Neuromorphic
Computing and
Engineering

PAPER • **OPEN ACCESS**

Device-to-logic variability propagation in RRAM-based logic-in-memory architectures

To cite this article: Ankit Bende *et al* 2026 *Neuromorph. Comput. Eng.* **6** 024009

View the article online for updates and enhancements.

You may also like

- (Invited) Physical Models of Program and Read Fluctuations in Metal Oxide Resistive RAM
Daniele Ielmini

- (Invited) Physical Models of Program and Read Fluctuations in Metal Oxide Resistive RAM
Daniele Ielmini

- *(Invited)* Resistive Memories (RRAM) Variability: Challenges and Solutions
Gabriel Molas, Gilbert Sassine, Cecile Nail et al.

This content was downloaded from IP address 134.94.123.63 on 11/06/2026 at 10:40

IOP Publishing Neuromorph. Comput. Eng. 6 (2026) 024009 https://doi.org/10.1088/2634-4386/ae65d5

NEUROMORPHIC
Computing and Engineering

OPEN ACCESS

RECEIVED
20 January 2026

REVISED
8 April 2026

ACCEPTED FOR PUBLICATION
28 April 2026

PUBLISHED
13 May 2026

Original content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

PAPER

Device-to-logic variability propagation in RRAM-based logic-in-memory architectures

Ankit Bende[1,*], Simranjeet Singh[1], Chandan Kumar Jha[2], Daniele Storelli[1], Dennis Nielinger[3], Rolf Drechsler[2,4], Regina Dittmann[1], Stephan Menzel[1], Farhad Merchant[5] and Vikas Rana[6]

[1] Peter Grünberg Institute–7 (PGI-7), Forschungszentrum Jülich GmbH, Jülich, Germany
[2] Institute of Computer Science, University of Bremen, Bremen, Germany
[3] Peter Grünberg Institute–4 (PGI-4), Integrated Computing Architectures, Forschungszentrum Jülich GmbH, Jülich, Germany
[4] German Research Centre for Artificial Intelligence (DFKI), Bremen, Germany
[5] Bernoulli Institute and CogniGron, University of Groningen, Groningen, The Netherlands
[6] Peter Grünberg Institute–10 (PGI-10), Forschungszentrum Jülich GmbH, Jülich, Germany
[*] Author to whom any correspondence should be addressed.

E-mail: a.bende@fz-juelich.de, si.singh@fz-juelich.de, chajha@uni-bremen.de, d.storelli@fz-juelich.de, d.nielinger@fz-juelich.de, drechsler@uni-bremen.de, r.dittmann@fz-juelich.de, st.menzel@fz-juelich.de, f.a.merchant@rug.nl and v.rana@fz-juelich.de

Keywords: RRAM, 1T1R array, CMOS integration, in-memory computing, logic-in-memory, MAGIC, variability.

Abstract

Logic-in-memory (LiM) has emerged as a promising paradigm to address the von Neumann bottleneck by integrating data storage and *in-situ* computation. Resistive random-access memory (RRAM) is a strong candidate for LiM owing to its non-volatility, fast switching characteristics, and compatibility with CMOS integration. However, intrinsic cycle-to-cycle and device-to-device variability fundamentally limits logic reliability in RRAM-based LiM architectures, necessitating well-defined device specifications and variability margins to ensure correct operation. In this work, we experimentally demonstrate a CMOS-integrated TaO_x-based 1T1R RRAM computing fabric that supports reconfigurable LiM operations, including a functionally complete Boolean set and in-memory arithmetic primitives. Through extensive statistical measurements and variability-aware statistical modeling, we systematically evaluate the impact of key variability sources, SET voltage (V_{SET}), low resistance state, and high resistance state, on logic correctness and identify the dominant contributors to computational failure. Finally, we derive the device specifications and requirements to achieve error-free stateful LiM operations in the 1T1R RRAM arrays.

1. Introduction

The growing demand for energy-efficient and scalable computing has intensified interest in architectures that minimize data movement between memory and processing units [1, 2]. Neuromorphic and in-memory computing (IMC) paradigms address this challenge by colocating computation with data storage, thereby alleviating the von Neumann bottleneck [3–7]. Logic-in-memory (LiM) represents a particularly attractive subset of IMC, as it enables Boolean logic operations to be executed directly within memory cells [8–10]. As the computation is increasingly performed directly within memory arrays, intrinsic device-level non-idealities propagate to the logic level, directly impacting computational correctness. Consequently, reliability emerges as a critical design constraint for in-memory and neuromorphic computing architectures.

The LiM approaches can be implemented in either stateful or non-stateful forms [11]. In stateful logic, both inputs and outputs are encoded as resistive states of memory cells, allowing computation to occur directly within the memory array and enabling in-place storage of results without explicit data movement [12, 13]. In contrast, non-stateful logic does not retain computation results within the memory cells and relies on peripheral circuits for writing, reading, or buffering intermediate values. Although both paradigms perform computation within the memory fabric, non-stateful schemes

© 2026 The Author(s). Published by IOP Publishing Ltd

incur additional overhead due to signal conversion and data transfer [3, 11, 14]. This in-place computation capability makes stateful LiM particularly well suited for cascaded logic and arithmetic operations, which are essential in application domains such as machine learning, cryptography, and signal processing [15].

Resistive random-access memory (RRAM) is a leading candidate for stateful LiM due to its non-volatility, CMOS compatibility, and intrinsic ability to represent logic states through distinct resistance levels [16–19]. There are various methods that have been explored in the literature for implementing stateful LiM using resistive memories, which include memristor-aided logic (MAGIC) [20], memristor-based material implication [21], fast and energy-efficient logic in memory [22], and majority logic [23]. Among the stateful LiM approaches, MAGIC has emerged as a prominent technique as it offers superior area, latency, and energy efficiencies [24, 25]. Although prior works have successfully validated individual logic gates and arithmetic primitives, most demonstrations either rely on discrete devices, passive crossbar arrays, or non-stateful logic [26–31] requiring peripheral signal conversion. These experimental studies primarily emphasized on functional validations, with limited quantitative analysis of how intrinsic RRAM variability propagates from the device level to logic-level reliability. In particular, stochastic variations in RRAM parameters such as SET voltage (V_{SET}) and resistive states (high resistance state (HRS) and low resistance state (LRS)) are inherent to filamentary switching mechanisms and introduce both cycle-to-cycle (C2C) and device-to-device (D2D) variability [32–34]. Although such variability is well documented at the device level, its direct impact on the correctness and scalability of the stateful LiM operations remains insufficiently understood. Consequently, the lack of quantitative design guidelines connecting device statistics to logic reliability fundamentally limits the practical deployment of RRAM-based stateful logic in neuromorphic and reconfigurable computing systems.

In this work, we address this gap through a combined experimental and analytical investigation of device-to-logic variability propagation in a CMOS-integrated RRAM-based LiM architecture. We demonstrate a reconfigurable TaO_x/Ta 1T1R RRAM computing fabric that supports a functionally complete Boolean logic set using the MAGIC logic and validate in-memory arithmetic operations through a compact half-adder implementation. Using extensive statistical characterization across 5000 switching cycles and 50 devices, we quantify the impact of V_{SET}, HRS, and LRS variability on the correctness of logic operations. By combining analytical voltage-division models with variability-aware Monte Carlo simulations, we identify the dominant sources of logic failure and derive explicit device-level requirements for near-error-free stateful LiM operation.

The remainder of the paper is organized as follows: section 2 introduces the reconfigurable RRAM-based LiM fabric, section 3 details the CMOS integration and device characterization, and section 4 presents the logic and arithmetic implementations. Sections 5–7 analyze logic reliability and variability propagation, and section 8 concludes the paper.

2. Reconfigurable RRAM-based LiM fabric

The LiM enables IMC paradigm by facilitating the direct mapping of applications onto memory crossbar arrays, where Boolean operations are executed *in-situ* within the array [24, 35, 36]. Initially, the application, described in a high-level language, is translated into a hardware description language (figures 1(a) and (b)). The resulting representation is then mapped onto a memory array using the available logic gates, as illustrated in figure 1(c) [37]. Each memory cell within the array can be reconfigured to execute different logic operations, allowing complex functions to be performed directly within the memory. In this study, the memory array is based on a TaO_x/Ta RRAM-based 1T1R crossbar architecture, which enables the implementation of logic gates by utilizing memory cells in a row with a specific voltage biasing scheme. The schematic layout of the fabricated TaO_x/Ta RRAM-based 1T1R array is shown in figure 1(f), while figure 1(e) presents the logic mapping on the array. At a lower level, logic functions are executed through voltage-controlled operations, which alter the logical state of targeted cells according to the required functionality [24]. The voltage levels necessary for implementing the logic-OR and logic-NIMP operations are highlighted in figure 1(e). Each logic operation requires three RRAM memory cells, each cell consisting of an RRAM stack in series with a transistor as shown in figure 1(d). This transistor facilitates cell selection and limits current flow. The RRAM devices are integrated in the back-end-of-line of the 180 nm CMOS process, as shown in the cross-sectional view in figure 1(g). The scanning electron microscope image in figure 1(h) highlights the nanocrossbar RRAM device, showing the top electrode (TE) and bottom electrode (BE) with dimensions of 200 nm × 200 nm. These devices are configured into an 8×4 crossbar structure, which is probed using a probe card setup to facilitate logic operations. The optical image of the crossbar with the 64-pin probe card setup is shown in

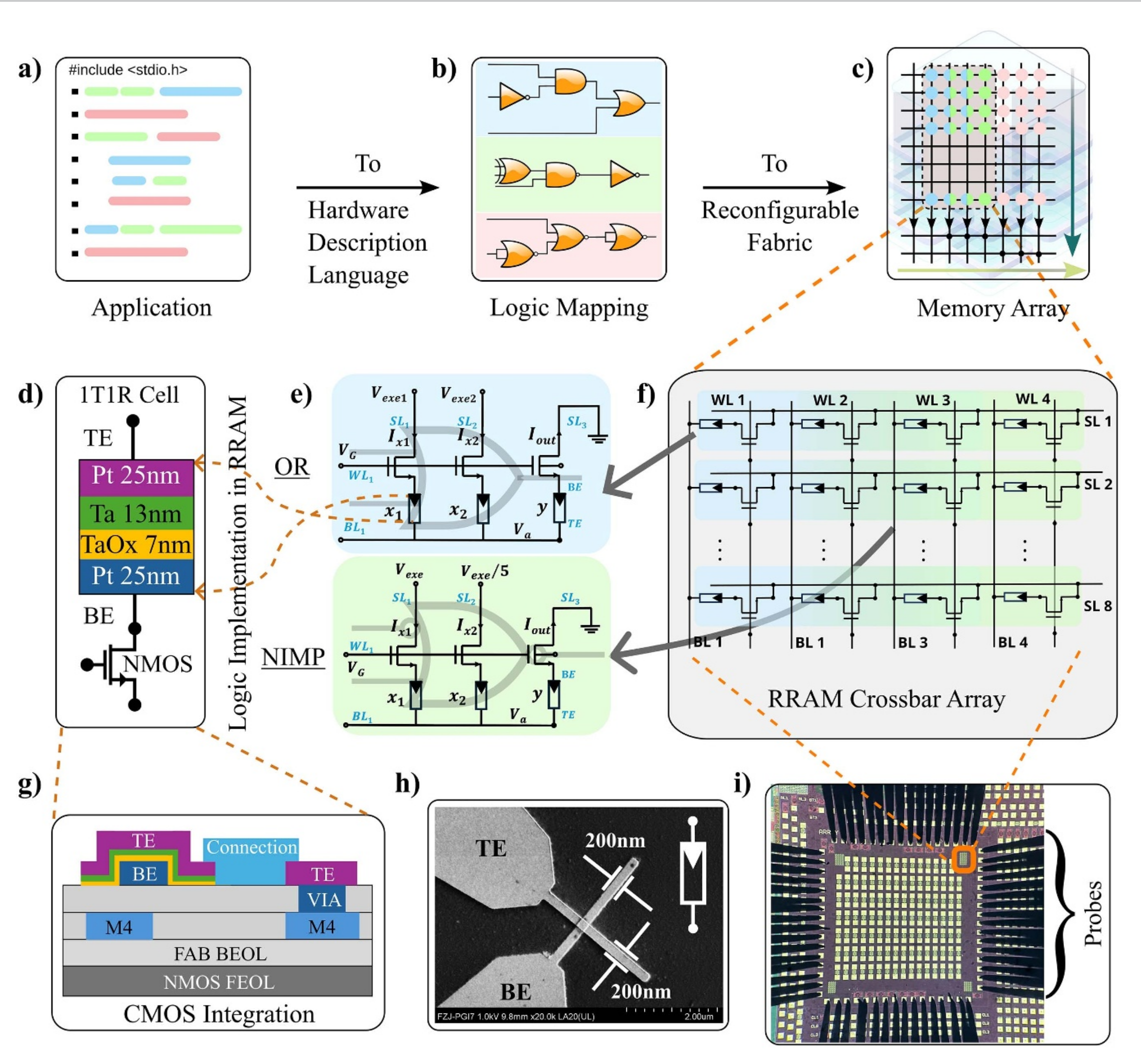

Figure 1. Reconfigurable computing using the RRAM crossbar array. (a) High-level schematic of the reconfigurable computing application. (b) Mapping of the application (a) to logic gates, analogous to the hardware description language. (c) Implementation of these logic gates on the reconfigurable fabric composed of the RRAM crossbar array. (d) Detailed view of a single RRAM device and its material stack. (e) Voltage configurations and operational sequence for in-memory logic operations, demonstrating OR and NIMP gates. (f) Magnified view of the memory array, highlighting an 8×4 RRAM crossbar. (g) Cross-sectional schematic of the RRAM-CMOS integration. (h) SEM image highlighting the physical dimensions of the fabricated RRAM device. (i) Optical micrograph of the 8×4 RRAM crossbar, corresponding to the region highlighted in (f).

figure 1(i). Overall, figure 1 illustrates the complete workflow, from application mapping to device integration, demonstrating how logic operations are implemented within the RRAM crossbar array for IMC applications.

3. CMOS integration of RRAM Arrays

This section covers the RRAM crossbar array fabrication process on the 180 nm CMOS chip, provided by XFAB. Following the integration, the characteristics of individual RRAM devices and the overall array performance are analyzed.

3.1. Array fabrication

For hardware realization of the LiM computations, TaO_x/Ta RRAM-based 1T1R arrays were fabricated on 180 nm CMOS substrate. For the RRAM integration, initially, the W plugs from the front end of the line processed wafers were exposed, followed by the deposition of a 25 nm-thick Pt layer as the BE using DC sputtering. The BE was then patterned through electron beam lithography and back etching via reactive ion etching (RIE). Next, a 7 nm-thick TaO_x layer was deposited using RF sputtering in Ar (77%) and O_2 (23%) gas mixture at 236 W RF power, followed by the deposition of a 13 nm-thick W

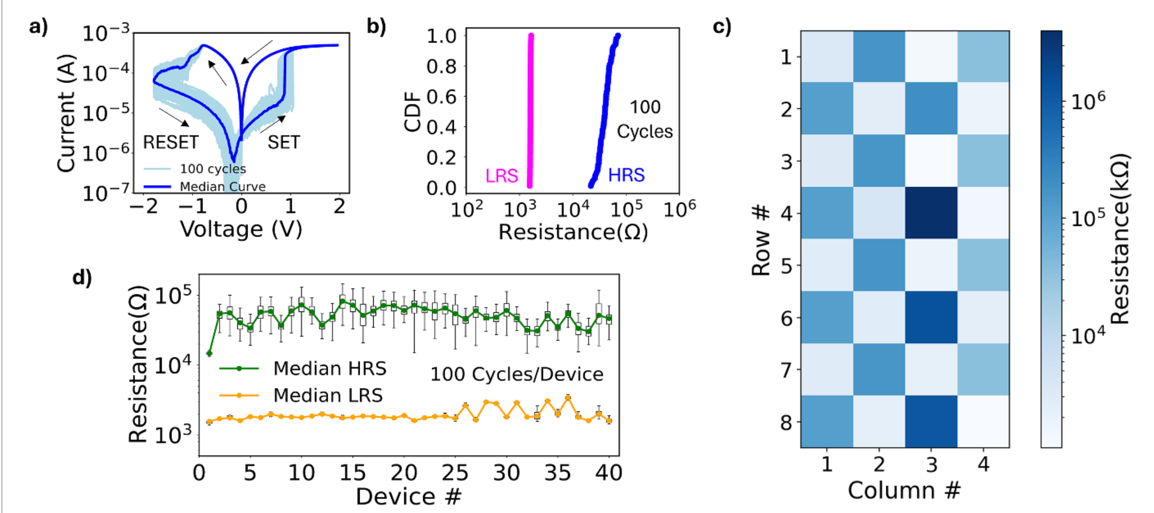

Figure 2. Electrical characterization of the TaO_x/Ta 1T1R RRAM devices: (a) I–V curves for 100 switching cycles, (b) C2C variation of LRS and HRS states for 100 switching cycles, (c) D2D variation of LRS and HRS states for 40 1T1R devices, where the box spans the interquartile range (IQR, 25th to 75th percentile) of the resistance distribution across 100 switching cycles per device, and the whiskers extend to the minimum and maximum values, and (d) checkerboard pattern programmed on the 8×4 1T1R array demonstrating controlled array programmability.

layer via DC sputtering. Subsequently, a 25 nm-thick Pt layer was deposited as the TE using DC sputtering. Finally, the patterning of the switching oxide and the TE stack was performed through electron beam lithography and RIE-based back etching.

3.2. RRAM switching characteristics

The TaO_x/Ta RRAM devices investigated in this work operate via a filamentary resistive switching mechanism. During the SET process, the application of a positive voltage to the TE drives the migration of oxygen vacancies within the TaO_x switching layer, leading to the formation of a localized conductive filament that bridges the top and BEs, thereby transitioning the device into the LRS. During the RESET process, the applied voltage drives partial dissolution of the filament through oxygen vacancy redistribution, rupturing the conductive path and returning the device to the HRS. The stochastic nature of filament nucleation and dissolution is the primary origin of the C2C and D2D variability observed in V_{SET}, LRS, and HRS, as is well established for filamentary oxide-based RRAM [32–34].

To qualify RRAM devices for the LiM operations, we analyzed their switching properties and reliability through a comprehensive electrical characterization of the 1T1R RRAM unit cells. The electrical characterization was conducted using a four-channel pulse and measure unit in a Keithley 4200 SCS semiconductor parameter analyzer. Figure 2(a) shows the current–voltage (I–V) switching curves for 100 cycles after a one-time forming step. During the SET process (switching the device to LRS), a positive voltage ramp (0–2 V) is applied to the TE, with the source and bulk terminals grounded and $V_{GS} = 1.5$ V to limit current to 500 μA. For the RESET process (switching the device in HRS), a positive voltage ramp (0–1.8 V) is applied to the source terminal, grounding the TE and bulk terminals thereby reversing the voltage polarity across the RRAM device. The RRAM device exhibits counterclockwise switching. The C2C variations are displayed as the cumulative distribution function plot for HRS and LRS over 100 cycles in figure 2(b). Figure 2(d) presents the D2D variation of HRS and LRS states for 40 devices each subjected to 100 switching cycles. The RRAM devices exhibit mean LRS, mean HRS and mean HRS/LRS ratio of 2 kΩ, 50 kΩ and 25, respectively. The statistical switching parameters are discussed in detail in section 6.1 and summarized in table 4. The observed switching characteristics, including V_{SET} and V_{RESET} below 2 V, and an HRS/LRS ratio exceeding 25, demonstrate the suitability of the TaO_x RRAM for crossbar memory array-based LiM applications. Leveraging the stable performance of the 1T1R devices, the 8×4 1T1R crossbar array was characterized, as discussed in the following section.

3.3. 1T1R array characterization

The 1T1R array was characterized using the FPGA-based crossbar memory tool Arc-2 (Arc Instruments), which facilitates programming, reading, and execution of logic operations within the array [38]. In the 8×4 array configuration illustrated in figure 1(f), devices in each column share a common bit line (BL) and word line (WL), while devices in each row share the same source lines (SLs). Programming and reading operations are performed sequentially by appropriately biasing the WLs, BLs, and SLs. To

program a specific device, the V_{SET} is applied to its corresponding BL, while all other BLs are held at ground potential. The selected device's SL is also grounded, whereas unselected SLs are pulled up to the V_{SET}. This approach ensures that only the targeted device experiences the full voltage difference of V_{SET}, while other devices within the same column remain at the same BL and SL potentials, preventing unintended programming. The WL voltage is then raised to $V_{G,SET}$, turning on all transistors along the column. Consequently, the selected 1T1R cell receives a potential difference of V_{SET} with a current compliance governed by $V_{G,S}$, thereby facilitating the controlled SET process. To RESET a particular device, the polarity of the voltage across the 1T1R device is reversed by reversing the BL and the SL voltages, and a high V_G is applied at the selected WL to fully open the series NMOS transistor. A READ operation follows the same methodology as the SET operation except that the read voltage (V_{READ}) is applied at the selected BL instead of V_{SET}. Using this methodology, a checkered board pattern was successfully programmed into the array, demonstrating precise and controllable programmability. Figure 2(c) presents the resistances of the devices following the programming of the checkered board pattern. Building on this read/write methodology, we extend our investigation to the LiM computing applications. The next section presents digital LiM concepts, including the functionally complete logic set and the arithmetic circuit block within the CMOS-integrated RRAM array.

4. Logic-in-memory implementation

The LiM enables efficient execution of complex computations directly within memory storage, eliminating the need to transfer data between logic and storage units. To demonstrate the LiM capabilities of fabricated TaO_x RRAM arrays, we first implement a functionally complete Boolean set consisting of NOT and OR gates. Functional completeness allows the implementation of any arbitrary Boolean logic function, enabling complex in-memory computation tasks. Subsequently, in-memory arithmetic operations are demonstrated by implementing a half-adder circuit based on the logic-NIMP gate in the 1T1R array.

4.1. Functionally complete logic set

To implement a functionally complete logic set, MAGIC-based logic NOT and OR operations are implemented within the CMOS-integrated TaO_x RRAM array. The NIMP gate realizes the NOT operation and serves as a fundamental building block for both logic inversion and arithmetic circuits [39]. The logic gates are constructed using three 1T1R devices within a single column of the 8×4 1T1R array, as shown in figure 3.

For executing the logic operations, inputs are first programmed into memristors x_1 and x_2, while the output memristor y is initialized to HRS. Both the programming and the initialization steps are carried out using square wave pulses of pulse width 1 μs. Logic '0' is encoded as the HRS, and logic '1' is encoded as the LRS. During execution, square wave pulses with pulse width of 1 μs and amplitudes V_{exe1} and V_{exe2} are applied to V_{SL1} and V_{SL2}, respectively, while V_{SL3} is grounded. A gate voltage (V_G) is applied to WL_1, with BL_1 left floating. To prevent unintended switching, all unselected BLs and WLs are also left floating. The selection of execution voltages at SL_1 and SL_2, along with the gate voltage at WL_1, determines the specific logic function being implemented. The correct execution of the logic operation depends on the switching behavior of the output memristor y for the applied inputs and under given execution voltages V_{exe1} and V_{exe2}. We now explain each of the logic gates implementations in detail.

4.1.1. Logic NIMP gate:

The NIMP gate is realized by applying the execution voltage V_{exe} at SL_1 and $V_{exe}/5$ at SL_2, while SL_3 is connected to ground and WL_1 has a voltage of V_G. BL_1 is kept at float. The truth table of the NIMP gate is shown in table 1. As evident from the truth table, the output memristor switches from its initialized HRS to the LRS only for the input combination '10' while for all other input combinations ('00', '01', and '11'), it remains in the initialized HRS state.

The node potential V_{out}, which corresponds to V_{NIMP} in the case of the NIMP gate, determines the switching behavior of the output memristor. Using Kirchhoff's current and voltage laws, the expression for V_{NIMP} can be derived as:

$$V_{NIMP} = \frac{V_{exe}}{5}\left(\frac{R_y(R_1+5R_2)}{R_1R_2+R_2R_y+R_1R_y}\right) \tag{1}$$

where R_1 and R_2 are the resistances of the programmed input memristors, and R_y is the resistance of the output memristor. Given an HRS/LRS ratio of n, an LRS of R, and an HRS of nR, the expressions

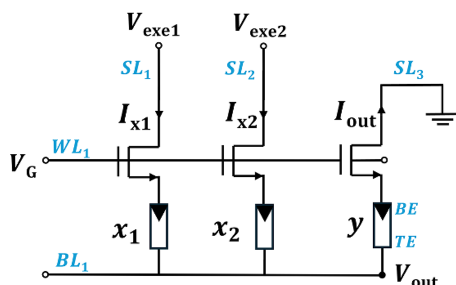

Figure 3. Schematic circuit for implementing the 1T1R RRAM-based logic NIMP, NOT, and OR gates. The logic gates require three 1T1R cells in a single column of the array to implement the logic operation. By selecting appropriate execution voltages, the respective logic operation is achieved.

Table 1. Truth table for NIMP, NOT, and OR gates together with their respective execution voltages.

NIMP Gate $V_{exe1} = 1.5$ V, $V_{exe2} = 0.3$ V BL_1 = Float				NOT Gate $V_{exe1} = 1.5$ V, $V_{exe2} = 0.3$ V BL_1 = Float				OR Gate $V_{exe1} = V_{exe2} = 1.5$ V BL_1 = Float			
x_1	x_2	y_{init}	y_{NIMP}	x_1	x_{in}	y_{init}	y_{NOT}	x_1	x_2	y_{init}	y_{OR}
0	0	0	0	—	—	—	—	0	0	0	0
0	1	0	0	—	—	—	—	0	1	0	1
1	0	0	1	1	0	0	1	1	0	0	1
1	1	0	0	1	1	0	0	1	1	0	1

Table 2. Input-dependent output voltages for the NIMP and OR gates. Here, n is the HRS/LRS resistance ratio of the RRAM devices. Values are also shown for $n = 25$, corresponding to the mean ratio measured in TaO_x-based 1T1R devices.

				NIMP		OR	
Input	R_1	R_2	R_y	V_{NIMP}	V_{NIMP} @ $n = 25$	V_{OR}	V_{OR} @ $n = 25$
00	nR	nR	nR	$\frac{2V_{exe}}{5}$	$0.4V_{exe}$	$\frac{2V_{exe}}{3}$	$0.67V_{exe}$
01	nR	R	nR	$\frac{V_{exe}}{5}\left(\frac{n+5}{n+2}\right)$	$0.2V_{exe}$	$V_{exe}\left(\frac{n+1}{n+2}\right)$	$0.96V_{exe}$
10	R	nR	nR	$\frac{V_{exe}}{5}\left(\frac{5n+1}{n+2}\right)$	$0.9V_{exe}$	$V_{exe}\left(\frac{n+1}{n+2}\right)$	$0.96V_{exe}$
11	R	R	nR	$\frac{V_{exe}}{5}\left(\frac{6n}{2n+1}\right)$	$0.6V_{exe}$	$V_{exe}\left(\frac{2n}{2n+1}\right)$	$0.98V_{exe}$

for V_{NIMP} corresponding to different input combinations are derived and summarized in table 2. As discussed in section 3.2, the fabricated device exhibits a mean HRS/LRS ratio of approximately 25. Table 2 also presents the corresponding output voltage values for this experimentally obtained ratio.

To obtain the correct logic NIMP operation the following equation must be satisfied:

$$\max\left(V_{NIMP,(00,01,11)}\right) < V_{SET} < V_{NIMP,10} \tag{2}$$

where $V_{NIMP,00}, V_{NIMP,01}, V_{NIMP,10}$ and $V_{NIMP,11}$ represents V_{NIMP} for inputs '00', '01', '10' and '11', respectively. Equation (2) ensures that the output memristor switches only for the input '10' while remaining in its initialized HRS state for all other input combinations. For the fabricated TaO_x RRAM devices with a mean HRS/LRS ratio of 25, the equation (2) can then be modified as:

$$0.6\,V_{exe} < V_{SET} < 0.9V_{exe}. \tag{3}$$

Equation (3) is then used to choose the execution voltage V_{exe} for correct NIMP operations. Figure 4(a) represents the experimental results of logic NIMP output for $V_{exe} = 1.5$ V and $V_G = 3$ V in the 1T1R crossbar array. With a threshold resistance of 10 kΩ, the correct logic NIMP output for various inputs can be obtained. The output memristor's resistance below the threshold resistance signifies logic '1', and vice versa.

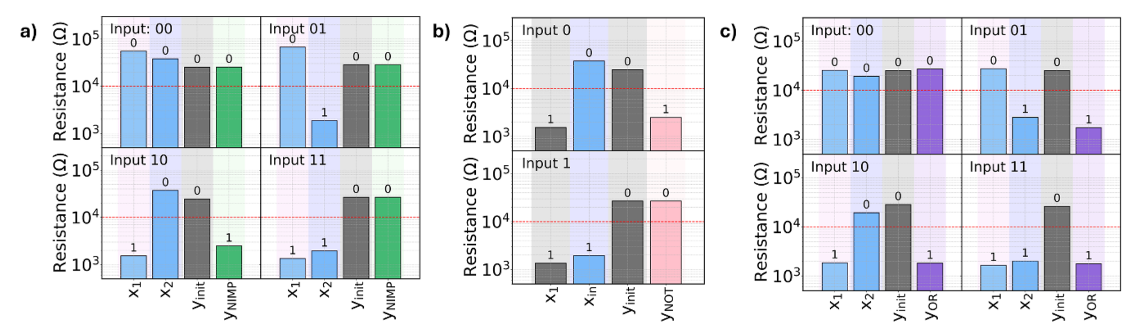

Figure 4. Logic NIMP, NOT and OR operations demonstration in the TaO_x/Ta 1T1R crossbar array. Bar graphs illustrate the resistances of input memristors x_1 and x_2, the initialized output memristor y_{init}, and the post-execution output memristors: (a) y_{NIMP} for the NIMP operation, (b) y_{NOT} for the NOT operation, and (c) y_{OR} for the OR operation, across various input combinations for a single cycle. The implementation of logic NOT and OR operations constitutes a functionally complete set, enabling the realization of complex Boolean algebra within the crossbar array architecture.

4.1.2. Logic NOT gate:

Logic NOT gate operates at the same principle as the logic NIMP gate with one of its inputs initiated to '1'. Firstly, the input memristor x_1 is initialized to '1', the 1-bit input is programmed in x_2 as x_{in}, and the output memristor y is initialized to HRS. Then, the NIMP execution operation is performed, resulting in the negation of x_{in}; the result is obtained in the output memristor as x_{NOT} . Figure 4(b) illustrates the experimentally obtained logic NOT gate output for $V_{\text{exe}} = 1.5\,\text{V}$ and $V_{\text{G}} = 3\,\text{V}$.

4.1.3. Logic OR gate:

For implementing the logic OR gate, the same execution voltage, V_{exe} is applied to both the SL_1 and SL_2 lines while grounding SL_3 line in the circuit shown in figure 3. The truth table of the OR gate is shown in table 1. The output memristor switches from its initialized HRS to the LRS for the input combinations of '01', '10', and '11'; whereas for the input combination '00', it remains in the initialized HRS state.

The node potential V_{out}, which corresponds to V_{OR} in the case of the OR gate operation, can be expressed by:

$$V_{\text{OR}} = V_{\text{exe}} \left(\frac{R_y (R_1 + R_2)}{R_1 R_2 + R_2 R_y + R_1 R_y} \right). \tag{4}$$

The condition for correct logic operation can be deduced as:

$$V_{\text{OR},00} < V_{\text{SET}} < \min \left(V_{\text{OR},01}, V_{\text{OR},10}, V_{\text{OR},11} \right) \tag{5}$$

where $V_{\text{OR},00}, V_{\text{OR},01}, V_{\text{OR},10}$ and $V_{\text{OR},11}$ represents V_{OR} for inputs '00', '01', '10' and '11', respectively. For the fabricated devices with $n = 25$, the above equation can be modified to:

$$0.67 V_{\text{exe}} < V_{\text{SET}} < 0.96 V_{\text{exe}}. \tag{6}$$

Equation (6) is then used to determine the execution voltage V_{exe} for correct OR operations. Figure 4(c) illustrates the logic OR output for $V_{\text{exe}} = 1.5\,\text{V}$ and $V_{\text{G}} = 3\,\text{V}$. With a threshold resistance of $10\,\text{k}\Omega$, the correct logic OR output for various inputs is achieved. The successful implementation of logic NOT and OR gates confirms the functionally complete logic set implementation and enables complex LiM capabilities of the TaO_x RRAM arrays. In the next section, we discuss the hardware implementation of an arithmetic circuit in the fabricated RRAM array.

4.2. In-memory arithmetic operations

A half-adder circuit is a fundamental building block for implementing arithmetic units such as adders and multipliers. Previously, Hoffer *et al* demonstrated a half-adder using RRAM-based passive crossbar arrays [39]. However, the passive crossbar arrays suffer from sneak path currents, which degrade the efficiency and accuracy of logic operations [40]. To address this limitation, we report the first implementation of the stateful MAGIC based half-adder circuit in the 1T–1R active crossbar array, shown in figure 5. In this architecture, a transistor integrated in series with each RRAM cell provides selective access to the addressed device, while isolating unselected cells, thereby suppressing sneak-path currents. The circuit consists of 2-bit inputs x_1 and x_2 and generates outputs, sum S_{XOR} and carry C_{OUT}. The sum

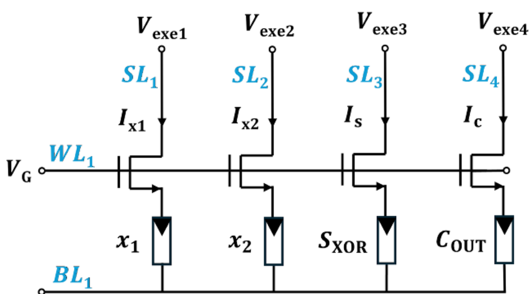

Figure 5. Schematic circuit of the 1T1R RRAM-based half adder. The half adder requires four 1T1R devices in a single column of the array. Inputs are stored in the memristors x_1 and x_2, and the sum and carry outputs are obtained in the S_{XOR} and the C_{OUT} memristors, respectively.

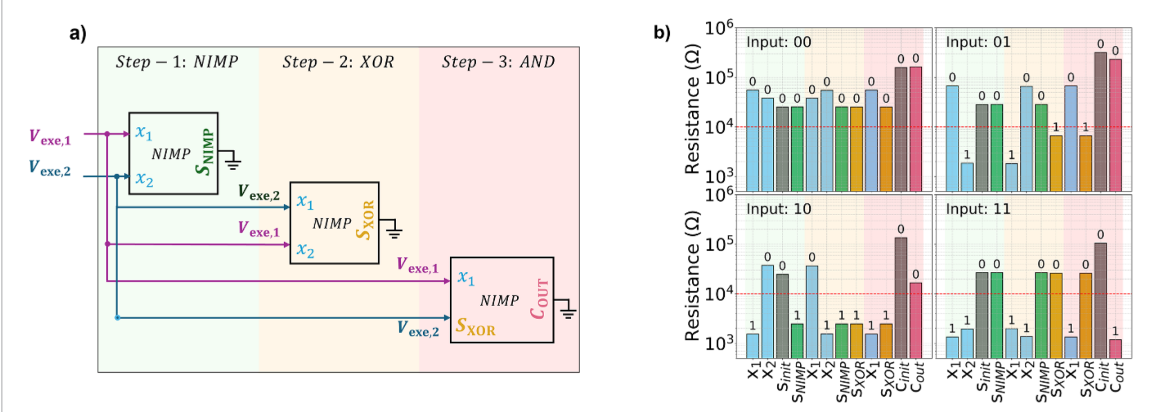

Figure 6. Demonstration of the half-adder operation in the CMOS integrated TaO_x/Ta 1T1R crossbar array. Block diagram in (a) illustrates the three NIMP logic steps necessary to compute the sum and carry outputs. Bar graphs in (b) represent the resistances of the input memristors, the initialized output memristor, and post execution output memristor across the three steps for various input combinations during a single cycle of the half-adder operation.

Table 3. Truth table for the half-adder operation. Three consecutive NIMP execution steps were used to implement the Sum and Carry outputs.

Step 1: NIMP $V_{exe1} = 1.5$ V, $V_{exe2} = 0.3$ V $V_{exe3} = 0$ V, $V_{exe4} =$ Float				Step 2: Sum (XOR) $V_{exe1} = 0.3$ V, $V_{exe2} = 1.5$ V $V_{exe3} = 0$ V, $V_{exe4} =$ Float				Step 3: Carry $V_{exe1} = 1.5$ V, $V_{exe2} =$ Float $V_{exe3} = 0.3$ V, $V_{exe4} = 0$ V			
x_1	x_2	S_{init}	S_{NIMP}	x_1	x_2	S_{NIMP}	S_{XOR}	x_1	S_{XOR}	C_{init}	C_{out}
0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	1	0	1	0	0
1	0	0	1	1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	0	1	0	0	1

output is generated via the XOR gate, while the AND gate produces the carry output. Both XOR and AND gate operations were realized using the MAGIC NIMP gate as the fundamental building block. The half-adder operation is implemented through three NIMP execution steps, as illustrated in figure 6(a). The corresponding truth table is presented in table 3, and the respective execution steps are described as follows:

(a) *Step*-1: NIMP operation. Program inputs in x_1 and x_2, initialize the memristor S as HRS and perform NIMP execution by applying voltages: $V_{SL1} = 1.5$ V, $V_{SL2} = 0.3$ V, $V_{SL3} =$ GND while keeping SL_4 and BL_1 at float. The NIMP output is obtained in S_{NIMP}.

(b) *Step*-2: XOR operation for *Sum*. Interchange the execution voltages at SL_1 and SL_2 by applying $V_{SL1} = 0.3$ V, $V_{SL2} = 1.5$ V, keep V_{SL3} at ground and perform second NIMP operation on x_1, x_2 and S_{NIMP}. SL_4 and BL_1 remain at float. The sum output is obtained in S_{XOR}.

(c) *Step*-3: AND operation for *Carry* : Initialize carry memristor as HRS (C_{INIT}) and perform third NIMP operation on x_1, S_{XOR} and C_{INIT} by applying $V_{SL1} = 1.5$ V and $V_{SL3} = 0.3$ V keeping V_{SL4} grounded, and SL_2 and BL_1 at float. The carry is obtained in C_{OUT}.

Figure 6(b) demonstrates the input and output resistance combinations for all three execution steps for all combinations of inputs. x_1 and x_2 serve as the inputs while S_{XOR} and C_{OUT} store the sum and carry outputs respectively. With the threshold of 10 kΩ, the output of the half adder for various input combinations can be correctly identified.

With the hardware implementation of the functionally complete Boolean logic set along with the arithmetic block, we have successfully validated the RRAM based fundamental primitives required for the reconfigurable LiM. However, since these logic functions are realized using the 1T1R RRAM cells, they are inherently susceptible to inaccuracies arising from device variability, including fluctuations in the SET voltages, resistance states, and HRS/LRS ratio. Such variations can impact the reliability and consistency of logic operations, necessitating a comprehensive evaluation of their effects. In the following section, we analyze the C2C and D2D variations of the NIMP and OR gates, which together form a functionally complete logic set and serve as the foundation for in-memory arithmetic operations.

5. Reliability of logic operations

This section investigates the reliability of RRAM-based logic operations across multiple switching cycles and across multiple devices. While appropriate execution voltages are selected to ensure correct nominal operation, stochastic switching dynamics and process-induced variations introduce significant C2C and D2D variability in key RRAM parameters such as V_{SET}, HRS, LRS, and the effective HRS/LRS ratio. These variations arise from the intrinsic filamentary nature of resistive switching as well as fabrication non-uniformities, and they can substantially influence logic accuracy across repeated executions and across different logic gates. Therefore, evaluating the logical correctness of the logic gates across multiple devices and repeated switching cycles is essential to ensure reliable and scalable operation.

5.1. Variations in logic NIMP operation

To evaluate the impact of the C2C and the D2D variations on the NIMP logic behavior, we performed 300 NIMP operation cycles across five distinct NIMP gates implemented in the 8×4 1T1R array. The measurements were conducted over a range of execution and gate-bias voltages, and biasing conditions that minimized logical errors were identified. In the NIMP operation, the '10' input case requires the output memristor to switch from the HRS to the LRS, i.e. the output voltage V_{out} must exceed the V_{SET}. In contrast, for the '00', '01', and '11' input cases, the output memristor must remain in the HRS, requiring $V_{out} < V_{SET}$. Among these, the '01' and '11' input combinations are the most critical due to their conflicting switching requirements and the close proximity of their output voltages, as summarized in table 2, making reliable discrimination challenging. The success of the '01' case strongly depends on the execution voltages: at low V_{exe} and V_G, the output voltage is insufficient to induce the SET transition, resulting in lower success rate, whereas higher voltages enable reliable switching. However, increasing the execution voltages also raises the output voltage for the '11' case, increasing the likelihood of unintended SET transitions and thereby degrading its correctness. This conflict significantly constrains the optimization of the execution voltages and is further exacerbated by variability in the V_{SET} of the RRAM devices. Figure 7 summarizes the success rates of individual input combinations and the overall success rate across the explored execution and gate-bias voltages. Here, the success rate is defined as the percentage of cycles producing the correct logic output. The inherent trade-off is evident in figures 7(c) and (d): the success rate of the '01' case improves with increasing execution voltages, whereas the '11' case remains stable at lower voltages and degrades at higher bias. Despite careful calibration of the V_G and V_{exe}, the overall success rate is limited to approximately 70%–80%, primarily due to variability in the V_{SET}, as discussed in section 5.4.

5.2. Variations in logic or operation

A similar voltage-dependent trend is observed in the OR gate as well. Figure 8 shows the success rates for the four OR input combinations across the measured execution and gate-bias voltages. For correct OR functionality, the output memristor must switch from the HRS to the LRS for the '01', '10', and '11' inputs, while it must remain in the HRS state for the '00' input. At lower values of V_{exe} and $V_{G,exe}$

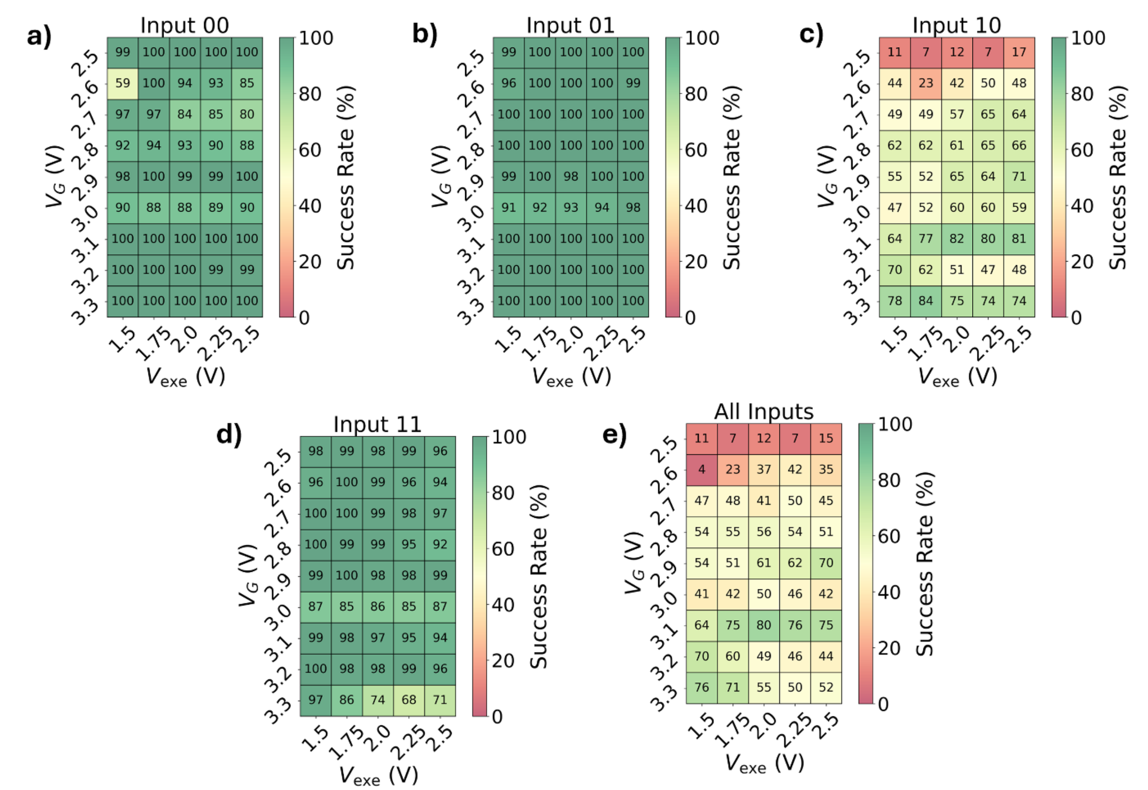

Figure 7. Variability in the logic NIMP operation. (a), (b), (c), and (d) show the success rate for 300 cycles of the NIMP operation across 5 different NIMP logic gates at different V_{exe} and V_G voltages for inputs '00', '01', '10', and '11', respectively. (e) Demonstrates the overall success rate for all inputs combined. A success rate of 70%–80% is achieved at the $V_G = 3.1$ V and the V_{exe} in the range from 1.75 to 2.5 V.

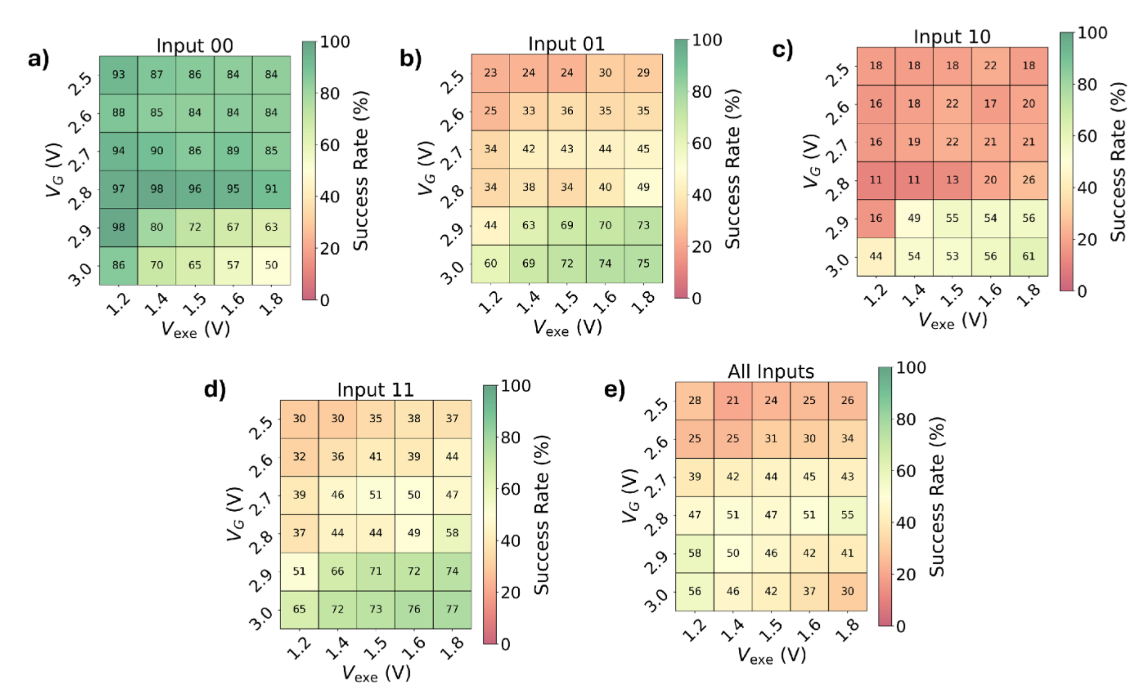

Figure 8. Variability in the logic OR operation. (a), (b), (c), and (d) show the success rate for 300 cycles of OR logic operation at different V_{exe} and V_G voltages for inputs '00', '01', '10', and '11', respectively, across five OR gates. The non switching input '00' and switching inputs '01', '10' and '11' exhibit contradictory dependence on V_{exe} and V_G voltages for achieving error-free OR operation. (e) Demonstrates the overall success rate for all inputs combined. The success rate of 40%–60% is achieved at $V_G = 3\,V$ and V_{exe} in the range from 1.2 to 1.5 V.

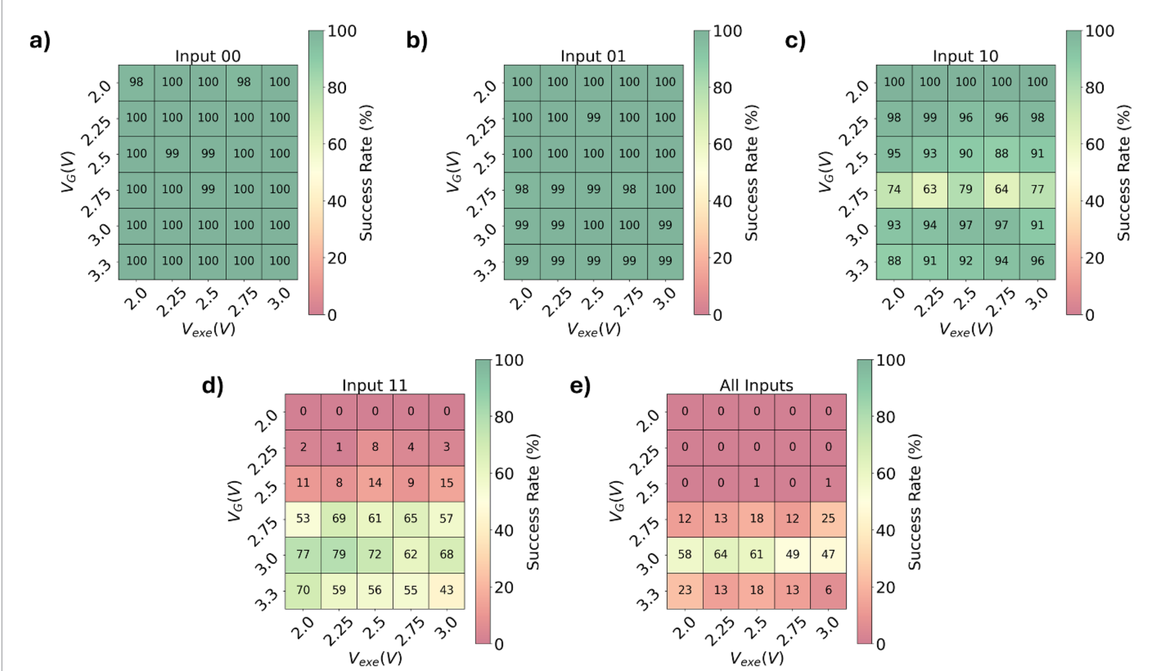

Figure 9. Variability in half-adder (HA) operation. (a)–(d) present the success rate over 300 cycles of HA logic execution at different V_{exe} and V_{G} conditions for input states '00', '01', '10', and '11', respectively. The HA implementation consists of three sequential NIMP stages; therefore, considering an individual NIMP gate success probability of $\sim$85%, the maximum achievable success rate for the complete HA operation is limited to 60%–65% due to cumulative error propagation.

(figure 8(a)), the '00' input exhibits more than 90% success rate because the applied bias is insufficient to induce the SET operation. However, under these same low-voltage conditions, the success rates for '01', '10', and '11' degrade significantly, as the required switching does not reliably occur. Increasing the voltages improves the success rates for the switching-required inputs, but simultaneously worsens the '00' case due to unintended SET events. These conflicting requirements make bias-point calibration particularly challenging. Even after optimization, the overall success rate of the logic OR gate remains limited.

5.3. Variability in half-adder operation

Figure 9 illustrates the impact of device variability on the reliability of the half-adder operation extracted through 300 cycles of the half-adder operation. The measured success rates for different input combinations ('00', '01', '10', and '11') reveal noticeable degradation in overall HA performance across operating conditions. Since the HA is implemented using three sequential NIMP stages, errors accumulate across stages, leading to a reduced overall success probability. To minimize errors in the HA operation, it is therefore essential to identify, analyze, and improve the success rate of the underlying NIMP logic operations, as they form the fundamental building blocks of the computation. We attribute this behavior to the high variability in the V_{SET} of the RRAM devices. In the next section, we mathematically correlate the experimentally extracted V_{SET} variability with the observed logic failures and estimate the maximum achievable success rate of the NIMP and OR operations in the fabricated 1T-1R array.

5.4. Correlation of V_{SET} distribution with success rate

In the previous section, we postulated that the reduction in the success rate of the logic operations is primarily driven by the variability in the V_{SET}. To validate this proposition, we first experimentally extracted the distribution of the V_{SET} for the fabricated devices and then correlated this distribution with the success rate of the logic operation at a given V_{exe}. The SET voltage V_{SET} is defined as the voltage at which a SET event occurs and is extracted from current–voltage (I–V) sweeps as the voltage corresponding to a current threshold of 100 μA. This analysis allows us to determine whether the observed spread in the V_{SET} can mathematically explain the logic errors that occur during the operation. Equation (3) defines the range of V_{SET} values required for error-free NIMP operation at a given V_{exe}. Furthermore, table 2 identifies the relationship between the V_{exe} and the corresponding input-dependent output voltages V_{NIMP} for the simplified NIMP circuit, as shown in figure 10(a). Using equation (2) along with the voltage mappings in table 2, we calculated the success rate of the NIMP operation across a range of the V_{exe} values based on the experimentally obtained V_{SET} distribution. This mathematically

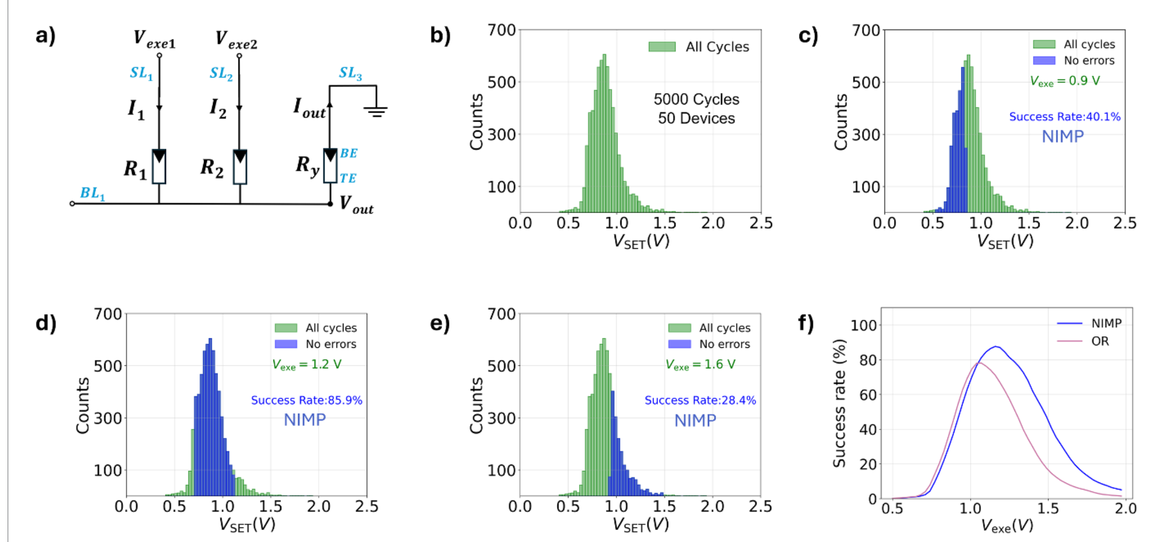

Figure 10. Mathematical modeling of the V_{exe} -dependent logic operation success rate. (a) Simplified circuit diagram of the NIMP gate. (b) Experimentally obtained V_{SET} distribution, derived from 5000 switching cycles across 50 1T1R devices. (c), (d), and (e) Overlapping range of the V_{SET} values enabling error-free NIMP operation at V_{exe} of 0.9 V, 1.2 V, and 1.6 V, respectively. The V_{SET} values outside the overlapping range lead to the NIMP logic errors. The success rate quantifies the percentage of V_{SET} values within the desirable range relative to the experimentally obtained V_{SET} distribution. (f) The mathematically calculated success rate as a function of execution voltage for both the NIMP and OR logic gates, showing a maximum success rate of approximately 85% for the NIMP gate and approximately 70% for the OR gate at $V_{exe} = 1.2$ V can be achieved for the given SET variability of the TaO_x 1T1R devices.

derived V_{exe}-dependent success rate represents the theoretical upper bound achievable for the fabricated 1T1R devices.
The detailed methodology used to compute this success rate is outlined below:

1. Extract the V_{SET} distribution from 5000 SET cycles measured across 50 devices.
2. For a given V_{exe}, we use table 2 to determine the corresponding output voltages $V_{NIMP,00}$, $V_{NIMP,01}$, $V_{NIMP,10}$, and $V_{NIMP,11}$.
3. Identify the subset of the V_{SET} values that satisfy equation. (2), ensuring correct execution of the NIMP logic operation.
4. Compute the success rate as the percentage of the V_{SET} values that meet this criterion relative to the total of 5000 measured switching cycles.

values.

Figure 10(b) shows the V_{SET} distribution obtained from the 5000 switching I–V cycles measured across the 50 fabricated 1T1R RRAM devices. Figures 10(c)–(e) illustrate the valid V_{SET} ranges that enable error-free NIMP logic operation for execution voltages of 0.9 V, 1.2 V, and 1.6 V, corresponding to the success rates of 40.1%, 85.9%, and 28.4%, respectively. The success rate of, for example, 85.9% at $V_{exe} = 1.2$ V indicates that, due to the broad distribution of V_{SET}, only 85.9% of switching cycles satisfy the conditions required for correct NIMP operation, while the remaining cycles fall outside the valid V_{SET} window and therefore resulting in logic operation errors. A similar analysis was performed for the OR logic gate. Figure 10(f) presents the mathematically calculated success rate as a function of V_{exe} for both NIMP and OR operations. For the measured V_{SET} distribution of the fabricated devices, the maximum achievable success rate is approximately 85% for the NIMP operations and about 70% for the OR operations at an execution voltage of 1.2 V. These theoretical values closely match the success rates trend observed experimentally. For the half-adder operation, which requires three sequential NIMP steps, the overall success rate is further reduced due to the multiplicative effect of per-operation reliability. With the individual NIMP success rate of approximately 85%, the maximum achievable success rate for the three-step half-adder is around 63.6%. These results indicate that the broad distribution of the V_{SET} is a dominant factor limiting logic reliability. This observation strongly motivates a systematic evaluation of logic correctness under different sources of device variability and the identification of RRAM design specifications required to achieve error-free operation. Accordingly, the following section investigates the impact of key variability sources in the RRAM devices on the reliability of logic operations.

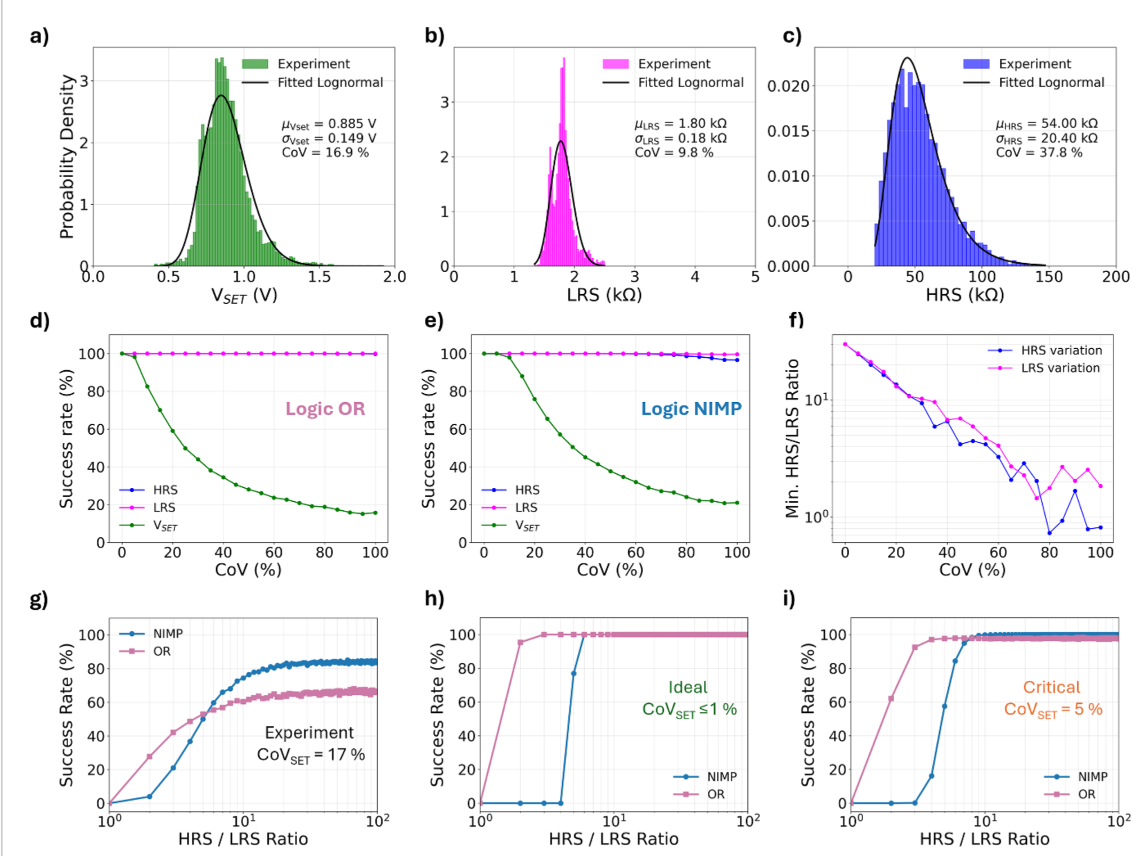

Figure 11. Effect of RRAM variabilities on logic operation. (a)–(c) Experimentally measured lognormal-distributed device parameters extracted from 5000 switching cycles across 50 1T1R devices in the array: (a) V_{SET}, (b) LRS, and (c) HRS. Success rate (SR) versus coefficient of variation (CoV) of HRS, LRS, and V_{SET} obtained from Monte-Carlo simulations for (d) MAGIC OR gate and (e) MAGIC NIMP gate, showing negligible impact of HRS and LRS variability but strong degradation from V_{SET} variability. (f) Minimum achievable HRS/LRS ratio for imposed CoV on HRS and LRS, demonstrating that the large nominal separation between HRS and LRS keeps the ratio above $\sim$2 even for large CoV values and therefore do not degrade the SR. Required HRS/LRS ratio and for reliable operation under different V_{SET} spreads: (g) experimental $\mathrm{CoV}_{V_{\mathrm{SET}}} = 17\%$, (h) ideal $\mathrm{CoV}_{V_{\mathrm{SET}}} \leqslant 1\%$, and (i) critical $\mathrm{CoV}_{V_{\mathrm{SET}}} \approx 5\%$. Near-error-free logic operation requires $\mathrm{CoV}_{V_{\mathrm{SET}}} \lesssim 5\%$ and HRS/LRS $\gtrsim 10$.

6. Impact of RRAM variability on logic operations

Due to inherent stochastic switching dynamics and process-induced variations, the RRAM devices exhibit significant variability in key parameters, including the V_{SET}, the HRS, the LRS, and the resulting resistance ratio. These variations directly influence the success rates of the RRAM-based logic operations. In this section, we systematically investigate how variability in these device parameters degrades the logic reliability. We first experimentally characterize and model the statistical distributions of the relevant RRAM properties and then correlate them with the observed logic success rates. This analysis identifies the dominant variability parameter that limits reliable logic execution. It also provides design guidelines for materials scientists and process engineers. These guidelines can help optimize RRAM material stacks and its fabrication processes for logic-in-memory operations.

6.1. RRAM variability characterization

To quantify the device-level stochasticity in our fabricated RRAM devices, we measured the distributions of the V_{SET}, the LRS, and the HRS values across 5000 switching events. These measurements were obtained from the 50 devices, each cycled 100 times. Figures 11(a)–(c) present the experimentally measured histograms along with fitted lognormal probability density functions. The degree of variability was quantified using the coefficient of variation (CoV), defined as

$$\mathrm{CoV} = \frac{\sigma}{\mu} \times 100$$

where σ and μ represent the standard deviation and the mean of the distribution, respectively. The statistical parameters obtained through the lognormal fit are summarised in table 4.

Table 4. Statistical parameters of the fabricated RRAM devices.

Parameter	Mean (μ)	Standard deviation (σ)	CoV ($\frac{\sigma}{\mu} \times 100$)
V_{SET}	0.885 V	0.149 V	16.9%
LRS	1.80 kΩ	0.176 kΩ	9.8%
HRS	54 kΩ	20.41 kΩ	37.8%
HRS/LRS	30	—	—

These extracted statistical parameters were then used as inputs to the Monte-Carlo simulation framework designed to evaluate the logic reliability of MAGIC-based operations under realistic device variability conditions.

6.2. Monte-carlo modeling framework

The behavior of the MAGIC logic gates was modeled using a three-resistor network, in which two input devices (R_1 and R_2) interact with an output device (R_y) initially programmed to the HRS, as shown in figure 10(a). The output node voltage V_{out}, which determines whether the output device undergoes a SET transition, was computed using Equations (1) and (4) for logic NIMP and OR, respectively. To evaluate the impact of variability in the HRS, the LRS, and the V_{SET} on logic reliability, Monte-Carlo simulations were performed by varying one parameter at a time while keeping the remaining parameters fixed at their nominal mean values. For the given variability parameter and the CoV value, 10 000 Monte-Carlo iterations were performed. In each iteration, the selected parameter was randomly sampled from its corresponding lognormal distribution, while the other parameters were held constant. For each sampled set of device parameters, the output voltage V_{out} was evaluated for all four input combinations. The logical correctness was determined by comparing V_{out} with the sampled V_{SET} value for that iteration. A Monte-Carlo trial was considered successful only if *all four* input combinations exhibited their expected switching or non-switching behavior.

6.3. Impact of HRS, LRS, and V_{SET} variability

Figures 11(d) and (e) show the success rate of the OR and the NIMP logic gates as a function of the imposed CoV on the HRS, LRS, and V_{SET} parameters. While assessing the effect of variation of a parameter, quantized by the CoV, the other two parameters were fixed at their mean values. The simulations reveal that variability in the resistive states has only a marginal effect on the logic reliability: even when the CoV of HRS or LRS is increased to 100%, the SR remains above 95% for both gates. This indicates that fluctuations in resistance do not significantly perturb the voltage division that governs the MAGIC switching. In contrast, the variability in the V_{SET} produces a steep degradation in the SR. As $\mathrm{CoV}_{V_{SET}}$ increases, the SR decreases rapidly, falling to approximately 40% for the OR gate and below 20% for the NIMP gate at 100% CoV. The minimum V_{SET} variability required to achieve 100% success rate is identified to be $\mathrm{CoV}_{V_{SET}} \leqslant 5\%$ for both logic gates. This pronounced sensitivity is consistent with our experimental observations, where the OR and the NIMP gates achieve only $\sim 60\%$ and $\sim 80\%$ success rates with experimental $\mathrm{CoV}_{V_{SET}}$ of 16.9%, respectively. These trends confirm that the statistical spread in the V_{SET} is the dominant device-level factor limiting reliable logic execution in the fabricated RRAM arrays. The limited influence of the HRS and the LRS variability on the logic performance can be understood by analyzing the evolution of the minimum HRS/LRS ratio under increasing CoV, as shown in figure 11(f). Even under large imposed fluctuations, the minimum ratio (the worst case scenario) within each sampled population remains higher than 2 for the CoV values below 80%. Since the MAGIC logic operation depends primarily on the ratio-dependent voltage division rather than the absolute resistance values, the random fluctuations in resistances rarely affect the switching behavior of the output memristor. Consequently, the HRS and the LRS variability make only a minor contribution to logic errors, while the V_{SET} variability remains the dominant source of the logic failure.

6.4. Impact of HRS/LRS ratio

To quantify the minimum HRS/LRS ratio required for reliable MAGIC logic operation, we analyzed the success rates of the logic gates as a function of the HRS/LRS ratio. In this analysis, the LRS was fixed at its experimentally extracted mean value, while the HRS/LRS ratio was swept from 1 to 100 by varying the HRS. Figures 11(g)–(i) summarize the results for three representative cases. For the experimentally measured V_{SET} variability ($\mathrm{CoV}_{V_{SET}} = 17\%$), the success rate saturates at approximately 60% for the OR and 80% for the NIMP logic, even at high HRS/LRS ratios. This behavior closely matches the experimental observations. In contrast, in an idealized scenario with nearly uniform V_{SET} ($\mathrm{CoV}_{V_{SET}} \leqslant 1\%$),

Figure 12. READ threshold sensitivity analysis: (a) correct classification success rate of LRS (bit '1') and HRS (bit '0') as a function of READ threshold. Threshold-dependent success rates for all input combinations of (b) the Logic OR gate, and (c) the Logic NIMP gate.

both logic gates achieve near-perfect success rates ($\approx 100\%$) with the modest HRS/LRS ratio of approximately 6. For the critical $\mathrm{CoV}_{V_{\mathrm{SET}}} \approx 5\%$, both OR and NIMP gates reach 100% success for HRS/LRS ratio exceeding approximately 10. These results confirm that the V_{SET} variability is the dominant bottleneck limiting logic operation reliability. Together, these results define the combined $(\mathrm{CoV}_{V_{\mathrm{SET}}}, R_{\mathrm{HRS}}/R_{\mathrm{LRS}})$ design space required to ensure the robust MAGIC logic operation.

6.5. READ threshold sensitivity on logic operations

In addition to the inherent RRAM variability parameters such as V_{SET}, HRS, and LRS, the correctness of logic operations is also dependent on READ threshold fluctuations. Significant variation in the READ margin can lead to false classification of either the input logic states (0/1) or the output of the logic gate. To demonstrate the robustness of the logic operations and to define a safe operable READ margin, a READ threshold sensitivity analysis was carried out to identify the range of read thresholds that minimizes false classification errors. To implement this, the extracted LRS and HRS distributions were used to characterize logic bit '1' and '0', respectively, and 10 000 Monte Carlo simulation trials of the Logic NIMP and OR gates were performed for READ thresholds swept from 1 kΩ to 100 kΩ. Figure 12(a) shows the correct classification success rate of LRS (bit '1') and HRS (bit '0') as a function of READ threshold. Below 3 kΩ, the LRS is misclassified as bit '0', while above 20 kΩ, the HRS is misclassified as bit '1', yielding a safe classification window of $3 < R_{\mathrm{TH}} < 20\,\mathrm{k}\Omega$ within which both states are correctly identified with a success rate of 100%. At the nominal threshold of 10 kΩ, which lies centrally within this window, both LRS and HRS are correctly classified, confirming a negligible misclassification probability and a robust read margin. Figures 12(b) and(c) present the simulated READ threshold-dependent success rates for various input combinations of the Logic OR and NIMP gates, respectively. For the Logic OR gate, the success rate for input combinations 01, 10, and 11 drops below a READ threshold of 3 kΩ, as the correct output logic '1' is misclassified as '0'. For the input combination 00, the success rate begins to degrade above 22 kΩ as in this case the correct output logic '0' is misclassified as '1'. Therefore, a READ threshold in the range $3 < R_{\mathrm{TH}} < 22\,\mathrm{k}\Omega$ ensures correct Logic OR operation. Similarly, for the Logic NIMP gate, the success rate for input combination 10 degrades below 3 kΩ, while the success rates for input combinations 00, 01, and 11 drop above 20 kΩ. Consequently, a safe READ threshold window of $3 < R_{\mathrm{TH}} < 20\,\mathrm{k}\Omega$ yields successful Logic OR and NIMP operation. , $(R_{\mathrm{HRS}}/R_{\mathrm{LRS}})$ design targets necessary for robust MAGIC operation.

7. Device requirements for error-free logic-in-memory operation

The experimental demonstrations and variability-aware modeling presented in sections 5 and 6 reveal that, even with careful calibration of execution and gate-bias voltages, the errors in logic operations cannot be fully eliminated in the fabricated RRAM arrays. Thus, we can conclude that intrinsic device variability is the main factor limiting reliable MAGIC-based logic-in-memory operation. To establish quantitative and actionable device-level requirements, we combine large-scale experimental characterization with Monte Carlo simulations in which the distributions of the V_{SET}, the HRS, and the LRS are independently controlled. This analysis yields two central insights. First, variability in the V_{SET} emerges as the dominant factor governing logic reliability. Even in the presence of a large nominal separation between the HRS and the LRS, a broad V_{SET} distribution severely restricts the overlap between the input-dependent output voltages and the permissible switching window required for the correct MAGIC operation. Consequently, for the experimentally observed $\mathrm{CoV}_{V_{\mathrm{SET}}} \approx 17\%$, both OR and NIMP logic

gates exhibit success rates below 100%, in close agreement with the measured behavior. This limitation persists regardless of bias optimization, indicating that voltage calibration alone cannot compensate for excessive V_{SET} dispersion. Second, the variability in the HRS and the LRS states has only a marginal impact on the logic correctness, provided that their nominal resistance ratio remains sufficiently large. Monte Carlo simulations show that even under large imposed fluctuations in the HRS and the LRS, the resulting HRS/LRS ratio remains high enough to preserve the voltage-division conditions governing MAGIC logic. As a result, the resistance variability by itself does not constitute a primary reliability bottleneck. Based on the combined experimental and modeling results, we derive the following device-level requirements for achieving near–error-free stateful logic-in-memory operation:

$$\mathrm{CoV}_{V_{\mathrm{SET}}} \lesssim 5\%, \qquad \frac{R_{\mathrm{HRS}}}{R_{\mathrm{LRS}}} \gtrsim 10. \tag{7}$$

Satisfying these conditions enables both OR and NIMP logic gates to achieve success rates approaching 100%, thereby supporting reliable cascaded logic and multi-step arithmetic operations. This requirement is particularly critical for reconfigurable computing fabrics and neuromorphic inference workloads, where logic depth amplifies per-operation error probabilities. Importantly, these results highlight that reducing V_{SET} variability is significantly more impactful than merely increasing the absolute HRS/LRS ratio. Reducing the V_{SET} variability is achievable through several complementary strategies. At the material level, interface engineering via a thin interlayer at the electrode–oxide interface stabilizes the oxygen exchange between the oxide and the ohmic electrode, yielding a reduction in V_{SET} variability compared to baseline stacks [41, 42]. At the programming level, avoiding deep RESET ensures the filament gap remains narrow and geometrically uniform, thereby reducing C2C V_{SET} variability in the subsequent SET operation [43]. Together, RRAM material stack engineering, improved forming protocols, and careful programming conditions provide a viable pathway toward the sub-5% variability required for reliable logic operation.

8. Conclusion

This work establishes a quantitative framework linking intrinsic RRAM device variability to logic level reliability in stateful LiM architectures. Using a CMOS-integrated TaO_x/Ta 1T1R fabric, we demonstrate a reconfigurable in-memory platform capable of implementing the MAGIC family of logic operations as well as a half-adder arithmetic unit. Variability-aware Monte Carlo simulations and statistical analysis reveal that the success rate of logic operations is primarily limited by V_{SET} variability, with $\mathrm{CoV}_{V_{\mathrm{SET}}} \lesssim 5\%$ and $R_{\mathrm{HRS}}/R_{\mathrm{LRS}} \gtrsim 10$ identified as critical targets for near–error-free logic operation. In contrast, variations in the HRS and LRS are less critical, provided that their nominal ratio remains sufficiently large.

These results provide actionable design guidelines for RRAM device optimization and for minimizing variability arising from process integration, highlighting the importance of co-design between device physics and logic architectures. Beyond individual logic demonstrations, this study outlines a roadmap toward robust, and energy-efficient IMC systems, and lays the foundation for future neuromorphic and reconfigurable computing platforms leveraging stateful LiM.

Acknowledgments

Special thanks go out to Dr Michael Schiek for his management work in the NEUROTEC II project.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Funding

This work was supported in part by the Federal Ministry of Education and Research (BMBF, Germany) in the project NEUROTEC II under Project 16ME0398K, Project 16ME0399, German Research Foundation (DFG) within the Project PLiM (DR 287/35-1, DR 287/35-2).

Conflict of interest

The authors declare that there are no conflicts of interest related to this work.

ORCID iDs

Ankit Bende 0009-0008-6434-7667
Simranjeet Singh 0000-0002-8297-1470
Chandan Kumar Jha 0000-0002-7237-5878
Daniele Storelli 0009-0004-5623-7442
Dennis Nielinger 0000-0002-7540-7062
Rolf Drechsler 0000-0002-9872-1740
Regina Dittmann 0000-0003-1886-1864
Stephan Menzel 0000-0002-4258-2673
Farhad Merchant 0000-0002-3708-5621
Vikas Rana 0000-0001-5432-0286

References

[1] Wulf W A and McKee S A 1995 Hitting the memory wall: implications of the obvious *SIGARCH Comput. Archit. News* **23** 20–24
[2] Gholami A, Yao Z, Kim S, Hooper C, Mahoney M W and Keutzer K 2024 Ai and memory wall *IEEE Micro* **44** 33–39
[3] Ielmini D and Wong H-S P 2018 In-memory computing with resistive switching devices *Nat. Electron.* **1** 333–43
[4] Marković D, Mizrahi A, Querlioz D and Grollier J 2020 Physics for neuromorphic computing *Nat. Rev. Phys.* **2** 499–510
[5] Zidan M A, Strachan J P and Lu W D 2018 The future of electronics based on memristive systems *Nat. Electron.* **1** 22–29
[6] Pedretti G and Ielmini D 2021 In-memory computing with resistive memory circuits: status and outlook *Electronics* **10** 1063
[7] Furber S 2016 Large-scale neuromorphic computing systems *J. Neural Eng.* **13** 051001
[8] Stone H S 1970 A logic-in-memory computer *IEEE Trans. Comput.* **19** 73–78
[9] Talati N, Ben-Hur R, Wald N, Haj-Ali A, Reuben J and Kvatinsky S 2020 mMPU—a real processing-in-memory architecture to combat the von Neumann bottleneck *Computing Systems: Architecture, Modeling and Simulation — 30th Int. Conf., SAMOS 2020 Samos, Greece July 20–23 2020 Revised Selected Papers* (Springer) pp 191–213
[10] Zhang Y, Xu Li, Dong Q, Wang J, Blaauw D and Sylvester D 2018 Recryptor: a reconfigurable cryptographic cortex-M0 processor with in-memory and near-memory computing for IoT security *IEEE J. Solid-State Circuits* **53** 995–1005
[11] Neuner T, Padberg H, Kornblum L, Yalon E, Amiri P K and Kvatinsky S 2026 A comparative study of digital memristor-based processing-in-memory from a device and reliability perspective *Adv. Electron. Mater.* **12** e00348
[12] Kim Y S, Son M W and Kim K M 2021 Memristive stateful logic for edge Boolean computers *Adv. Intell. Syst.* **3** 2000278
[13] Xu N, Park T, Yoon K J and Hwang C S 2021 In-memory stateful logic computing using memristors: gate, calculation and application *Phys. Status Solidi RRL* **15** 2100208
[14] Zahedi M, Lebdeh M A, Bengel C, Wouters D, Menzel S, Gallo M L, Sebastian A, Wong S and Hamdioui S 2022 Mnemosene: tile architecture and simulator for memristor-based computation-in-memory *J. Emerg. Technol. Comput. Syst.* **18** 1–2
[15] Eliahu A, Hur R, Ali A H and Kvatinsky S 2021 *mMPU: building a memristor Based General Purpose In Memory Computation Architecture* pp 119–31
[16] Wong H-S P, Lee H-Y, Yu S, Chen Y-S, Wu Y, Chen P-S, Lee B, Chen F T and Tsai M-J 2012 Metal–oxide RRAM *Proc. IEEE* **100** 1951–70
[17] Siemon A *et al* 2015 Realization of Boolean logic functionality using redox-based memristive devices *Adv. Funct. Mater.* **25** 6414–23
[18] Linn E, Rosezin R, Tappertzhofen S, Böttger U and Waser R 2012 Beyond von Neumann—logic operations in passive crossbar arrays alongside memory operations *Nanotechnology* **23** 305205
[19] Wang C, Shi G, Qiao F, Lin R, Wu S and Hu. Z 2023 Research progress in architecture and application of RRAM with computing-in-memory *Nanoscale Adv.* **5** 1559–73
[20] Kvatinsky S, Belousov D, Liman S, Satat G, Wald N, Friedman E G, Kolodny A and Weiser U C 2014 MAGIC—memristor-aided logic *IEEE Trans. Circuits Syst. II Express Briefs* **61** 895–9
[21] Borghetti J, Snider G S, Kuekes P J, Yang J J, Stewart D R and Williams R S 2010 Memristive' switches enable 'stateful' logic operations via material implication *Nature* **464** 873–6
[22] Gupta S, Imani M and Rosing T 2018 Felix: fast and energy-efficient logic in memory *2018 IEEE/ACM ICCAD* (IEEE) pp 1–7
[23] Deb A, Datta K, Hassan M, Shirinzadeh S and Drechsler R 2023 Automated equivalence checking method for majority based in-memory computing on ReRAM crossbars *2023 ASP-DAC* pp 19–25
[24] Bende A *et al* 2024 Experimental validation of memristor-aided logic using 1T1R TaO_x RRAM crossbar array *2024 37th Int. Conf. on VLSI Design and 2024 23rd Int. Conf. on Embedded Systems (VLSID)* pp 565–70

[25] Talati N, Gupta S, Mane P and Kvatinsky S 2016 Logic design within memristive memories using memristor-aided logic (MAGIC) *IEEE Trans. Nanotechnol.* **15** 635–50
[26] Wang Z-R, Su Y-T, Li Y, Zhou Y-X, Chu T-J, Chang K-C, Chang T-C, Tsai T-M, Sze S M and Miao X-S 2017 Functionally complete Boolean logic in 1T1R resistive random access memory *IEEE Electron Device Lett.* **38** 179–82
[27] Ding Z, Li X, Jin C, Yu X, Chen B, Cheng R and Han G 2024 Experimental demonstration of non-volatile boolean logic with field configurable 1FeFET-1RRAM technology *IEEE Electron Device Lett.* **45** 1084–7
[28] Padberg H, Regev A, Piccolboni G, Bricalli A, Molas G, Nodin J F and Kvatinsky S 2024 Experimental demonstration of non-stateful in-memory logic with 1T1R oxram valence change mechanism memristors *IEEE Trans. Circuits Syst. II Express Briefs* **71** 395–9
[29] Brackmann L, Ziegler T, Wouters D J and Menzel S 2024 Experimental verification and evaluation of non-stateful logic gates in resistive ram *IEEE Trans. Circuits Syst. I Regul. Pap.* **72** 1–10
[30] Wang Z-R, Li Y, Su Y-T, Zhou Y-X, Cheng L, Chang T-C, Xue K-H, Sze S M and Miao X-shui 2018 Efficient implementation of boolean and full-adder functions with 1T1R RRAMs for beyond von Neumann in-memory computing *IEEE Trans. Electron Devices* **65** 4659–66
[31] Wang Z, Li Y, Su Y-T, Zhou Y, Yin K, Cheng L, Chang T-C, Xue K, Sze S and Miao X 2018 Implementation of functionally complete boolean logic and 8-bit adder in CMOS compatible 1T1R RRAMs for in-memory computing *2018 IEEE Int. Memory Workshop (IMW)* pp 1–4
[32] Zhang M *et al* 2014 An overview of the switching parameter variation of RRAM *Chin. Sci. Bull.* **59** 5324–37
[33] Waser R, Dittmann R, Staikov G and Szot K 2009 Redox-based resistive switching memories – nanoionic mechanisms, prospects and challenges *Adv. Mater.* **21** 2632–63
[34] Roldán J B *et al* 2023 Variability in resistive memories *Adv. Intell. Syst.* **5** 2200338
[35] Singh S, Bende A, Jha C K, Rana V, Drechslert R, Patkar S and Merchant F 2024 In-memory mirroring: cloning without reading *2024 IFIP/IEEE 32nd Int. Conf. on Very Large Scale Integration (VLSI-SoC)* pp 1–6
[36] Singh S, Jha C K, Bende A, Rana V, Patkar S, Drechsler R and Merchant F 2024 Memspice: automated simulation and energy estimation framework for magic-based logic-in-memory *2024 29th Asia and South Pacific Design Automation Conf. (ASP-DAC)* pp 282–7
[37] Ben-Hur R, Ronen R, Haj-Ali A, Bhattacharjee D, Eliahu A, Peled N and Kvatinsky S 2020 Simpler magic: synthesis and mapping of in-memory logic executed in a single row to improve throughput *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **39** 2434–47
[38] ArC TWO | ArC Instruments — arc-instruments.co.uk 2024 (www.arc-instruments.co.uk/products/arc-two/) (Accessed 22 October 2024)
[39] Hoffer B, Rana V, Menzel S, Waser R and Kvatinsky S 2020 Experimental demonstration of memristor-aided logic (MAGIC) using valence change memory (VCM) *IEEE Trans. Electron Devices* **67** 3115–22
[40] Zidan M A, Fahmy H A H, Hussain M M and Salama K N 2013 Memristor-based memory: the sneak paths problem and solutions *Microelectron. J.* **44** 176–83
[41] Bende A, Paul G, Solfronk O, Hoffmann-Eifert S, Dittmann R and Rana V 2025 Variability reduction in RRAM through device stack engineering *MikroSystemTechnik KONGRESS 2025; Mikroelektronik/Mikrosystemtechnik und Ihre Anwendungen – Nachhaltigkeit und Technologiesouveränität* pp 478–80
[42] Paul G, Glint T, Menzel S and Rana V 2025 Bilayer ohmic electrode engineering in TaO_x ReRAM devices *IEEE Electron Device Lett.* **46** 1537–40
[43] Fantini A, Goux L, Degraeve R, Wouters D J, Raghavan N, Kar G, Belmonte A, Chen Y-Y, Govoreanu B and Jurczak M 2013 Intrinsic switching variability in HfO_2 RRAM *2013 5th IEEE Int. Memory Workshop* pp 30–33