Performing Memristor-Aided Logic (MAGIC) using STT-MRAM

Jeffry Louis*, Barak Hoffer, and Shahar Kvatinsky
Viterbi Faculty of Electrical Engineering, Technion – Israel Institute of Technology, Haifa, Israel
*Also Birla Institute of Technology and Science, Pilani, India

***Abstract*—Modern computers suffer from a limited data transfer rate between the memory and the processing units. One of the attractive potential solutions to overcome this bottleneck is to combine processing and memory by performing computing in the same location where the data is stored. Processing-in-memory (PIM) has been demonstrated by memristor-aided logic (MAGIC) operations using resistive random access memory (RRAM) memristive devices within crossbar arrays. Nevertheless, RRAM devices are relatively slow and suffer from limited endurance. Spin-transfer torque magnetoresistive random access memory (STT-MRAM) is another memristive technology, which is faster and has practically unlimited endurance and is therefore considered to be an attractive technology for cache-level memories. In this paper, we demonstrate MAGIC operations within an STT-MRAM array by supplying voltages suitable to achieve the current required for MRAM device switching. The proposed circuit was evaluated in SPICE simulations with the GlobalFoundries 22nm CMOS-MRAM process, including Monte Carlo simulations to verify the proposed design in the presence of process variation and device mismatch. The circuit showed more than 90% chance of functioning for the {1 1} and {0 0} input cases while it was about 85% correct for the {0 1} case.**

Keywords—STT-MRAM; MAGIC; memristor; Processing-in-memory; mMPU

I. INTRODUCTION

The von Neumann bottleneck refers to the difference in the speeds of operation of the processor and the memory, which limits the performance of a computing system [1]. This has serious repercussions in data-centric applications prevalent today. Several methods have been devised to deal with this bottleneck and improve the performance of modern computer architectures. One approach builds application-targeted architectures (known as accelerators) such as Google's Tensor Processing Unit, IBM's TrueNorth and Intel's Loihi [2], [3], which provide better performance for specific applications but are essentially still limited by the von Neumann bottleneck.

Another attractive means of overcoming this bottleneck involves device-level innovations, such as monolithic 3D integration that places memory above or below the processing unit, *e.g.*, Intel's Teraflops Research Chip (Polaris), 3D-MAPS and N3XT architectures [4], [5]. However, these approaches do not fully eliminate the von Neumann bottleneck.

Clearly, a better solution is the processing-in-memory (PIM) approach, which processes data in the memory itself. While performing PIM with conventional memory technologies is difficult, emerging memristive memory technologies enable efficient PIM. In memristive memory, the data is stored within the memristors in the form of resistance. The logical operations are performed by connecting structurally distinct memristors arrays to form different logic gates. As this method involves no data movement, the von Neumann bottleneck is absent.

Memristor-aided logic (MAGIC) is a PIM technique that implements voltage-controlled logic gates on memristors using the stored data as the logical states [6], [7]. So far, MAGIC has only been demonstrated on a voltage-controlled resistive RAM (RRAM) [8]. While the primary advantage of the RRAM is its density, it suffers from limited endurance and relatively low speed, which limits the benefits provided by MAGIC. STT-MRAM is another emerging memristive technology with eight orders of magnitude higher endurance [9] and faster switching time as compared to RRAM [10], [11]. Hence, STT-MRAM appears to be better suited for PIM systems with small memories.

To date, the scientific literature has only reported on logic gates on STT-MRAM using an externally applied magnetic field and current signals for switching [12]–[15]. In this paper, we demonstrate MAGIC gates on STT-MRAM using voltage signals. The results were generated using realistic device models from the CMOS-MRAM GlobalFoundries 22nm process. We demonstrate and evaluate the MAGIC gates within a 1T-1MTJ (MRAM memristor) memory array, including Monte Carlo simulations.

II. MAGIC USING STT-MRAM

A. MTJ Device

The memristive device used in an STT-MRAM is the magnetic tunnel junction (MTJ), which is composed of a fixed polarization layer and a free layer, which are separated by an insulating MgO layer, as shown in Fig. 1(a) [16]. If the magnetization of both layers is parallel to each other, a high current is obtained upon application of a voltage across the device. This low resistive state (LRS), also known as the parallel resistance state (R_p), is due to the tunneling of electrons through the MgO layer with reduced scattering. Anti-parallel magnetization of the layers leads to a lower current, due to a high resistance state (HRS), also known as the anti-parallel resistance state (R_{ap}). The spin-transfer torque (STT) effect is responsible for switching the magnetization of the free layer of the MTJ upon application of an electric current [14]. If the applied current is greater than the current threshold (I_c), then the magnetization is altered, which switches the resistance state.

The parameters of the MTJ were taken from GlobalFoundries 22nm process (Table I). To switch the device from HRS to LRS, the current must exceed I_{set}, and its direction must be from the free layer of the device to the fixed layer. Similarly, currents that exceed I_{reset} in the opposite direction switch the MTJ from LRS to HRS.

This research is partially supported by the ERC under the European Union's Horizon 2020 Research and Innovation Programme (grant agreement no. 757259), and by the Israel Science Foundation grant no. 1514/17.

TABLE I. PARAMETERS OF THE MTJ

Parameter	Value
R_p	2.8 KOhms
R_{ap}	6.2 KOhms
I_{set}	91 µA
I_{reset}	134 µA

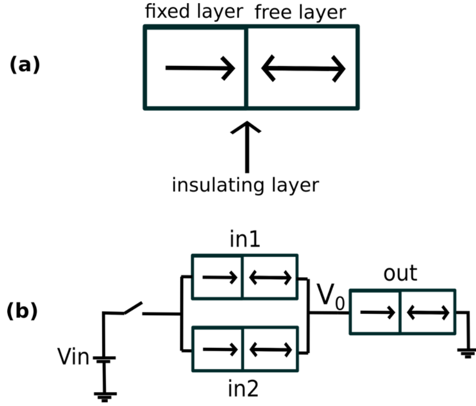

Fig. 1. (a) Schematic of a magnetic tunnel junction (MTJ) device, which consists of fixed and free layers separated by an isulating MgO layer. (b) Schematic of a MAGIC NOR gate using STT-MRAM. Three MTJ/STT-MRAM memristive devices are used, two as inputs (*in1*, *in2*) and a single output (*out*).

TABLE II. SWITCHING REQUIREMENTS FOR A NOR GATE

in1	*in2*	*out*
HRS (0)	HRS (0)	LRS (1)
HRS (0)	LRS (1)	HRS (0)
LRS (1)	HRS (0)	HRS (0)
LRS (1)	LRS (1)	HRS (0)

B. MAGIC NOR using MTJs

The two-input MAGIC NOR gate consists of two input MTJ memristive devices (*in1, in2*) connected in parallel and an additional MTJ memristor connected as an output (*out*). In the first step, the output memristor is initialized to the LRS, and then a voltage V_{in} is applied to the circuit. This configuration is shown in Fig 1(b).

The switching of the output MTJ depends on the input values as listed in Table II. In the case where both inputs are logical 0, both input MTJs (*in1* and *in2*) are in HRS, and, therefore, when applying V_{in}, the voltage across the output MTJ (*out*) will be too low to switch, and will remain LRS. In all the other input combinations (11, 10, and 01), at least one input MTJ is in LRS; when applying V_{in}, the voltage across the output MTJ is sufficiently large to switch its state to HRS.

C. 1T – 1MTJ Crossbar Array

A previous work by Zabihi *et al.* [12] proposed to use an additional transistor in a 2T-1MTJ array to support logic and memory operations. In the 2T-1MTJ structure, one transistor is

TABLE III. MTJ- BASED NOR GATE

Input	Output	Current in *out* during switching [µA]	Resistance of *out* [KOhms]
0 0	1	111	2.79
0 1	0	134.64	5.8
1 0	0	134.64	5.8
1 1	0	134.69	5.7

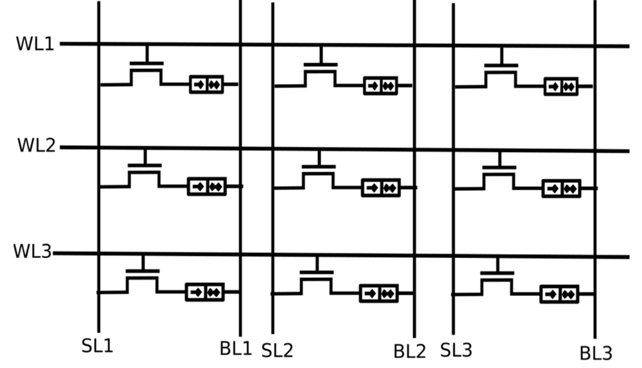

Fig. 2. Schematic of the standard 1T-1MTJ memory array.

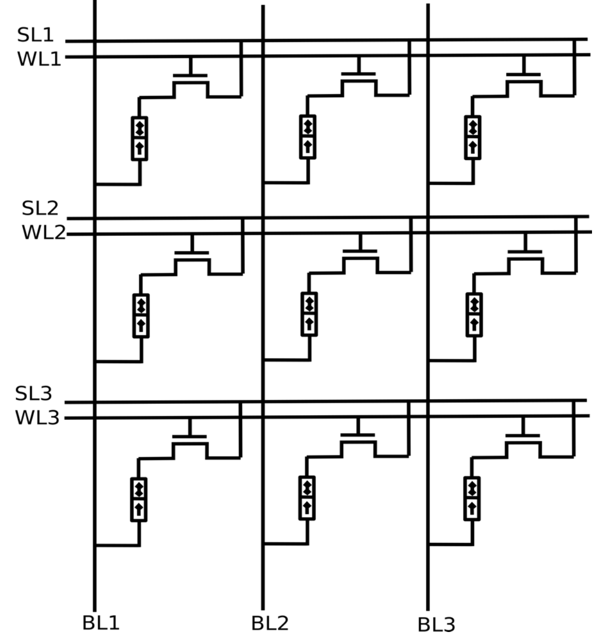

Fig. 3. Schematic of the proposed 1T-1MTJ memory array.

used for writing and reading data, including writing multiple values along the row at the same time, while the other transistor is connected to the logic line, which connects MTJs along the row to enable logic operation. Due to the area overhead in this configuration compared to the 1T-1MTJ array, we modified the 1T-1MTJ structure to support logic operations in memory.

We propose an STT-MRAM memory array using one transistor and one MTJ (1T-1MTJ) memory cell, as shown in Fig. 3. This is not the standard array structure (shown in Fig. 2) [17], as MAGIC operations could not be performed in such arrays. In the standard array, select lines (SL) are shared between MTJs of the same column, which means that MAGIC operations

cannot be performed along rows of the standard array structure, since there is no common line between the MTJs. MAGIC operations across columns are also impossible in the standard array since the input and output MTJs are connected between the same lines (both *Select Line* and *Bit Line* are shared). Hence, SLs were defined as shared lines along the row to enable MAGIC operations, as shown in Fig. 4. The common voltage at the drain of all three transistors is V_c, and V_o is the voltage across the output MTJ, and now depends on the voltage across the source-drain of the transistor as well.

The memory and logic operations are performed within the array itself. For writing data, the WL (*Word Line*) is pulled high and the voltage corresponding to the bit value is applied across the BL (*Bit Line*) and SL (*Select Line*). Multiple bits can be written along a row at the same time by grounding the SL, and applying positive or negative voltages in each BL, according to the required logical value. For reading data, WL is pulled high and a voltage lower than the writing voltage is applied across the BL and SL. The currents obtained through the BLs can be measured to read multiple bits along a row.

To perform logic operations, WL is pulled high and the MTJs are connected by the SL in each row (the SL is floating). To realize logic along the row, appropriate voltages are applied to the BLs, as illustrated in Fig. 4. If V_o is sufficiently high to generate a current greater than I_c, then the resistance of the output MTJ is switched.

III. ANANLYSIS AND EVALUATION

A. MTJ MAGIC NOR Gate Operation

The original work on MAGIC gates relied on voltage-controlled memristors [6]. For proper logic operation in such devices, the circuit must fulfill the following condition:

$$\frac{LRS}{LRS+HRS||LRS} V_{in} > V_{th}, \tag{1}$$

where V_{th} is the threshold voltage of the memristor. However, MTJs are current-controlled devices through the STT mechanism. Hence, different circuit conditions must be determined, to ensure proper logic behavior. The value of V_{th} was obtained by multiplying the current threshold I_{reset} with LRS. From Table I and (1), we obtained that the equivalent memristor threshold V_{th} should be -0.375V, and select V_{in} to be -0.65V in our design. V_{th} and V_{in} are negative due to the polarity of the current. I_{reset} flows from the fixed layer to the free layer of the *out* MTJ. Table III summarizes the results of a single NOR gate simulated in SPICE for the MTJ parameters in Table I. From Table III, we can see that when the current is higher than the current threshold I_{reset}, the output MTJ switches to HRS.

B. 1T-1MTJ MAGIC NOR Evaluation

In this sub-section, we derive the gate voltages needed to perform MAGIC NOR using STT-MRAM memory arrays. The main constraint faced when deriving the gate voltages, is the W/L ratio of the transistor. To enable a higher voltage drop across the MTJ, the transistor should have a lower resistance value. The transistor resistance can be decreased by increasing the W/L ratio, which is the same throughout the array. Another constraint is that the gate voltage is identical for all input cases.

To compensate for the voltage drop caused by the resistance of the transistors, we fixed V_{in} at -1V for this array, which is higher than the V_{in} used in the previous section. V_{WL}, the gate voltage of the transistors, was set at 2V. As an example,

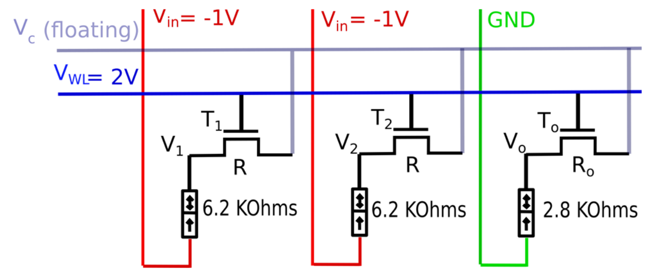

Fig. 5. Equivalent schematic of the {0 0} input case.

TABLE IV. SWITCHING CURRENT FOR MAGIC NOR GATE WITHIN A 1T-1MTJ ARRAY

Input	Output	Current in *out* during switching [μA]	*Out* Resistance [KOhms]
0 0	1	127	2.8
0 1	0	134.55	6.5
1 1	0	134.72	6.6

we show the voltages applied at the {0 0} input combination in Fig. 5. Transistors T_1, T_2 and T_o are required to operate in the linear region as they offer higher resistance in the saturation region. In the following analysis, we show that the transistors are in the linear region for the chosen V_{WL}. The conditions for the linear region for T_1, T_2 and T_o are

$$V_{WL} - V_1 - V_{th_{trans}} > V_c - V_1, \tag{2}$$

$$V_{WL} - V_2 - V_{th_{trans}} > V_c - V_2, \tag{3}$$

$$V_{WL} - V_c - V_{th_{trans}} > V_o - V_c. \tag{4}$$

From (2) and (3), we obtain:

$$V_{WL} - V_{th_{trans}} > V_c. \tag{5}$$

From (5), it can be concluded that T_1 and T_2 will always be in the linear region as V_c is negative and $V_{WL} - V_{th_{trans}}$ is 1.5V, since $V_{th_{trans}}$ is 0.5 V. From (4), we obtain:

$$V_{WL} - V_{th_{trans}} > V_o\ . \tag{6}$$

Since V_o is always negative, (6) is true and T_o is also in the linear region.

This circuit was simulated for all input cases and the results, listed in Table IV, demonstrate correct execution of the NOR gate within the 1T-1MTJ memory array.

C. Monte Carlo Simulations

To verify the functionality of the 1T-1MTJ logic gate in practical devices with process variation, we performed Monte Carlo simulations using statistical models from the Global Foundries 22nm process. Each input pattern was simulated 500 times and in each simulation, the device diameter, current density threshold and RA (resistance area product) were deviated in a Gaussian distribution with standard deviations that were 0.03 times their mean values.

Table V presents the Monte Carlo results. The last column, *Pe*, states the percentage of error in the operation of the logic gate for each input case. *Pe* was higher for the {0 1} case due

TABLE V. MONTE CARLO SIMULATION RESULTS

Input	Switching from 1 to 0	Not switching from 1 to 0	*Error Pe*
0 0	28	472	5.6%
0 1	423	77	15.4%
1 1	494	6	1.2%

to a lower V_0 than the {1 1} case, which enabled switching from LRS to HRS.

As seen in Table V, the error rate is low but not negligible. When performing logic operations, this rate must be considered zero, in order to receive reliable outputs. In future work, we will further develop this model to achieve this goal.

IV. CONCLUSIONS

In this paper, we have demonstrated that MAGIC gates could be implemented using STT-MRAM within 1T-1MTJ memory arrays. Furthermore, we demonstrated the vulnerability of their functionality to process variation and device mismatches. In future work, we plan to investigate how to improve their robustness to process variation and evaluate the dynamics of the gates in the presence of parasitic effects, such as device and line capacitances.

REFERENCES

[1] P. Kogge *et al.*, “ExaScale Computing Study : Technology Challenges in Achieving Exascale Systems,” *Gov. Procure.*, 2008.

[2] N. P. Jouppi *et al.*, “In-Datacenter Performance Analysis of a Tensor Processing Unit,” *44th Int. Symp. Comput. Archit.*, pp. 1–17, 2017.

[3] P. Merolla *et al.*, “A Million Spiking-Neuron Integrated Circuit with a Scalable Communication Network and Interface,” *Science*, Vol. 41, pp.1–5, Aug. 2009.

[4] D. H. Kim *et al.*, "Design and Analysis of 3D-MAPS (3D Massively Parallel Processor with Stacked Memory)," in *IEEE Transactions on Computers*, Vol. 64, No. 1, pp. 112-125, Jan. 2015.

[5] M. M. Shulaker *et al.*, “Three-dimensional Integration of Nanotechnologies for Computing and Data Storage on a Single Chip,” *Nature*, Vol. 547, No. 7661, pp. 74–78, July 2017.

[6] S. Kvatinsky *et al.*, "MAGIC—Memristor-Aided Logic," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 61, No. 11, pp. 895-899, Nov. 2014.

[7] N. Talati, S. Gupta, P. Mane and S. Kvatinsky, "Logic Design Within Memristive Memories Using Memristor-Aided loGIC (MAGIC)," in *IEEE Transactions on Nanotechnology*, Vol. 15, No. 4, pp. 635-650, July 2016.

[8] H. Bae *et al.*, “Functional Circuitry on Commercial Fabric via Textile-Compatible Nanoscale Film Coating Process for Fibertronics,” *Nano Letters*, Vol. 17, No. 10, pp. 6443–6452, Oct. 2017.

[9] J.-P. Wang *et al.*, “A Pathway to Enable Exponential Scaling for the Beyond-CMOS Era,” *Proceedings of the 54th Annual Design Automation Conference*, pp. 1–6, June 2017.

[10] R. Sbiaa and S. N. Piramanayagam, “Recent Developments in Spin Transfer Torque MRAM,” *Physica Status Solidi - Rapid Research Letters*, Vol. 11, No. 12, pp. 1–8, Dec. 2017.

[11] H. Y. Lee *et al.*, “Low Power and High Speed Bipolar Switching with A Thin Reactive Ti Buffer Layer in Robust HfO2 Based RRAM ,” *2008 IEEE International Electron Devices Meeting*, pp. 1–4, Dec. 2008.

[12] M. Zabihi *et al.*, "In-Memory Processing on the Spintronic CRAM: From Hardware Design to Application Mapping," in *IEEE Transactions on Computers*, Vol. 68, No. 8, pp. 1159-1173, Aug. 2019.

[13] Z. I. Chowdhury *et al.*, “Computational RAM to Accelerate String Matching at Scale,” *arXiv Preprint*, 1812.08918

[14] A. Lyle *et al.*, “Direct Communication Between Magnetic Tunnel Junctions for Nonvolatile Logic Fan-out Architecture,” *Applied Physics Letters*, Vol. 97, No. 15, pp. 2008–2011, Oct. 2010.

[15] S. Patil *et al.*, “Spintronic Logic Gates for Spintronic Data Using Magnetic Tunnel Junctions,” *2010 IEEE International Conference on Computer Design* , pp. 125–131, Oct. 2010.

[16] J. Kim *et al.*, "Spin-Based Computing: Device Concepts, Current Status, and a Case Study on a High-Performance Microprocessor," in *Proceedings of the IEEE*, Vol. 103, No. 1, pp. 106-130, Jan. 2015.

[17] H. Zhang, W. Kang, and T. Pang, “Dual Reference Sensing Scheme with Triple Steady States for Deeply Scaled STT-MRAM,” *2016 IEEE/ACM International Symposium on Nanoscale Architecture*, pp. 1–6, July 2016.