Processing within a Memristive Memory

Rotem Ben-Hur and Shahar Kvatinsky

Electrical Engineering Department, Technion – Israel Institute of Technology, Haifa, Israel 3200003

Memory accesses are the bottleneck of modern computer systems both in terms of performance and energy. This barrier, known as "the Memory Wall", can be break by utilizing memristors. Memristors are novel passive electrical components with varying resistance based on the charge passing through the device [1]. In this abstract, the term "memristor" covers also an extension of the definition, memristive devices, which vary their resistance depending on a state variable [2]. While memristors are naturally used as memory cells, they can also be used for other applications, such as logic circuits [3].

We present a novel architecture that redefines the relationship between the memory and the processor by enabling data processing within the memory itself. Our architecture is based on a memristive memory array, in which we perform two basic logic operations: Imply (material implication) [4] and False. These operations create a complete logic structure, thus enabling any desired processing of data within the memory, such as image processing, as shown in Figure 1.

Data-processing as well as standard memory tasks are managed and performed by a designated memory controller. This sophisticated controller operates according to instructions originated in the processor, using a pre-defined instruction set. The controller translates logic tasks to voltage values that are applied on the desired memristors to changes their resistances.

We present a memory controller that is able to perform any logic and arithmetic operation inside the memory, regardless of the size, location and orientation of the data. This concept, implemented and simulated in ModelSim environment, dramatically improves performance and energy by significantly reducing the memory access necessity, especially in intensive applications, such as image processing.

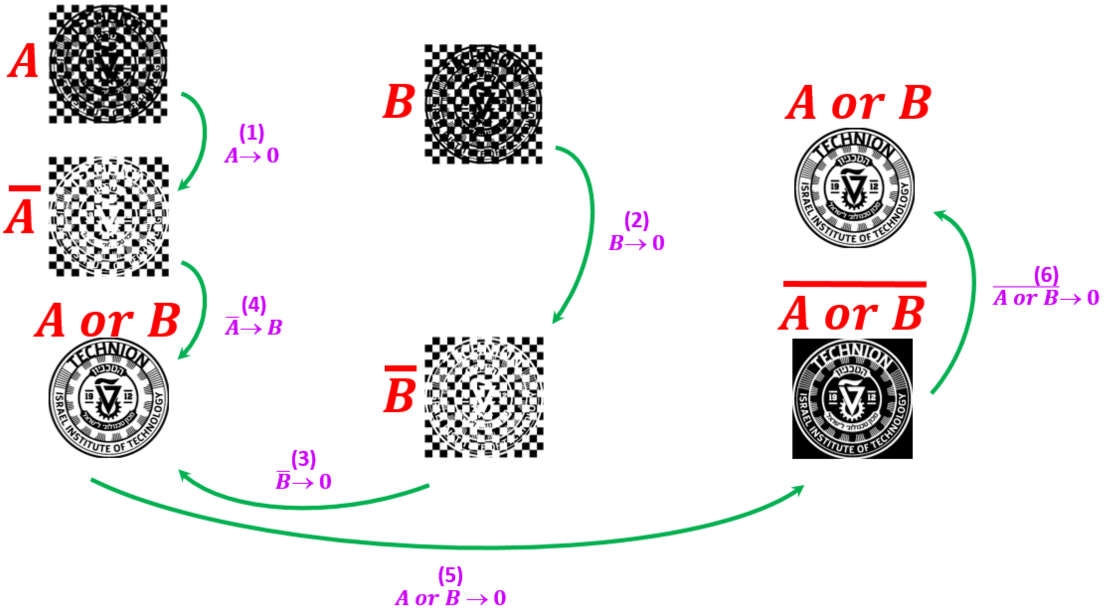

Figure 1. Performing OR operation of MxN images within a memristive memory: $\boldsymbol{A \text{ or } B} = (\boldsymbol{A} \to \boldsymbol{0}) \to \boldsymbol{B}$

(1) $\boldsymbol{A} \to \boldsymbol{0} = \boldsymbol{A'}$: Takes M cycles and MxN operations.

(2) $\boldsymbol{B} \to \boldsymbol{0} = \boldsymbol{B'}$: Takes M cycles and MxN operations.

(3) $\boldsymbol{\bar{B}} \to \boldsymbol{0} = \boldsymbol{B}$: Takes N cycles and MxN operations.

(4) $\boldsymbol{\bar{A}} \to \boldsymbol{B} = \boldsymbol{A \text{ or } B}$: Takes M cycles and MxN operations.

(5) $\boldsymbol{A \text{ or } B} \to \boldsymbol{0} = \overline{\boldsymbol{A \text{ or } B}}$: Takes N cycles and MxN operations.

(6) $\overline{\boldsymbol{A \text{ or } B}} \to \boldsymbol{0} = \boldsymbol{A \text{ or } B}$: Takes M cycles and MxN operations.

Takes a total of: 4M+2N cycles, 6MxN operations

REFERENCES

[1] L. O. Chua, “Memristor—The missing circuit element,” IEEE Trans. Circuit Theory, vol. 18, no. 5, pp. 507–519, Sep. 1971.

[2] L. O. Chua and S. M. Kang, “Memristive devices and systems,” Proc. IEEE, vol. 64, no. 2, pp. 209–223, Feb. 1976.

[3] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, “Memristor-based material implication (IMPLY) logic: Design principles and methodologies,” IEEE Trans. VLSI, vol. 22, no. 10, pp. 2054–2066, Oct. 2014.

[4] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, "Memristive Switches Enable 'Stateful' Logic Operations via Material Implication," Nature, Vol. 464, pp. 873-876, April 2010