SCIENCE CHINA
Information Sciences

• LETTER •

July 2024, Vol. 67, Iss. 7, 179403:1–179403:2
https://doi.org/10.1007/s11432-023-4037-4

Efficient implementation of majority-inverter graph logic and arithmetic functions with memristor arrays

Zhouchao GAN[1†], Dongdong ZHANG[1†], Chenyu ZHANG[1], Yinghao MA[1], Xiangshui MIAO[1,2] & Xingsheng WANG[1,2*]

[1]*School of Integrated Circuits & Wuhan National Lab. for Optoelectronics, Huazhong University of Science and Technology, Wuhan* 430074*, China;*
[2]*Hubei Yangtze Memory Laboratories, Wuhan* 430205*, China*

Received 15 December 2023/Revised 1 March 2024/Accepted 14 May 2024/Published online 24 June 2024

Memristors are characterized by several outstanding features that permit them to function both as memory and logic units, enabling logic-in-memory (LIM) computing that potentially breaks the von Neumann bottleneck. Typically, in-memory logic is categorized into stateful logic and non-stateful logic based on the physical quantities of inputs and outputs. For stateful logic, inputs and outputs are represented as the resistive states of memristors, which is advantageous for logic cascading and parallel computing. However, the limited reconfigurability of stateful logic necessitates the use of more memristors and logic operations. On the contrary, non-stateful logic employs voltage signals as inputs, providing greater flexibility for the implementation of logic primitives. However, a "cascading issue" arises, as the resistive output from a current logic operation needs to be transformed into a voltage input for the next, thereby incurring additional costs in peripheral circuitry. Majority-inverter graph (MIG), comprising three-input majority (MAJ) logic and inverter (INV) logic, has been proven to form a complete set within its data structure and outperforms traditional AND/OR/INVERTER graphs (AOIGs) during logic optimization [1]. Currently, the two primary approaches for implementing MAJ logic with memristors are categorized as stateful logic [2] and non-stateful logic [3], each presenting distinct limitations.

This study introduces a novel MIG logic mapping strategy that enables the realization of MIG logic within a memristor array by applying designated voltages from the set {0, V_{COND}, V_p, $2V_p$} at the control terminals. The proposed logic scheme integrates the advantages of stateful logic and non-stateful logic, achieving a balance between cascading capabilities and flexibility. Employing the proposed MIG logic, two economical 1-bit full adder (FA) schemes were designed and validated through experiments, which were then individually adapted for serial and parallel N-bit FA architectures. The N-bit parallel FA demonstrated a significant improvement in speed efficiency, with a 4-bit parallel adder and a 4-bit Wallace tree multiplier instance being verified experimentally. The power consumption and robustness of logic operations are further analyzed. This study lays a solid foundation for the future development of high-performance LIM systems based on memristors.

Device characteristics. Bipolar memristors transition from a high resistance state (HRS) to a low resistance state (LRS) when a forward voltage exceeds the threshold voltage V_{set}, known as the set process. Conversely, a reverse resistance transition occurs when the applied voltage falls below the threshold voltage V_{reset} for the reset process. After initializing the device at a forming voltage of 2.5 V, 100 consecutive DC cycles were performed, indicating bipolar switching of the Ti/HfO_2/TiN memristor device, as shown in Figure 1(a). Figure 1(b) illustrates that the memristor can endure 10^8 cycles under set (1 V, 50 ns) and reset (−1.7 V, 50 ns) pulse operations while maintaining a resistance state window of about ten times. The fabrication steps, resistance switching speed, retention, and resistance distribution characteristics of the devices are shown in Appendix B.1. The devices exhibit rapid resistive switching, a sufficiently large window, and robust endurance, making them suitable for executing logic operations.

Proposed MIG logic scheme. The MIG logic consists of three-input MAJ gates and inverters, together representing a complete logic family. The expressions for the MAJ gate and INV logic are as follows:

$$M_3(a,b,c) = a \cdot b + b \cdot c + a \cdot c, \tag{1}$$

$$INV(a) = \bar{a}. \tag{2}$$

In Figure 1(c), an MAJ gate configuration is depicted, which employs three memristors and a serial resistor. The two inputs of the MAJ gate are the resistive states of memristors M_1 and M_2, while the third input, denoted by "V", is linked to the operating voltage at the terminal of the memristor. The output is represented by the resistive state of memristor M_3, as demonstrated in Figure 1(e). Figure 1(g) describes the variable mapping relationship, where the HRS and LRS of the memristor correspond to the logical "0" and "1", respectively. The voltage variable V takes on the values 0 and V_p when the logic input is "0" and "1", respectively. The voltage at control terminal T_4 is $2V_p$-V, so depending on

* Corresponding author (email: xswang@hust.edu.cn)
† Gan Z C and Zhang D D have the same contribution to this work.

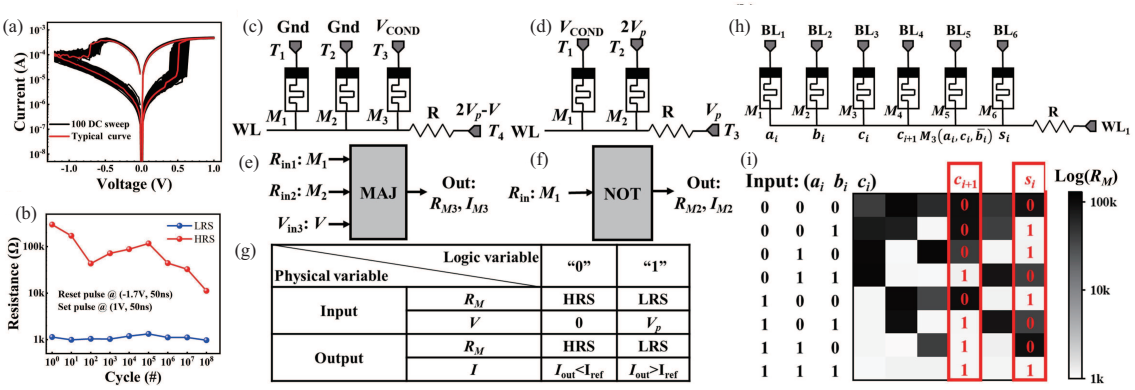

Figure 1 (Color online) (a) Measured resistive switching behavior in DC IV sweeping mode; (b) more than 10^8 cycling endurance under voltage pulses; the proposed schematic of the memristor-based (c) three-input MAJ logic and (d) INV logic circuit; the instance diagram of (e) the MAJ logic gate and (f) INV gate showing the port configuration; (g) mapping relationship between physical and logical variables; (h) scheme 2 for1-bit full adder; (i) experimental results for eight logic input conditions.

the logic input "0" or "1", the port voltage is selected from the set $\{2V_p, V_p\}$. The MAJ logic is conducted through one write step and one logic operation step:

Step 1. Write the information of the two input variables into memristors M_1 and M_2, initializing output memristor M_3 to a HRS;

Step 2. Apply corresponding operation voltage pulses to control terminals T_1, T_2, T_3, T_4 (with T_1 and T_2 connected to Gnd, T_3 connected to a fixed voltage $V_{\rm COND}$, and the voltage T_4 is determined by the third input variable of the MAJ gate).

The INV logic is implemented in the structure depicted in Figure 1(d), where M_1 serves as the input for the logic, and M_2 is the output. The port configuration for this INV logic is presented in Figure 1(f). Similarly, the INV logic can be executed in two steps:

Step 1. Write the input information into memristor M_1, initializing the output memristor M_2 to an HRS;

Step 2. Operation voltages are imposed on control terminals T_1, T_2, and T_3, which are set to the fixed voltages $V_{\rm COND}$, $2V_p$, and V_p, respectively.

Through the above voltage operations, the logic results are stored in-situ in the memristors. Appendix B.2 describes the logical experimental results.

Realization of full adders and multipliers. Adders are fundamental arithmetic operators in digital systems, forming the basis of complex arithmetic functions. For a 1-bit FA, there are three inputs (augend a_i, addend b_i, and carry-in c_i) and two outputs (sum s_i and carry-out c_{i+1}). The results can be represented iteratively using MAJ and INV logic as follows:

$$c_{i+1} = M_3(a_i, c_i, b_i), \tag{3}$$

$$s_i = M_3(\overline{M_3(a_i, c_i, b_i)}, M_3(a_i, c_i, \overline{b_i}), b_i). \tag{4}$$

Thanks to the flexibility of logic, two economical 1-bit FA schemes are designed to pursue area and speed efficiency, respectively. Scheme 2 for 1-bit FA only requires 6 memristors and 4 steps, as depicted in Figure 1(h), with experimental results presented in Figure 1(i). The execution details of the two FA schemes can be found in Appendix C.1. Based on the above 1-bit FA designs, the implementation costs for the N-bit FA were derived, demonstrating significant performance improvements, detailed in Appendix C.2. A case of 4-bit Wallace tree multiplication is given in Appendix C.3, which shows the efficiency of the proposed logic design. Further analyses of the power consumption of addition operations and the impact of non-ideal factors of memristors on logic operations are also discussed in Appendix D.

Conclusion. In this study, a novel MIG logic implementation scheme based on memristors is proposed. The proposed logic scheme combines the advantages of stateful and non-stateful logic, ensuring direct deployment into memristor arrays and providing superior efficiency in logic cascades. On this basis, the serial and parallel implementation methods for N-bit full adders were studied, achieving the best performance compared with previous work. The experimental validation for a 4-bit parallel adder and Wallace tree multiplier demonstrates significant improvements in latency, which suggests the potential of the logic scheme for efficiently realizing complex arithmetic tasks. The power consumption performance of the proposed logic in 32-bit addition operations was evaluated, showing minimal power consumption overhead compared with IMPLY logic. The robustness of the logic operation to the memory window and threshold voltage is analyzed to guarantee the correct execution of the logic. The presented logic design and arithmetic function demonstration constitute a substantial step toward the realization of an efficient and practical memristor-based LIM computing system.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 62274070, U2341221), Hubei Province Science and Technology Major Project (Grant No. 2022AEA001), Huazhong University of Science and Technology (Grant Nos. 2023JCYJ042, 2019kfyXJJS048), and Hubei Key Laboratory of Advanced Memories.

Supporting information Appendixes A–D. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Amaru L, Gaillardon P E, de Micheli G. Majority-inverter graph: a new paradigm for logic optimization. IEEE Trans Comput-Aided Des Integr Circuits Syst, 2016, 35: 806–819
2 Reuben J. Binary addition in resistance switching memory array by sensing majority. Micromachines, 2020, 11: 496
3 Gaillardon P-E, Amarú L, Siemon A, et al. The Programmable logic-in-memory (PLiM) computer. In: Proceedings of Design, Automation & Test in Europe Conference & Exhibition (DATE), 2016. 427–432