Journal of Computational Electronics (2018) 17:1370–1381
https://doi.org/10.1007/s10825-018-1215-8

# NMLSim: a Nanomagnetic Logic (NML) circuit designer and simulation tool

Thiago R. B. S. Soares[1] · João G. Nizer Rahmeier[2] · Vitor C. de Lima[1] · Lucas Lascasas[1] · Luiz G. Costa Melo[2] · Omar P. Vilela Neto[1]

Published online: 18 July 2018


**Abstract**
Nanomagnetic Logic (NML) is a new technology based on nanomagnets. Logic operations are performed at room temperature through dipolar magnetostatic interactions. The low switching energies 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. Although intense work has been done on the physical behavior of the NML systems, simulation tools for the study of large and complex NML architectures are lacking. In this paper, we present a fast and robust computational tool that allows the design and simulation of complex NML circuits. We developed a new fast simulation engine based on single-domain approximation able to capture both the static and dynamic behaviors of the system. The tool simulates the clock, the magnet influence in between neighbors and the thermal effect on the circuit. Initial results show a good agreement with experimental and micromagnetic simulations works. To the best of our knowledge, this is the first open-source NML simulator available and serves as a complete tool to design and evaluate complex NML circuits.

**Keywords** CAD tool · Nanomagnetic Logic · Simulation · Nanocomputing

## 1 Introduction

CMOS technology is close to its physical limits while at the same time reliability and power issues are rising at an alarming pace. The biggest challenges are the fabrication process and leakage currents [1]. To overcome these problems and to continue the trend of increasing integration densities, several new technologies were proposed in recent years [3]. One of them is Nanomagnetic Logic (NML) which is a new technology that operates with ultra-low energy dissipation [12,14]. An NML circuit is composed of arrays of nanomagnets placed on a plane interacting through the magnetostatic dipolar coupling. Also, NML is nonvolatile and presents majority logic, enabling the realization of ultra-high density computing systems.

NML circuits rely on the dipolar coupling between neighbor particles. Information that comes from one or more input magnets is propagated in the circuit as antiferromagnetic and ferromagnetic coupling. Particles geometries and position define their coupling interaction. Besides the magnetostatic coupling energy, there is also the shape anisotropy energy (which is of magnetostatic nature), that influences the final magnetization direction of a given output nanomagnet. The magnetic polarizations are then associated with '0' and '1' binary logic states, allowing the performance of Boolean logic operations. Some simple NML circuits have been experimentally demonstrated [12,20,21].

The functionality of NML circuits is extremely dependent on the geometry of the nanomagnetic particles as well

✉ João G. Nizer Rahmeier
joaonizer@ufmg.br

Thiago R. B. S. Soares
thiagorbss@dcc.ufmg.br

Vitor C. de Lima
vitorcezar@dcc.ufmg.br

Lucas Lascasas
lucas.freitas@dcc.ufmg.br

Luiz G. Costa Melo
lgmelo@ufmg.br

Omar P. Vilela Neto
omar@dcc.ufmg.br

[1] Department of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil

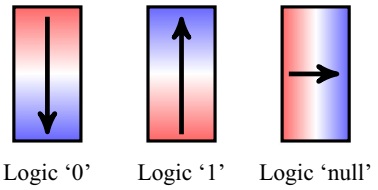
[2] Graduate Program in Electrical Engineering, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

as their relative positions, turning the design of new logical devices into a nontrivial task and also leading to a topology which is quite different from the traditional CMOS circuits. To implement a circuit that entirely works, correct placement of the magnets is necessary, avoiding undesired interactions among neighbor cells, thus creating an appropriate behavior, essential to the global functioning of the circuit. However, the evaluation of NML circuits demands the use of simulators because the physical implementation of large devices is still difficult to accomplish.

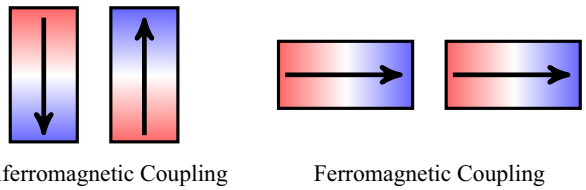
The simulation of an NML circuit can be performed by solving the Landau–Lifshitz–Gilbert equation (LLG) for every single spin in a magnetic device [11]. Although this method is appropriate to explore magnetic phenomena or the behavior of a small circuit, it is computationally prohibitive for large circuits. On the other hand, high-level HDL model can also be used to simulate NML circuits [9,10], but the simulation is approximated and neglects information about the design. Also, there is a SPICE-like model, but the design information is also missed; besides, just small circuits were presented [7]. Finally, the ToPoliNano tool offers details about the circuit layout and allows the simulation of large circuits [18]. However, the designer has no full control over the circuit layout (described in high-level HDL), and the behavioral model simulation engine is quite simple, bypassing important physical aspects such as thermal fluctuations. Thus, there is a lack of CAD tools to allow the full design and simulation of large NML circuits, such as QCADesigner [22] applied to simulate electronic quantum-dot cellular automata circuits. To bridge this gap, in this paper we present a fast and robust new CAD tool and simulation engine, named NMLSim.

The main contributions of this paper are as follows. First, we proposed and implemented a new simulation engine based on the single-domain approximation able to capture both the static and the dynamic behaviors of the system, enabling the simulation of large NML circuits quickly. We demonstrated the functionality and validated the proposed model comparing with micromagnetics simulations, providing performances evaluations of NMLSim against other tools available in the literature. Second, we developed and released a friendly CAD tool interface, allowing the designer to create any NML circuit easily. Third, we extended the engine to consider the effects of thermal noise and validated the results compared with theoretical and experimental results. Finally, we implemented different circuits to prove the flexibility and efficiency of NMLSim.

The remainder of this paper is organized as follows. Section 2 provides background about NML technology. Section 3 discusses the state of the art, while Sect. 4 proposes the novel simulation engine and CAD tool for NML circuits design and simulation. Section 5 introduces some case studies and discusses their results. Finally, Sect. 6 concludes this work.




**(a)** Logic representation




**(b)** Coupling between magnets

**Fig. 1** **a** The magnetization of nanomagnets represented by arrows are used to represent logic values. **b** Positioning two nanomagnets side by side its possible to achieve two different coupling patterns

## 2 Nanomagnetic logic

In this section, we present background on Nanomagnetic Logic (NML), going from basic devices through logic gates and how we can use clock networks to control more complex circuits functionality.

### 2.1 Basic devices

In this work, we assume a nanosized rectangular-shaped magnet, with single-domain behavior, as the basic NML device (other geometries are also possible [15]). Here the magnetization of these nanomagnets is assumed uniform and is mathematically represented by a vector of constant amplitude and variable angle. In NML circuits, the magnetization direction of a nanomagnet accounts for a bit of information.

The magnetic polarization of an isolated elongated nanomagnet is likely to lie along its longer axis (also known as easy axis), in order to minimize the shape energy. This energy has two stable minimum, yielding the magnetization vector to point in any of the two possible directions along the length of the rectangle. We then may arbitrarily define the logical values '1' and '0' when magnetization points 'up' and 'down,' respectively, as shown in Fig. 1a. In contrast, while applying an external magnetic field to the short axis of the particle, it goes to a metastable state, which maps a 'null' logic state. Concerning the correct coupling between two particles, it may occur in two different ways: antiferromagnetic and ferromagnetic. As shown in Fig. 1b, the former configuration, antiferromagnetic, presents antiparallel direction of the magnetization vectors, while the latter, ferromagnetic, presents a parallel orientation.

### 2.2 Simple logic gates

In this subsection, we describe some practical details on two types of nanomagnetic wires, a majority gate and an inverter. All these gates and wires had been tested experimentally [12,20,21].

Figure 2a, b shows examples of wire structures formed by the superposition of several nanomagnets to the situations shown in Fig. 1b. For the configuration in Fig. 2b, an even number of particles yields an inverter wire circuit.

An important gate in field-coupled nanocomputing technology is the majority gate (MG), shown in Fig. 2c. By setting the inputs 'A,' 'B' and 'C' while they couple with the output particle 'O,' a majority operation occurs. Both coupling types are present here; 'A' and 'C' act upon 'O' through a ferromagnetic coupling, while 'B' and 'O' present an antiferromagnetic coupling. So, considering such coupling behaviors the final Boolean equation of this kind of majority is $(MAJ(A, \overline{B}, C))$ [12]. AND and OR gates can be implemented by assigning one of the MG inputs to a fixed state, i.e., C = 0 executes an AND operation, while C = 1 performs an OR logic gate.

### 2.3 Clocking circuits

As NML circuits get complex (association of several nanomagnetic gates), the usage of clocking schemes becomes necessary. In this work, we have considered a three clock zone scheme [10,16] to address three major issues:

- Information directionality;
- Signal errors in long arrays of magnets;
- Nonadiabatic change of the magnetization.

In order to model the clock signal in NML, for simplicity, one can consider it as an external magnetic field that controls particle's magnetization. The magnetic field is applied perpendicularly to the long axis of the nanomagnet to force it into a logic 'null' state, as shown in Fig. 1a. When these particular nanoparticles are oriented on the horizontal, they reach an unstable energy state. By removing the external magnetic field, the magnets get oriented vertically, assuming a minimum state of energy, corresponding to a ground state. Finally, the magnetization of a given nanomagnet will choose to point up or down depending on the polarization of its neighbors.

At room temperature, NML circuits composed by few nanomagnets reproduce the coupling pattern of Fig. 2a, b. However, the level of ordering in a long array of nanomagnets is likely to present errors due to the influence of non-nearest neighbor coupling and thermal noise [5]. In this case, the clocking is necessary to ensure ordering and to minimize errors, as well as to allow control over magnetization of

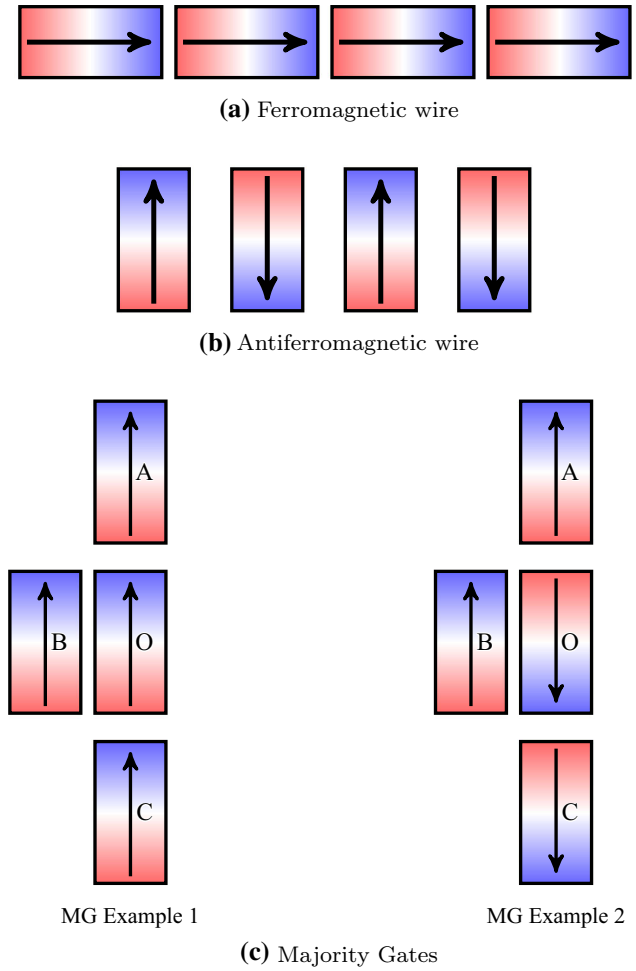



**Fig. 2** **a** The superposition of ferromagnetic coupled nanomagnets forms a ferromagnetic wire. **b** The superposition of antiferromagnetic coupled nanomagnets forms a antiferromagnetic wire. An antiferromagnetic wire with an even number of nanomagnets works as an inverter. **c** Two examples of inputs and outputs of the majority logic gate: its answer is the majority of the logical inputs

individual particles. Signal propagation and synchronization are achieved by splitting the circuit into groups called clock zones and by submitting them to different external magnetic fields (clock signals) [10].

A Hold phase represents an instant when no clocking field acts upon the particle; it remains in its current ground state. On the Reset phase, there is an applied field strong enough to force particle's magnetization into a metastable state, a 'null' logic state. The Switch phase characterizes a gradual removal of the applied magnetic field. This removal allows a particle to assume a minimum energy state based on the interaction with its neighbors. Figure 3a shows three clock signals and their corresponding zones. In the same clock step, different clock signals are in different clock phases, to guarantee information directionality.

The application of the presented clock scheme is observed in the example of Fig. 3b. The first particle is a fixed input

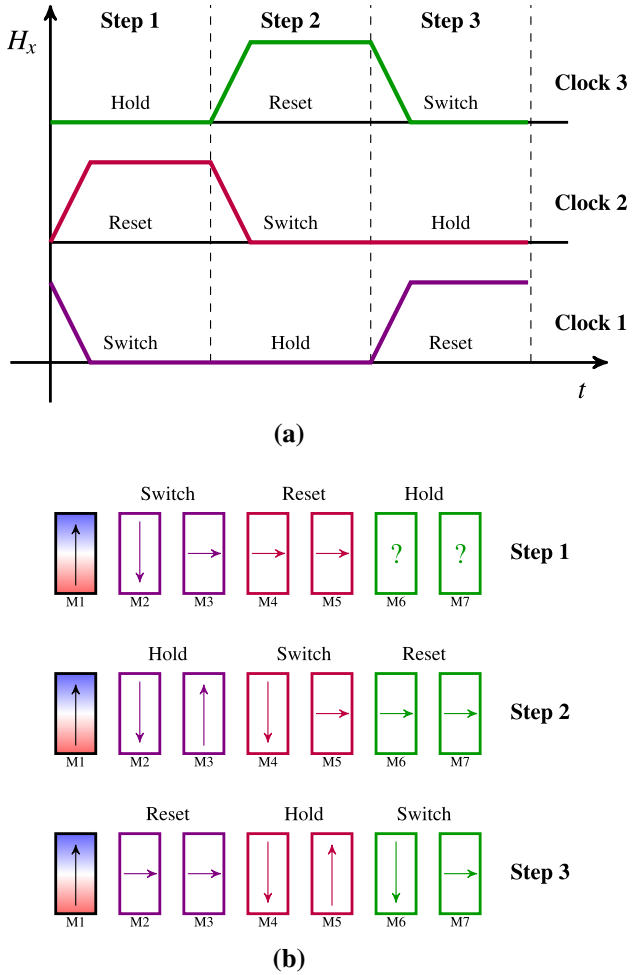



**Fig. 3** Clocking in NML. The designer splits the NML circuit into three clock zones. Each clock is controlled by a periodic clock signal. **a** Clock signals and its phases for Nanomagnetic Logic circuits. $H_x$ represents the external magnetic field and $t$ the time. **b** Antiferromagnetic wire divided into three clock zones. The left magnet is the input

magnet that drives an antiferromagnetic wire. In Step 1, we observe magnets M2 and M3 subject to Clock 1, switching and coupling with the input magnet, while the ones in Clock 2 are in a Reset state. Then, in Step 2, magnets M2 and M3 are in a Hold phase keeping their orientation. Also, magnets M4 and M5 are in a Switch phase and start to couple with magnets at Clock 1 region, while magnets M6 and M7 reach a Reset state on Clock 3. Finally, at Step 3, Clock 1 region resets again, M4 and M5 Hold their values, and M6 and M7, at Clock 3 region, start to Switch into their minimum energy state based on the coupling with the Clock 2 region magnets.

In addition to the adiabatic switch, the clock in NML is very important to signal synchronization. To solve this, the designer must correctly split the circuit into clock zones, ensuring that each critical part of the circuit (such as logic gates) will operate only when all inputs arrive.

## 3 Related work

The most suitable way to simulate NML circuits and explore their magnetic phenomena is solving the Landau–Lifshitz–Gilbert equation (LLG) for every single spin of a magnetic device. The LLG equation can be performed using the well-known micromagnetic simulation tools such as the Object Oriented MicroMagnetic Framework (OOMMF) [8] or MuMAX [19]. However, this approach is time-consuming and just feasible for small NML circuits.

In another work, Csaba et al. [4,6] developed a SPICE macro-model for the simulation of interacting nanomagnet arrays. The SPICE model is based on the single-domain approximation, which is a limiting case of the micromagnetic equations for nanometer particles. More recently, the authors introduced a numerical technique to derive several geometry-dependent parameters (demagnetization and coupling matrices) required by the single-domain model [7]. In the same work, they included in the model the effects of thermal fluctuations described by a stochastic magnetic field. Although the SPICE model allows the designer to simulate such magnetic devices within the same framework as conventional electronic circuits, just some small circuits, such as wires and majority gates, have been demonstrated.

Some works proposed the application of HDL models to simulate the architectural behavior of NML circuits in a faster way [9,10]. However, the HDL model neglects a lot of information on the circuit layout. Aiming to overcome this limitation, Vacca et al. [18] presented the ToPoliNano, a robust tool for synthesis and simulation of NML circuits. In this case, the tool emulates the top-down approach used in CMOS design. The circuit is described in VHDL language, and the tool automatically generates the layout, knowing exactly the circuit area and the precise placement of every element (the designer has no full control about the circuit layout). The circuits are then simulated using a simulation engine based on a behavioral model. However, this engine is quite simple and ignores some important features of NML circuits, such as nanomagnets geometry and thermal effects. More recently, an upgraded version of ToPoliNano has been presented, improving the automatically design generation [16]. Also, the previous model proposed by Csaba et al. [7] was adapted to the matrix-exploring simulation algorithm of the tool, and the fault analysis of small circuits have been presented [17].

This being said, there is a lack of CAD tools for NML circuits to satisfy all the following aspects: a control of the placement of any nanomagnet in the circuit by the designer; a fast simulation engine based on single-domain approximation; simulation of errors due to thermal fluctuations; the possibility of creation and reuse of logic blocks. In this paper, we propose a CAD tool which complies with all these require-

ments in a user-friendly interface. This tool may complement the NML designers tools available in the literature.

## 4 NMLSim

One of the main stages in a hardware project is the simulation phase. In order to increase the reliability of the simulation and to guarantee satisfactory results, it is necessary to understand the tools used. In this section, we will detail the computational model and the simulation algorithm implemented in NMLSim.

The magnetostatic energy between nanomagnets has been calculated according to the model presented in [13]. The total magnetostatic energy includes the demagnetizing energy of individual magnetic specimens and the dipolar coupling between two or more nanomagnets. For the simulations, we have considered that the magnets are made of Permalloy (Ni80Fe20).

### 4.1 NMLSim: modeling

The interaction between two nanomagnets depends on their size and the relative distance between them. Thus, these will be the first defined parameters. Next, the computational model of the circuit and the clock will be detailed. Finally, the simulation algorithm will be explained.

#### 4.1.1 Circuit modeling

For this version of the tool, the basic cell is a prism-shaped nanomagnet with 50 nm × 100 nm × 10 nm, as shown in Fig. 4a. The shape and dimensions of this configuration are the same applied in other works [7,15]. The spacing between two ferromagnetically and antiferromagnetically coupled nanomagnets was 24 and 10 nm (see Fig. 4b), respectively, since the ferromagnetic coupling is stronger than the antiferromagnetic one [7] and these values allow the equilibrium coupling energy. It is important to highlight that the simulator proposed here can be easily changed to new dimensions and distances.

After defining the shape and dimensions of the nanomagnet, the next step is to model the circuit computationally. The chosen data structure was a weighted directional graph. The graph is represented as an adjacency list in order to optimize the simulation algorithm. This way, an NML circuit is represented by a graph $G = (V, E)$, with $V$ vertices and $E$ edges. The algorithm maps each nanomagnet to a vertex $v$, and each edge $e$ connects two neighboring nanomagnets. Figure 5 shows an example of a majority gate represented as a graph with $V = \{A, B, C, M\}$ and $E = \{e_1, e_2, e_3\}$. To represent the magnetizations 'up,' 'down' and 'null,' a vertex can assume the values 1, − 1 or 0, respectively.

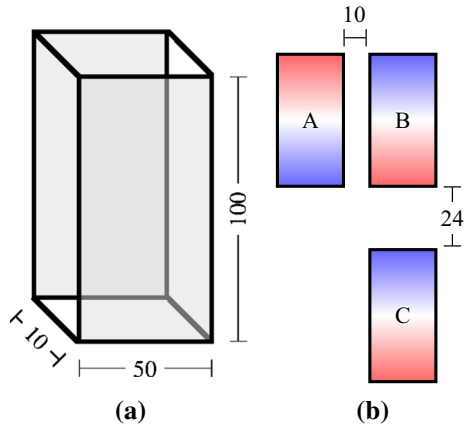



**Fig. 4** Details on shape and arrangement of the nanomagnets. **a** The nanomagnets are 50 × 100 × 10 nm. **b** The spacing between two nanomagnets antiferromagnetically and ferromagnetically coupled is 10 and 24 nm, respectively

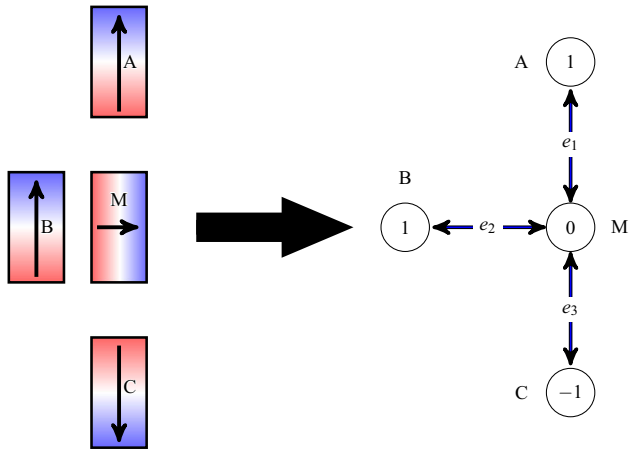



**Fig. 5** Majority gate represented as a graph. The vertex value represents the magnetization numerically, and the value of each edge correspond to the coupling energy between two nanomagnets

The weight of an edge $e$ quantifies the magnetostatic interaction between two neighboring nanomagnets. To determine the edge's weight, we calculated the stable magnetization configuration that minimizes the total magnetostatic energy of two nanomagnets, composed of shape plus dipolar coupling. This calculation is performed by fixing the magnetization of a nanomagnet and finding the magnetization angle of the second nanomagnet that minimizes the energy [13]. An example is presented in Fig. 6. The nanomagnet 'A' has its magnetization fixed at 90°, while nanomagnet B has no defined magnetization. Thus, the angle $\theta_2$ that minimizes the total energy of the system (demagnetization energy plus coupling energy) was determined. For instance, when the angle $\theta_2$ is 90° the coupling energy ($E_{cp}$) is equal to $-0.65eV$. To quantify the interaction between two nanomagnets, we used the $yy$ component of the coupling tensor $N_c$, as calculated in [13], to determine the edge weight. This calculation was per-

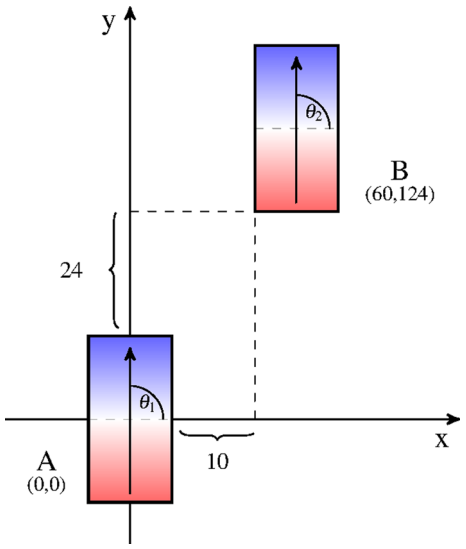



**Fig. 6** Configuration for determining the angle $\theta_2$ that minimizes the magnetostatic energy of the system. In this arrangement, with $\theta_1 = \theta_2 = 90°$, the coupling energy between the nanomagnets 'A' and 'B' is $-0.65\ eV$

formed for each possible pair of nearby nanomagnets, and the results are presented in Table 1. In this work, we considered that a nanomagnet has a two-nanomagnet influence radius, so a nanomagnet can have up to 24 neighbors.

As shown in Table 1, the magnetization of the nanomagnet 'A' was always fixed in 90° ($\theta_1$ angle) and, for each possible neighbor (configurations 1, 2, . . ., 8), the magnetization of the nanomagnet 'B' ($\theta_2$ angle) that minimizes the energy and the respective coupling energy ($E_{cp}$) between 'A' and 'B' were determined.

The $E_{cp}$ terms on Table 1 are only negative, meaning that these energy components minimize the total energy when particles are oriented in such $\theta_1$ and $\theta_2$ configurations. While analyzing the $\mathrm{Nc}_{yy}$ component of all magnets, we conclude that configurations 1 and 2 present an antiferromagnetic coupling between A and B since $\mathrm{Nc}_{yy}^{A,B} > 0$, while the remaining configurations present ferromagnetic coupling with $\mathrm{Nc}_{yy}^{A,B} < 0$. The weight $w$ used in the graph is also presented in Table 1. The value $w$ was determined by normalizing the $\mathrm{Nc}_{yy}$ terms by the highest absolute value among the set of coupling components. The signal of $w$ is negative when an antiferromagnetic coupling occurs. This way, an edge's weight maximum value is 1 (ferromagnetic coupling), and the minimum value is − 1 (antiferromagnetic coupling).

Considering the neighborhood radius equal to two and the calculated $w$'s, the complete graph representation of a majority gate is shown in Fig. 7. During the graph creation, whenever one of the eight configurations shown in Table 1 is identified between a pair of nanomagnets in the circuit, the weight of the edge between these vertices is the corresponding $w$ in the table. The purpose of implementing a neighborhood radius of two is to make the simulation more accurate to allow the design of more complex circuits. We calculated the energies for a neighborhood radius of three and four nanomagnets, but the values found were considered small, so we decided to simplify the calculations by removing them.

**Table 1** The eight coupling combinations between a pair of nanomagnets considered in this work

| Config | $\theta_1$ | $\theta_2$ | $E_{cp}$ (eV) | $\mathrm{Nc}_{yy}$ ($10^{-3}$) | $w$ |
|---|---|---|---|---|---|
| 1 | 90° | 270° | − 2.16 | 8.71 | − 0.98 |
| 2 | 90° | 270° | − 0.42 | 1.71 | − 0.19 |
| 3 | 90° | 90° | − 2.19 | − 8.83 | 1.00 |
| 4 | 90° | 90° | − 0.65 | − 2.63 | 0.30 |
| 5 | 90° | 90° | − 0.07 | − 0.28 | 0.03 |
| 6 | 90° | 90° | − 0.15 | − 0.61 | 0.07 |
| 7 | 90° | 90° | − 0.12 | − 0.49 | 0.06 |
| 8 | 90° | 90° | − 0.07 | − 0.28 | 0.03 |

$\theta_1$ is the magnet 'A' magnetization angle, $\theta_2$ is the magnet 'B' magnetization angle, $E_{cp}$ is the coupling energy between magnets 'A' and 'B.' $E_{cp}$ is the coupling energy related to such configurations, and $w$ is the edge's weight in the graph, obtained by normalizing the corresponding coupling tensor component $\mathrm{Nc}_{yy}$

#### 4.1.2 Clock modeling

Having the circuit modeled as a graph, we will explain the computational model developed for the clock and how it works on NMLSim. It is important to highlight that the clock developed in this tool is a generic approach. The clock was considered only as a way of conducting, adiabatically, the magnetization change in a nanomagnet. Nevertheless, here the clock will be represented as an external magnetic field.

In order to promote the shift from one stable magnetization to another, first the nanomagnet must be forced into

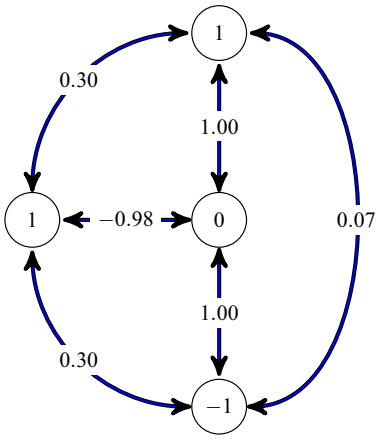



**Fig. 7** Complete representation of the MG shown in Fig. 5. In this example, the neighborhood radius causes all nanomagnets to be neighbors with each other

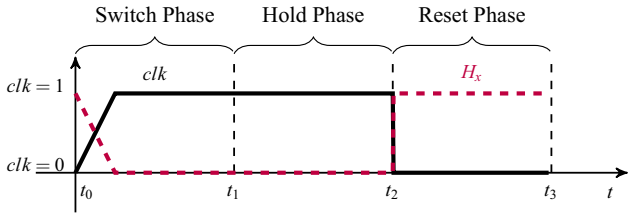



**Fig. 8** Representation of the *clk* waveform in NMLSim. Each phase lasts 300 time steps. If *clk* is equal to 1, it means that there is no external magnetic field being applied. Otherwise, if *clk* is equal to 0, it means that the external magnetic field is at its maximum amplitude

an unstable magnetization through, for example, an external magnetic field applied in the $x$ direction. During the removal of this magnetic field, the nanomagnet magnetization aligns according to its neighborhood, going to a stable magnetization state. This periodically inserted and removed external magnetic field is the clock in NML. To model something similar computationally, considering the model of the circuit previously explained, we created a variable *clk* that represents the clock. Figure 8 shows the *clk* behavior and its waveform.

Analogously to the waveform of the external magnetic field shown in Fig. 8, assuming that the external magnetic field amplitude $H_x$ varies from 0 to 1 (0 being the absence of the field and 1 the field at maximum intensity), the *clk* value is equal to $1 - H_x$. Thus, when the external magnetic field has the maximum amplitude ($H_x = 1$), *clk* is equal to 0 and when the external magnetic field is absent ($H_x = 0$), *clk* is equal to 1.

The waveform shown in Fig. 8 illustrates the variation of *clk* as a function of time. Nanomagnets in a clock zone controlled by *clk* may be on a Switch, Hold or Reset phases. The change from one phase to another lasts 300 time steps in this example. The number of time steps has a direct impact on the computational time spent during a simulation, although it does not change the complexity of the simulation algorithm.

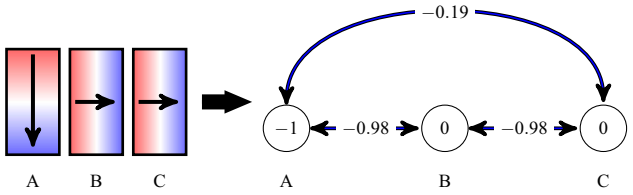



**Fig. 9** Antiferromagnetic wire with three nanomagnets and its graph representation

#### 4.1.3 Calculating the vertex value

The simulation algorithm implemented in this work calculates the dynamics of a nanomagnet magnetization in the Switch phase. The simulation algorithm implemented for NMLSim calculates a nanomagnet magnetization using Eq. (1).

$$M_t^i = \left( M_{t-1}^i + \sum_{j=0}^{n} M_{t-1}^j \times w_{ij} \right) \times clk_t + P_{\text{thermal}}, \quad (1)$$

where $M_t^i$ is the magnetization of a nanomagnet $i$ in the time step $t$. The $M_{t-1}^i$ is its previous magnetization, $M_{t-1}^j$ is the magnetization of the neighbor nanomagnet $j$ in time step $t-1$ and $w_{ij}$ is the weight of the edge between nanomagnets $i$ and $j$ in the graph (the normalized coupling energy). The $clk_t$ is the clock in the time step $t$, and $P_{\text{thermal}}$ is the thermal perturbation factor. Thus, the new magnetization of a nanomagnet is composed by its previous magnetization, added the influence of all its neighbors, limited by the external magnetic field (clock), plus a random thermal perturbation, which will be explained latter in this section. The $M_t^i$ can assume values between $-1$ and 1.

To better illustrate the simulation, consider the antiferromagnetic wire shown in Fig. 9. Here, the random thermal factor of Eq. (1) was removed (temperature equal to 0 K) in order to ensure reproducibility. The nanomagnet A has a fixed magnetization, and the magnetizations of nanomagnets B and C will be calculated. The expected values for nanomagnets B and C are 1 and $-1$, respectively. Figure 10 shows the values of vertices B and C during the simulation.

As can be seen in time step 0 (t0) (Fig. 10), if *clk* is equal to 0 (equivalent to the external magnetic field with maximum amplitude), the value of node B and C will be equal to 0 (null magnetization), which is the expected magnetization. If *clk* is equal to 1 (equivalent to the absence of the external magnetic field), the value of the vertices B and C will be determined by A, which is also expected when there is no external magnetic field. Thus, the clock model can simulate the presence and absence of an external magnetic field.

As a initial condition, the input represented by vertex A is pointing down with value $-1$; vertices B and C are pointing to

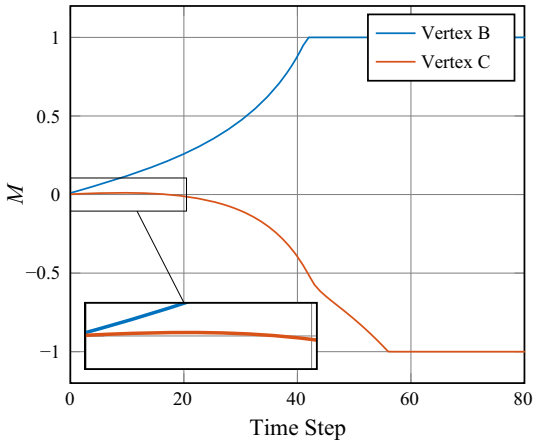



**Fig. 10** Values of vertex B and C calculated by Eq. 1 at 0 Kelvin. The inset magnifies the condition when vertex C goes slightly up

the right, both with value 0, representing that the external field applied is in its maximum value. As long as we start to remove the applied field (setting the clock signal toward one), B and C start to evolve their values. Before the 20th step, C is going slightly upwards, influenced by the input particle (see inset in Fig. 10). But, as soon as vertex B gets near to 1 its influence over C gets stronger and C start to move downwards. At the end, the system stabilize with a $\{-1, 1, -1\}$ configuration, dictated by the magnetic coupling among particles.

#### 4.1.4 Magnet evaluation and thermal effect

The investigation of errors is very important in simulation tools, and noise coming from thermal fluctuations is one of the main sources of errors in NML circuits. All properties of the thermal noise for single-domain particles are shown in [2]. Here we considered this noise as a perturbation term applied on each nanomagnet at each time step. First we calculated the standard deviation of the thermal magnetic field:

$$\sigma = \frac{1}{\sqrt{V \Delta t}} \sqrt{\frac{2kT\alpha}{\mu_0 \gamma M_s}}. \tag{2}$$

In Eq. (2), $V$ is the volume of a single nanomagnet ($5 \times 10^{-23}$ m$^3$), $\Delta t$ is the time scale of the simulation (we choose $10^{-12}$ s), $k$ is the Boltzmann constant ($k = 1.38064852 \times 10^{-23}$), $T$ is the temperature in Kelvin, $\alpha$ is the damping coefficient ($\alpha = 1$), $\mu_0$ is the permeability in vacuum ($\mu_0 = 4\pi \times 10^{-7}$ H/m), $\gamma$ is the gyromagnetic ratio ($\gamma = 2,211 \times 10^5$ m/A $\times$ s), and $M_s$ is magnetization saturation ($M_s = 796 \times 10^3$ A/m). For room temperature ($T = 300K$), $\sigma = 8596,8$ A/m. As defined in [23], the damping factor for Permalloy particles, at room temperature, is in the order of 0.01. In this work, we consider it as 1 to achieve a faster convergence into a steady state of the magnetization, reducing the computational efforts.

As presented in [5], the thermal field in A/m can be calculated as:

$$H_{\text{thermal}} = \sigma \times \mathcal{N}(0, 1) \tag{3}$$

where $\mathcal{N}(0, 1)$ is a white Gaussian distributed random value. Since the magnetization values in Eq. (1) are normalized from $[-1, 1]$ the thermal field described above need to be normalized by the saturation magnetization $M_s$, resulting in the thermal perturbation factor:

$$P_{\text{thermal}} = \frac{H_{\text{thermal}}}{M_{rms}}. \tag{4}$$

Considering the conditions stated earlier in this section, $P_{\text{thermal}}$ has a normalized standard deviation of 0.0108.

### 4.2 NMLSim: simulation algorithm

The simulation algorithm was implemented in C++ and is shown in Algorithm 1. The algorithm has as input the number of clock cycles ($n$) to be simulated, the list of vertices ($Magnets$) and the clock ($Clock$). The $Magnets$ list is divided into four sublists: a list with all input nanomagnets ($Magnets.Inputs$); a list with all nanomagnets in clock zone one ($Magnets.Zone1$); a list with nanomagnets in clock zone two ($Magnets.Zone2$); and, finally, a list with nanomagnets in clock zone three ($Magnets.Zone3$). The clock variable ($Clock$) contains the three clock signals ($clk1, clk2, clk3$) that control the phase of their respective clock zone.

**Algorithm 1** NMLSim's Simulation Algorithm

```
1: procedure SIMULATION(n, Magnets, Clock)
2:     repeat
3:         newInputs(Magnets.Inputs)
4:         repeat
5:             repeat
6:                 if Clock.clk1 == Switch then
7:                     Reset(Magnets.Zone2);
8:                     Switch(Magnets.Zone1, Clock.clk1);
9:                 else if Clock.clk2 == Switch then
10:                    Reset(Magnets.Zone3);
11:                    Switch(Magnets.Zone2, Clock.clk2);
12:                else
13:                    Reset(Magnets.Zone1);
14:                    Switch(Magnets.Zone3, Clock.clk3);
15:                Clock.clk1.nextValue();
16:                Clock.clk2.nextValue();
17:                Clock.clk3.nextValue();
18:            until d
19:        until p
20:    until n
```

The simulation algorithm has three nested loops: the first counts the number of clock cycles ($n$ cycles) to be

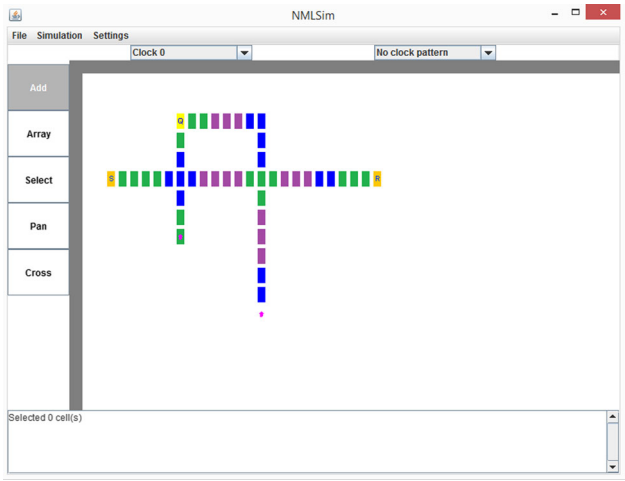



**Fig. 11** The NMLSim interface. The interface was developed in Java and has simple options to aid the designer in the construction of NML circuits

simulated; the second counts the number of phases in a clock cycle ($p$); the third loop counts the phase duration ($d$). The number of clock cycles that will be simulated is defined by the user. Initially, the algorithm introduces a new set of values in the input nanomagnets (function $newInputs(Magnets.inputs)$), and this set of values will be simulated for a clock cycle. The designer can set the input values through the interface. In the third loop, the clock zone in the Switch phase is identified (and by consequence the clock zone in the Reset phase) by its respective *clk* signal. First, the value of all nodes in the Reset phase is set to 0. Second, the magnetization of all nodes in the Switch phase is calculated. After the calculations, the clock signals are updated. The algorithm has a time complexity of $O(n \times m)$, where $n$ is the number of clock cycles and $m$ is the number of nodes in the graph (or the number of nanomagnets in the circuit).

### 4.3 NMLSim: GUI

We developed a friendly user interface in Java, shown in Fig. 11. The designer creates a circuit by placing each nanomagnet in a plane, and by selecting the input/output nanomagnets, the nanomagnets with fixed magnetization, and clock zones. In the superior menu 'Simulation,' it is possible to change the number of clock cycles that will be simulated along with other configurations. In the superior menu 'Settings,' the designer can change the substrate dimension, the auxiliary grids and the colors used in the tool.

## 5 Results

To exemplify NMLSim features, four circuits are discussed in the following. A simple majority gate, a Latch SR, a 4-bit ripple-carry adder, and a simple antiferromagnetic wire in order to show the effect of noise generated by the thermal fluctuations. All simulations were performed considering a temperature equal to 300 K in a 2.0 GHz Core i7-4510U CPU with 8 Gb DDR3 RAM.

### 5.1 Majority gate

In order to compare out tool with OOMMF, we simulated a majority gate composed of 4 nanomagnets spaced 10 nm horizontally and 25 nm vertically, similar to Fig. 2c, in both tools. All nanomagnets are $100 \times 50 \times 10$ nm, and the inputs were set in OOMMF by changing the initial magnetization `m0` and evolving the system until the output was stable. The material of the nanomagnets is Permalloy, and we used the standard variables values as pointed out earlier in this article. The cell size was set to $5 \times 5$ nm. The simulation on OOMMF took 17 s. The same circuit was simulated on NMLSim setting a fixed magnetization on the inputs. We set one nanomagnet as output to properly evaluate the correctness of the layout. The correct simulation performed in NMLSim took 8 ms, showing a speedup greater than 2000 (two thousand) times. This simple test shows that our tool is much faster than a standard tool used for simulation of NML circuits. It is important to highlight that our tool does not replace OOMMF since it does not solve the LLG equation, but it is suited for simulation of large NML circuits.

### 5.2 SR latch

The SR latch was implemented to demonstrate the feedback features of NMLSim. Figure 12a shows the circuit of the implemented latch and the NML implementation. The fixed polarization nanomagnets are required to generate an AND/OR gates using an MG. The green nanomagnets are in clock zone 1; the blue ones are in clock zone 2; and the purple ones in clock zone 3. In both figures, the arrows indicate the direction of the data through the feedback loop which retains the previous data when both inputs ($S$ and $R$) are 0. Since the information is transmitted from nanomagnet to nanomagnet, and during the Reset phase their magnetization is erased, memory units must be created using loops of nanomagnets that continuously circulate the stored data.

Figure 12b depicts the simulation results for SR latch, whereas the latch has the following sequence: set, reset, hold, set, hold and reset. This indicates the correct operation of the latch for all allowed input values. The circuit has 46 nanomagnets and an area of 27,456 $\mu m^2$. To set the latch, the inputs '$S$' must be 1 for two clock cycles due to the delay caused by the clock zones between the inputs and the output '$Q$.' The same applies to the reset operation.

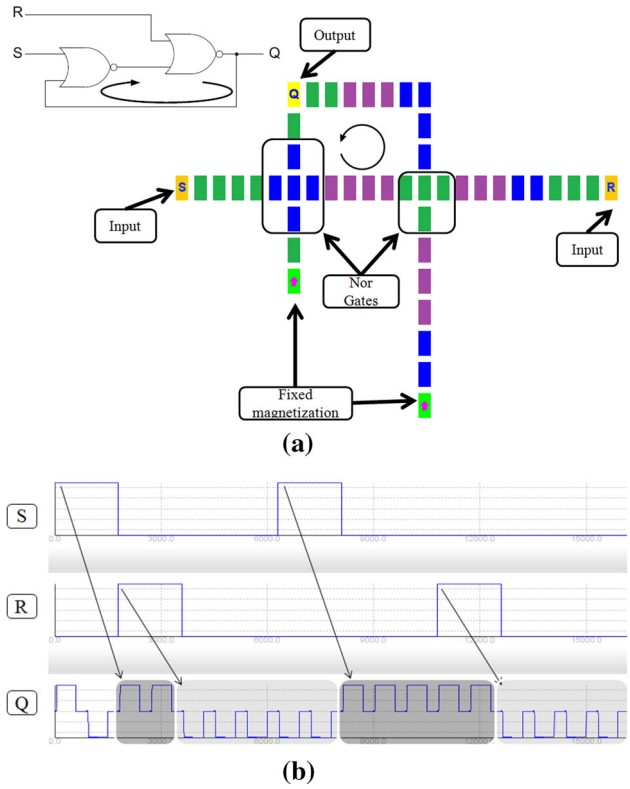



**Fig. 12** Implementation and simulation of an SR latch. **a** SR latch schematic and its implementation in NML with a small feedback loop and **b** simulation results for the SR Latch

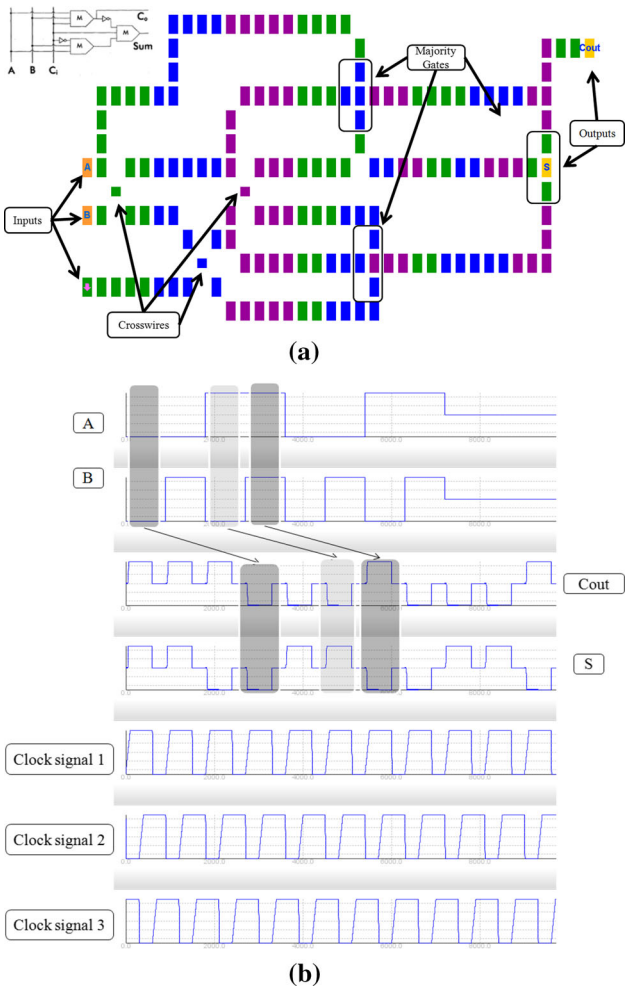



**Fig. 13** Schematic and simulation of a 1-bit adder. **a** 1-bit adder schematic and its implementation in NML and **b** simulation results for the full adder

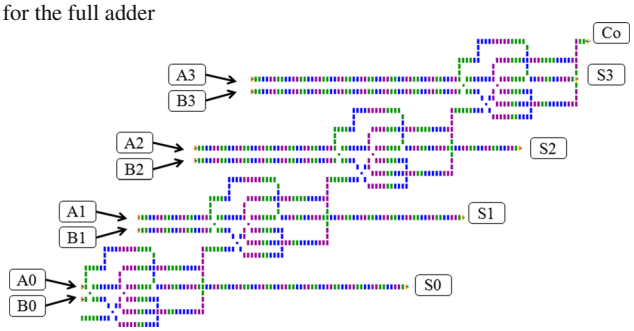



**Fig. 14** Implementation of a 4-bit full adder in NML

### 5.3 4-Bit ripple-carry adder

Aiming to explore large circuits, a full adder and a 4-bit ripple-carry adder were implemented. The schematic and the NML circuit for the full adder is depicted in Fig. 13a. The full adder is composed of three majority gates, two inverters and three cross-wire cells. We implemented a generic square cross-wire cell to enable the design of large complex circuits. The cross-nanomagnet only connects two neighboring nanomagnets to aid the in-plane cross-magnetization operation. The simulation result of the full adder is shown in Fig. 13b. We highlighted three set of input values and the output generated by it. As can be seen in the simulation, the first correct value will only reach the output after three clock cycles due to the delay generated by the clock zones. The circuit work as expected.

The circuit has 155 nanomagnets and an area of 4.5968 $\mu m^2$. The adder has three inputs: $C_{in}$, $A$ and $B$, which represent the bit carried in from the previous less significant stage in an adder cascade (set to 0 in this case), and the two bits to be added. The outputs are the sum and a carry out bit, which should be passed along to the next most significant stage.

The 4-bit ripple-carry adder is shown in Fig. 14. The circuit has 945 nanomagnets and an area of 63.6272 $\mu m^2$. The adder has 8 inputs: $A0$ to $A3$, $B0$ to $B3$. The bits from the inputs whose label begin with $A$ should be added to their counterparts $B$ bits. There are four outputs for the sum result and another one for the carry out bit. We simulated all 256 input combinations in 270 clock cycles in approximately 59 s. We do not present the simulation results for the 4-bit ripple-carry adder due to lack of space. However, we emphasize that the results are all correct.

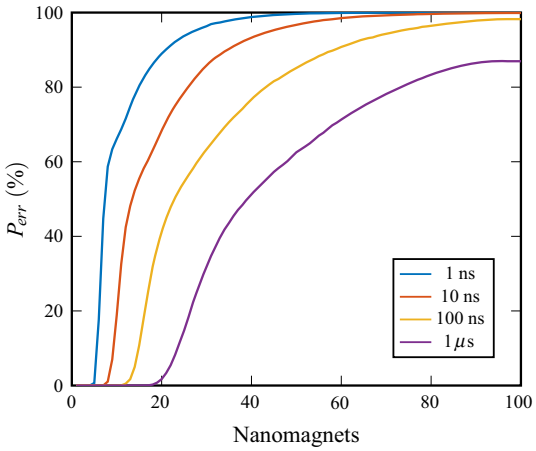



**Fig. 15** The probability of errors in a antiferromagnetic wire as a function of wire length with temperature equals to 300 K

To emphasize the NMLSim robustness, we also implemented a 16-bit ripple-carry adder. Due to the size of the circuit, the layout and the simulation results will not be shown here. The circuit has 8961 nanomagnets and an area of 979.2848 $\mu m^2$. We simulated a random set of input values for 100 clock cycles in approximately 3 min and 30 s.

### 5.4 Thermal effect analysis

The influence of the thermal fluctuations in long arrays of nanomagnets is an important topic that must be addressed in our tool. In this subsection, we will exemplify how the thermal fluctuation can affect the circuit functionality. As discussed earlier, the thermal effect is modeled in NMLSim simulation algorithm by a stochastic perturbation therm. Although this therm value is small, nanomagnets far from the input are very sensitive to random thermal fluctuations.

In order to analyze the influence of thermal noise in wires, we simulated a 100 nanomagnets wire at 300 K with 4 different removal times of the external field. We took note were the error start to occur on a set of 10,000 simulations for each removal time. After all simulation, the cumulative probability error was calculated for each nanomagnet in the chain; Fig. 15 depicts the results. Similar behavior is presented in [5] using OOMMF. While increasing the removal time of the external field, more nanomagnets accommodate on their global minimum of total energy. Since we implement an approximation of the nanomagnets dynamics, the number of correct coupled nanomagnets, probability error equal 0%, goes over five nanomagnets.

The example depicted in Fig. 16 is an antiferromagnetic wire composed of one nanomagnet with fixed magnetization and ten nanomagnets. The white nanomagnets present their magnetization pointing down, while the gray ones present their magnetization pointing up. For this simulation, the clock

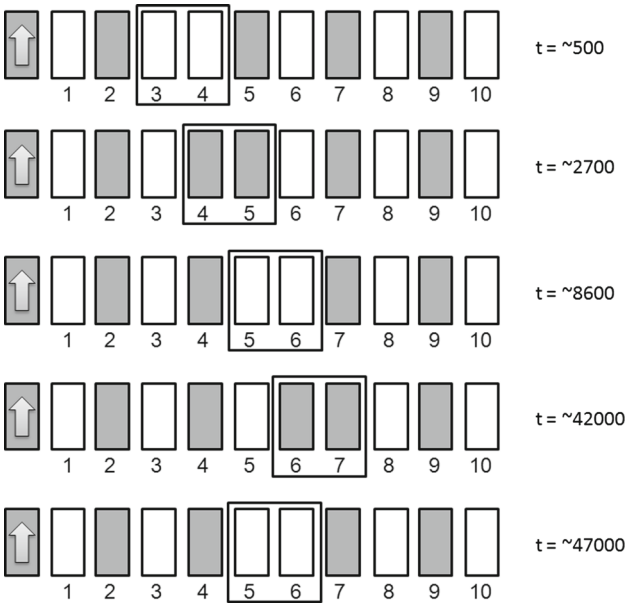



**Fig. 16** Wire submitted to a fast removal of the clock field is likely to present magnetic ordering

was removed in only one time step; this is called the soliton clock, and the simulation lasts 10 $\mu s$. As can be seen in the figure, for this clock removal rate, initially the wire presents an error in nanomagnet four. As the simulation advances, the ordering error 'walks' through the wire.

## 6 Conclusion and future work

In this work, we presented NMLSim, a CAD tool designed specifically for NML logic design and simulation. The tool allows the design and simulation of NML circuits, where the designer can place each nanomagnet on a plane to obtain the desired logic circuit and behavior. To the best of our knowledge, this is the first open-source simulation tool based on the nanomagnets coupling energy. The simulator is very fast, allowing the analysis of complex circuits in a few minutes. Since thermal noise is one of the main sources of errors in NML circuits, we have also introduced the effects of thermal fluctuations in the tool. To demonstrate the simulator functionality, we have successfully examined four different circuits. The simulator has successfully produced the expected behavior for all of them.

In future works, we intend to include the simulation of different sizes of nanomagnets automatically. Although we simulate only rectangular-shaped particles, a three-dimensional version of the graph will allow the tool to simulate different geometries via dipolar and coupling tensor coefficients. Also, we expected to allow NMLSim to simulate both in-plane and perpendicular NML (pNML) in the near future.

**Acknowledgements** The authors would like to thank CNPq, CAPES and FAPEMIG for the support.

## References


1. Abbas, Z., Olivieri, M.: Optimal transistor sizing for maximum yield in variation-aware standard cell design. Int. J. Circuit Theory Appl. **44**(7), 1400–1424 (2016)
2. Brown, W.F.: Thermal fluctuations of a single-domain particle. Phys. Rev. **130**, 1677–1686 (1963). https://doi.org/10.1103/PhysRev.130.1677
3. Cavin, R.K., Lugli, P., Zhirnov, V.V.: Science and engineering beyond Moore's law. Proc. IEEE **100**(Special Centennial Issue), 1720–1749 (2012)
4. Csaba, G., Porod, W.: Simulation of field coupled computing architectures based on magnetic dot arrays. J. Comput. Electron. **1**(1), 87–91 (2002)
5. Csaba, G., Porod, W.: Behavior of nanomagnet logic in the presence of thermal noise. In: 2010 14th International Workshop on Computational Electronics (IWCE), pp. 1–4 (2010). https://doi.org/10.1109/IWCE.2010.5677954
6. Csaba, G., Imre, A., Bernstein, G.H., Porod, W., Metlushko, V.: Nanocomputing by field-coupled nanomagnets. IEEE Trans. Nanotechnol. **99**(4), 209–213 (2002)
7. Csaba, G., Becherer, M., Porod, W.: Development of cad tools for nanomagnetic logic devices. Int. J. Circuit Theory Appl. **41**(6), 634–645 (2013). https://doi.org/10.1002/cta.1811
8. Donahue, M.J., Porter, D.G.: OOMMF user's guide. US Department of Commerce, Technology Administration, National Institute of Standards and Technology (1999)
9. Graziano, M., Chiolerio, A., Zamboni, M.: A technology aware magnetic qca ncl-hdl architecture. In: 9th IEEE Conference on Nanotechnology, 2009. IEEE-NANO 2009. IEEE, pp. 763–766 (2009)
10. Graziano, M., Vacca, M., Chiolerio, A., Zamboni, M.: An ncl-hdl snake-clock-based magnetic qca architecture. IEEE Trans. Nanotechnol. **10**(5), 1141–1149 (2011). https://doi.org/10.1109/TNANO.2011.2118229
11. Hubert, A.: Magnetic Domains: The Analysis of Magnetic Microstructures. Springer, Berlin (2008)
12. Imre, A., Csaba, G., Ji, L., Orlov, A., Bernstein, G.H., Porod, W.: Majority logic gate for magnetic quantum-dot cellular automata. Science **311**(5758), 205–208 (2006). https://doi.org/10.1126/science.1120506
13. Melo, L.G.C., Soares, T.R.B.S., Neto, O.P.V.: Analysis of the magnetostatic energy of chains of single-domain nanomagnets for logic gates. IEEE Trans. Magn. **53**(9), 1–10 (2017). https://doi.org/10.1109/TMAG.2017.2704913
14. Niemier, M., Bernstein, G., Csaba, G., Dingler, A., Hu, X., Kurtz, S., Liu, S., Nahas, J., Porod, W., Siddiq, M., et al.: Nanomagnet logic: progress toward system-level integration. J. Phys. Condens. Matter **23**(49), 493,202 (2011)
15. Niemier, M.T., Varga, E., Bernstein, G.H., Porod, W., Alam, M.T., Dingler, A., Orlov, A., Hu, X.S.: Shape engineering for controlled switching with nanomagnet logic. IEEE Trans. Nanotechnol. **11**(2), 220–230 (2012). https://doi.org/10.1109/TNANO.2010.2056697
16. Riente, F., Turvani, G., Vacca, M., Roch, M.R., Zamboni, M., Graziano, M.: ToPoliNano: a CAD tool for nano magnetic logic. IEEE Trans. Comput. Aided Design Integr. Circuits Syst. **36**(7), 1061–1074 (2017). https://doi.org/10.1109/TCAD.2017.2650983
17. Turvani, G., Riente, F., Cairo, F., Vacca, M., Garlando, U., Zamboni, M., Graziano, M.: Efficient and reliable fault analysis methodology for nanomagnetic circuits. Int. J. Circuit Theory Appl. **45**(5), 660–680 (2016)
18. Vacca, M., Frache, S., Graziano, M., Zamboni, M.: Topolinano: a synthesis and simulation tool for nml circuits. In: 2012 12th IEEE Conference on Nanotechnology (IEEE-NANO). IEEE, pp. 1–6 (2012)
19. Vansteenkiste, A., Leliaert, J., Dvornik, M., Helsen, M., Garcia-Sanchez, F., Van Waeyenberge, B.: The design and verification of mumax3. AIP Adv. **4**(10), 107,133 (2014)
20. Varga, E., Liu, S., Niemier, M.T., Porod, W., Hu, X.S., Bernstein, G.H., Orlov, A.: Experimental demonstration of fanout for nanomagnet logic. In: Device Research Conference (DRC), Vol. **2010**, 95–96 (2010). https://doi.org/10.1109/DRC.2010.5551852
21. Varga, E., Csaba, G., Bernstein, G.H., Porod, W.: Implementation of a nanomagnetic full adder circuit. In: 2011 11th IEEE Conference on Nanotechnology (IEEE-NANO), pp. 1244–1247 (2011). https://doi.org/10.1109/NANO.2011.6144445
22. Walus, K., Dysart, T.J., Jullien, G.A., Budiman, R.A.: Qcadesigner: a rapid design and simulation tool for quantum-dot cellular automata. IEEE Trans. Nanotechnol. **3**(1), 26–31 (2004)
23. Zhao, Y., Song, Q., Yang, S.H., Su, T., Yuan, W., Parkin, S.S., Shi, J., Han, W.: Experimental investigation of temperature-dependent Gilbert damping in permalloy thin films. Sci. Rep. **6**(22), 890 (2016)