Modeling, Design, and Analysis of MagnetoElastic NML Circuits

Davide Giri, Marco Vacca, Giovanni Causapruno, Maurizio Zamboni, and Mariagrazia Graziano, *Member, IEEE*

***Abstract*—With the predicted end of CMOS scaling process, researchers started to study several alternative technologies. Among them NanoMagnet Logic (NML) offers advantages complementary to MOS transistors especially for its magnetic nature. Its intrinsic memory capability makes it suitable for zero stand-by power and logic-in-memory applications. NML requires a clock system that, if based on a magnetic field, highly increases the circuit dynamic power consumption. We have recently proposed a solution based on the magnetoelastic effect (ME-NML) [1] and on currently available fabrication processes, which drastically reduces dynamic power consumption. However, many questions still remain unanswered. Which kind of applications are best suited for this technology? How can we effectively design, analyze, and compare ME-NML circuits? Does it really offer advantages over state-of-the-art CMOS transistors? In this paper, we provide answers to all these questions and the results prove that this technology offers indeed extremely good performance. We have designed a Galois field multiplier with a systolic array structure to reduce interconnection overhead. We developed a new RTL model that allows us to easily describe and simulate circuits of any complexity, evaluating at the same time the performance and keeping into account technology constraints. We approach for the first time in the NML scenario the design of ME-NML circuits adopting the standard-cell method used in standard technologies and fulfill the design down to the physical level. The same circuit is designed also with NML technology based on magnetic fields and with a 28 nm low power CMOS bulk technology for comparison. The CMOS circuit is obtained through physical place&route with a commercial tool, providing, therefore, the most accurate comparison ever presented in literature. Power analysis shows that ME-NML circuits have a considerable advantage over both NML and state-of-the-art CMOS bulk technology. As a further by-product results clearly highlight which kind of architectures can better exploit the true potential of NML technology.**

***Index Terms*—Magnetoelastic clock, nano-magnet logic, parallel architectures.**

I. Introduction

QUANTUM dot Cellular Automata (QCA) [2] is an emerging technology that has drawn in recent years considerable amount of attention. QCA circuits are based on cells that can have only two stable polarization states, representing logic binary values "1" and "0" [3]. Each cell interacts with neighbor ones to propagate the information and implement logic functions.

There are three main implementations of QCA principle: Molecular QCA [5], NanoMagnet Logic (NML) [6] and Silicon Atomic QCA [7]. In molecular QCA technology, complex molecules are used as base cells. These molecules can switch at very high frequency, making this kind of QCA interesting for building extremely high speed circuits (1 THz) [8]–[10]. In the NanoMagnet Logic (NML) case, single domain nanomagnets are instead used as base cell [11]. The main advantages of NML technology is the very low power consumption [1], [12], [13]. Finally Silicon Atomic QCA aims at reproducing the QCA principle using individual atoms as quantum-dot, showing until now extremely promising experimental results [14]. Among these QCA implementations, NML logic offers some specific advantage. Particularly, circuits can be fabricated with current technological processes [15], work at room temperature and posses an intrinsic memory capability [16]. The basic unit of a NML circuit is a single domain nanomagnet with a rectangular shape and sizes smaller than 100 nm. Only two stable states are possible (Fig. 1(a)) which are therefore used to represent logic values [6].

To propagate signals through a NML circuit, a multiphase clock system is required: four clock signals with different phases (with 90° shift between one signal and the successive one) are applied to small areas of the circuit called clock zones (Fig. 1(c)). The need of a clock system is given by the two following reasons: 1) Theoretically information should propagate through the circuit thanks to magnetic interaction among neighbor magnets, but actually this interaction is not sufficient. Magnets must be forced in an unstable state through an external mean, like a magnetic field, lowering the barrier between the two stable states [17] (RESET state in Fig. 1(d)). When the clock field is removed magnets are free to switch according to the input element, propagating therefore the information; 2) Due to thermal noise [18], only a limited number of elements can be cascaded and can therefore switch together, otherwise the probability of error in propagating the information notably increases. With the adoption of a clock system, only a limited numbers of magnets in a clock zone, during the SWITCH phase (Fig. 1(d)), will flip. These magnets will polarize according to neighbor magnets in a stable (HOLD) state, while magnets in RESET state have no

Manuscript received August 27, 2016; accepted October 12, 2016. Date of publication October 19, 2016; date of current version November 8, 2016. This work was supported by the project MEKIMI, Marie-Sklodowska-Curie Intra-European Fellowship action, REA (EU). The review of this paper was arranged by Associate Editor Garrett S. Rose.

D. Giri is with the Department of Electronics and Telecommunications, Politecnico di Torino, Torino I10129, Italy, and also with Columbia University, New York, NY 10027 USA (e-mail: davide.giri@studenti.polito.it).

M. Vacca, G. Causapruno, and M. Zamboni are with the Department of Electronics and Telecommunications, Politecnico di Torino, Torino I10129, Italy (e-mail: marco.vacca@polito.it; giovanni.causapruno@polito.it; maurizio.zamboni@polito.it).

M. Graziano is with the Department of Electronics and Telecommunications, Politecnico di Torino, Torino I10129, Italy, and also with the London Centre for Nanotechnology (UCL), London WC1H 0AH, U.K. (e-mail: mariagrazia.graziano@polito.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNANO.2016.2619377

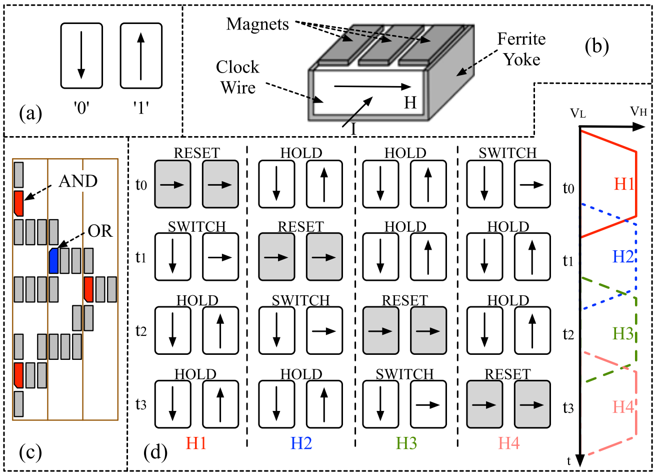


Fig. 1. (a) Single domain nanomagnets are used to represent logic values. (b) Magnetic field clock. The magnetic field is generated by a current flowing through a wire placed under the magnets plane. (c) Example of circuit layout based on the magnetic field clock. AND/OR gates are used as basic logic gates [4]. (d) Circuits are divided in small areas called clock zones. At every clock zone is applied one of many clock signals. Thanks to this mechanism in every time step magnets of a clock zone switch according to magnets of a neighbor clock zone which are in a stable state.

influence on signal propagation. In this way signal propagation direction is exactly defined. In Fig. 1(c) a 4-phases system is depicted, but it is possible also to adopt a 3-phases clock system as we proposed in [19].

From the implementation point of view, if a magnetic field is used as a clock mechanism, a current flowing through a wire placed under the magnets plane can be employed to generate it, as shown in Fig. 1(b). While this clock mechanism was demonstrated both theoretically and experimentally [6], [15], its main drawback is the high power losses due to the Joule power dissipation in the clock wires, thus strongly reducing the predicted possibility to achieve low power circuits.

Recently we have developed an innovative solution based on an electric field instead of a magnetic field, the magnetoelastic clock [12]. This solution allows to reach a very low power consumption, taking into account all power losses in the clock generation network. While the technological solution is similar to the one proposed in [20], our approach is technology-friendly, developed accordingly to current fabrication processes limitations. The particular circuits structure derived by this solution leads to the definition of a limited amount of possible basic structures, defined as “Standard Cells” [12], [21], that we adopt in this work. This predefined set of Standard Cells can be easily used to design circuits both with custom layout and using automated tools [22]–[24] greatly enhancing the development of NML technology.

From the application point of view, the use of an ultra low power clock system might be wasted if appropriate architectures are not chosen. This is due to the intrinsic pipelined behavior of a QCA circuit subjected to the clock system. Moreover, from a methodology point of view, in order to reliably capture the real circuit behavior and performance, the correct modeling technique and simulation environment should be defined. The model must be simple but faithful to the circuit physical structure. At the same time it should enable the description and simulation of circuits of any complexity.

Most importantly, it is fundamental to run a fair comparison between circuits based on ME-NML and highly scaled CMOS transistors. Too often in literature CMOS data are simply extracted from ITRS roadmap leading to a very imprecise and limited analysis.

In this paper we address all these concerns, evaluating the effectiveness of ME-NML circuits. Three important contributions are presented in this paper.

1) We demonstrate that ME-NML enables the design of effectively very low power circuits, compared to circuits based on ultra scaled CMOS transistors. Moreover we prove that, with an appropriate choice of circuits architecture that better exploit NML circuit characteristics, power consumption can be further reduced. The circuit that we use as testbench is a Galois Field Multiplier with a systolic array structure. Two versions of the Galois Field Multiplier are presented, with and without preskew and deskew networks on input and output signals. These unavoidable networks often are not considered in QCA literature, while we demonstrate that they add a lot of area and power overhead.
2) We develop a new simulation methodology for ME-NML circuits at Register Transfer Level (RTL), based on VHDL language. The simulation method is based on a set of Standard Cells described with an accurate model and can be easily used to build any kind of circuits. Moreover this simulation environment can be easily integrated in ToPoliNano [24], our design tool for NML circuits.
3) We perform the most accurate comparison with CMOS transistor ever presented in literature. The NML layout takes into account both technological constraints and clock generation network implementation. The same NML circuit is described in CMOS and its physical layout is obtained through Cadence Encounter using 28 nm low power CMOS technology.

Our aim is to provide clearer information on how further it is possible to go with NML technology. To reach this goal we rely on a complete analysis of the ME-NML circuits and a throughout comparison with CMOS technology.

II. Magnetoelastic Clock

The general structure of ME-NML circuits is shown in Fig. 2(a) Magnets are placed on a piezoelectric substrate, made of PZT (Lead-Zirconate-Titanate). Two electrodes are located at the boundaries of the cell. When a voltage is applied to the electrodes, an electric field is generated. This electric field induces a strain in the piezoelectric substrate, and the correspondent magnets mechanical deformation induces a variation in the magnetization thanks to the Magnetoelastic effect. Therefore the application of an electric field effectively forces magnets in the RESET state. When the electric field is removed, the shape anisotropy becomes predominant again and magnets switch to a stable state propagating the information. The complete theoretical analysis is reported in [21], while in [12] a possible

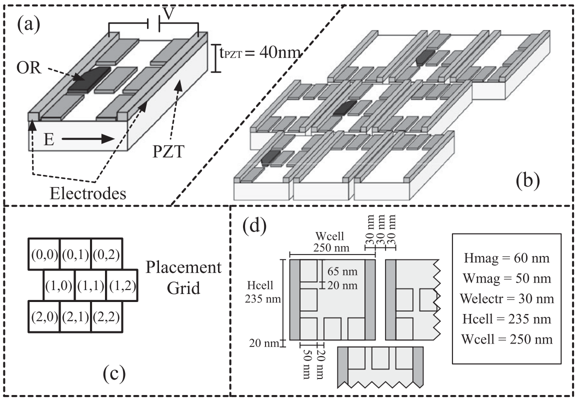


Fig. 2. (a) Magnetoelastic clock. Magnets are forced in the RESET state by the application of a voltage to electrodes placed on both sides of the clock zone. The correspondent electric field creates a mechanical deformation of the piezoelectric substrate (PZT) changing therefore magnets state. (b) Example of circuit layout based on the magnetoelastic clock. AND/OR gates are used as basic logic gates. (c) Placement Grid: magnetoelastic cells are placed in the circuit through the assignment of two indexes (row, column). (d) Size of the ME-NML 3 × 3 cell.

TABLE I
ENERGY DISSIPATION OF A ME-NML CELL WITH DIFFERENT PIEZOELECTRIC MATERIAL

Material	PVDF	ZnO	BT	PZT max	PZT min
Energy (fJ)	0.005	0.423	0.088	0.117	0.059

fabrication process is described. The maximum clock frequency that can be used for NML circuits is limited by the time necessary to reset magnets and their successive switching. According to the analysis in [1], this can be set to 100 MHz to guarantee a proper functioning of the circuit.

In current-based NML clock systems [6], [25], there is a very high power consumption due to the Joule losses in clock wires. Using a voltage instead of a current as driving technique greatly reduces power consumption. In this case the main source of power consumption is the energy lost during the charge and discharge of parasitic capacitances. This power consumption (CV^2) depends on the applied voltage, which is lower than 1V and the capacitance value, which is normally in the order of few hundreds of fF. This leads to a very low power consumption, typically 10 times lower than a 28 nm low power transistor [1]. The power consumption can be further reduced improving the piezoelectric material. PZT has optimal piezoelectric characteristics but it also has an extremely large dielectric constant, which leads to a high capacitance value. Choosing or developing new materials could further reduce power consumption of 10–100 times as shown in Table I. In Section IV we provide also further details on the equations used to compute energy consumption and included in our RTL model.

An example of ME-NML circuit is shown in Fig. 2(b): each cell is mechanically isolated from the others through patterning of the PZT obtained with lithography. In this way an electric field can be applied to each cell and the strain will not influence neighbor cells. Each cell represents a single clock zone. The size of a cell can vary between 3 and 5 nanomagnets, depending on the maximum number of magnets allowed in the critical path [18]. Communication among cells can be achieved only through top and bottom borders of each cell, since electrodes are placed on left and right sides of the cells. Logic circuits can be created using AND/OR gates as described in [4]. To standardize the design process, cells are placed to create the circuit adopting a "Placement Grid" (Fig. 2(c)). This solution leads to a very regular layout where every cell can be identified by a row and column number. The sizes of each nanomagnet and of an entire cell are reported in Fig. 2(d).

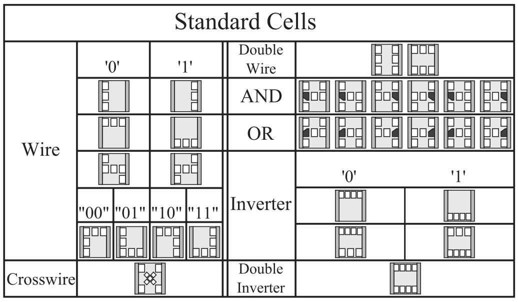


Fig. 3. Elements of the Standard Cells library (3 × 3 size only).

III. STANDARD CELLS

Due to the limited size of each cell, the number of possible magnet patterns is limited. This feature leads to the definition of a ME cells library enclosing all the conceivable magnets configurations. The full set of 3 × 3 size cells has been tabulated in Fig. 3. Circuits can be assembled simply by selecting the desired cells from the library and placing them in a grid-like fashion as shown in Fig. 2(c). Since a propensity for automation is in no doubt, this approach is particularly interesting in the perspective of a future *ad hoc* simulation and synthesis tool for ME-NML circuits. We are already working toward the integration of the layout editor in ToPoliNano, our design and simulation tool for emerging technologies.

The size of nano-magnets used in ME-NML circuits is 50 × 65 nm^2, with 20 nm interstice (2.D). As explained in [12] and in [1], magnets size can be increased to simplify the fabrication process. The value of 50 × 65 nm is chosen because it provides the best immunity to process variations. Electrodes are instead 40 nm wide. Their size is compatible with the minimum width of metal-1 wires in current CMOS technology.

In Fig. 3 each row defines a different type of cell. Each type can have different orientations. All possible permutations are not reported here for space reasons, but for each cell the other versions can be derived with horizontal and/or vertical flipping. *Wire* cells do not carry any logic function and they must have an odd number of horizontal magnets to avoid signal inversion. Every *wire* cell can host up to two wires, leading to *Double Wire* cells, allowing the propagation of two independent signals. The same is true for the *Crosswire* [6], but the difference is that here signals cross each other. The *Crosswire* is a particular logic block that allows to cross two wires on the same plane. Note that this kind of interference-immune crossing is essential because NML

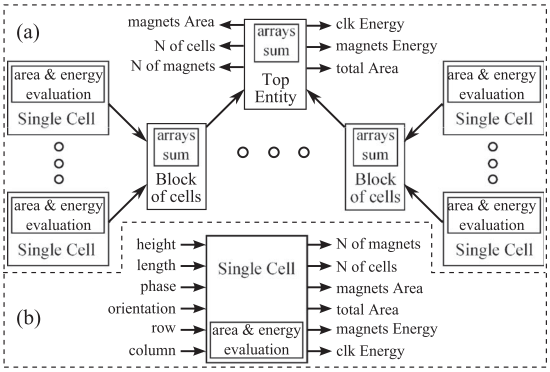


Fig. 4. Hierarchical model for estimating number of magnets and cells, occupied area and power dissipation. (a) Area and power value of every clock zone are added up by each Processing Element (PE) and finally from the multiplier entity, obtaining the final results. (b) Outline of inputs and outputs for a single standard cell.

is a planar technology, where it is not possible to use additional layers for interconnections. The set of logic gates counts *AND*, *OR* and *Inverter*. The *Inverter* is simply realized by an even number of horizontal adjacent magnets. *AND* and *OR* logic gates can be obtained cutting a corner of a magnet (Fig. 1(c)). The different shape of the cut magnets gives them a preferential state, which they will leave only when both inputs are up or down depending on the position of the cut, implementing as a consequence *AND* and *OR* ports [4].

IV. VHDL Model

We developed a new RTL model, written in VHDL language, whose purpose is twofold: Simulating a circuit verifying the correctness of the design and evaluating the occupied area and power consumption. Modeling the behavior of a cell is straightforward thanks to the clock system. Every cell samples a new data every clock cycle, therefore each standard cell can be modeled using a register plus, if needed, an ideal logic gate. So the propagation delay of a signal through a cell is equivalent to the behavior of a D-Latch. Every type of standard cell has its own VHDL description, and many *generic* parameters allow to differentiate cells of the same type and to set their relative position within the circuit. The parameters are cell length and width (in terms of nano-magnets), cell orientation (when needed), clock phase and cell position in the placement grid (Fig. 4(b)).

The VHDL model also evaluates area occupation and power dissipation for each cell and then sums these values up throughout the hierarchy of the circuit. In the following we provide an overview of the principles and equations used to compute the dissipated power in each cell and hierarchically in the entire circuit. The complete analysis with further details is provided in [1]. Formally, in ME-NML, there are two sources of power consumption: the losses in the clock generation systems and the nanomagnets switching. The former represents the main cause of power losses. In particular, this is the energy dissipated on the parasitic resistance when the parasitic PZT capacitance is charged and discharged. [1]. This energy can be computed as expressed in equation (1), where it is also approximated considering that the time constant value is much smaller than the integration period. The same amount of energy is dissipated in the discharging process, so finally the total energy will be doubled as reported in equation (1). This is a conservative choice because it provides a very pessimistic approximation of the real energy consumption, which is lower. This simplification also allows to calculate the energy consumption without the need of knowing the parasitic resistance

$$E = \int_{t1}^{t2} \frac{V^2}{R} \left(e^{-2t/RC}\right) dt \approx \frac{1}{2} C V^2 \Rightarrow E_{clock} = C V^2 \tag{1}$$

V is the applied voltage and C the equivalent capacitance, and they are computed as in equation (2):

$$V = \frac{W_{cell} \cdot \sigma}{Y \cdot d_{33}} \qquad C = \frac{\epsilon_0 \cdot \epsilon_r \cdot t_{PZT} \cdot H_{cell}}{W_{cell}} \tag{2}$$

where $\sigma = 28$ MPa is the applied stress [1], $Y = 80$ GPa is the Young modulus for Terfenol, and $d_{33} = 150$ pm/V is the piezoelectric coefficient of the PZT. ϵ_0 is the absolute dielectric constant, ϵ_r is the relative dielectric constant of PZT, $t_{PZT} =$ 40 nm is the thickness of the PZT substrate, $W_{cell} = 250$ nm and $H_{cell} = 235$ nm are the width and the height of a Standard Cell (Fig. 2(d)). The applied stress is computed starting from the physical characteristics of the single nanomagnet. The minimum value of applied stress is the one that generates a stress anisotropy at least equal to the shape anisotropy [20], and it is computed as in equation (3):

$$\sigma_{MIN} = \frac{\mu_0 \, N_d \, {M_s}^2}{3\lambda_s} \tag{3}$$

where N_d is the demagnetization factor, M_s is the saturation magnetization and λ_s is the magnetostrictive coefficient [1]. If the applied stress is greater than the minimum one, the behavior of a nanomagnet can be modeled as a bistable switch.

Table I highlights how, changing the piezoelectric material it is possible to obtain lower values of energy consumption. Among the piezoelectric materials we consider Polyvinylidenfluoride (PVDF), Zinc Oxide (ZnO), Barium Titanate (BT) and two types of PZT [1].

The VHDL model evaluates capacitance and voltage starting from cell dimensions and materials properties. A specific block sums the energy consumption of each cell and passes these values to the blocks at an higher hierarchical level. Blocks at higher hierarchical levels evaluate their energy consumption summing the energy consumed by lower level blocks. This approach is repeated recursively starting from the lowest level (each standard cell) until it reaches the top block in the design hierarchy. Thanks to this bottom-up computation the top entity provides the total energy (and therefore power) consumption for the whole circuit. Fig. 4(a) depicts how the model propagates area and power values through a three layers circuit: Single Cells, Cell Blocks, Top Entity. The function *arrays_sum* within a block sums area and power values of all the cells enclosed in such block. The model provides the area occupied by nano-magnets only and the area filled by all the cells, comprehensive of the separation space among them. For each cell the number of magnets and the cell dimensions are evaluated considering: type of Standard Cell, height and width (in terms of magnets). It is important to note that the values of area and power are exact, because no approximation are used in the layout creation. Circuit layout correspond to the exact physical mapping of the circuit, as it will be in case it is fabricated.

As mentioned before, nanomagnets switching is the other source of power consumption. This is the intrinsic energy consumption required to force magnets in the reset state. If an abrupt switching is used, it is equivalent to the height of the energy barrier between stable and reset states. The nanomagnets used in the ME-NML implementation, with chosen dimensions of 50 nm × 65 nm × 10 nm, have an energy barrier of just about 180 K_bT. Using an adiabatic switching it is possible to lower this value down to 30 K_bT, with the cost of worse performances in terms of clock frequency. We adopted the abrupt switch solution because this power component is in any case still much lower than the clock generation network consumption.

V. Galois Field Multiplier

To show the benefits of the proposed technology, we use as case study a Galois Field Multiplier (GFM). This architecture is chosen for its wide application in cryptography, coding theory, switching theory and digital signal processing.

A Galois Field is a field enclosing a finite number of elements together with the definition of its own addition and multiplication between two elements. For a Galois Field to exist and be unique the number of elements must be $q = p^m$, where q is the number of field elements, p a prime number and m a positive integer. Here we focus on the Binary Galois Fields arithmetic ($p = 2$), as it is the most suitable for VLSI implementation. For $m = 1$ the addition and multiplication rules are the ordinary ones, modulo p. However that is not true when m is greater than one. First of all, each element can be univocally associated to a polynomial $p(t)$ with binary coefficients and degree up to $m - 1$. The multiplication (modulo $p(t)$) follows the Montgomery algorithm reported below, where a and b are the inputs, r is the result and p corresponds to an irreducible polynomial of degree $m - 1$.

```
r(t) := 0
for i = m-1 downto 0 do
  r(t) := t*r(t) + a_i*b(t)
  if degree(r(t)) = m then r(t) :=
  r(t)-p(t)
return r(t)
```

The GFM is here implemented with a systolic array structure. Systolic arrays are particular architectures where arrays of identical processing elements are connected together without the need of long interconnection wires [26], [27]. The use of this kind of architectures is mandatory in NML (but it is also advised in QCA technology in general), since it is a planar technology. Without the possibility to use additional layers for interconnections as in CMOS, in NML circuits area tend to explode with the increasing of complexity due to the interconnections overhead. In [28] it is possible to see that, without choosing a proper architecture, the interconnections overhead can be roughly 99% of circuit area. In that case, also a low power clock system leads inevitably to a circuit with a higher power consumption than CMOS technology. It is therefore mandatory to choose appropriate architectures for NML (and QCA) technology, to exploit their true potential. To provide a better evaluation, we compare three different implementations: CMOS, NML based on the classic magnetic field clock and magnetoelastic NML. Some work has been presented about the analysis of perpendicular NML (pNML) performance with respect to CMOS [29], [30]. In the future we will also extend our analysis taking into account the pNML implementation of this circuit.

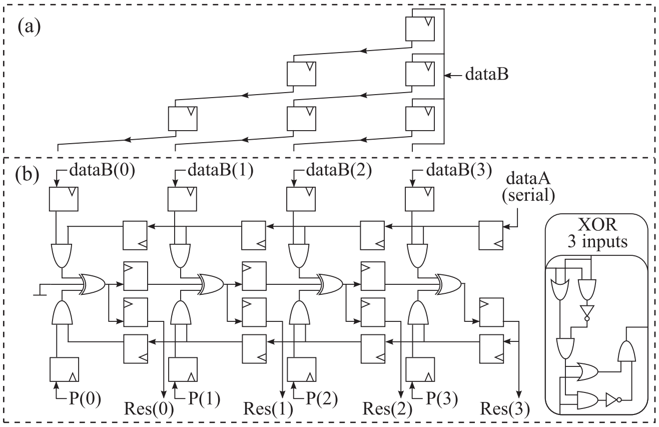


Fig. 5. (a) Circuitry for dataB bits synchronization. (b) Circuit schematic of a CMOS fully pipelined bit-serial Galois Field Multiplier. On the right the detail of the XOR logic function realized with AND, OR and INVERTER gates.

A. CMOS GFM

The schematic in Fig. 5(b) is a possible CMOS version of the bit-serial GFM for the case GF(2^4) [31]. The AND and XOR ports perform multiplication and addition in any binary Galois field, respectively. This circuit can be thought as a Systolic Array where every vertical block, composed of 2 AND and 1 XOR gates, plus a number of registers, is a Processing Element (PE). This makes the circuit very modular, composed of m identical PES, where m is the number of bits of parallelism. Therefore, increasing the parallelism, the circuit will simply grow horizontally, adding as much blocks as the parallelism increase. Of course the first and last block are slightly different from the others. We chose this fully pipelined version of the multiplier for two reasons. 1) Without the pipeline the *dataA* and feedback propagation could have long critical paths, growing proportionally to the circuit parallelism. The pipeline guarantees a constant critical path for any circuit parallelism, thus implying a greater throughput, at the cost of an area increase due to additional registers. 2) Since NML circuits are intrinsically pipelined, the comparison between this CMOS implementation and those based on NML technology is straightforward in terms of timing.

The timing protocol of this structure is strongly dependent on the pipeline stages. *DataA* must be given serially one bit every 2 clock cycles starting from the MSB. *DataB*, *P* and the

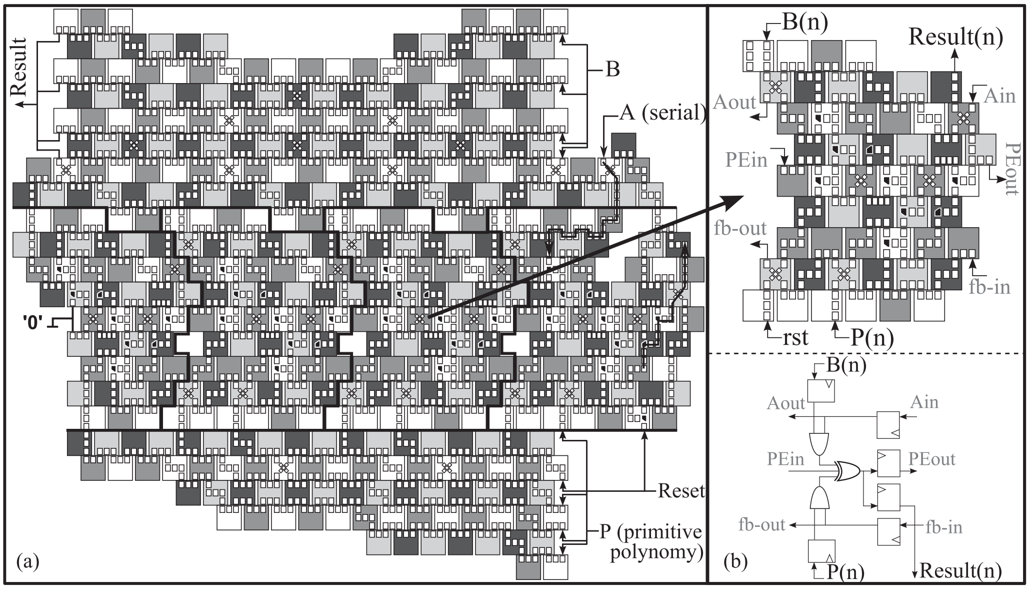


Fig. 6. (a) NML implementation of a 4-bit serial Galois Field Multiplier based on magnetoelastic NML circuits. Cells electrodes are not depicted for image clarity. (b) Detail of a basic block of the ME-NML GFM and its related CMOS implementation.

Result signals have the same behavior. To supply or acquire all inputs and outputs simultaneously, a preskew and deskew networks of registers are required. Fig. 5(A) shows the preskew network for *DataB*. Unfortunately with this additional circuitry the multiplier area grows quadratically instead of linearly with the circuit parallelism. However, analyzing the circuit without considering preskew and deskew network, leads to an important underestimation of circuit area and power consumption, as will be clear from the results provided in Section VI.

The requirements of sending a new data every two clock cycles derives from the two clock cycle delay of the loop inside the circuit. When feedback loops are pipelined a new data cannot be sent every clock cycles, because it is necessary to wait for the back propagation of the previous result. This reduces the circuit throughput.

The circuit detail on the right of Fig. 5(b) shows the implementation of a 3-input XOR logic function, exploiting only the ports at our disposal for ME-NML circuits: AND, OR and Inverter. However, this equivalent circuit is used only in the NML and ME-NML implementations. The CMOS circuit uses directly XOR gates, because it results in a more optimized layout.

B. Magneto-Elastic GFM

Using the standard cells library, we designed the magnetoelastic (ME) version of the GFM, from 4 to 64 bits. Fig. 6 reports the layout of a 4 bit Galois multiplier. Notice from Fig. 6 that 4 clock phases were used, represented with 4 different shades of gray. The white phase is the first phase (phase 1) and the phase progression continue from light gray to the darkest gray, which represents phase 4. Cells are taken from the Standard Cell Library (Fig. 3), however for sake of clarity electrodes are not depicted. Arrows in Fig. 6 shows how signals propagate through the circuit. Given the characteristics of ME NML technology and circuit layout, data bits are provided with 6 clock cycles of delay. This is caused by the intrinsic pipelined nature of the circuit.

Fig. 6 is divided into three parts. The middle one corresponds to Galois Multiplier alone, while the top and bottom sections representing preskew and deskew networks. The central part is further divided into blocks, each one representing a processing element: The central ones are identical, while the first and last blocks are slightly different. The GFM body is very compact and perfectly scalable, as the multiplier parallelism can be increased by simply copying and pasting processing elements equal to the central blocks. Even though the preskew and deskew networks are less regular than the multiplier body, they can be generalized too for any number of bits.

C. Magnetic Clock GFM

To provide a broader comparison, we implemented the GFM also using NML technology based on magnetic field clock and the snake-clock mechanism. The snake clock mechanism [19] uses a 3-phase clock system. Each clock phase is generated by the current flowing through a wire placed under or over the magnets plane. Since NML signals must traverse clock phases in the right order (1 then 2 then 3), to propagate in a specific direction wires 2 and 3 must be twisted to allow feedback signals. As a consequence clock wires corresponding to clock phases 2 and 3 are placed on different planes, to allow the twisting. The snake clock structure is depicted in Fig. 7 to better understand the circuit layout. Fig. 7 shows the 2-bit version of the GFM, the 4-bit implementation was not included due to its size. Similarly to the ME-NML Galois multiplier the middle region represents the Galois multiplier itself, with repeating processing elements. Top and bottom sections of Fig. 7 represent the preskew and deskew networks, which are relatively smaller than their equivalent in the ME-NML implementations.

While the principle of signal propagation through nanomagnets and the set of logic ports available are the same as in the

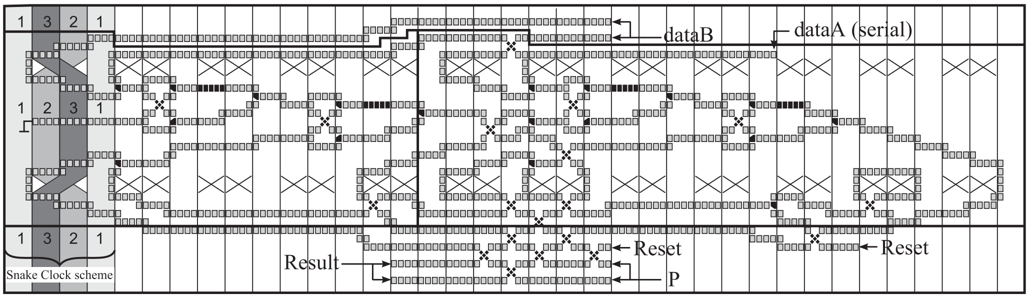


Fig. 7. NML implementation of a 2-bit serial Galois Field Multiplier based on a magnetic field and the snake-clock mechanism.

ME-NML case, the snake clock method leads to a distinctly different circuit organization. Each vertical stripe is a clock zone and it is driven by one of the three clock phases. In Fig. 7 the X are the areas correspondent to the clock wires twist, no magnet can be placed there. The two rows of Xs divide the circuit in 3 horizontal stripes; as pointed out by the numbers on the left, the central one propagates signals from left to right, while the others from right to left.

To model this Galois multiplier implementation we used a previously developed RTL model [32], still written using VHDL language. The model is different from the one used in the ME-NML case, because no standard cells are present. However it works in a similar way, modeling the propagation delay with registers and using ideal logic gates to implement logic functions. The area is evaluated as the rectangle circumscribed to the circuit. The power dissipation is instead the sum of two components: Power consumption due to nanomagnets switching, and power dissipation of clock wires. Using an adiabatic clock, the average energy dissipated by a single nanomagnet is equal to 30 $k_B T$. The main contribution are however the losses in the clock generation network. The current necessary to generate a magnetic field able to force a reset is high. Clock losses are therefore evaluated as the power dissipation by a 3 mA current flowing through a copper wire with a length estimated starting from circuit area. For more details on the model refer to [32]. Due to longer feedback loops, compared to ME-NML implementation, a new data can be sent to circuit inputs only every 10 clock cycles.

VI. Results

In this final Section we compare circuit performance of the three implementations, in terms of area and power. The analysis is obtained varying the number of bits from 4 to 64. We first consider the body of the GFM only, without considering the synchronization circuitry. Then, we compare the three implementations considering also preskew and deskew networks.

To obtain the most accurate comparison, for the CMOS implementation we performed the physical Place&Route with Encounter 13.1 by Cadence, exploiting a CMOS 28 nm FDSOI standard cell library. Fig. 8 shows two examples of post route layouts: A single processing element (Fig. 8(A)) and a Galois multiplier with 4 processing elements (Fig. 8(b)). Both cells and interconnections can be observed in Fig. 8. Area and power consumption are calculated automatically by Encounter. While the

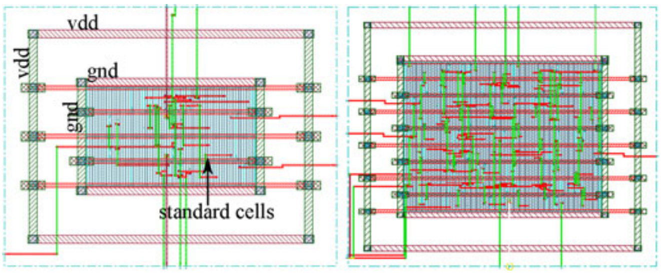
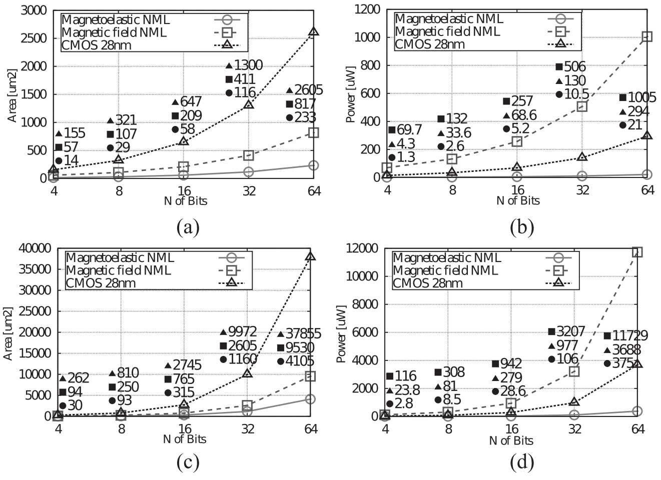


Fig. 8. Post route layout of the 28 nm CMOS implementation. (a) Single processing element. (b) 4-bit GFM.

operating frequency of the CMOS implementation can reach up to 7 GHz, for the power evaluation the frequency was limited to 100 MHz, the same frequency of NML circuits. This was done to get a fair comparison between the two technologies. At 7 GHz the power consumption of the CMOS circuit will be much higher. It is then clear that NML technology cannot completely replace CMOS technology, since its speed is intrinsically limited. NML technology can only provide benefits in terms of area and power consumption, coupled with the intrinsic memory ability. For its characteristics NML is therefore ideal to implement those algorithms that can be parallelized to have high throughput even if the latency for a single result is high. In particular, NML is ideal for circuits that would require too much power if implemented in CMOS.

Fig. 9(A) shows the comparison in terms of area among the three implementations of the Galois multiplier, without considering preskew and deskew networks. Clearly the area increases with the number of bit, but surprisingly, both NML implementations beat the CMOS implementation. This is a very interesting outcome, since CMOS is a multilayer technology while NML is a planar technology. The consequences are easy to understand: With the proper choice of application (and therefore circuit architecture) NML technology has a considerable advantage over CMOS in terms of area occupation. Without a proper choice of architecture, there can be no gain at all as shown in [28]. ME NML shows particularly good performance, having an area 4 times smaller then magnetic field NML Galois multiplier and 11 times smaller than the CMOS implementation. It can be argued that CMOS transistors can be scaled but the same apply to NML technology. Moreover, also considering a 14 nm transistor (2 times smaller), the area will decrease approximatively 4 times. ME NML still holds a considerable advantage also with these magnet sizes.



(a) (b) (c) (d)

Fig. 9. Comparison among the three Galois multiplier implementations. (a) Area without considering interconnection networks. (b) Power without considering interconnection networks. (c) Area considering interconnection networks. (d) Power considering interconnection networks.

Fig. 9(b) depicts instead the comparison in terms of power consumption, without considering preskew and deskew networks. The grow trend is similar to the area, but now the worst performance are obtained by the NML implementation. While the CMOS area is bigger, its power consumption is 4–5 times lower than the NML circuit. The current required to generate the magnetic field kills NML performance in terms of power consumption. ME NML power consumption is instead amazing low, about 13 times lower than the CMOS circuit, with the gap increasing with the bit number. Results are really promising for the future development of this technology.

As stated in Section V, the Galois field multiplier requires an external synchronization circuitry. This is a common requirements in many QCA circuits [16] , due to the intrinsic pipelining of this technology. However this additional circuits have a huge cost in terms of area and power consumption. Not often this cost is considered in literature, but here we want to deliver the best possible comparison between these three technologies. Fig. 9(c) shows the area comparison considering preskew and deskew networks. The trend and the differences among the three implementations is similar. The CMOS circuit is still the worst in terms of area occupation. To implement the synchronization network in CMOS a huge amount of registers is required. The only difference is that the gap between magnetic field NML and ME NML is reduced of 2 times. As described in Section V, the magnetic field implementation is more efficient when it comes to preskew and deskew networks. In terms of absolute performance instead, the influence of synchronization networks is heavy. Considering the 64 bits implementation there is an increment in area of 10 times, compared to the case with only the processing elements (Fig. 9(A)). The increment grows to 15 times in the CMOS case and to 20 times in the ME NML. ME NML technology seems the worst of the three in terms of implementing synchronization networks.

Fig. 9(d) highlights the power consumption considering synchronization networks. The general trend is similar to the one shown in Fig. 9(b), with magnetic field NML providing the worst performance of the three, while ME NML is the best. The increment of power consumption in absolute terms, considering preskew and deskew networks, is notable. Similar to the area the power increases of 10-20 times, depending on the implementation. From these results two conclusions can be drawn. First, it is mandatory to consider synchronization networks in QCA circuits, if they are required. They have a huge impact on performance and must be considered to get an accurate area and power evaluation. Second, ME-NML technology clearly leads to an incredible reduction in circuit area and power consumption over CMOS technology, provided that a proper circuit architecture is chosen.

TABLE II
AREA, ENERGY AND POWER COMPARISON OF THE 16-BIT GFM

Technology	Frequency	Area (μm^2)	Energy (fJ)	Power (μW)
ME-NML	100 MHz	58	7.28	5.2
Magnetic NML	100 MHz	209	603.95	257.0
CMOS	100 MHz	647	32.33	68.6
Optimized CMOS	1 GHz	261	6.42	217.8

To provide further comparisons between the three technologies, in Table II we have reported the energy and power comparison of the 16-bit GFM implemented with the three different technologies. The results are computed for circuit withoud preskew and deskew network to have a significant comparison with the fourth solution, called "Optimized CMOS". This represents the Galois Field Multiplier executing the Montgomery algorithm implemented in the optimum way in CMOS technology. It has less pipeline stages with respect to the version shown in Fig. 5(b) and it has been synthesized to run at a higher frequency of 1 GHz. Results show that the Optimized CMOS version achieves slightly better results in Energy consumption with respect to the ME-NML. Nevertheless, ME-NML is still the best technology in terms of Area occupation and Power dissipation.

VII. Conclusion

This article demonstrates that the introduction of magnetoelastic clock greatly enhances the potential of NML technology.

We have introduced several achievements to deal with ME-NML circuits. 1) We have proposed an advanced design and simulation methodology based on a set of Standard Cells and an RTL model, which is also able to estimate exactly the occupied area and power dissipation. 2) We have used as a testbench a Galois multiplier with a systolic array structure, demonstrating that these kind of circuits can greatly benefit from NML technology. 3) We highlighted the benefits of this technology against Magnetic field NML and CMOS, physically mapping the CMOS circuit with Cadence Encounter on 28 nm bulk technology.

Results show that, with the proper choice of architecture, ME-NML technology provides an outstanding advantage over ultra scaled CMOS transistors in terms of area and power consumption. Both are more than 10 times lower in the ME NML case.

Furthermore in our case study we analyzed also the overhead due to synchronization circuitry that increases area and power up to 17 times in the worst case.

References

[1] M. Vacca *et al.*, "Magnetoelastic clock system for nanomagnet logic," *IEEE Trans. Nanotechnol.*, vol. 13, no. 5, pp. 963–973, Sep. 2014.

[2] C. Lent, P. Tougaw, W. Porod, and G. Bernstein, "Quantum cellular automata," *Nanotechnology*, vol. 4, pp. 49–57, 1993.

[3] P. Tougaw and C. Lent, "Dynamic behavior of quantum cellular automata," *J. Appl. Phys.*, no. 80, pp. 4722–4736, 1996.

[4] M. Niemier *et al.*, "Shape engineering for controlled switching with nanomagnet logic," *IEEE Trans. Nanotechnol.*, vol. 11, no. 2, pp. 220–230, Mar. 2012.

[5] C. Lent and B. Isaksen, "Clocked molecular quantum-dot cellular automata," *IEEE Trans. Electron Devices*, vol. 50, no. 9, pp. 1890–1896, Sep. 2003.

[6] M. Niemier *et al.*, "Nanomagnet logic: Progress toward system-level integration," *J. Phys.: Condens. Matter*, vol. 23, Nov. 2011, Art. no. 493202.

[7] M. B. Haider, J. L. Pitters, G. A. DiLabio, L. Livadaru, J. Y. Mutus, and R. A. Wolkow, "Controlled coupling and occupation of silicon atomic quantum dots at room temperature," *Phys. Rev. Lett.*, vol. 102, Jan. 2009, Art. no. 046805.

[8] M. Liu, C. Lent, and Y. Lu, "Molecular electronics - From structure to circuit dynamics," in *Proc. 6th IEEE Conf. Nanotechnology*, Cincinnati, OH, USA, 2006, pp. 62–65.

[9] A. Pulimeno, M. Graziano, D. Demarchi, and G. Piccinini, "Towards a molecular QCA wire: Simulation of write-in and read-out systems," *Solid-State Electron.*, vol. 77, pp. 101–107, 2012.

[10] A. Pulimeno, M. Graziano, V. Cauda, A. Sanginario, D. Demarchi, and G. Piccinini, "Bis-ferrocene molecular QCA wire: Ab-initio simulations of fabrication driven fault tolerance," *IEEE Trans. Nanotechnol.*, vol. 12, no. 3, pp. 498–507, Jul. 2013.

[11] R. Cowburn and M. Welland, "Room temperature magnetic quantum cellular automata," *Science*, vol. 287, pp. 1466–1468, 2000.

[12] M. Vacca *et al.*, "Electric clock for nanomagnet logic circuits," in *Field-Coupled Nanocomputing* (series Lecture Notes in Computer Science), N. G. Anderson and S. Bhanja, Eds. New York, NY, USA: Springer, 2014, pp. 73–110.

[13] C. Augustine, X. Fong, B. Behin-Aein, and K. Roy, "Ultra-low power nano-magnet based computing: A system-level perspective," *IEEE Trans. Nanotechnol.*, vol. 10, no. 4, pp. 778–788, Jul. 2011.

[14] R. Wolkow *et al.*, "Silicon atomic quantum dots enable beyond-CMOS electronics," in *Field-Coupled Nanocomputing*, N. G. Anderson and S. Bhanja, Eds. New York, NY, USA: Springer, 2014, pp. 33–58.

[15] M. Alam, M. Siddiq, G. Bernstein, M. Niemier, W. Porod, and X. Hu, "On-chip clocking for nanomagnet logic devices," *IEEE Trans. Nanotechnol.*, vol. 9, no. 3, pp. 348–351, May 2010.

[16] M. Vacca *et al.*, "Nanomagnet logic: An architectural level overview," in *Field-Coupled Nanocomputing* (series Lecture Notes in Computer Science) N. G. Anderson and S. Bhanja, Eds. New York, NY, USA: Springer, 2014, pp. 223–256.

[17] M. Vacca, M. Graziano, and M. Zamboni, "Majority voter full characterization for nanomagnet logic circuits," *IEEE Trans. Nanotechnol.*, vol. 11, no. 5, pp. 940–947, Sep. 2012.

[18] G. Csaba and W. Porod, "Behavior of nanomagnet logic in the presence of thermal noise," in *Proc. IEEE Int. Workshop Comput. Electron.*, Pisa, Italy, 2010, pp. 1–4.

[19] M. Graziano, M. Vacca, A. Chiolerio, and M. Zamboni, "An NCL-HDL snake-clock-based magnetic QCA architecture," *IEEE Trans. Nanotechnol.*, vol. 10, no. 5, pp. 1141–1149, Sep. 2011.

[20] M. S. Fashami, J. Atulasimha, and S. Bandyopadhyay, "Magnetization dynamics, throughput and energy dissipation in a universal multiferroic nanomagnetic logic gate with fan-in and fan-out," *Nanotechnology*, vol. 23, no. 10, Feb. 2012, Art. no. 105201.

[21] D. Giri, M. Vacca, G. Causapruno, W. Rao, M. Graziano, and M. Zamboni, "A standard cell approach for MagnetoElastic NML circuits," in *Proc. EEE/ACM Int. Symp. Nanoscale Archit.*, Jul. 2014, pp. 65–70.

[22] S. Frache, D. Chiabrando, M. Graziano, M. Graziano, L. Boarino, and M. Zamboni, "Enabling design and simulation of massive parallel nanoarchitectures," *J. Parallel Distrib. Comput.*, vol. 74, no. 6, pp. 2530–2541, 2014.

[23] S. Frache, D. Chiabrando, M. Graziano, F. Riente, G. Turvani, and M. Zamboni, "ToPoliNano: Nanoarchitectures design made real," in *Proc. IEEE/ACM Int. Symp. Nanoscale Archit.*, Amsterdam, The Netherlands, 2012, pp. 160–167.

[24] M. Vacca *et al.*, "ToPoliNano: NanoMagnet logic circuits design and simulation," in *Field-Coupled Nanocomputing* (series Lecture Notes in Computer Science), N. G. Anderson and S. Bhanja, Eds., Berlin, Germany: Springer, 2014, pp. 274–306.

[25] J. Das, S. Alam, and S. Bhanja, "Low power magnetic quantum cellular automata realization using magnetic multi-layer structures," *J. Emerging Sel.. Topics Circuits Syst.*, vol. 1, no. 3, pp. 267–276, Sep. 2011.

[26] H. Kung and C. Leiserson, *Systolic Arrays for VLSI* (series CMU-CS). Pittsburgh, PA, USA: Carnegie-Mellon University, 1978.

[27] L. Lu, W. Liu, M. O'Neill, and E. Swartzlander, "QCA systolic array design," *IEEE Trans. Comput.*, vol. 62, no. 3, pp. 548–560, Mar. 2013.

[28] M. Awais, M. Vacca, M. Graziano, M. R. Roch, and G. Masera, "Quantum dot cellular automata check node implementation for LDPC decoders," *IEEE Trans. Nanotechnol.*, vol. 12, no. 3, pp. 368–377, 2013.

[29] X. Ju, M. Niemier, M. Becherer, W. Porod, P. Lugli, and G. Csaba, "Systolic pattern matching hardware with out-of-plane nanomagnet logic devices," *IEEE Trans. Nanotechnol.*, vol. 12, no. 3, pp. 399–407, May 2013.

[30] A. Papp *et al.*, "Threshold gate-based circuits from nanomagnetic logic," *IEEE Trans. Nanotechnol.*, vol. 13, no. 5, pp. 990–996, Sep. 2014.

[31] J. Grossschadl, "A low-power bit-serial multiplier for finite fields GF(2m)," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2001, vol. 4, pp. 37–40.

[32] M. Vacca, M. Graziano, and M. Zamboni, "Nanomagnetic logic microprocessor: Hierarchical power model," *IEEE Trans. Very Large Scale Intgr. (VLSI) Syst.*, vol. 21, no. 8, pp. 1410–1420, Aug. 2012.

Davide Giri received the M.Sc. degree in electronic engineering from the Politecnico di Torino, Turin, Italy, and from the University of Illinois at Chicago, Chicago, IL , USA, in 2014. He is currently working toward the Ph.D. degree in Computer Science at Columbia University, New York, NY, USA. His research interests cover emerging technologies, circuit architectures, and heterogeneous System-on-Chip.

Marco Vacca received the Dr. Eng. degree in electronics engineering from the Politecnico di Torino, Turin, Italy, in 2008. He received the Ph.D. degree in electronics and communications engineering from Politecnico di Torino, Turin, Italy, in 2013, and where he is now a Research Assistant. His research interests include Nanomagnet Logic and others beyond-CMOS technologies. He is also an expert of innovative and unconventional computer architectures.

Giovanni Causapruno received the Dr.Eng. degree in electronics engineering from Politecnico di Torino, Turin, Italy, in 2012, where he is currently working toward the Ph.D. degree in electronics and communications engineering. He works on parallel processing architectures for nanotechnologies.

Maurizio Zamboni received the Electronics Eng. and Ph.D. degrees from the Politecnico di Torino, Turin, Italy, in 1983 and 1988, respectively, where he is now a Full Professor. His research activity focuses on multiprocessor architectures design, in IC optimization for Artificial Intelligence, Telecommunication, low-power circuits, and innovative beyond CMOS technologies.

Mariagrazia Graziano (M'07) received the Dr.Eng. and Ph.D degrees in electronics engineering from the Politecnico di Torino, Turin, Italy, in 1997 and 2001, respectively. Since 2002, she has been an Assistant Professor in the Politecnico di Torino. Since 2008, she has been an adjunct Faculty in the University of Illinois at Chicago and since 2014, she has been Marie-Curie Fellow at the London Centre for Nanoelectronics. She works on "beyond CMOS" devices, circuits, and architectures.