Experimental Validation of Memristor-Aided Logic Using 1T1R TaOx RRAM Crossbar Array

Ankit Bende*§, Simranjeet Singh*¶§, Chandan Kumar Jha‡, Tim Kempen‖, Felix Cüppers‖,
Christopher Bengel‡‡, André Zambanini×, Dennis Nielinger×, Sachin Patkar¶,
Rolf Drechsler‡††, Rainer Waser*‖, Farhad Merchant†, Vikas Rana‖

*PGI-7, ‖PGI-10, & ×ZEA-2, Forschungszentrum Jülich GmbH, Germany, ‡‡IWE-2 RWTH Aachen, Germany,
¶Indian Institute of Technology Bombay, India, ‡University of Bremen, Germany,
††DFKI GmbH, Germany, †Newcastle University, UK

{a.bende, t.kempen, f.cueppers, a.zambanini, d.nielinger, v.rana, r.waser}@fz-juelich.de, bengel@iwe.rwth-aachen.de,
{simranjeet, patkar}@ee.iitb.ac.in, {chajha, drechsler}@uni-bremen.de, farhad.merchant@newcastle.ac.uk

Abstract—Memristor-aided logic (MAGIC) design style holds a high promise for realizing digital logic-in-memory functionality. The ability to implement a specific gate in a MAGIC design style hinges on the SET-to-RESET threshold ratio. The TaOx memristive devices exhibit distinct SET-to-RESET ratios, enabling the implementation of OR and NOT operations. As the adoption of the MAGIC design style gains momentum, it becomes crucial to understand the breakdown of energy consumption in the various phases of its operation. This paper presents experimental demonstrations of the OR and NOT gates on a 1T1R crossbar array. Additionally, it provides insights into the energy distribution for performing these operations at different stages. Through our experiments across different gates, we found that the energy consumption is dominated by initialization in the MAGIC design style. The energy split-up is 14.8%, 85%, and 0.2% for execution, initialization, and read operations, respectively.

Index Terms—MAGIC, RRAM, logic-in-memory, fabrication

I. INTRODUCTION

Memristive devices, such as resistive random access memory (RRAM), offer a solution to the von Neumann bottleneck by implementing operations within memory itself [1], [2]. One approach to implementing operations in memory is designing digital logic gates exploiting two distinct states - the high resistive state (HRS) and low resistive state (LRS) of an RRAM. These states are correspondingly mapped to logic "0" and logic "1" respectively. Depending on the input combination stored as a resistive state, the output memristor state can switch from one state to another, representing a logical output value. Several methods for achieving digital logic-in-memory (LiM) have been suggested in the literature. Various stateful and non-stateful logic techniques have been presented in the literature such as IMPLY [3], FELIX [4], majority logic [5], and memristor-aided logic (MAGIC) [6]. Amongst all the techniques, MAGIC stands out as a popular choice because it stores the output in the form of the memristor's state itself, representing stateful logic.

Experimental validation of MAGIC gates has recently been achieved using fabricated valence change memory (VCM) devices [7], [8]. However, this study specifically focuses on passive crossbar architectures, which suffer from sneak path currents and scalability challenges [9]. The passive crossbars
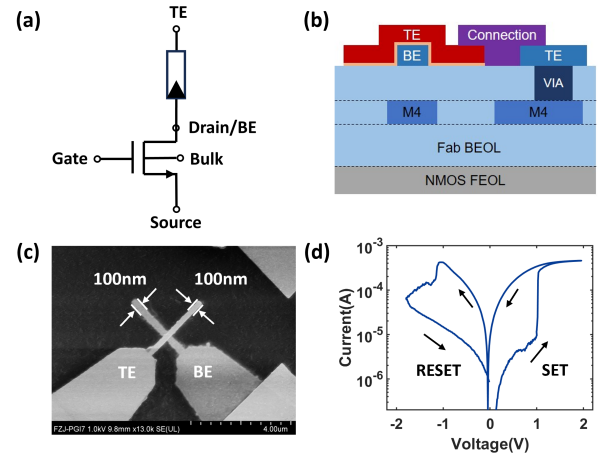
§Equal contribution

Fig. 1. (a) 1T1R cell schematic (b) schematic cross-section of CMOS integrated memristive device, (c) SEM image of fabricated memristive devices, and (d) I-V switching characteristics of a 1T1R cell after forming.

also encounter difficulties during device formation, requiring significant initial current. To address these issues and enhance forming capabilities, a solution involves incorporating a transistor in series with the memristive device. This configuration creates a 1T1R cell, effectively mitigating sneak path problems. The 1T1R cell enables precise current control at the individual device level, offering enhanced control for the MAGIC operations. Despite the increase in physical footprint, the 1T1R configuration renders the overall system more scalable and allows for better control [9].

The MAGIC design style offers the potential for a variety of logic gates. However, the availability of specific logic gates is contingent upon the SET and RESET switching thresholds, which are directly influenced by the material stack used in the fabrication process [10]. The RRAM device based on Pt/TaOx/W/Pt stack offers implementation of OR, NIMP, and 2-cycled XOR gates and has been demonstrated using 1R passive devices [7]. Additionally, in this specific stack configuration, the output device is initialized to the HRS (logic "0") state, and the input combination, along with execution voltage, determines its transition to the LRS ("1") from the HRS state.

To the best of our knowledge, for the first time, this paper demonstrates the implementation of MAGIC gates on a *1T1R*
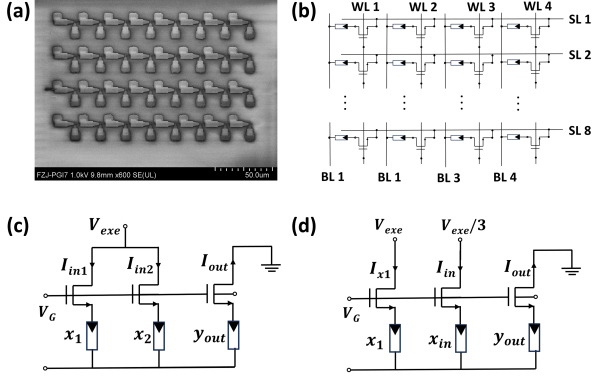
Fig. 2. Logic gates mapping on fabricated 8x4 1T1R array. (a) SEM image of the fabricated 8x4 array (array orientation rotated by 90°), (b) Schematic of 8x4 1T1R array, (c) schematic of two input OR gate implemented in the array, (d) schematic of NOT gate implemented in the array.

TaO_x *RRAM crossbar array* and illustrates the realization of both OR and NOT gates, which can be effectively combined to implement any Boolean operation as they are functionally complete. Additionally, the paper offers a detailed breakdown of energy consumption during each operation phase within the MAGIC design style. Given the potential application of the MAGIC design style in creating general-purpose processing units, it becomes imperative to examine its design from an energy consumption perspective [11]. Such an analysis is crucial for gaining insights into its viability in future technologies. The following are the contributions of this paper:

- Fabrication of TaO_x RRAM devices and their integration with CMOS to make the 1T1R crossbar array.
- Experimental validation of OR and NOT gates based on MAGIC design style on fabricated 1T1R crossbar array.
- Comprehensive energy assessment and breakdown of energy consumption during the MAGIC operations.

The rest of the paper is organized as follows. We provide a background of the used technique in Section II. Section III discusses the experimental methods used for logic implementation and energy estimations. The results obtained from the study are explained in Section IV. Finally, we conclude the paper in Section V.

II. BACKGROUND AND RELATED WORK

A. Memristive Devices

Memristive devices have emerged as a significant advancement in non-volatile memory technology. Initially proposed as a concept by Professor Leon Chua in 1971 [12], RRAM has gained prominence due to its unique ability to store data by modulating resistance states [13]. The resistance modulation is achieved by applying a voltage across the terminals of the RRAM. In response, the resistance of the devices changes based on the magnitude and direction of the current flow. Remarkably, the memristor preserves its resistance value even when devoid of power, safeguarding its data until a new voltage is applied, firmly establishing its status as a non-volatile memory element [1].

These memristive devices can be interconnected to form a crossbar structure. However, when individual memristive devices are connected in a passive crossbar configuration, issues related to forming and sneak-path currents can arise. To mitigate these concerns, memristive devices are fabricated with a CMOS transistor in series, resulting in what is known as a 1T1R cell. In Fig. 2 (a), the SEM image of the fabricated 1T1R cell is shown, with multiple cells interconnected in an 8x4 (rows × columns) crossbar structure (fabrication detailed is discussed in Section III). Fig. 2 (b) provides a schematic layout of the 1T1R crossbar array. The word lines (WLs) from 1 to 4 are linked to the gates of the devices connected in columns, while the source lines (SLs) from 1 to 8 are connected in a row-wise fashion, shorted to all the source pins of the transistors within the same row. The bit lines (BLs) from 1 to 4 are connected to the top electrode (TE) pin of each memristor within a column, as illustrated in Fig. 2 (b).

B. MAGIC Design Style

MAGIC represents a stateful logic methodology that employs crossbar-connected memristive devices to execute logic operations. Each memristor is programmable to two distinct states, HRS and LRS, which are subsequently mapped to logic "0" and "1" respectively. An initialization step is necessitated to facilitate the MAGIC operations, configuring the output memristor to its initial state. The range of achievable gates within these devices is contingent upon the SET-to-RESET switching threshold ratio. The SET-to-RESET voltage ratio for the device stack used in our study made logic OR, NIMP, and NOT gates attainable, diverging from the original NOR and NOT gates proposed in [6]. The output memristor (y_{out}) is always initialized to the HRS state rather than the LRS state.

In the OR operation, an execution voltage (V_{exe}) is applied to the input memristors (x_1 and x_2). Simultaneously, the output memristor, initially set to the HRS state, is grounded, as depicted in Fig. 2 (c). Furthermore, the NOT operation necessitates three memristors but with different voltage values compared to the OR operation. In this context, one of the inputs is consistently initialized to LRS (designated as x_1 for clarity) in conjunction with the output memristor as shown in Fig. 2 (d). Additionally, distinct execution voltages (V_{exe} and $V_{exe}/3$) are employed for the x_1 and x_{in} memristors, respectively.

C. Related Work

The experimental validation of MAGIC design style on TaO_x RRAM devices using *passive crossbars* has been demonstrated in existing literature [7]. Nonetheless, passive crossbars are plagued by the issue of sneak-path currents, which can disrupt the accurate reading of the final state. Additionally, forming processes in passive crossbars poses challenges. Non-stateful logic (e.g., scouting and majority logic) has been demonstrated using 1T1R cells [14]. This paper represents the pioneering demonstration of *stateful logic on a 1T1R TaO_x RRAM crossbar*.

Prior studies have indicated that energy consumption in the MAGIC design style is predominantly influenced by initialization energy [15]. However, it's essential to note that these findings are primarily derived from simulation studies. Furthermore, the presented energy figures are based on simulation models and pertain specifically to NOT and NOR gates based on the MAGIC design style. This paper takes strides towards calculating the energy consumption of OR and NOT gates in fabricated TaO_x RRAM devices.

III. EXPERIMENTAL METHODS

A. Device Fabrication

For the experimental validation of MAGIC gates, 1T1R-based active memristive arrays were fabricated and integrated with CMOS 180 nm technology provided by X-FAB. The dimensions of the memristive devices in the arrays are 100 nm x 100 nm and consist of $Pt/TaO_x/W/Pt$ stack. Fig. 1 (b) shows the schematic vertical cross-section of the fabricated device. Firstly, the W plugs from the processed wafers were exposed, and the 25 nm thick Pt layer was deposited as the bottom electrode (BE) with DC sputtering. The BE layer was then patterned using electron beam lithography and back etching using reactive ion etching (RIE). A 7 nm thick TaO_x layer was then deposited by RF sputtering in Ar (77%) and O_2 (23%) gas mixture at 236W RF power followed by deposition of 13 nm thick W electrode using the DC sputtering. Subsequently, a 25 nm thick Pt layer was deposited as TE using the DC sputtering. Finally, the deposited switching oxide and TE stack were patterned using electron beam lithography and RIE-based back etching. Fig. 1 (c) shows the SEM image of the fabricated 1T1R TaO_x RRAM device.

B. Electrical Characterization Setup

The electrical characterization of the 1T1R memristive devices was carried out using Keithley 4200 SCS. Fig. 1 (a) shows the schematic of the fabricated 1T1R memristive device with different terminals. The memristive device in its pristine state needs a one-time forming step, which involves the creation of a conductive filament in the switching oxide by applying a positive voltage across the memristor. The current through the memristor during the electroforming process is controlled by applying an appropriate DC gate voltage to avoid permanent breakdown of the oxide. To realize digital (logic in memory) LiM using these devices, four distinct operations are necessary for any logic operation, which are discussed below.

- **SET Operation:** The SET operation involves changing the device's state from HRS to LRS. This is achieved by applying a positive ramp voltage from 0 to 1.8V at the TE electrode while maintaining a constant DC voltage of 1.6V on the gate terminal to limit the current flowing through the memristive device to $500\mu A$. The drain and bulk pins of the transistor are connected to the ground.

- **RESET Operation:** The RESET operation switches the device from LRS to HRS by applying a positive ramp voltage from 0 to 2V at the source terminal, while keeping TE and bulk grounded. A minimum current requirement is crucial
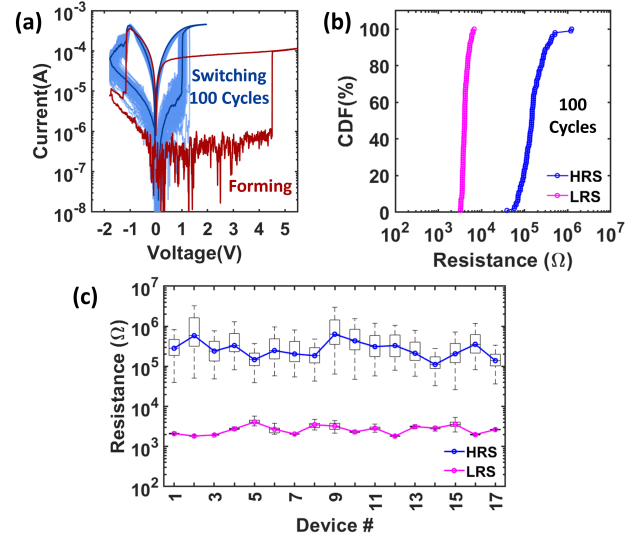

Fig. 3. Electrical characterization of TaO_x RRAM devices. (a) Forming and median I-V curve for 100 switching cycles. (b) Cycle-to-cycle variability for 100 switching cycles. (c) Device-to-device variability for 17 devices with 100 switching cycles each.

to dissolve the filament, and if not met, the device remains in LRS. Achieving this current requirement entails applying the maximum allowed voltage of 5V at the NMOS gate, opening the channel for current conduction, and affecting the minimum transistor size usable with memristors for SET and RESET processes.

- **Execution Operation:** During the execution operation, specific voltage configurations are applied to memristors for executing the OR and NOT operations. Once the input memristors are loaded with the input, the execution voltage applied across them decides the logic operation being executed on them. During this operation, the output memristor is connected to the ground. The detailed voltage value is discussed in Section IV-B and IV-C.

- **Read Operation:** The read operation serves to determine the current state of the memristor. A read voltage of 0.5V is applied at the TE and the source and the bulk terminals are pinned to the ground. A gate voltage of 3.3V is applied at the gate to open the NMOS channel during read operation.

Through skillful control of these operations, it becomes possible to design any arbitrary logic configuration on the crossbar. In this specific instance, these voltage sequences are combined to generate OR and NOT gates on the crossbar. However, these voltage patterns can also be harnessed to execute complex circuits sequentially or implement architectures resembling single instruction multiple data. Subsequently, we look into the outcomes achieved by sequentially applying these voltages to achieve the desired operations.

IV. RESULTS AND DISCUSSION

Within this section, we unveil the outcomes derived from our experimental examinations and analyses, offering valuable insights into the energy consumption associated with the implementation of MAGIC OR and NOT gates.
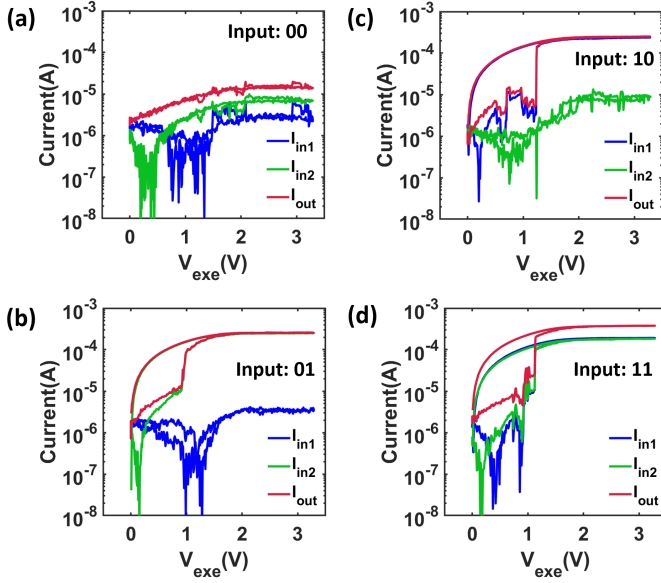
Fig. 4. Execution operation in OR gate for (a) input "00", (b) input "01", (c) input "10" and (d) input "11". For input "00" there is no sharp change in I_{out} implying no change of y_{out} state. For the rest of the input combinations, y_{out} undergoes the SET process by drawing the majority of the current through the input memristors that are in the LRS state.
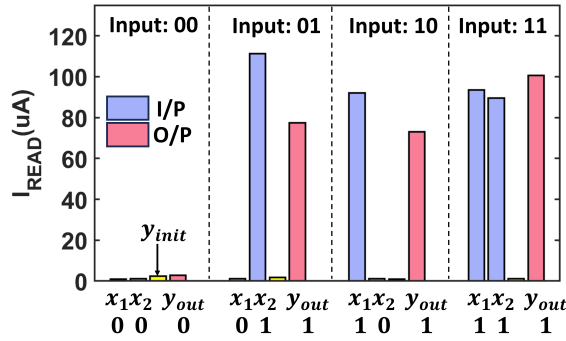

Fig. 5. Logic OR operation. Output read currents for the different combinations of input read currents.

A. 1T1R TaO$_x$ RRAM Switching Characteristics

The initial forming and subsequent 100 switching cycles are shown in Fig. 3 (a). The memristive devices exhibit counterclockwise switching characteristics. Fig. 3 (b), shows the CDF plot of HRS and LRS states obtained by switching the 1T1R devices 100 times. The device exhibits low cycle-to-cycle (C2C) variation with a consistent HRS/LRS ratio of 10. For evaluating device-to-device (D2D) variability, 17 devices were switched 100 times each as shown in Fig. 3 (c).

From the C2C and D2D variability plots, it is evident that the HRS state exhibits higher variability with resistances distributed over two orders of magnitude ranging from around 100kΩ to 1MΩ. The high variability in HRS can be attributed to the variable oxygen vacancy concentration profile once the dynamical RESET process has concluded [16]. Although the devices exhibit variability in the HRS state, a *consistent HRS/LRS ratio of 10* is obtained across all the tested devices and suffices for the implementation of logic gates. Next, we will discuss the execution of logic OR and NOT operations
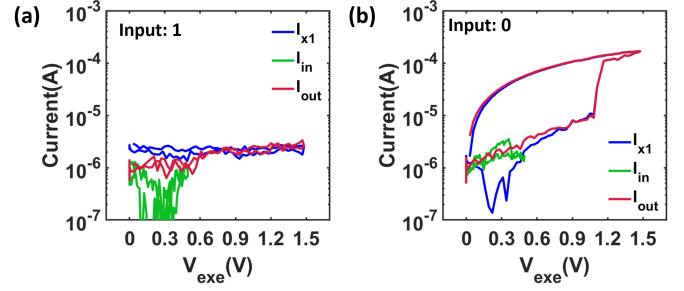

Fig. 6. Execution operation in NOT gate for (a) input "1", (b) input "0".

on these devices.

B. MAGIC OR Implementation

Fig. 2 (c) shows the schematic to implement the logic OR gate on the crossbar. To implement OR gate only three memristors are required so this needs to be mapped to the fabricated crossbar with a size of 8x4 as shown in Fig. 2 (c). To map the OR gate on the crossbar, three memristive devices sharing a common WL and a BL in the array are used to implement the gates. Firstly, the inputs of the OR gate are stored as the resistance state in x_1 and x_2. Subsequently, the output memristor (y_{out}) is initialized to the "0" state. Next, an execution voltage (V_{exe}) sweep from 0 to 3.3V at the TE terminal of x_1 and x_2 and the current at y_{out} is monitored. During this cycle, the BL shared by three x_1, x_2 and y_{out} is kept floating. While WL/V_G is connected to a DC voltage of 3.3V and the SL of the output memristor is grounded. All the other unused WLs, SLs, and BLs are kept floating. Post-execution cycle, the state of y_{out} is obtained by performing a READ operation. The memristor output currents during execution cycles for different inputs are shown in Fig. 4.

The truth table for the OR gate with the input and output states of the memristor are shown in Table I. In Table I, y_{init} column shows the initialization state of the output memristor before the execution cycle. It can be inferred from the truth table that for successful OR gate operation, the y_{out} changes its state for all combinations of inputs except for the input "00" during the execution cycle. Fig. 4 shows the execution cycles for different input combinations. For the input "00" case, both x_1 and x_2 are in HRS state represented by the current value I_{in1} and I_{in2}, respectively. When an execution voltage is applied at the inputs, the current through the y_{out} is limited by the parallel combination of x_1 and x_2. This current is not sufficient to drive the y_{out} to the SET state. This is evident from the fact that no sharp change in output current

TABLE I
TRUTH TABLE FOR OR AND NOT LOGIC GATE OPERATION

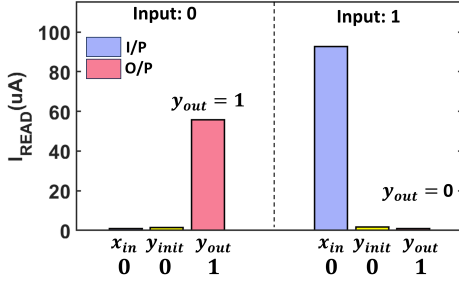| OR				NOT			
x_1	x_2	y_{init}	y_{out}	x_1	x_{in}	y_{init}	y_{out}
0	0	0	0	-	-	-	-
0	1	0	1	-	-	-	-
1	0	0	1	1	0	0	1
1	1	0	1	1	1	0	0

Fig. 7. Logic NOT operation. Output read currents for the different combinations of input read currents.

(I_{out}) w.r.t applied V_{exe} is observed in the y_{out} as shown in Fig. 4 (a).

On the other hand, for the input combination of "01", "10", and "11", either one or both of the input memristors are in LRS states. This allows a sharp increase in I_{out} during the execution cycle, contributing to the change of state of y_{out} as shown in Fig. 4 (b), Fig. 4 (c) and Fig. 4 (d), respectively. In Fig. 5, the read currents of output memristors for different combinations of input memristors are displayed, showing a successful logic OR operation.

C. MAGIC NOT Implementation

The schematic of the MAGIC NOT gate is shown in Fig. 2 (d). For performing NOT logic operation, memristor x_1 and y_{out} are initialized to LRS and HRS state, respectively. Input memristor x_{in} is initialized in accordance with the input. An execution ramp voltage of V_{exe} from 0 to 1.5V is applied at the source terminal of x_1 whereas 1/3 V_{exe} voltage is ramped at the x_{in}. Table I shows the truth table for NOT gate with different input combinations. For the successful operation of NOT gate, y_{out} memristive device should change its state when input "0" is applied and should remain in "0" state for input "1". The currents in the output memristor during the execution cycle for different inputs are shown in Fig. 6.

The input voltage applied at x_{in} is 1/3rd of the voltage applied at x_1. Because of this, during the input "0", the potential difference across the y_{out} memristor (currently in HRS state) is higher as compared to the x_{in}. This higher potential across y_{out} allows it to undergo a SET process by drawing current through x_1 (currently in LRS state). This can be seen as a sharp rise in output current as depicted in Fig. 6 (b). On the other hand, for the case of input bit "1", the y_{out} memristor path becomes higher in resistance as compared to x_{in} path and does not receive enough current to drive it into

SET from RESET state as shown in Fig. 6 (a). This allows y_{out} to change its state only when x_{in} is in the HRS state. Fig. 7 summarizes the output read currents for different input read current combinations and successfully demonstrates NOT gate.

D. Energy Calculations

The energy consumption of logic operations is heavily dependent on the initialization as well as execution energies. In earlier works, researchers have typically calculated the energy consumption of a LiM system through a coarse-grained approach by multiplying the average energy of operations by the number of operations [17]. However, this method has been found to underestimate the actual energy consumption as it ignores the initialization energy involved during the operations [15], [18]. Moreover, these results are in the simulation and with different memristor models. Therefore, in the current study, a similar approach has been considered for real devices for calculating the energy consumption of logic operations by taking execution as well as initialization energy into account.

During OR operation, the different initialization steps involve numerous SET and RESET operations as mentioned in Table III. In the study, triangular wave sweep voltages are used to perform, SET, RESET, and READ as well as execution operations. The energy is obtained by multiplying the voltage waveforms with the sensed current and then integrating the product over the measurement time as per $\int_0^t v(t) \times i(t)\,dt$, where t is the pulse time considered for energy calculations. The I,V-t curves corresponding to median SET and RESET operations are shown in Fig. 8. Two approaches have been used for calculating the energy: a) using full voltage ramp cycle time and b) using optimum times. In the first technique, energy is calculated for the whole triangular voltage sweep. In contrast, in the case of optimum energy calculation, the SET, RESET, and execution times have been derived from the experimental data, and energy is calculated for the same as shown in Fig. 8.

The former technique gives actual energy consumption numbers but is an overestimate. Therefore, for more realistic energy values, the latter technique is used. Table IV summarises the energy consumption for OR and NOT gate operation through both techniques. It is quite evident that for both the optimal and full voltage ramp cycle-based energy calculations, depending on the inputs of the OR operation, around 35-85% of the energy consumption is constituted by the initialization energy, execution energy constitutes around 64.8-14.8% while read energy only constitutes around 0.2%

TABLE II
NUMBER OF SET AND RESET OPERATIONS REQUIRED DURING
INITIALIZATION FOR OR GATE IMPLEMENTATION

	No. of initialization steps	
Input	RESET	SET
00	3	0
01	2	1
10	2	1
11	1	2

TABLE III
ENERGY CONSUMPTION IN SET, RESET, AND READ OPERATIONS FOR
FULL VOLTAGE RAMP CYCLE

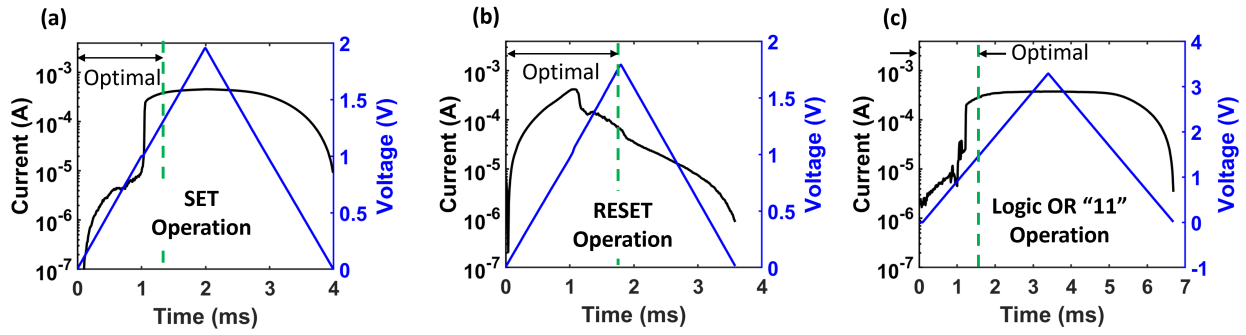| Operation	Voltage(sweep)	Pulse duration (μ s)	Energy(nJ)
SET	0 to 2 V	4 ms	1300
RESET	0 to 1.8 V	3.6 ms	312
LRS Read	0 to 0.3 V	0.6μ s	5.4
HRS Read	0 to 0.3V	0.6μ s	0.056

Fig. 8. I,V-t curves for calculating the energy consumption of (a) SET process (b) RESET process (c) Logic OR "11" input execution.

TABLE IV
COMPARISON OF INITIALIZATION ENERGY TO THE TOTAL ENERGY CONSUMED DURING DIFFERENT INPUT COMBINATIONS FOR LOGIC OR OPERATIONS.

OR	Full Voltage Ramp cycle					Optimal				
Input	Initialization Energy (nJ)	Execution Energy (nJ)	Read Energy (nJ)	% Initialization Energy	% Read Energy	Initialization Energy (nJ)	Execution Energy (nJ)	Read Energy (nJ)	% Initialization Energy	% Read Energy
00	3900	139	0.1	97	0.002	**696**	**8**	0.07	99	0.003
01	2912	2455	5.4	54	0.1	**738**	**108**	2.8	87	0.1
10	2912	2300	5.4	56	0.1	**738**	**73**	2.8	91	0.1
11	1924	3531	10.8	35	0.2	**780**	**134**	5.6	85	0.2

of energy Evidently, as the accuracy of energy calculation increases, the gap between total energy and execution energy also increases. Therefore, for accurate calculation of energy consumption for logic operations using the MAGIC technique, the initialization energy must be taken into consideration.

V. CONCLUSION

In this paper, TaO_x RRAM devices were fabricated and integrated on a 180 nm CMOS substrate to create a 1T1R crossbar array. The fabricated devices consistently exhibited an HRS/LRS ratio of approximately 10, making them eligible for the implementation of MAGIC gates. Subsequently, we demonstrated the implementation of logic OR and NOT gates, and along with energy consumption values. Energy consumptions for logic OR and NOT operations were calculated by evaluating both the initialization and execution energies. It was found that the initialization energy played a significant contributing role in the overall energy consumption during logic implementation.

REFERENCES

[1] A. Sebastian *et al.*, "Memory devices and applications for in-memory computing," *Nature nanotechnology*, vol. 15, no. 7, pp. 529–544, 2020.
[2] T. Kempen *et al.*, "50x endurance improvement in taox rram by extrinsic doping," in *2021 IEEE International Memory Workshop (IMW)*, 2021, pp. 1–4.
[3] S. Kvatinsky *et al.*, "Memristor-based material implication (IMPLY) logic: Design principles and methodologies," *IEEE TVLSI*, vol. 22, no. 10, pp. 2054–2066, 2013.
[4] S. Gupta *et al.*, "Felix: Fast and energy-efficient logic in memory," in *2018 IEEE/ACM ICCAD*. IEEE, 2018, pp. 1–7.
[5] A. Deb *et al.*, "Automated Equivalence Checking Method for Majority based In-Memory Computing on ReRAM Crossbars," in *2023 ASP-DAC*, Jan. 2023, pp. 19–25.
[6] S. Kvatinsky *et al.*, "MAGIC—Memristor-Aided Logic," *IEEE TCAS-II: Express Briefs*, vol. 61, no. 11, pp. 895–899, Nov. 2014.
[7] B. Hoffer *et al.*, "Experimental demonstration of memristor-aided logic (MAGIC) using valence change memory (VCM)," *IEEE Transactions on Electron Devices*, vol. 67, no. 8, pp. 3115–3122, 2020.
[8] B. Hoffer *et al.*, "Stateful logic using phase change memory," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 8, no. 2, pp. 77–83, 2022.
[9] M. A. Zidan *et al.*, "Memristor-based memory: The sneak paths problem and solutions," *Microelectronics Journal*, vol. 44, no. 2, pp. 176–183, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0026269212002108
[10] N. Wald *et al.*, "Design methodology for stateful memristive logic gates," in *2016 IEEE International Conference on the Science of Electrical Engineering (ICSEE)*, 2016, pp. 1–5.
[11] A. Eliahu *et al.*, *mMPU: Building a Memristor-based General-purpose In-memory Computation Architecture*, 04 2021, pp. 119–131.
[12] L. Chua, "Memristor-the missing circuit element," *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971.
[13] D. Strukov *et al.*, "The missing memristor found," *Nature*, vol. 453, pp. 80–3, 06 2008.
[14] H. Padberg *et al.*, "Experimental demonstration of non-stateful in-memory logic with 1t1r oxram valence change mechanism memristors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2023.
[15] S. Singh *et al.*, "Should we even optimize for execution energy? rethinking mapping for magic design style," arxiv 2023 (accepted).
[16] A. Marchewka *et al.*, "Nanoionic resistive switching memories: On the physical nature of the dynamic reset process," vol. 2, no. 1, 2016, p. 1500233.
[17] P. L. Thangkhiew *et al.*, "Efficient mapping of boolean functions to memristor crossbar using magic nor gates," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 8, pp. 2466–2476, 2018.
[18] S. Singh *et al.*, "MemSPICE: Automated simulation and energy estimation framework for magic-based logic-in-memory," 2023.