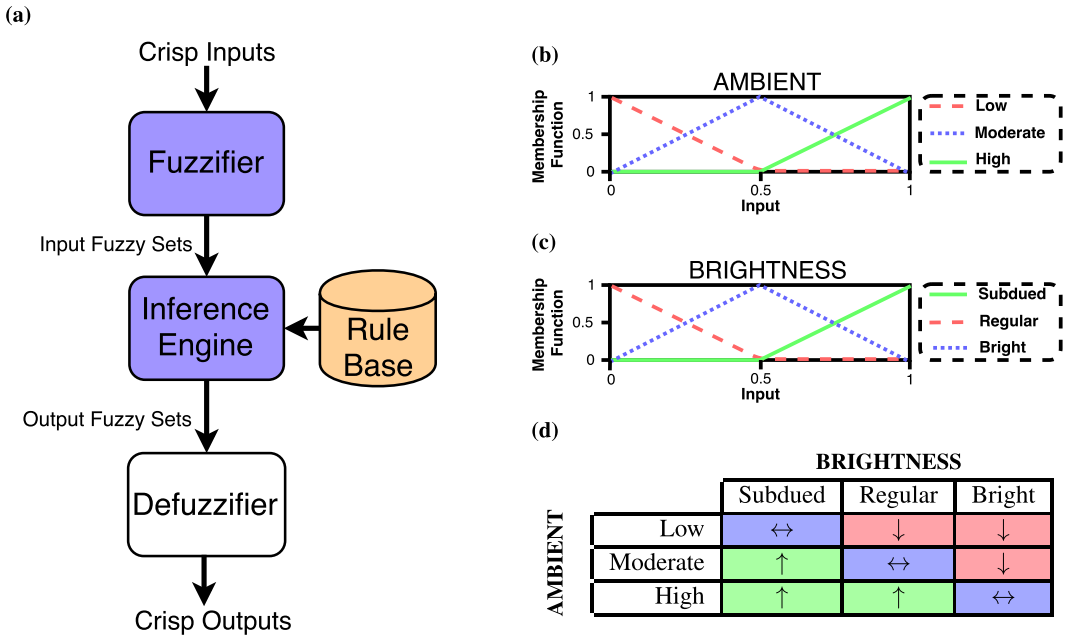


Figure 5. Fuzzy logic control. (**a**) Processing blocks used for fuzzy logic control are shown. The violet coloured blocks are implemented using the multi-state ReRAM devices. The rule base is stored as control/instruction steps. (**b**) Membership functions for variable *AMBIENT*. (**c**) Membership functions for variable *BRIGHTNESS*. (**d**) Rule table for fuzzy brightness controller.

then set to *R*2 in cycles *t*5 and *t*6. The computation of $0 \rightarrow 1$ is complete. To verify the correctness of computation, we read out the state of the device D_2 in the last cycle *t*7. The state R_{D2} is *R*2 (corresponding to logic 1) which is the correct result for $0 \rightarrow 1$.

Proof-of-concept. We demonstrate the realization of a fuzzy logic controller as a representative application of Łukasiewicz logic. A conventional fuzzy controller has three major steps, as shown in Fig. 5(a).

1. Fuzzify input variables using fuzzy membership functions. If the inputs to the system are analog, analog-to-digital converters would be used to convert the inputs to the corresponding values in Łukasiewicz logic. Then, these multi-valued inputs are *fuzzified* using the membership functions.
2. Execute all the fuzzy inference rules from the rule database to determine the fuzzy output functions.
3. Defuzzify the fuzzy output functions to get *crisp* output value i.e a single multi-valued output value.

To illustrate the working of a fuzzy logic control, we consider a fuzzy logic controller for regulating screen brightness. The screen brightness has to be regulated based on the ambient light *AMBIENT* and screen brightness *BRIGHTNESS*. Each of the variables can be represented using three gradations.

- $AMBIENT \in$ {Low (L), Moderate (M), High (H)}
- $BRIGHTNESS \in$ {Subdued (S), Regular (R), Bright (B)}

We use the fuzzy membership functions shown in Fig. 5(b) and (c) to fuzzify the input variables *AMBIENT* and *BRIGHTNESS* respectively. The fuzzy membership functions can be expressed in terms of Łukasiewicz operators[32,33]. Therefore, we can realize the "Fuzzifier" block of a fuzzy control system (shown in Fig. 5(a)) using Łukasiewicz logic operations only. Let us consider the inverted notch membership function $f_{(LOW,AMBIENT)}$ for the grade *LOW* of variable *AMBIENT*. $f_{(LOW,AMBIENT)}$. It can be expressed as $(\neg v \rightarrow v) \rightarrow 0$, where v is the input. Similarly, the flipped notch membership function $f_{(SUBDUED,BRIGHTNESS)}$ for the grade *SUBDUED* of variable *BRIGHTNESS* can be written as $(v \rightarrow \neg v) \rightarrow 0$. These membership functions can be simplified and expressed in terms of $min(u, v)$ and $\neg u$ $(1 - u)$ operation as shown below.

$$f_{(LOW,AMBIENT)} = (\neg v \rightarrow v) \rightarrow 0 = 1 - min(1, 2v) \tag{4}$$

$$f_{(SUBDUED,BRIGHTNESS)} = (v \rightarrow \neg v) \rightarrow 0 = 1 - min(1, 1 - v + 1 - v) \tag{5}$$

The detailed procedure to simplify the functions is provided in Supplementary Discussion S1. The series of steps required to realize the inverted notch and flipped notch membership function using multi-state ReRAM is presented in Fig. 6(a) and (b) respectively. We experimentally verified the correctness of computation. Figure 7 shows the experimental results when the input variable *AMBIENT* is 0 and also for value 1 for input *BRIGHTNESS*. In

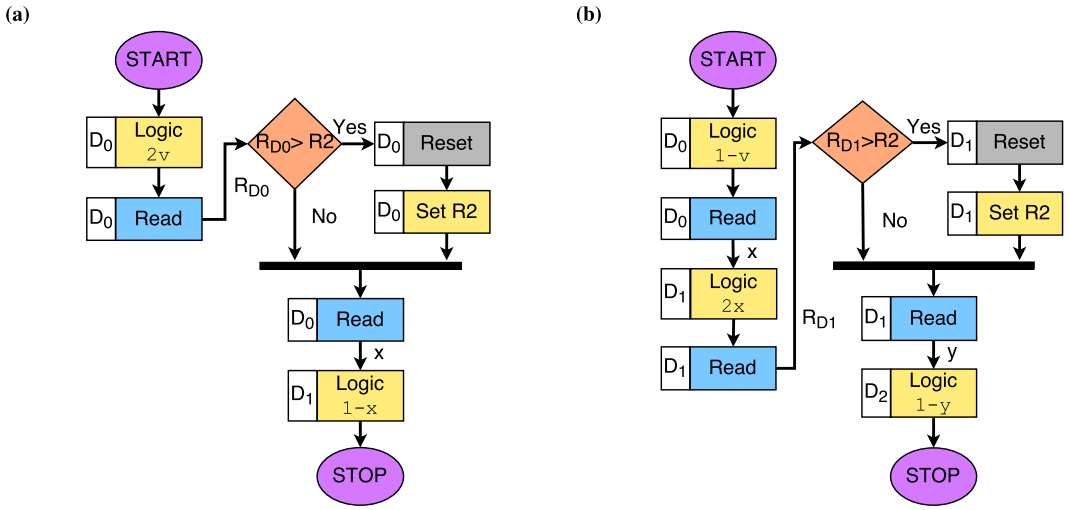


Figure 6. Realization of two fuzzy membership functions. (**a**) Steps to compute inverted notch[$(\neg v \rightarrow v) \rightarrow 0$]. (**b**) Steps to compute flipped notch[$(v \rightarrow \neg v) \rightarrow 0$].

Supplementary Discussion S2, we have shown the experimental validation of membership function for other values of input *AMBIENT*.

The fuzzy inference engine determines the *ACTION* to be taken based on fuzzified inputs and be stated as a set of rules. The *ACTION* can be increase brightness (↑), decrease brightness (↓) or no action (↔). A rule for example can the following–if *AMBIENT* is Low and *BRIGHTNESS* is Subdued, then *ACTION* is no action (↔). Figure 5(d) presents multiple such control rules to determine the *ACTION* represented compactly as a rule table. To evaluate the output of a fuzzy rule, a suitable T-norm function is used. We use Łukasiewicz T-norm–$max(0, u+v-1)$ for evaluation of the fuzzy controller rules. The Łukasiewicz T-norm can be expressed in terms of *neg* and *min* functions as $\neg min(1, (1-u)+(1-v))$. Figure 8(a) show the sequence of steps to realize Łukasiewicz T-norm using the ReRAM devices. As a representative example, Fig. 9 shows the state transitions of the ReRAM devices during computation of T-norm for inputs $u=1$ and $v=1$.

Once all the rules have been evaluated, the outputs of these is combined using a suitable T-conorm. The T-conorm should be the dual of the T-norm used for rule evaluation. Therefore, we use Łukasiewicz T-conorm which is $min(1, a+b)$, for combining the output of the rules. Figure 8(b) shows the steps for computation of the Łukasiewicz T-conorm using the multi-state memristive devices. T-conorm evaluation for inputs $u=1$ and $v=0.5$ is given in Supplementary Fig. S3. To obtain *crisp* output, the combined fuzzy output in the end has to be defuzzied. The defuzzifier block shown in Fig. 5(a) computes *crisp* output using an appropriate defuzzification method[34]. Note that the defuzzifier block has not been implemented in the presented prototype, which can be realized using conventional methods.

Discussion

Knowledge-based system is capable to reason with judgmental, imprecise, and qualitative knowledge as well as with formal knowledge of established theories. The design of such systems is an important challenge in the realm of Artificial Intelligence (AI). The incompleteness and uncertainty associated with the knowledge-base in is handled through fuzzy logic. Fuzzy logic allows linguistic variable[35] to be assigned inexact or partial truth values for modeling logical reasoning. In this work, we have shown the realization of fuzzy logic control in terms of three-valued Łukasiewicz logic operands. We have demonstrated the feasibility of implementation of three-valued Łukasiewicz logic L_3 using the multi-states memristive devices.

Our demonstrated method can be scaled up for arbitrary n-valued Łukasiewicz logic L_3, $n \geq 3$, depending on the number of resistive states available. For the realization of L_n, the memristive device should support at least $2n$ states. From the perspective of area, the implementation of a higher-valued logic system does not increase the area per device since it is dependent on the number of resistive states. However with increase in number of resistive states, the peripheral circuitry has to be more robust.todo.

Regarding the representation of numbers, it is well understood that for higher radix, the number of literals reduce in logarithmic order in comparison to lower-radix. For example, the efficiency of a n-valued representation of a truth-value N compared to its corresponding Boolean representation is equal to $\frac{log_n(N)+1}{log_2(N)+1}$. Implementation of a given fuzzy system in Boolean logic requires the treatment of every member with varied degree in a separate set and performing Boolean logic operations on those sets. Therefore, the computation steps do also increase in logarithmic proportion when using the Boolean logic in comparison to the fuzzy logic. Traditionally, Mamdani-type fuzzy systems use *min* and *max* functions for evaluation of fuzzy rules and combining the output of the rules[11]. *min* and *max* functions can be expressed as in terms of Łukasiewicz logic operators:

$$min(u, v) = (u \rightarrow v) \rightarrow v \tag{6}$$

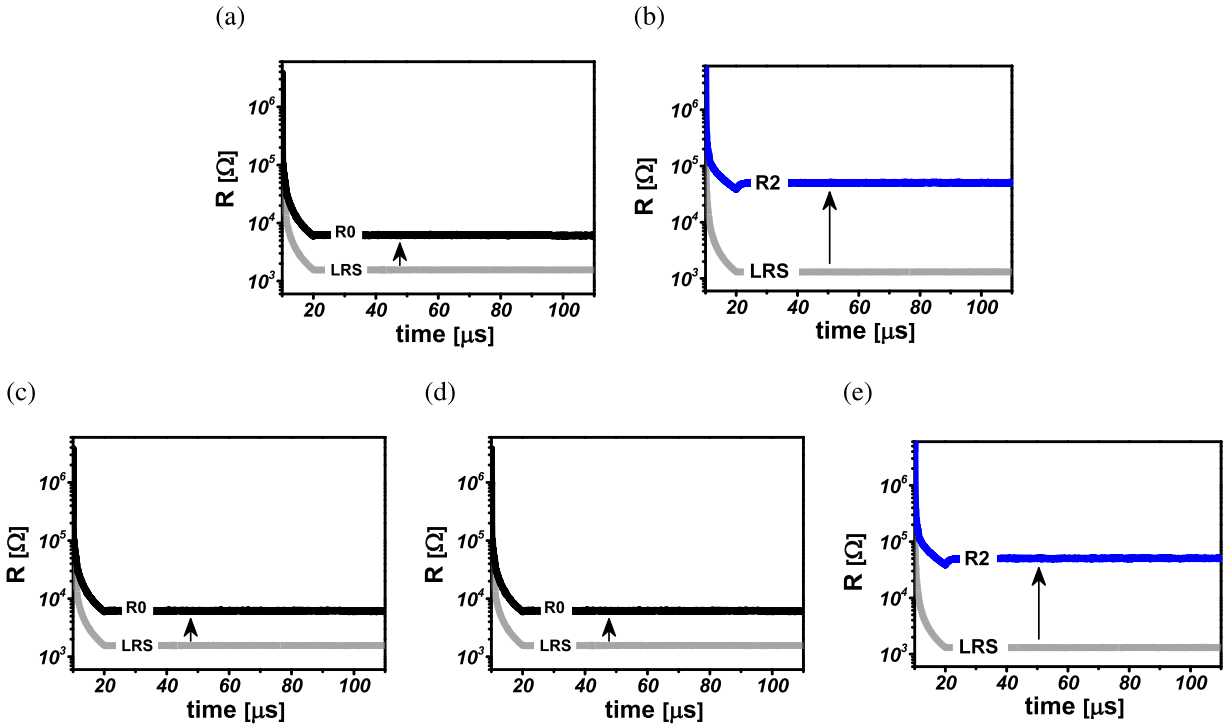


Figure 7. Membership function realization using multi-state ReRAM devices. (**a**,**b**) State transition of device D_0 and D_1 for realization of inverted notch membership function for $v = 0$. (**c**–**e**) State transition of devices D_0, D_1 and D_2 for realization of flipped notch membership function for $v = 1$.

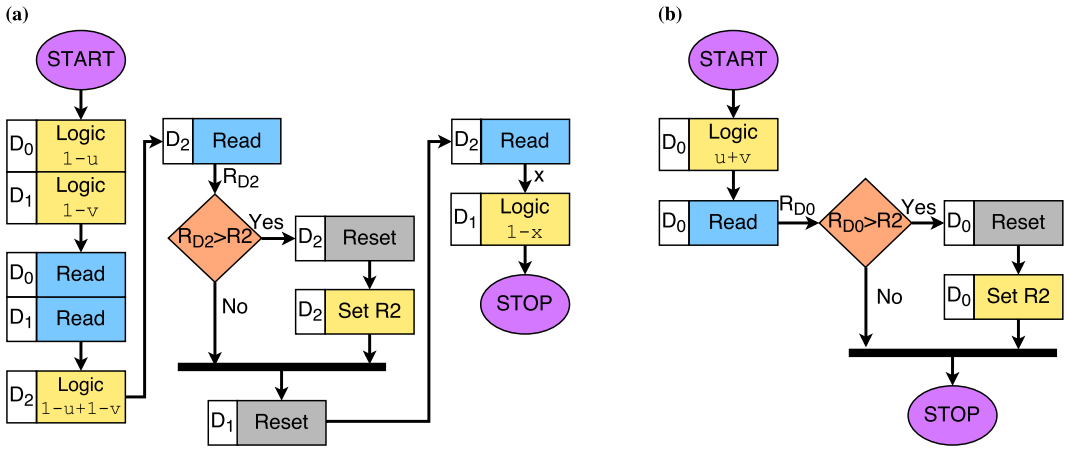


Figure 8. Fuzzy rule evaluation. (**a**) Computation of Łukasiewicz T-norm. (**b**) Computation of Łukasiewicz T-conorm.

$$max(u, v) = \neg(min(\neg u, \neg v)) \tag{7}$$

Therefore, it is possible to use the multi-valued Łukasiewicz logic for realization of Mamdani type fuzzy systems as well. In general, Łukasiewicz logic is capable of dealing with a wide range of approximate reasoning paradigms, since it can express evaluation function of multi-valued logic classes described in terms of $+$, $-$, *min* and *max*[35–37].

In past, Łukasiewicz logic arrays has been demonstrated for realization of fuzzy inference engines and expert systems[38–40] with CMOS-based circuitry. However, such realizations did not have any multi-valued storage devices for storing the intermediate results thereby requiring costly conversions to-and-from binary representation. In this work, we have experimentally realized a working fuzzy system by using Łukasiewicz logic, that does not use any intermediate binary/Boolean representation. Although implementation of fuzzy logic gates have been reported in the DNA computing paradigm[41], this is the first experimentally reported work on multi-valued

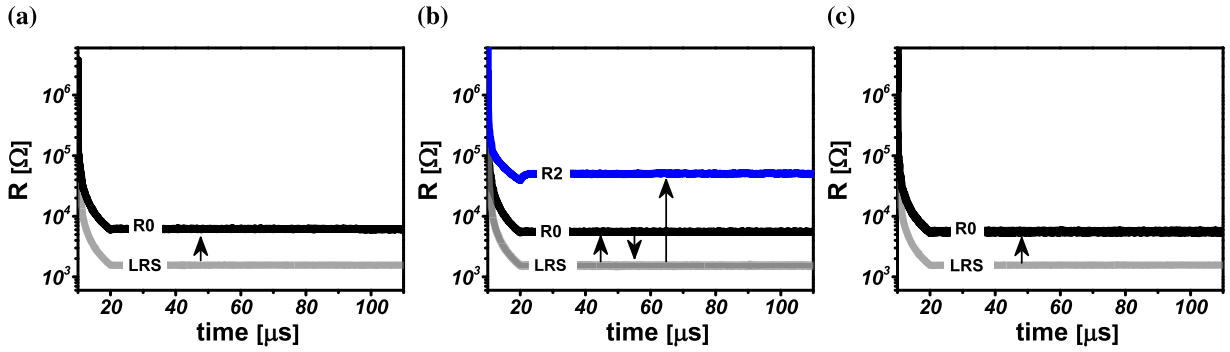


Figure 9. Łukasiewicz T-norm computation for $u = 1$ and $v = 1$. State transitions for (**a**) D_0 (**b**) D_1 and (**c**) D_2 ReRAM device respectively.

logic operators as well as a demonstrative application of that in fuzzy inference engine using memristive devices. Recently, an implementation of Boolean minimum and maximum gate has been demonstrated using memristive devices[42] with their application restricted to the implementation of sorting networks.

Here, we have successfully demonstrated Łukasiewicz logic operation on $2\,\mu m \times 2\,\mu m$ ReRAM devices. However, these devices are fully compatible to $4F^2$ configuration in crossbar array in conjunction with a selector device and can be scaled down to $5\,nm$[43,44]. The integration of the selector device would prevent the problem of sneak paths in the crossbar array. Ultra-dense large-scale multi-state ReRAM crossbars can be controlled by peripheral control circuitry, as shown in Supplementary Fig. S4. The approach of implementing Łukasiewicz logic operation within the resistive memory device using the available multi-level states is a highly attractive option for future hybrid CMOS/ReRAM chips for enhancing its present functionality. Each multi-valued operation requires a constant number of steps, 1 step for negation and 7 steps for implication (depicted in Fig. 4), to be realized, irrespective of the value of n in an n-valued logic system. For Boolean realization of the implication and negation operators, the number of steps would increase with the value of n[29,45,46]. Furthermore, parallel operations across multiple devices that share the same wordline, can be enabled by carefully packing operations that have the same input, similar to the strategy proposed by Bhattacharjee *et al.*[47]. In contrast, to leverage such parallelism, the Boolean circuits corresponding to the implication and negation operations need to be replicated. Multi-level ReRAM devices reduces the complexity of state representation and thus, brings fundamental benefits across arithmetic and logical primitives. This capability has far-reaching implications in modern Internet-of-Things (IoT) systems, which promotes local computing due to the bandwidth scarcity. By having multi-valued and fuzzy logic primitives at the device level, efficient processing-in-memory can be undertaken for application domains like public key cryptography, error correcting codes, industrial control and security. Note that, the energy-efficiency can be further boosted by having short-pulse (sub-*ns*) operations.

Conclusion

Fuzzy set allows its elements with certain degrees of membership, in contrast to a crisp set. Operators on the fuzzy set can realistically model real-life applications in, for example, industrial control, linguistics, decision variables and bio-informatics, and therefore, have grown in usage over last half century. The logic operations on the fuzzy set is performed through fuzzy inference system. In this manuscript, we demonstrated a practical fuzzy inference system, by realizing the fuzzy logic operations using the multi-state TaO_x devices. Further, each fuzzy operation is mapped to a series of the multi-valued logic primitives, namely, Łukasiewicz logic. We showed a practical fuzzy inference system through a limited number of logical steps and 1×3 memristive crossbar array. The multi-state TaO_x devices enable computation entirely using multi-valued elements for the operations, without need for any intermediate representations. Therefore, these devices provide a natural platform to undertake multi-valued logic and thus, fuzzy inference operations. We believe that these results can greatly benefit scientific community and provide a direction to move forward in the field of fuzzy logic.

Methods

Device Fabrication. Cross-point based Ta_2O_5 ReRAM was fabricated on thermally grown SiO_2 samples. In our design, each device shares a common bottom electrode (BE). The BE was patterned in $30\,nm$-thick platinum (Pt) layers, grown by the sputtering process. After patterning the BE, switching layer of $7\,nm$-thick TaO_x, $13\,nm$-thick tungsten (W), and $25\,nm$-thick platinum (Pt) were sequentially deposited by the sputtering process. The TaO_x layer was grown with reactive sputtering process with 76.6% Argon and 23.3% Oxygen at partial pressure of $2.3 \times 10^{-2}\,mbar$. The W ohmic electrode, and the Pt were grown with DC sputtering method. For the top electrode (TE) patterning, photo-lithography and reactive ion etching steps were performed. These steps lead to the $Pt/W/TaO_x/Pt$ memristive device stack. Figure 1 shows the scanning electron microscopy (SEM) of 1×3 crossbar array with $2\,\mu m \times 2\,\mu m$ size cell with cross-sectional tunneling electron microscopy (TEM) image and its corresponding schematic diagram. More experimental details can be found in reference[48].

Measurement Set-up. The pristine state of the memristive devices was highly resistive ($G\Omega$) and therefore an electroforming process was required. This process was carried out by applying a positive DC voltage on the

TE for a given current compliance, while keeping the BE grounded. This turned the devices into low resistance state (LRS). Now, the memristive devices were sequentially switched to high resistance state (HRS) by the 'reset process' and low resistance state (LRS) by the 'set process'. In this experiment, the 'set' process is performed with DC voltage while the reset operations were performed with 200 *ns* pulse width and 120 μs long pulse width at $0.1V$ was used to read the respective resistance states. More measurement details can be traced in reference[48].

Data availability. No datasets were generated or analysed during the current study.

References

1. Shannon, C. E. A symbolic analysis of relay and switching circuits. *Electrical Engineering* **57**, 713–723 (1938).
2. Boole, G. The mathematical analysis of logic (*Philosophical Library*, 1847).
3. Aristotle. Complete Works of Aristotle, Volume 1: The Revised Oxford Translation (*Princeton University Press*, 2014).
4. Łukasiewicz, J. Selected Works (*Amsterdam: North-Holland Pub. Co.*, 1970).
5. Post, E. L. Introduction to a general theory of elementary propositions. *American journal of mathematics* **43**, 163–185 (1921).
6. Wajsberg, M. Axiomatization of the 3–valued sentential calculus. . *CR Soc. Sci. Lettr. Varsovie* **24**, 126–148 (1931).
7. Chang, C. C. A New Proof of the Completeness of the Łukasiewicz Axioms. *Transactions of the American Mathematical Society* **93**, 74–80, http://www.jstor.org/stable/1993423 (1959).
8. Novák, V. A formal theory of intermediate quantifiers. *Fuzzy Sets and Systems* **159**, 1229–1246 (2008).
9. Roth, J. P. Diagnosis of automata failures: A calculus and a method. *IBM journal of Research and Development* **10**, 278–291 (1966).
10. Zadeh, L. A. Fuzzy sets. *Information and control* **8**, 338–353 (1965).
11. Mamdani, E. H. Application of fuzzy algorithms for control of simple dynamic plant. In *Proceedings of the Institution of Electrical Engineers*, vol. 121, 1585–1588 (IET, 1974).
12. Warren, J., Beliakov, G. & Van Der Zwaag, B. Fuzzy logic in clinical practice decision support systems. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, 10–pp (IEEE, 2000).
13. Dutta, S. Fuzzy logic applications: Technological and strategic issues. *IEEE Transactions on Engineering Management* **40**, 237–254 (1993).
14. Yager, R. R. Expert systems using fuzzy logic. In *An introduction to fuzzy logic applications in intelligent systems*, 27–44 (Springer, 1992).
15. Moudgal, V. G., Kwong, W. A., Passino, K. M. & Yurkovich, S. Fuzzy learning control for a flexible-link robot. *IEEE Transactions on Fuzzy Systems* **3**, 199–210 (1995).
16. Zadeh, L. A. Soft computing and fuzzy logic. *IEEE software* **11**, 48–56 (1994).
17. Ozer, E., Sendag, R. & Gregg, D. Multiple-valued logic buses for reducing bus energy in low-power systems. *IEEE Proceedings-Computers and Digital Techniques* **153**, 270–282 (2006).
18. Dubrova, E. Multiple-valued logic in VLSI: challenges and opportunities. *In Proceedings of NORCHIP* **99**, 340–350 (1999).
19. Hayes, B. Computing science: Third base. *American scientist* **89**, 490–494 (2001).
20. Hurst, S. L. Multiple-valued logic its status and its future. *IEEE Transactions on Computers* **33**, 1160–1179 (1984).
21. Moaiyeri, M. H., Sedighiani, S., Sharifi, F. & Navi, K. Design and analysis of carbon nanotube FET based quaternary full adders. *Frontiers of Information Technology & Electronic Engineering* **17**, 1056–1066 (2016).
22. Yang, J. J., Strukov, D. B. & Stewart, D. R. Memristive devices for computing. *Nature nanotechnology* **8**, 13–24 (2013).
23. Waser, R., Dittmann, R., Staikov, G. & Szot, K. Redox-based resistive switching memories–nanoionic mechanisms, prospects, and challenges. *Advanced materials* **21**, 2632–2663 (2009).
24. Wilson, L. International technology roadmap for semiconductors (ITRS). *Semiconductor Industry Association* (2013).
25. Lee, M.-J. *et al.* A fast, high-endurance and scalable non-volatile memory device made from asymmetric Ta_2O_{5-x}/TaO_{2-x} bilayer structures. *Nature materials* **10**, 625–630 (2011).
26. Wei, Z. *et al.* Retention model for high-density ReRAM. In *Memory Workshop (IMW), 2012 4th IEEE International*, 1–4 (IEEE, 2012).
27. Kim, W., Menzel, S., Wouters, D., Waser, R. & Rana, V. 3-Bit Multilevel Switching by Deep Reset Phenomenon in $Pt/W/TaO_X/Pt$-ReRAM Devices. *IEEE Electron Device Letters* **37**, 564–567 (2016).
28. Torrezan, A. C., Strachan, J. P., Medeiros-Ribeiro, G. & Williams, R. S. Sub-nanosecond switching of a tantalum oxide memristor. *Nanotechnology* **22**, 485203 (2011).
29. 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).
30. Linn, E. Memristive nano-crossbar arrays enabling novel computing paradigms. In *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*, 2596–2599 (IEEE, 2014).
31. van de Burgt, Y. *et al.* A non-volatile organic electrochemical device as a low-voltage artificial synapse for neuromorphic computing. *Nature Materials* **16**, 414–418 (2017).
32. McNaughton, R. A theorem about infinite-valued sentential logic. *The Journal of Symbolic Logic* **16**, 1–13 (1951).
33. Mills, J. W. & Daffinger, C. A. Cmos vlsi lukasiewicz logic arrays. In *Application Specific Array Processors, 1990. Proceedings of the International Conference on*, 469–480 (IEEE, 1990).
34. Van Leekwijck, W. & Kerre, E. E. Defuzzification: criteria and classification. *Fuzzy sets and systems* **108**, 159–178 (1999).
35. Zadeh, L. A. Fuzzy logic and approximate reasoning. *Synthese* **30**, 407–428 (1975).
36. Gaines, B. Fuzzy reasoning and the logics of uncertainty. In *Proceedings of the sixth international symposium on Multiple-valued logic*, 179–188 (IEEE Computer Society Press, 1976).
37. Katz, M. Two systems of multi-valued logic for science. In *Proceedings of the 11th International Symposium on Multiple-valued Logic*, 175–182 (1981).
38. Yamakawa, T. & Miki, T. The current mode fuzzy logic integrated circuits fabricated by the standard CMOS process. *IEEE Transactions on Computers* **35**, 161–167 (1986).
39. Yamakawa, T. High-speed fuzzy controller hardware system: The mega-FIPS machine. *Information Sciences* **45**, 113–128 (1988).
40. Giles, R. A resolution logic for fuzzy reasoning. In *Proceedings of IEEE 17th International Symposium on Multiple-Valued Logic*, 60–67 (1985).
41. Zadegan, R. M., Jepsen, M. D. E., Hildebrandt, L. L., Birkedal, V. & Kjems, J. Construction of a Fuzzy and Boolean Logic Gates Based on DNA. *Small* **11**, 1811–1817, https://doi.org/10.1002/smll.201402755 (2015).
42. Breuer, T. *et al.* Realization of Minimum and Maximum Gate Function in Ta_2O_5-based Memristive Devices. *Scientific reports* **6** (2016).
43. Govoreanu, B. *et al.* $10 \times 10\,nm^2$ Hf/HfO_x crossbar resistive RAM with excellent performance, reliability and low-energy operation. In *Electron Devices Meeting (IEDM), 2011 IEEE International*, 31–6 (IEEE, 2011).
44. Li, K.-S. *et al.* Study of sub-5 nm RRAM, tunneling selector and selector less device. In *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, 385–388 (IEEE, 2015).
45. Borghetti, J. *et al.* 'Memristive' switches enable 'stateful' logic operations via material implication. *Nature* **464**, 873–876 (2010).

46. Kvatinsky, S. *et al*. Memristor-based material implication (imply) logic: Design principles and methodologies. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **22**, 2054–2066 (2014).
47. Bhattacharjee, D., Devadoss, R. & Chattopadhyay, A. ReVAMP: ReRAM based VLIW architecture for in-memory computing. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, 782–787, https://doi.org/10.23919/DATE.2017.7927095 (2017).
48. Kim, W. *et al*. Multistate memristive tantalum oxide devices for ternary arithmetic. *Scientific reports* **6** (2016).

Acknowledgements

This work was supported by the German Research Foundation (DFG) within the framework of SFB 917, Nanoswitches and the authors thank Dr Daesung Park and Mr Sebastian Zischke from Central Facility for Electron Microscopy (GFE), RWTH Aachen for the cross-sectional TEM images, and Anne Siemon from RWTH Aachen for critical review of the manuscript.

Author Contributions

Debjyoti Bhattacharjee designed the experiments, interpreted the data and wrote the manuscript. Wonjoo Kim prepared the devices, performed the measurements. Anupam Chattopadhyay conceived the idea, initiated and supervised the research and co-wrote the manuscript; Rainer Waser initiated and supervised the research; Vikas Rana conceived the idea, supervised the research and co-wrote the manuscript. All authors discussed the results and implications at all stages and contributed to the improvement of the manuscript text.

Additional Information

Supplementary information accompanies this paper at https://doi.org/10.1038/s41598-017-18329-3.

Competing Interests: The authors declare that they have no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.