NML-Logic-Based ALU

Vitor C. de Lima*, Thiago R. B. S. Soares*, Luiz G. C. Melo†, Omar P. Vilela Neto*
*Department of Computer Science
Federal University of Minas Gerais, Belo Horizonte, Minas Gerais
Email: [vitorcezar, thiagorbss, omar]@dcc.ufmg.br
†Department of Electrical Engineering
Federal University of Minas Gerais, Belo Horizonte, Minas Gerais
Email: lgmelo@ufmg.br

***Abstract*—The CMOS technology is reaching its size and energy limits while new technologies try to overcome these problems. Nanomagnetic Logic (NML) is one of these technologies where logic operations are performed through dipolar magnetostatic interactions between nanosized magnetic specimens at room temperature. The low switching energy involved in the operation of NML circuits, in addition to the possibility of higher integration density are remarkable advantages of this technology over CMOS-based systems. For the development of such technology the design of complex circuits must be addressed. In this paper we propose an one bit NML-based ALU. The ALU implements the logic operations AND, OR, XOR and their inverses on two data inputs. To the best of our knowledge, this is the first ALU designed in NML. A recently proposed NML simulator was improved to test the ALU.**

***Index Terms*—NML, ALU, Nanocomputing, Nanomagnet, Simulation**

I. INTRODUCTION

Gordon E. Moore predicted that the number of transistors would double every two years, which has been known as Moore's Law [1]. His prediction proved accurate for several years due the transistor miniaturization, increasing the devices density and clock rate. However, this growth is close to its limits and problems regarding the high energy leakage and power consumption in the scaling of CMOS caused researches along with the industry to seek for alternatives [2]. One of the most promising is Nanomagnetic Logic (NML), which consists of bistable magnets locally connected through field effect forces [3]. NML is a nanoscale technology with ultralow power consumption, and promising high clock rate [4].

The idea of NML circuits is to use the magnetic ”stray” field produced by the nanomagnets to change the magnetic polarization of its neighbors. This influence is governed by a long-range magnetostatic coupling, which depends on the magnetization direction and on the relative distance between the magnetic particles. The dependence on the relative position of the nanomagnets of a given circuit makes the design of new logical devices a nontrivial task and also leads to a topology which may be quite different from those of the traditional CMOS circuits. In order to implement the desired logic circuit, each magnet must be placed in such a way that it leverages the interaction with its neighboring magnets, thus creating the desired logical behavior.

An ALU (*Arithmetic and Logic Unit*) is a circuit that performs logic operations and it is used in many architectures due its importance on computing applications. The ALU proposed here has two data inputs and implements the logic operations AND, OR, XOR, NAND, NOR and XNOR. Three control bits are used to select which logic function will propagate to the output magnet. The ALU was designed using an improved version of the recently proposed NML simulator [5].

The rest of this paper is divided as follows: section II discusses how NML circuits work. Section III presents the ALU implementation and the results. Finally, section IV concludes this work.

II. NANOMAGNETIC LOGIC (NML)

Here is discussed how the logic in NML works. First we show how a bit is represented and how the coupling between nanomagnets occurs. Also, the functioning of a majority gate is shown, which is one the most important gates in NML. Finally, the clock issue and how it works are examined.

A. Nanomagnetic Devices

In this work, we assume a nanosized rectangular-shaped nanomagnet, with single-domain behavior, as the basic NML device (other geometries are also possible [6]). Here the magnetization of these nanomagnets are assumed uniform and are mathematically represented by a vector of constant amplitude and variable angle. In NML circuits, the magnetization direction of a nanomagnet represents a bit of information.

The magnetic polarization of an isolated elongated nanomagnet is likely to lie along its longer axis, in order to minimize the shape energy. This energy is degenerated, yielding the magnetization vector to point in any of the two possible directions. We then may arbitrarily define the logical values 1 and 0 when magnetization points ”up” and ”down”, respectively, as shown in Fig. 1 A). In contrast, under the influence of a strong external magnetic field applied along the shortest side of the magnet, the magnetization is brought to a unstable energy level, which is associated to the logic ”null” state. The coupling between two rectangular-shaped magnets can present either an antiferromagnetic or a ferromagnetic symmetry, as shown in Fig. 1 B). The alignment of the magnetic polarization is antiparallel in the former and it is parallel in the latter.

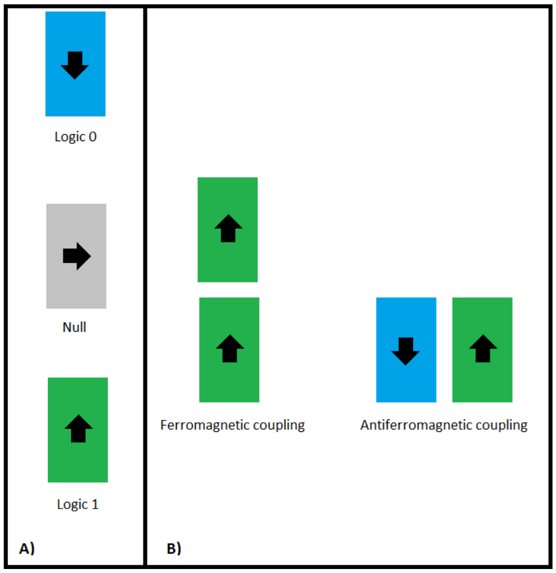
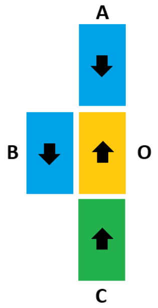


Fig. 1: A) Logic representation. B) Coupling between magnets.



A	B	C	O
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Fig. 2: Majority gate and its truth table: the output magnet O has the most influenced magnetization of its neighbors.

A wire in NML is represented by an alignment of nanomagnets following the same coupling pattern shown in Fig. 1 B), thus is possible to create a ferromagnetic or an antiferromagnetic wire. For the antiferromagnetic wires, an even number of particles yield an inverter wire circuit.

One of the most important gates in NML is the Majority Gate (MG), Fig. 2. Magnets A, B and C are the inputs and magnet O is the output. Magnetic coupling between A and O and between C and O will force the latter to polarize ferromagnetically, while the influence from input particle B over O favors antiferromagnetic coupling. The MG takes three inputs as polarized nanomagnets and retrieves the polarization of the majority of these inputs $(MAJ(A, \neg B, C)$ [7]. By ascribing A or C to a fixed value of 0 or 1, the MG can work as an ”AND” or ”OR” gate, respectively.

B. Clocking in NML

The clock in NML circuits have three purposes: to yield an adiabatic change of magnetization, to avoid signal error in long arrays of magnets and to ensure signal synchronization. In the following, the clocking mechanism is explained.

In NML circuits, the clock is given by an external magnetic field to control the magnetization of a set of magnets. The magnetic field is applied perpendicularly to the long axis of the magnet in order to force the magnet into a ”null” magnetization, as Fig. 1 A) shows. This is an unstable energy state, which is likely to change to a stable, local minimum state. As a consequence, when the external field is removed, the magnetization of the magnet shall to switch towards a vertical direction, corresponding to a ground state. Finally, the magnetization of a given nanomagnet will chose to point up or down depending on the polarization of its neighbors. Thus, the magnet is first driven into an unstable state, then the clock field is removed and the new magnetic polarization is set according to the magnetization of its neighbors.

For NML circuits composed by few magnets, the coupling pattern of Fig. 1 B) are always reproduced. However, the level of ordering in large array of magnets is likely to present an error due to the influence of non-nearest neighbor coupling. In large array of nanomagnets, the magnetic field is necessary to ensure ordering and to minimize errors, as well as to allow control over magnetization of individual particles. Signal propagation and synchronization [8] are achieved by splitting the circuit into groups called clock zones, and by submitting them to different magnetic fields (clock signals).

The clock mechanism adopted here is based on three phases [5]: RESET, SWITCH and HOLD. On the first phase (RESET) the magnetic field is applied, therefore the magnets in that phase are in the null state. In the SWITCH phase, the magnetic field is removed and then the null-state magnets have their values defined by their neighboring magnets. Finally, on the HOLD phase, the magnets magnetization is stable and can influence the next magnets. The clocking is possible by breaking the circuit in clock zones and each one is set in a different phase by the clock signals. A sequence of bits can be propagated one after another when the zones change their phases.

III. Methodology and Results

In this section we review the tool used in this work, explain the proposed ALU and show the NML-based ALU with its simulation waveform.

A. NML Simulator

Nowadays, there are a few ways to design and simulate NML circuits. The most accurate way is by low level micromagnetic simulators, such as OOMMF [9]. Although very precise, simulating circuits with a few dozens of magnets in OOMMF is computationally expensive, being more suitable for the study of small magnetic circuits. Another way to simulate NML circuits is the ToPoliNano tool (Torino Politecnico Nanotechnology tool) [10]. The circuit is described in VHDL and synthesized using predetermined logic gates, then a simulation algorithm based on a behavioral model simulate the circuit. Despite these advantages, the tool does

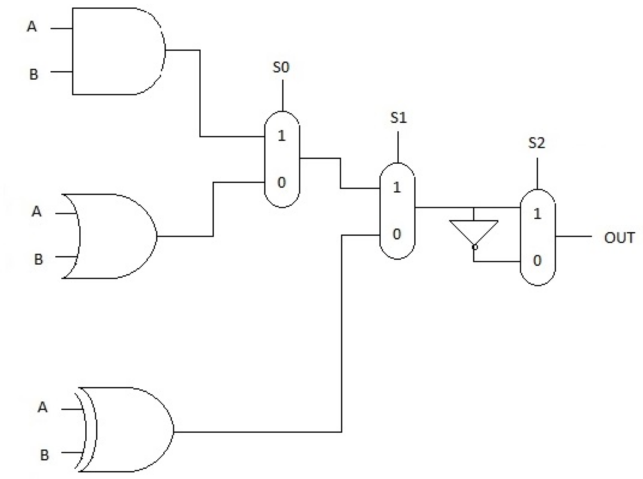


Fig. 3: ALU schematic: A and B are the inputs, OUT is the output and S0, S1 and S2 the control signals.

not allow the designer to change the final layout and the simulation algorithm is slower ($O(n^3)$) when compared with the developed in [5].

The NML simulator used in this work is also based on a behavioral model. In this model a magnet can present only three states, logic ’0’, logic ’1’ and ”null”, and can only influence its direct neighbors. The circuit is represented as a weighted graph in which each magnet is a vertice and the edges connect neighboring magnets. If two magnets present a ferromagnetic coupling the edge weight between them is 1. On the other hand, if the coupling is antiferromagnetic the weight is -1. The simulation algorithm performs a walk through the graph evaluating each magnet ($O(n^2)$). To simulate a sequence of inputs the original algorithm was improved.

B. ALU Schematic

The ALU is used to perform arithmetic and logic operations on many computer architectures. Its operations may vary from architecture to architecture but the ones considered here are the most common. The design has two bits of data and three bits for control.

The two input bits, A and B, are evaluated with the logic operations AND, OR and XOR. To select the required operation, the three control bits, S0, S1, and S2, are used to select which signal will pass forward.

Figure 3 shows the ALU schematic. First, operations AND, OR and XOR are executed on the inputs A and B. Then, two multiplexers controlled by S0 and S1, select one of the three operations. Finally, the third multiplexer controlled by S2, select if the output value will be the previous selected logic or its inverse.

The AND and OR gates were implemented using majority gates. The Multiplexer design follows the schematic shown in Fig. 4 A). The control input S select if ”I1” or ”I0” (S = ’1’ and S = ’0’, respectively) will be the output signal. The XOR gate was implemented using two AND gates, two NOT and one OR gate, as shown in Fig. 4 B).

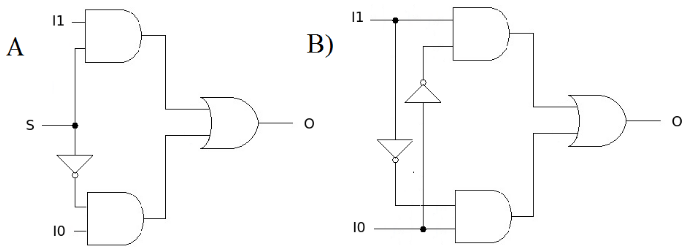


Fig. 4: A) Multiplexer schematic. B) XOR gate schematic.

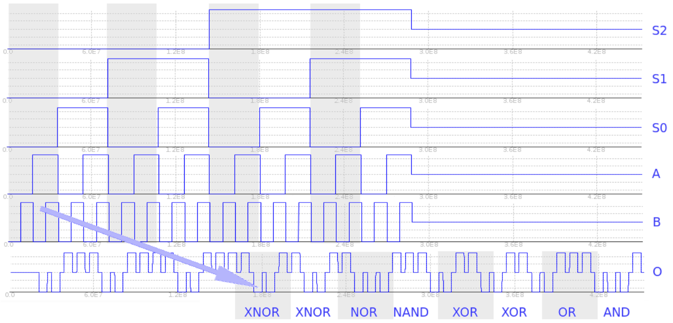


Fig. 5: Simulation signals: all combinations for A, B, S0, S1 and S2 were tested and the output is represented by the signal O.

The designed ALU is shown in Fig. 6. The signal goes from left (inputs and control signals) to right (output), passing through the circuit. The logic gates AND, OR, XOR and the three Multiplexers are highlighted in the figure.

C. Results

The ALU simulation waveform is presented in Fig. 5. The circuit has a latency of 20 clock cycles between the first set of inputs and the first valid output due the number of clock zones in the circuit. All possible input combination were tested and in all cases the expected output was achieved. As can be seen in the waveform, S2 initially is ’0’, thus the first outputs will be the inverse and s1 select the XOR logic twice due the exhaustive simulation.

In total, 685 nanomagnets were used on the ALU design: 5 as inputs, 11 with fixed-magnetization, 70 on majority gates, 1 as output and the remaining 598 just as signals propagation.

IV. CONCLUSION

In this paper a Nanomagnetic Logic ALU was proposed. It has two inputs and the operations AND, OR, XOR, NAND, NOR and XNOR are executed on them. Three control signals select which logic will be the final output. To the best of out

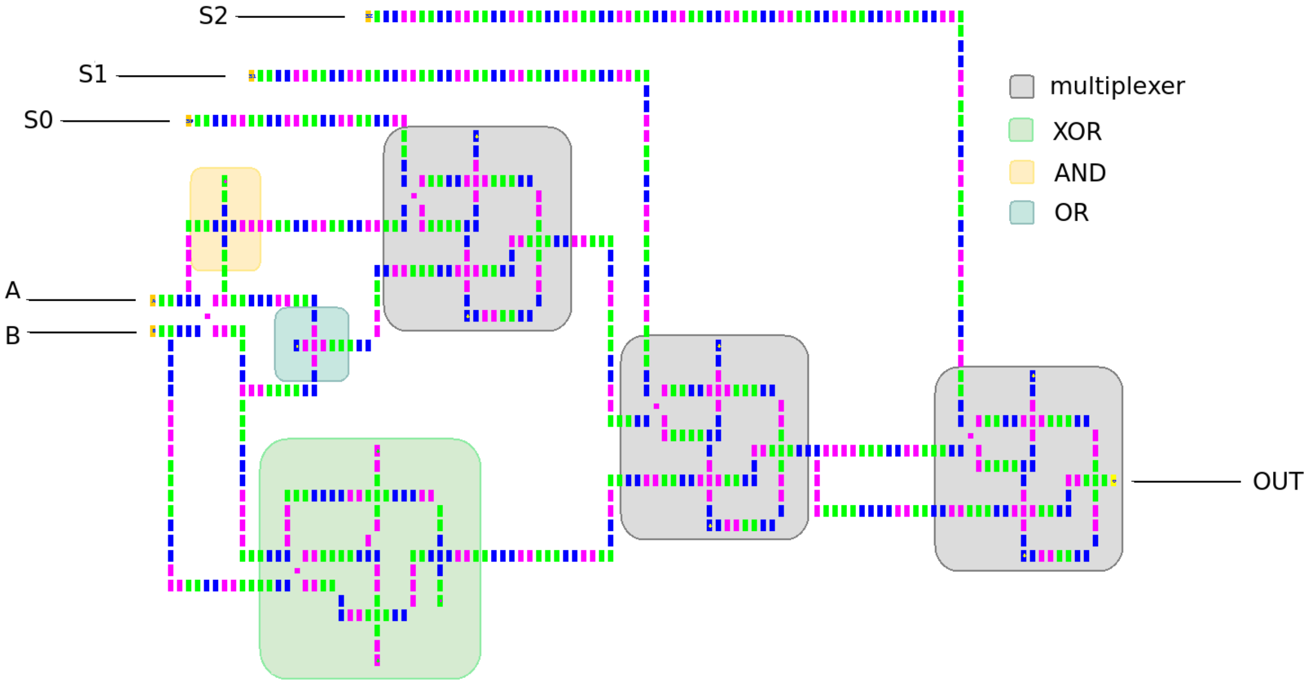


Fig. 6: ALU implemented in NML: green magnets are in clock zone 1, blue magnets in Clock Zone 2 and the purple magnets in Clock Zone 3.

.

knowledge, this is the first attempt to design a NML-based ALU.

Acknowledgment

The authors would like to thank CNPq and CAPES for the support.

References

[1] G. E. Moore, "Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp. 114 ff." *IEEE Solid-State Circuits Newsletter*, vol. 3, no. 20, pp. 33–35, 2006.

[2] R. K. Cavin, P. Lugli, and V. V. Zhirnov, "Science and engineering beyond moore's law," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1720–1749, 2012.

[3] G. Csaba, A. Imre, G. H. Bernstein, W. Porod, and V. Metlushko, "Nanocomputing by field-coupled nanomagnets," *IEEE Transactions on Nanotechnology*, vol. 1, no. 4, pp. 209–213, Dec 2002.

[4] J. Hong, B. Lambson, S. Dhuey, and J. Bokor, "Experimental test of landauer's principle in single-bit operations on nanomagnetic memory bits," *Science advances*, vol. 2, no. 3, p. e1501492, 2016.

[5] T. R. B. S. Soares, I. F. Silva, L. G. C. Melo, and O. P. V. Neto, "A new methodology for design and simulation of nml circuits," in *2016 IEEE 7th Latin American Symposium on Circuits Systems (LASCAS)*, Feb 2016, pp. 259–262.

[6] M. Niemier, E. Varga, G. H. Bernstein, W. Porod, M. T. Alam, A. Dingler, A. Orlov, and X. S. Hu, "Shape engineering for controlled switching with nanomagnet logic," *IEEE Transactions on Nanotechnology*, vol. 11, no. 2, pp. 220–230, 2012.

[7] A. Imre, G. Csaba, L. Ji, A. Orlov, G. H. Bernstein, and W. Porod, "Majority logic gate for magnetic quantum-dot cellular automata," *Science*, vol. 311, no. 5758, pp. 205–208, 2006.

[8] G. Csaba and W. Porod, "Behavior of nanomagnet logic in the presence of thermal noise," in *International Workshop on Computational Electronics*, 2010, pp. 1–4.

[9] M. J. Donahue and D. G. Porter, *OOMMF User's guide*. US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 1999.

[10] M. Vacca, S. Frache, M. Graziano, and M. Zamboni, "Topolinano: a synthesis and simulation tool for nml circuits," in *2012 12th IEEE Conference on Nanotechnology (IEEE-NANO)*. IEEE, 2012, pp. 1–6.