POLITECNICO DI TORINO
Repository ISTITUZIONALE

ToPoliNano: a CAD Tool for Nano Magnetic Logic

Original
ToPoliNano: a CAD Tool for Nano Magnetic Logic / Riente, F., Turvani, G., Vacca, M., RUO ROCH, M., Zamboni, M., Graziano, M.. - In: IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. - ISSN 0278-0070. - ELETTRONICO. - 36:7(2017), pp. 1061-1074. [10.1109/TCAD.2017.2650983]

Availability:
This version is available at: 11583/2666386 since: 2018-02-20T12:22:10Z

Publisher:
IEEE

Published
DOI:10.1109/TCAD.2017.2650983

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

11 September 2026

ToPoliNano: a CAD Tool for Nano Magnetic Logic

F. Riente, G. Turvani, M. Vacca, M. Ruo Roch, M. Graziano, M. Zamboni
Politecnico di Torino, Department of Electronics and Telecommunications, Torino 10129, Italy

***Abstract*—In the post-CMOS scenario, Field Coupled Nanotechnologies represent an innovative and interesting new direction for electronic nanocomputing. Among these technologies, NanoMagnet Logic (NML) makes it possible to finally embed logic and memory in the same device. To fully analyze the potential of NML circuits, design tools that mimic the CMOS design-flow should be used for circuit design.**

We present, in this manuscript, the latest and improved version of ToPoliNano, our design and simulation framework for Field Coupled Nanotechnologies. ToPoliNano emulates the top-down design process of CMOS technology. Circuits are described with a VHDL netlist and layout is then automatically generated considering in-plane NML (iNML) technology. The resulting circuits can be simulated and performance can be analyzed. In this work, we describe several enhancements to the tool itself, like a circuit editor for custom design of Field Coupled Nanodevices, improved algorithms for netlist optimization and new algorithms for the place and route of iNML circuits. We have validated and analyzed the tool by using extensive metrics, both by using standard circuits and ISCAS 85 benchmarks. This contribution highlights the improvements of ToPoliNano, which is now a innovative and complete tool for the development of iNML technology.

I. Introduction

Quantum-dot Cellular Automata (QCA) [1] is a low power emerging technology where the interaction between electrons of different identical quantum cells enables logic operations (Fig. 1.A). Different QCA implementations that make use of novel materials, are currently under investigation. In Molecular QCA [2] [3] [4], molecules act as quantum dots operating at very high frequencies. Another interesting implementation is Nano Magnetic Logic (NML) [5]. In particular, in the in-plane NML (iNML), the elementary cell is characterized by a rectangular shaped magnets with typical dimensions of (50x100x20)nm or (60x90x20)nm [6]. Magnet sizes can be further reduced. In [7], we estimated that magnets of (15x30x5)nm have an energy barrier between stable states of $30K_bT$, allowing room temperature operation. Thanks to their anisotropy and nanoscale dimensions, iNML cells have the capability to store binary information by exploiting their intrinsic bistable magnetization (Fig. 1.B). The magnetic interaction among iNML devices makes it possible to propagate information through planar circuits. The true beauty of NML technology derives from its ability to offer features that are not available in MOS technology. NML has no stand-by power consumption (one of the biggest problem of MOS transistors), it is immune to radiations and it it based on a device that is both a memory and a logic element. The lack of leakage power consumption is a particularly appealing feature of the technology. All the applications that need to stay in stand-by for long time can greatly benefit from NML technology. NML technology can therefore complement MOS transistors very effectively [8] [9]. As depicted in Fig. 1.C, logic gates can be combined in order to perform digital functions. In the literature, several studies of NML have been presented: experimental results on elementary logic gates [10] [11] [12], simulations [13] [14] and architectural analysis [15].

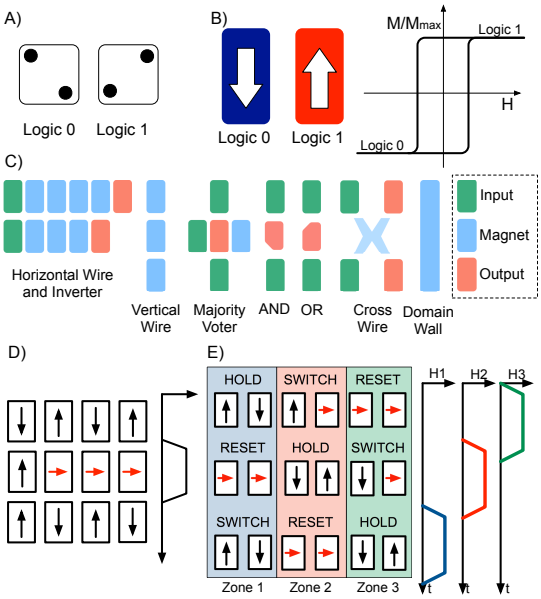

Figure 1: A) QCA elementary cells; B) iNML elementary cells and related hysteresis cycle; C) NML-logic Gates: wires, inverter, majority voter, and, or and cross wire; D) Reset mechanism; E) Clocking mechanism.

The information propagation is obtained thanks to the magneto-dynamic interaction of neighboring magnets. Thus, if the magnetization vector of the input magnet flips, all the other magnets of the wire chain should flip in a domino-like fashion. However, the shape energy barrier involved in the switch is too high to be won by the dipole-dipole energy supplied by the left magnet. To allow the information to propagate, in iNML, an external agent is required to rotate the magnetization vector of 90°, along the short axis [16]. This external agent, called *clock*, is the most important element in QCA technologies [17]. When the clock field is applied, magnets are forced into an intermediate unstable state. As depicted in Fig. 1.D, once the clock is removed, the magnets realign themselves according to the small energy provided by the dipole-dipole interaction. However, simulation results show that the number of magnets

that can be cascaded above a clocked zone is limited to only 4 or 6 magnets [6]. This is mainly due to the influence of the thermal noise as a result of working at room temperature [18]. In the literature, several clocking schemes have been proposed to guarantee the correct information propagation in iNML circuits. The first proposed solutions were based on a four-phase clock system [19]. However, in [16] [20] a more simple solution that uses only three partially overlapped phases has been proposed. The clock system is crucial in iNML technology. This ensures data flow direction and correct information propagation by alternating the phases of the clock zones as shown in Fig. 1.E. Different clock mechanisms were studied in literature. A current-generated magnetic field was the first mechanism proposed [6]. A STT-current clock, where magnetotunnel junctions are used in place of plain nanomagnets, was also proposed [21]. As demonstrated in [21] these second clock system is less efficient, with respect to a magnetic field clock, in case of large circuits. A more efficient system was proposed in [7]. Magnets are controlled by a mechanical stress applied through a piezoelectric substrate. Since the mechanical stress is generated with an electric field, power consumption is greatly reduced.

Due to these kind of constraints, the manual design of iNML-based architectures is rather complex. Moreover, the increasing interest in NML technologies has created the need for sophisticated tools that make it easier to design and study of circuits based on these technologies. In this paper, we introduce for the first time the complete flow of our tool, called ToPoliNano (Torino Politecnico Nanotechnology). This has been envisioned to meet the need for a software able to automatically design iNML circuits and to simulate them.

Here we present new and refined physical design algorithms [38] which have been conceived and tailored specifically for the iNML technology. Besides this, different techniques for the layout optimization have been implemented within the software. Simulation and fault analysis features, already presented in [38] [13] [22] have been recently enriched and refined in order to obtain more accurate results. These new algorithms, based on the LLG equations [23] [14], make it possible to reduce the gap between switch level and micromagnetic simulations. Our algorithms have been tested and analyzed through a benchmarking process based on ISCAS 85 [24] circuits. Moreover, in [25] simulation results have been validated by using oommf as a term of comparison.

II. Background and Motivations

A. Overview on existing tools

CAD tools for emerging technologies are becoming increasingly attractive; studies on these new technologies must be supported by ad-hoc designed tools, which enable to design and study the behavior of complex architectures. The impact of the technological constrains which characterize each emerging technology, dramatically affect the algorithm's implementation needed to design the layouts and to perform simulations. The following focuses on the iNML technology. iNML devices can be studied at different abstraction layers by using standard tools such as low level simulators. Beside this, switch level analysis of iNML architectures can be performed with high-level simulators. Micromagnetic simulators like oommf [26] and mumax^3 [27] are widely adopted to simulate the magnetic behavior of such nanostructures. Those tools enable accurate results with the possibility of observing the magnetic evolution during time. With micromagnetic simulators it is possible to set up and personalize multiple physical parameters; the modification of materials, shape-dimensions, external fields allow the evaluation of their impact on the structure's behavior. Furthermore, micromagnetic simulators can be used, if well supported with adequate experimental activities and measurements, to extract the physical parameters needed to create simplified models. Indeed, a first approach which makes possible the study of iNML architectures can be identified in the description of a compact model. For example, VHDL models can encapsulate and reflect the logic behavior of each single nanomagnet which composes a system [28] [29]. Moreover, the same approach can be exploited to study this technology by designing an equivalent electrical model. Exploiting compact models, commercial software like Modelsim or Cadence can be then "adapted" to work with emerging technologies. High level and low level simulators created the basis in the study of iNML technology, but a greater interest in this topic has raised the need for a more sophisticated tool able to follow the same top-down approach well established for the traditional CMOS technology. As mentioned above, low-level simulators enable accurate simulations but are extremely time-consuming, the complexity of large architectures would require enormous resources in terms of computational time. On the other hand, high-level simulators based on a finite number of compact models allows the testing of the logic behavior of complex iNML circuits, but in this case, a significant loss of physical information must be accepted. In this paper we present for the first time, the complete flow of ToPoliNano, an innovative tool envisioned to work with iNML obtaining accurate result with performances optimized for this specific target technology. Indeed, ToPoliNano not only allows the simulation of complex iNML structures with a remarkable accuracy in short period of time, but also introduces the capability of automatically design the final circuit layout optimized taking into account all technological constraints. The potentiality of ToPoliNano are multiple:

- The same top-down approach adopted for CMOS technology can be followed. Indeed, circuits can be described in a textual form simply using the VHDL standard language.
- Specifically tailored algorithms and optimizations allow the automatic generation of the final layout.
- Circuits can be simulated thanks to ad-hoc studied algorithms. Simulations are extremely fast and accurate.
- The effect of faults typically derived from the fabrication process of this technology, can be considered.
- The software organization is intended to be flexible with the objective of being extended to other emerging technologies.

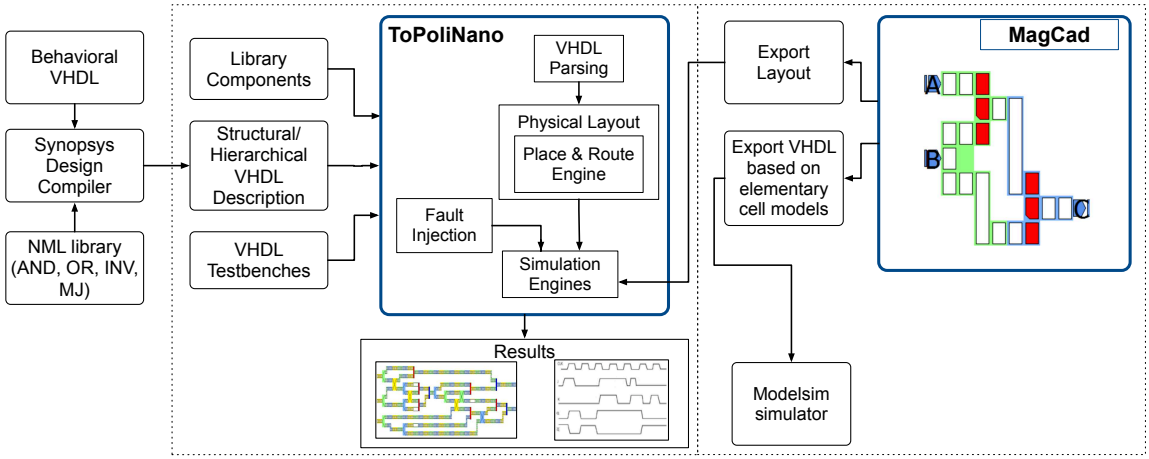

Figure 2: ToPoliNano and MagCAD design flow

B. *iNML and DML, technological constrains and design rules*

The flexibility of ToPoliNano resides in its intrinsic characteristic of distinguishing between technology-independent parts and others technology-dependent parts. In the following, a short background about iNML will be given to better understand the main constrains which characterize this technology and how they can impact the formulation of specific algorithms.

In iNML technology, single domain nanomagnets are used to represent binary information. The logic '1' and '0' is encoded in the two stable magnetizations of the magnet, respectively parallel and antiparallel to the easy axis. This is possible due to the shape anisotropy of rectangular magnets. As explained in Section I a clocking mechanism is needed to ensure a correct signal propagation [19]. The clock zone layout of iNML technology introduces tight constraints during the physical design phase. The clock zone layout defines the performance and the final timing of the whole circuits. The clock zone layout does not depend on the clock mechanism adopted, any of the three main clock solutions can be employed. As a consequence, choosing a different clock solution will lead to different performance, in terms of timing and power, however the layout of circuits will be always the same.
The maximum number of magnets that can be chained in a clock zone is limited to 4-6 [18]. This is a required to reduce thermal noise influence. In ToPoliNano the width of clock zones is however a parameter. It can be set to 4 or 6 or can be further reduced to improve circuits behavior in presence of thermal noise. The maximum number of vertical magnets that can be placed is limited to two. This is a very tight constraint when long vertical connections must be routed. Indeed, with this limitation vertical connections assume a stair-like behavior [30]. However, as demonstrated in [31], the use of domain walls for vertical interconnection can solve the problem. The domain wall is a long magnet with a minimum height of around 300nm. Summarizing, it is possible to claim that the clock zone layout is crucial for the correct propagation of the information inside the circuit. Moreover, a limited number of magnets (4 or 6) can placed in each clock zone to guarantee the correct signal propagation. Another problem that is difficult to address is the presence of loop inside the circuit. Different solutions have been proposed, like snake clock [16], but unfortunately none of them at the moment is able to solve the problem definitively from a technological point of view. This issue is related to the intrinsic pipelining of the technology. Therefore, the layout engine proposed in Section III-C of this paper does not address circuits with loops, only combinatorial circuits are considered.

While we do not have the possibility to experimentally demonstrate the circuits generated by ToPoliNano, we base them on two solid foundations. First, we employ the clock wire structure that was experimentally demonstrated in [10]. This clock structure is indeed very simple and can be easily extended to circuits of any complexity. The chip layout based on this clock solution is far more simple than the layout of MOS chips, so we are confident that the circuits here presented can be fabricated. The second foundation is that we base our design on experimental results [6], or on physical simulations obtained with micromagnetic simulators. Some examples among many can be found in [32] and [31]. The combination of this two principles makes us pretty confident not only that these circuits can be fabricated, but also that they will work as expected.

III. ToPoliNano

A. *Tool Overview*

The ToPoliNano (Torino Politecnico Nanotechnology) software has been completely developed by the VLSI group of Politecnico di Torino. This software, entirely written in C++, is able to design, simulate and test circuits based on emerging technologies.

Fig. 2 summarizes the ToPoliNano's working principle. The framework is composed of two parts: MagCad and ToPoliNano. The former is a stand-alone software which makes it possible to graphically design (custom) circuits based on emerging technologies. The latter is a CAD Tool which is able to design, test and simulate circuits based on the iNML technology. Here, the structural description of the circuit can

be given through a VHDL file or using MAGCad as an entry point.

MagCad can be seen as an entry point of ToPoliNano or a standalone software which enable designers to compose iNML-circuits and to test them using external softwares.
Indeed, once the circuit is completely designed, it can be exported following two approaches: first, extract a standard VHDL file, which can be used with standard tools like Modelsim. A compact VHDL behavioral model of each iNML elementary cell (single magnet, and, or etc) has been inserted in order to describe its logical behavior. Indeed, once the matrix-like topology is created the VHDL model of each cell is instantiated and properly connected into a top-level file. This file, fully compliant with standard formats can be directly used with external softwares like Modelsim in order to verify its logic correctness.
Second, a compatible data structure can be exported into ToPoliNano in order to use its internal simulation engine (eventually also considering the fault analysis) and verify its correctness.

In **ToPoliNano** the flow starts with the parsing of the input VHDL file (or files). At this stage all useful information related to inputs, outputs, interconnection between components and their relative technological implementation are extracted.
This information, stored through graphs and other supportive data structures, is given as input to the Place & Route engine to generate the final layout. From this second step on, algorithms are developed according to many technological constraints, and in this manuscript only the iNML is presented.
In the next stage the simulation of the circuit is performed also with the possibility to consider faults derived from the manufacturing process.

B. Parser

The ToPoliNano mainstream starts with the parsing stage. The VHDL files given as input are analyzed and translated into the corresponding internal data structure.
The provided input files must be structural and synthesized using only four fundamental gates: and, or, inverter and majority voter. In case of behavioral VHDL description, the user has to pre-process the circuit description by using a synthesis tool (e.g. Synopsys Design Compiler), provided that only basic iNML gates are used during the synthesis (and, or, inverter, majority voter).

Indeed, the output of the parser should allow the representation of the final data as structural, maintaining its hierarchical correspondence. In this stage, the text is divided into tokens, then, thanks to a massive use of semantic actions, the HDL Graph is generated. In this hierarchical structure, each node represents a basic element of the circuit while edges represent their interconnections. In the last step, the final data structure is created and prepare to be given to the Place and Route engine.

C. Automatic Layout Generation

The ToPoliNano's layout engine takes as input the graph generated during the parsing phase. Users can tune the design rules and choose different optimization algorithms according to their needs. Moreover, it is possible to choose among three layout approaches: i) fully hierarchical, ii) flat and iii) partially hierarchical. The first approach exploits the components already available in the user library in order to speed up the layout process. Then, the graph hierarchy is analyzed and an internal layout is generated for each sub-circuit not yet available. The use of a hierarchical design provides two more advantages: i) the possibility to design and test sub-modules of the circuit and ii) to reuse the tested sub-modules in order to make more complex designs. In the flat approach, the circuit's hierarchy is instead flattened to generate the layout. The last technique represents a trade-off between the two approaches. Users can choose on which level to stop the optimization.

In this section, the flat method is described. A comparison with the other two approaches is provided in Section IV. Before going into the details of the physical design, it is important to recall the layout constrains of the iNML technology: i) the I/O terminals are located on the top/bottom boundary of the circuit; ii) the adopted clock mechanism sets the directionality of the signal flow from inputs to outputs; iii) the number of magnets that can be cascaded in a clock zone is limited; iv) signas synchronization is a critical issue which defines the performance and the final timing of the whole circuit; and v) since iNML is a planar technology, the cross wire minimization is an important issue. The aim is to perform the routing between two layers of the graph by minimizing the number of crossings. The core of the layout engine (Fig. 3) is divided in two main parts: i) the graph elaboration and ii) the physical mapping phase.

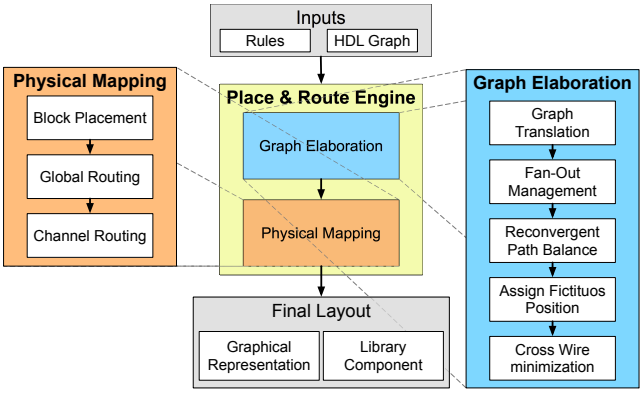

Figure 3: iNML layout engine flow in ToPoliNano

Graph Elaboration: The first part of the layout process can be done at a higher level of abstraction, whereas in the second part, the particular shape of each block is considered.
As a first step, the HDL Graph, which is technology independent, is translated into a new Directed Acyclic Graph G(V,E), called iNML Graph, which represents the circuit by taking into account the iNML technology constraints. Indeed, the first step of the graph elaboration phase is in charge of checking the gates used in the VHDL description. If the gates are compliant with the one available in the iNML technology, the layout process can continue to the next steps, otherwise an error

message is reported to the user.
The iNML Graph G(V,E) is a k-layered bipartite graph with V vertices and E edges (Fig. 4.A). Therefore, the graph G is composed by k number of p disjoint partitions with an assigned level, denoted $lev(p)$. After the graph translation, the iNML

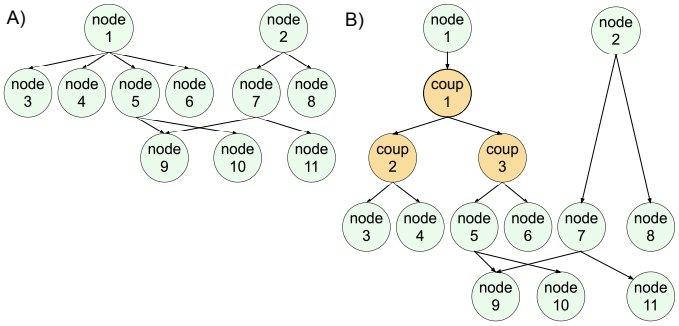

Figure 4: A) Graph before the execution of the fan-out control routine; B) Final graph after the execution of the fan-out limitation function

Graph is analyzed by the *fan-out management* routine. As for CMOS technology, the fan-out of a logic gate output represents the number of gates inputs that it can feed. Therefore, there is a limited number of fan-out that a logic gate can support. This concept has been extended to iNML technology in a similar way, . The task of this step is to map the initial graph, which contains functional nodes with an arbitrary fan-out into a new one compliant with the maximum fan-out affordable for the iNML technology. The nodes within the iNML Graph are visited, and if the number of children is higher than the maximum fan-out, additional levels and nodes are added according to the following inequality:

$$maxFanOut^{l} < n \leq maxFanOut^{l+1}, \qquad (1)$$

where n is the number of children that are fed by the parent node. The inequality is solved iteratively starting from l=0. Specifically, the additional elements are *coupler* nodes. An example of a graph before and after the execution of the fan-out management routine is depicted in Fig. 4.

In the iNML Graph generated from the VHDL netlist, many reconvergent paths may occur. As stated in [33], two paths (p) and (q) are called reconvergent if they diverge from and reconverge to the same blocks. An example is depicted in Fig. 5.A. This is a common situation in CMOS-based electronic circuits, whereas this is a big issue for iNML circuits due to their intrinsic pipelined behavior and the multi-phase clocking system. This is known as the ”layout=timing” problem [34]. Therefore, all reconvergent paths must be balanced in order to guarantee the correct information propagation within the circuit. The graph shown in Fig. 5.A represents an unbalanced iNML Graph. Here, two reconvergent paths can be identified. The *input1* runs through two branches before reaching the output node. Since the paths are not balanced, one of the two branches reaches

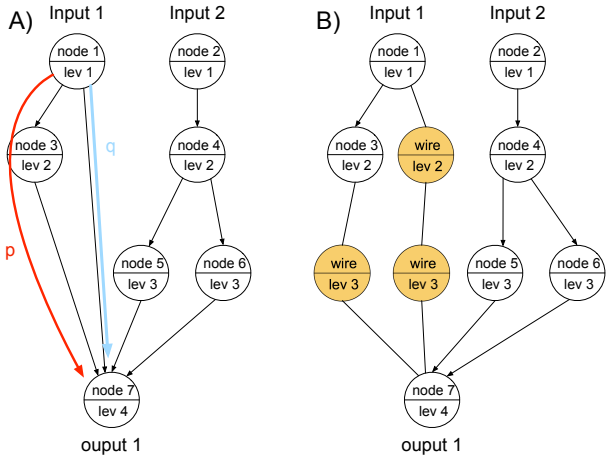

Figure 5: A) iNML Graph unbalanced; B) iNML Graph after wire blocks insertion

the output node before the other. The problem can be solved by adding wire blocks, which represent simple connections between two adjacent nodes. The same iNML Graph, this time with balanced paths is shown in Fig. 5.B. From this picture, it is evident that the paths' synchronization will increase the overall area of the circuit. However, the algorithm tries to share wire blocks when possible in order to reduce the area overhead. At the end of this step, the correct signal propagation is ensured over the all netlist.

The next objective of the layout process consists in the evaluation of the positions of each node with respect to the 2D space. This is performed in order to reduce the number of crossings etc... To achieve this goal, a position to each node must be assigned. The node position represents the *virtual* column of the node which belongs to. Two different approaches have been developed to assign position to nodes. The first assigns a fictitious position by scanning the iNML Graph level by level. The position is a number ranging from 1 to N, where N is the number of nodes of the level under examination. As an example, the assigned position for a 2-to-1 multiplexer netlist is reported in Fig. 6.A. However, this method follows the natural ordering of nodes in memory, according to the circuit netlist.
Better results can be obtained when the numbering of the nodes is carried out with the Breadth-Fist-Search (BFS) algorithm. Indeed, using this approach to assign the positions, it is possible to automatically obtain a reduction of total number of crossings. The BFS explores the neighboring nodes (children) before moving to the next level. The *pos* variable starts from 1 to N, where N is the sum of the children of each node at level i. Since multiple input nodes can be present inside the circuit, a simple trick has been applied to create a root in the iNML. A dummy node with level 0 is introduced on top of the inputs, before the BSF is applied. The final result considering the same 2-to-1 multiplexer is reported in Fig. 6.B.

The objective of the last step of the graph elaboration

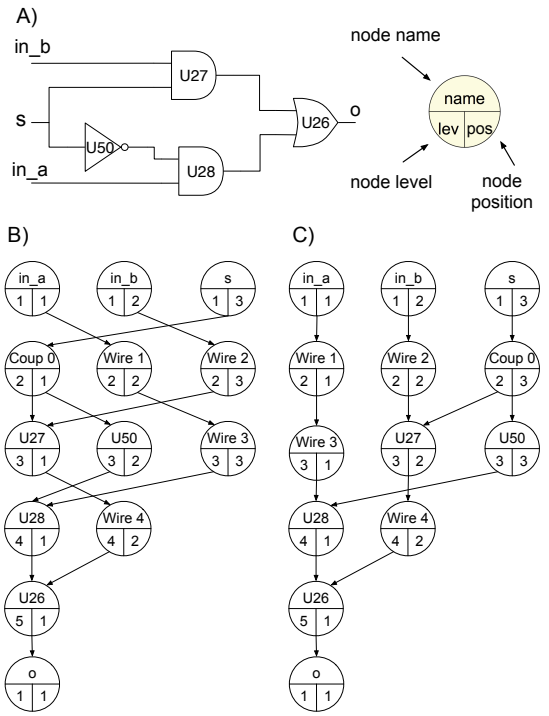

Figure 6: A) RTL view of a 2 to 1 multiplexer; Position assignment using the: B) fictitious position; C) Breadth-first-search algorithm

phase is to reduce the number of crossings among graph edges in order to minimize the routing area. The total number of crossings affects the final area of the circuit. Up to now, only planar implementation of iNML technology have been proposed. Therefore, logic functions and interconnections belong to the same physical plane. As a consequence, the connectivity must be ensured within a single layer, where both logic gates and routed signals are placed. Due to the planarity of the technology, rearranging the nodes is the only way to reduce wire crossings. The problem of finding the minimum number of crossing is NP-complete [35]. Here, we employ two heuristic algorithms: i) the *Barycenter* and ii) *Kerninghan-Lin* [36]. Another interesting technique is the fan out duplication [37], this basically tries to reduce the length of long wires by duplicating nodes. For the sake of brevity, this approach is not analyzed here eve if ToPoliNano can support it.

The Barycenter method implemented in ToPoliNano is applied to each layer of the iNML Graph. The position associated to nodes is used as weighted contribution for the computation of the barycenter value. Therefore, nodes are rearranged according to the calculated barycenter. In particular, the algorithm tries to place nodes directly above their children (fan-out) or parent (fan-in) in order to reduce the final number of cross wires (XW). Two versions of the barycenter method have been implemented: i) the *down-bary* considers children nodes and ii) the *up-bary* considers parent nodes.

To achieve better results, the algorithm is not applied in the classical way by scanning the graph levels from outputs to inputs [37]. In this version of the barycenter, as a first step, the down-bary is applied to the level that contains the highest number of crossings. After that, the down-bary is executed on the remaining layers in the upper side of the graph, whereas the up-bary is executed to the layers belonging to lower side of the iNML Graph.

The Kerninghan-Lin (KL) heuristic is one of the most popular algorithms for graph partitioning. In the following, an adaptation for the iNML technology is presented. For the sake of brevity, only the differences with respect to the standard version of the algorithm is highlighted.

The general objective of a partitioning algorithm is to partition a circuit into two parts such that the number of connections among the sub-circuits is minimized. The algorithm takes as input a DAG $G(V,E)$, this represents the circuit with $V = 2n$ nodes. The graph consists of vertices ($v \in V$) that have the same weights and edges ($e \in E$) characterized by non-negative weights. The aim of the algorithm is to find two disjoint partitions A and $B \in V$ with minimum *cut cost* and equal size ($|A| = |B| = n$). The algorithm is iterative; thus it tries to find an acceptable solution during each step (m). At the iteration m, the algorithm tries to swap pairs of nodes (each one from different partition) that generate the smallest increase in the cut size. The algorithm stops when no further improvements are possible. It is important to remember that in iNML technology,

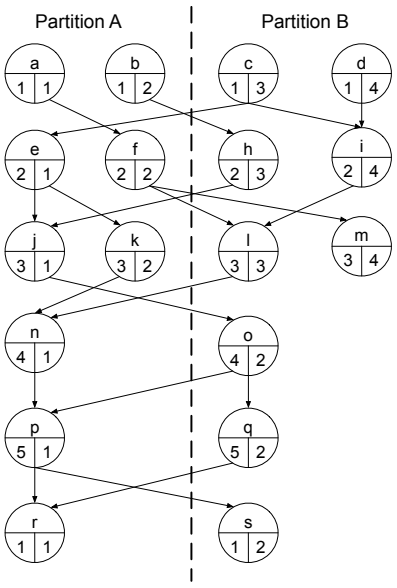

Figure 7: Partition example of the iNML Graph

circuits and as consequence graphs, are organized in levels. Nodes within the same level cannot have connections among each other as may happen in CMOS logic circuits. Therefore, it is not possible to move nodes from one level to another due to clock layout constrains of the technology. Due to technological constrains, nodes can be moved only within the same level. As a consequence, the standard implementation of KL algorithm must be modified to determine the gain obtained from each couple of nodes in the different levels. Before starting with the execution of the KL, it is important to change node positions

in order to have two disjoint partitions (Fig. 7). Moreover, the iNML Graph must be balanced if an odd number of nodes is present. This is done by analyzing the graph level by level, and if necessary, a dummy node is added in the level under examination. At the end of the execution, the dummy nodes are removed. At this point, the core of the KL can start. Once the gain of each level is computed, the nodes that produce the highest gain are swapped. This definitely increases the final computation complexity of the algorithm compared to the standard version. The algorithm is applied recursively within each sub-partition of A and B

The aim of the last step of the graph elaboration phase, named cross-node creation, is to identify edges intersections within the iNML Graph and to map them as specific blocks. Therefore, the iNML Graph is analyzed level by level and if a crossing is identified, a cross-wire node is introduced. This step has been introduced to simplify the routing of the interconnection in the physical mapping phase.

Physical Mapping: The iNML Graph, already optimized during the graph elaboration phase, is now taken as input in order to complete the physical design. Due to the high complexity of ICs, the physical mapping cannot be completed in a single phase. Therefore, the main flow of the physical mapping has been divided in four parts which are summarized in Fig. 2. The procedure starts with the *Block Placement* where each node of the iNML Graph is translated into its corresponding logic gate. Once all the blocks have been placed, the *Global Routing* phase takes place. This means that the final position of each element is defined. At this point, the layout can be completed by generating the interconnections among the blocks in the *Channel Routing* phase. As last step, the final data structure, which represents the layout, is translated into graphical objects in order to be shown to the user.

The block placement represents the first step of the physical mapping. Here, the iNML Graph is analyzed level by level and every node is translated into its corresponding logic gate. For each basic element, a block with proper shape and size has been defined. During the node translation, blocks are placed starting from the origin of the row (0,0). Moreover, they are aligned by considering a minimum spacing, equivalent to the width of one magnet (Fig. 8.B). This first attempt of placement is necessary to evaluate the maximum width of the circuit which corresponds to the widest row. Moreover, it is important to remember that at this point all the rows are overlapped since no vertical information is present. In other words, only the maximum height of the row is saved, this depends on the highest block instantiated. In order to minimize the overall wire length, each Row object is analyzed and blocks are shifted of an amount equal to the half of the distance with respect to the system barycenter (Fig. 8.C). Moreover, during this phase top and bottom pins are defined for each row. They do not refer to the physical position of pins but to their absolute position (x) within the row. At this point, since all the blocks and their absolute position are defined, it is possible to define the netlist. The netlist is used afterwards to refine the block position and define the channel before the routing takes place.

Before moving to the routing phase, it is important to refine the position of each block within the rows. Different techniques

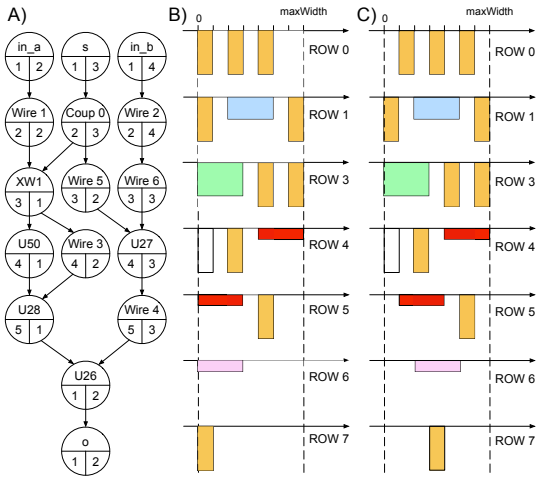

Figure 8: A) iNML Graph of a 2 to 1 multiplexer after the graph elaboration phase; B) fist attempt of placement; C) barycentered placement

can be applied to shift the blocks. In [38], simulated annealing and ghost net minimization have been proposed. However, the algorithm proposed in the following allows to achieve the highest compaction with respect to the aforementioned methods. The algorithm tries to move blocks towards the sides of each row. The algorithm seeks the widest row and then visits rows below and above it, one at time, by trying to shift blocks starting from the center. During the movement of the elements, the local position for consequent rows that contains cross wire blocks is preserved. This means that the algorithm tries to keep the local compactness of subsequent rows that have a high density of cross wire blocks. This algorithm is quite similar to the local barycenter considering leaf blocks. However, the regions characterized by a high density of crossing are managed in a different way. Indeed, in these regions, better results can be obtained by placing cross wire blocks in a compact way, by leaving between them the minimum horizontal distance. However, if other blocks are present in rows with higher density of cross wires, they are shifted towards the sides. At this point, the final positions of the blocks and the pins positions are defined. However, since channels are rectangular, it is important to fill the empty space that may be present inside the row inserting wires. These wires are used to link the pins of the output port of a block with the corresponding position in the imaginary bottom border of the row.

The Channel Routing represents the last step of the physical design. It performs the physical connections between rows. To route the interconnections, a specific algorithm has been implemented. It takes as a input the previously defined netlist, with the definition of top and bottom pins. The routing is performed by means of horizontal and vertical wires. Cross wires are not used during the channel routing phase. The defined netlist is already cross wire free by construction. The channel is defined according to the top and bottom pin position of the netlist previously defined. The routing is performed according to the following rules: i) the horizontal minimum

distance is set to 1 magnet, ii) vertical minimum distance is set to 1 magnet, iii) maximum number of horizontal and vertical magnets can be tuned by the user. In iNML technology, there is a limit to the maximum number of element that can be cascaded in a single clock zone. However, in order to be flexible, this technological constraint can be tuned by the user. As a consequence, this limitation, combined with the clock zone constraint implies that the routed signals follow a stair-like behavior when long connections have to be generated. An example of layout generated by ToPoliNano is reported in Fig. 10.

The hierarchical approach follows the same sequence of operations. Notwithstanding, with this method sub-circuits already present within the library are adopted in order to compose larger circuits. As an example, while designing a RCA with a fully hierarchical approach, FA library components are instantiated and considered as nodes of the graph. In this way, further high-level optimizations can be performed. In case no sub-circuits are found in the library, ToPoliNano recognize smaller sub-circuits and generates first their layouts and uses them as components in the final design.

D. Simulation & Fault Analysis

With the simulation engine [25] it is possible to verify the behavior of the circuit. The layout generated by the previous step is manipulated and a flattened matrix-like structure is created. An overview of the working principle is depicted in Fig. 9.

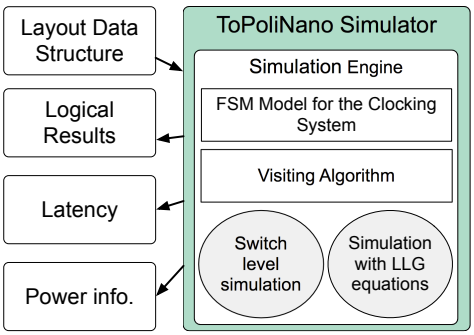

Figure 9: Simulator working principle

The control of the simulation algorithm has been captured and encapsulated in a finite-state machine (FSM). The general simulation algorithm follows the FSM as it switches periodically through its states.

This mechanism is adopted to represent the time evolution within the simulator. This makes it possible to reproduce the way information propagates through the circuit. The periodic behavior of the clocking mechanism is particularly suitable to be captured and encapsuled as a FSM; hence, this works as a controller for the clock signals distribution. Transitions among states determine whether the clock zones are in *Reset, Hold, Switch*conditions. Each clock zone where the *Switch* state is active is scanned thanks to a specific exploration algorithm. Here, the new value of magnetization of each magnet is calculated.

To evaluate the magnetization status of each element in the switch zone, two engines have been developed. First, the **High level simulator** is a very fast switch level simulator. It allows the verification of the logic behavior of the circuit. The working principle is based on the ferromagnetic and antiferromagnetic interaction among neighboring cells. The evaluation of the logic status of each cell is approximated by limiting the neighbors influence to the 8 adjacent magnets. This algorithm performs a weighted sum of the magnetization contributions given by the 8 possible neighbors. Initially, magnets coupled at north, south, west and east are considered. Then, if they do not exist, diagonal coupled magnets are considered since their contribution is meaningful only if there are no stronger contributions.

Second, with the **Low level simulator** it is possible to obtain more accurate information. Another engine has been developed based on a simplified formulation of the LLG equations, which describe the dynamic interaction between two magnetic nanoparticles [23] [14]. This simulation, which represents an intermediate level between switch level simulators and micromagnetic ones, better reflects the physical behavior of circuits. Additionally, it is 600 times faster than traditional micromagnetic simulators.

The intrinsic flexibility of the low level simulation algorithm makes it possible to evaluate the impact of misalignment error caused during the fabrication process. In [25] a detailed description of the implemented algorithm is given. In order to verify the logic correctness of circuits, exhaustive simulations are performed. Results of each circuit have been compared to a previously realized golden model.

IV. Results

The layout engine of ToPoliNano has been tested with several adders with different data parallelism. In addition, the well-known benchmark from ISCAS 1985 has been used as a test. The layouts reported in the following have been obtained running ToPoliNano on a laptop with CentOS 6.7, Intel Core i5 and 12 GB of RAM. Firstly we present the results obtained with different implementations of a ripple carry adder, and then, we report the data obtained with the ISCAS 85 benchmark. For the circuits here considered, the information extracted can be grouped into three categories: i) iNML Graph parameters, ii) the execution time of the physical design, iii) layout based data. The following parameters have been taken into account with the iNML Graph:

- number of initial crossings (#CW)
- number of crossings after Breath-First-Search (BFS)
- number of crossings after Barycenter method
- number of crossings after Kerninghan-Lin (KL)
- percentage of reduction considering each algorithm

The time-related parameters are the main steps of the physical design and the processing time of the cross wire minimization algorithms. They are the time taken to run the BFS, Barycenter method, KL, block placement routine and channel routing routine. The relevant data about the final layout are:

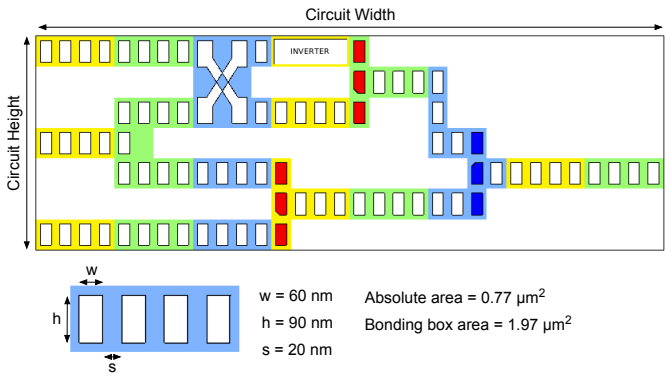

Figure 10: Example of iNML layout generated with ToPoliNano where the main geometrical metrics are reported

- total number of magnets
- total number of clock zones
- bounding box area, i.e. the area of the minimum rectangle that bounds the layout
- absolute area, i.e. the area occupied by all the magnets plus the inter-magnet space
- percentage of occupation

Moreover, we analyze and discuss the impact of the signals' synchronization. For all benchmarks, the total circuit area has been computed considering 60nm x 90nm nanomagnets, with 20nm spacing among each magnet in both the vertical and the horizontal direction. The drawing depicted in Fig. 10 summarizes the meaning of the most important geometrical quantities.

A. *Ripple Carry Adders*

This section reports layouts of ripple carry adders with different design constraints. In the first part, we adopted the classical implementation of the full adder, whereas in the second part we generated the ripple carry adders using full adders made by majority gates. This enabled the exploration of the majority function by increasing the final compactness of the design. In the final part of this section, a comparison between the number of magnets and the execution time is given to highlight the benefit of a fully hierarchical approach with respect to the flat one.

1) Standard RCA implementation: As a first benchmark, we designed RCAs with a different number of bits (1 to 64), by using a maximum of 4 magnets per clock zone and 2 magnets for vertical connections. Table I reports all data related to the graph representation of these circuits. In particular, the effectiveness of all the crossing minimization algorithms is compared.
The benchmarks were run by applying the two sequences of operations listed in the following:

- a fictitious position of nodes is set. After that, the BFS was applied, followed by the Barycenter method

Nbits	#CW Init	#CW BFS	#CW Bary	#CW KL	% Reduc. BFS	% Reduc. Bary	% Reduc. KL
1	29	16	5	15	44.83	68.75	6.25
2	111	47	32	46	57.66	31.91	2.13
4	375	165	139	179	56	15.76	-8.48
8	1237	556	381	673	55.05	31.47	-21.04
16	4249	2156	1544	2898	49.26	28.39	-34.42
32	17793	11721	6667	12981	34.13	43.12	-10.58
64	66907	46484	27452	54759	30.52	40.94	-17.8

Table I: Comparison between cross wire reduction algorithms for different RCAs

Nbits	# Magnets	# Clock Zones	Bound. Area [mm^2]	Abs. Area [mm^2]	% Occ.
1	711	30	1.9E-05	4.12E-06	21.7
2	3567	95	9.7E-05	2.07E-05	21.3
4	14987	219	4.24E-04	8.69E-05	20.5
8	70891	582	2.07E-03	4.11E-04	19.9
16	403132	1616	1.12E-02	2.34E-03	20.9
32	2652140	5535	7.58E-02	1.54E-02	20.3
64	19739987	20250	5.51E-01	1.14E-01	20.8

Table II: RCAs layout comparison using 4 magnets per clock zone and a maximum of 2 magnets for vertical connections. All the layouts have been obtained combining BFS and Barycenter methods during the graph processing phase

- a fictitious position of nodes is set. After that, the BFS was executed, followed by the Kerninghan-Lin algorithm

The number of crossings is reduced by 47% just applying the BFS. Further crossing reductions are obtained by applying the Barycenter after the BFS, in this case the average reduction proves to be an additional 37%. Indeed, by combining the BFS and the Barycenter methods together, it is possible to achieve a remarkable (67%) reduction in the number of cross wire. On the other hand, the sequence BFS plus KL is less effective since only a 45% reduction can be achieved.
As expected, the total number of cross wires directly affects the final area of the circuit. Table II and table III show the bounding box area, the absolute area and the percentage of occupation for all the RCAs of the layouts generated using the BFS plus Barycenter and the BFS plus KL. It is possible to see that the bounding box area is smaller than 1 mm^2 for all the

Nbits	# Magnets	# Clock Zones	Bound. Area [mm^2]	Abs. Area [mm^2]	% Occ.
1	919	35	2.22E-05	5.33E-6	24
2	4238	102	1.04E-04	2.46E-05	23.6
4	20146	301	5.62E-04	1.17E-04	20.8
8	127115	1023	3.64E-03	7.37E-04	20.3
16	867853	3625	2.62E-02	5.03E-03	19.2
32	6064901	12831	2.43E-01	3.52E-02	14.5
64	61749022	72475	2.7	3.58E-01	13.28

Table III: RCAs layout comparison using 4 magnets per clock zone and a maximum of 2 magnets for vertical connections. All the layouts have been obtained combining BFS and Kernighan Lin methods during the graph processing phase

Nbits	# Magnets	# Clock zones	Bound. Area [mm^2]	Abs. Area [mm^2]	% Occ.
1	716	29	1.74E-5	4.15E-06	23.9
2	2685	66	7.2E-05	1.56E-05	21.6
4	11923	169	3.15E-04	6.92E-05	21.9
8	57726	448	1.59E-03	3.35E-04	21
16	284660	1149	7.97E-03	1.65E-03	20
32	1385741	3019	4.13E-02	8.04E-03	19.4
64	11171240	11713	3.19E-01	6.48E-02	20.3

Table IV: RCAs layout comparison using 4 magnets per clock zone and domain wall for vertical connections. All the layouts have been obtained combining BFS and Barycenter methods during the graph processing phase

Nbits	# Magnets	# Clock zones	Bound. Area [mm^2]	Abs. Area [mm^2]	% Occ.
1	704	28	1.68E-05	4.08E-06	24.4
2	2629	67	6.84E-05	1.52E-05	22.2
4	15274	221	4.12E-04	8.86E-05	21.5
8	80373	589	2.16E-03	4.66E-04	21.6
16	702907	2689	1.92E-02	4.08E-03	21.2
32	4267616	9255	1.79E-01	2.48E-02	13.9
64	47576166	52299	1.99	2.76E-01	13.86

Table V: RCAs layout comparison using 4 magnets per clock zone and domain wall for vertical connections. All the layouts have been obtained combining BFS and Kernighan-Lin methods during the graph processing phase

RCAs except the 64-bit adder optimized with the KL. For all the circuits, the average percentage of occupation ranges from 20% to 24%. This is mainly due to the clock zone layout and to the fact that iNML is a planar technology.
Other important parameters that we analyzed are the graph processing time and the time required by the main steps of the design process.

The most time consuming physical design steps are the placement and the routing phases.
The above analysis has considered very tight layout constraints, i.e. 4 magnets per clock zone and a maximum of 2 magnets for vertical connections. In the following, the already mentioned RCAs have been redesigned using the same number of magnets per clock zone, but using DWs instead of only 2 magnets for vertical interconnections. However, the maximum height of the domain wall has been limited to ten times the height of the magnet. The first thing that it is possible to notice by looking at the tables IV-V, is that the area has been greatly reduced. The data show that approximately 37% of the bounding box area can be saved when compared to the previous results obtained by applying the Barycenter method.
On the other hand, by comparing data obtained employing the KL it is possible to observe around 30% reduction. The introduction of domain walls for routing the channel greatly increases the circuit compaction.

2) Fully hierarchical vs. Flat layout: In this paragraph, the results already discussed, obtained with the Flat method are compared with the Fully hierarchical approach. The same

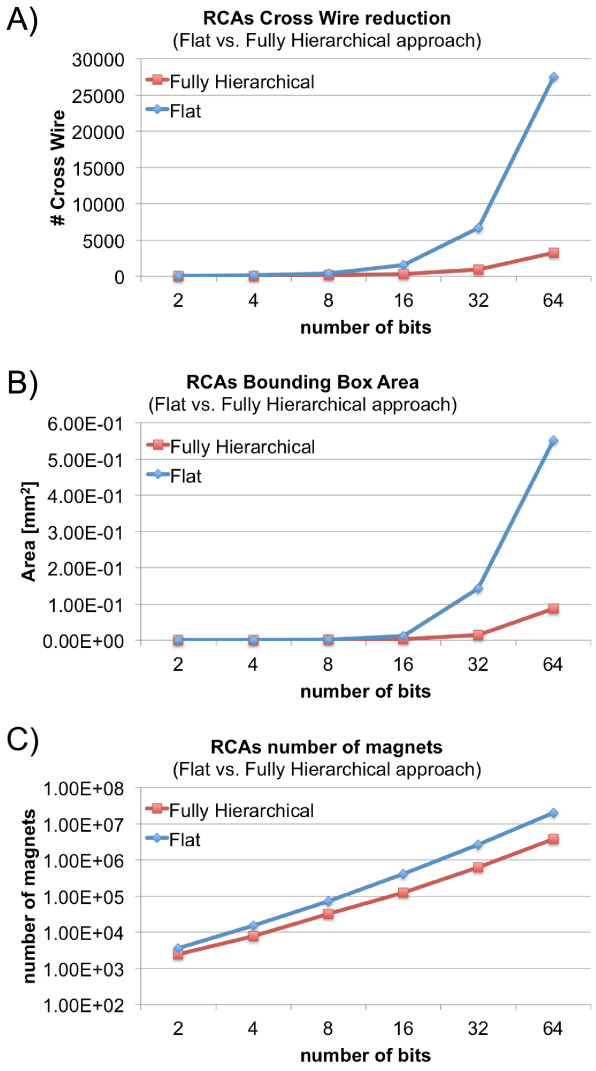

Figure 11: RCAs comparison between Flat and Fully Hierarchical approaches. The layouts have been designed considering 4 magnet per clock zone and 2 magnets for vertical interconnections. The graphs report: A) the total number of cross wire; B) the bounding box area; C) the total number of magnet.

RCAs are considered by varying the number of bits from 2 to 64. All adders have been designed by applying the algorithm sequence BFS plus Barycenter method. From Fig. 11.B and Fig. 11.C, it is possible to notice that the Fully hierarchical approach presents a remarkable improvement in terms of occupied area and total number of magnets, if compared to the flat approach.
Hence, with the second approach approach, interconnections are optimized determining a higher compactness. The average cross wire reduction achieved with the hierarchical method is 74.5% reaching a peak of 88% with the 64-bit adder

(Fig. 11.A). In a similar way, the bounding box area is greatly reduced. Here, the flat layouts are in average 63% larger than one obtained with the hierarchical method.

3) RCA based on Majority Voters: In the last part of the analysis of RCA circuits, the benefit of the introduction of the majority based circuit is presented [39]. Indeed, the following results exploit the higher compactness of the full added built using majority gates (Fig. 12). Indeed, it consist only of three majority voters and two inverters. This full adder has been used to design RCAs with different number of bits (from 2 up to 64). It is evident that, adopting a simple full adder, it is possible to obtain much more compact circuits. Therefore, the iNML Graphs processed are smaller if compared to the previous one (with an equal number of bits). This can be observed by looking at table VI where the number of cross nodes before and after the optimization are reported. A more clear comparison of the number of crossings between the two RCAs implementations is given by the bar chart in Fig. 13. For the sake of clarity, data on the y axis are reported on a logarithmic scale. The results about the area occupation are reported in tables VII-VIII. As expected, in both cases the final area is reduced if compared to the standard implementation of the full adder. However, in some cases, the percentage of occupied area is lower with respect to the previous versions of the circuits. The last important aspect to be considered is the impact of paths synchronization. This can be influenced by both the clock zone layout and the intrinsic pipelining of the iNML technology. Fig. 14 shows the amount of nodes inside the graph after the first steps of the graph elaboration phase, i.e. the fan out management (FOC) and the reconvergent paths balance (RPB) routines. It is possible to observe that the number of nodes introduced by the RPB grows very quickly

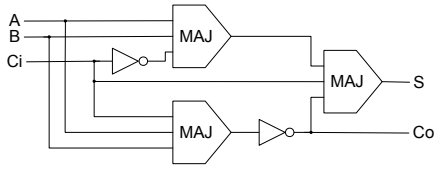

Figure 12: RTL view of a full adder made by majority voters

Nbits	#CW Init	#CW BFS	#CW Bary	#CW KL	% Reduc. BFS	% Reduc. Bary	% Reduc. KL
2	58	25	22	21	62.06	12	16
4	140	85	48	81	65.71	43.53	4.7
6	242	107	58	107	76.03	45.79	0
8	420	251	113	243	73.09	54.98	3.18
10	604	323	205	325	66.05	36.53	-0.61
16	1216	725	543	732	55.34	25.1	-0.95
28	3233	1825	1229	1866	61.98	32.66	-2.24
32	4294	3150	990	3231	76.94	68.57	-2.57
48	8997	7576	4077	7968	54.68	46.19	-5.17
64	14372	10200	5735	10668	60.09	43.77	-4.5

Table VI: Comparison between cross wire reduction algorithms for different RCAs based on majority gates

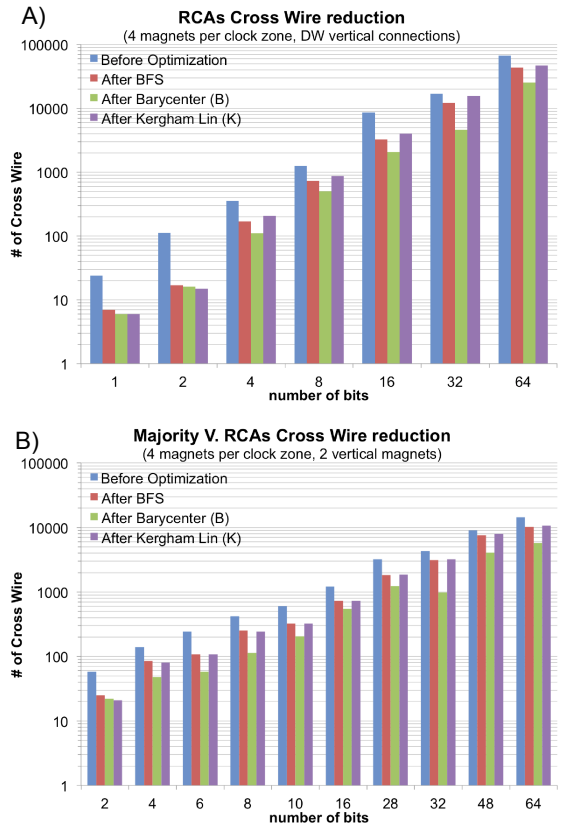

Figure 13: Comparison between cross wire minimization algorithm considering: A) standard RCAs; B) majority voter based RCAs

Nbits	# Magnets	# Clock Zones	Bound. Area [mm^2]	Abs. Area [mm^2]	% Occ
2	1385	44	3.72E-05	8.04E-06	21.6
4	4486	98	1.21E-04	2.6E-05	21.6
6	6681	115	1.9E-04	3.87E-05	20.4
8	14925	218	4.14E-04	8.66E-05	20.9
10	26937	330	7.09E-04	1.56E-04	22
16	99046	813	2.83E-03	5.74E-04	20.2
28	302749	1605	8.7E-03	1.76E-03	20.2
32	322847	1560	1.04E-02	1.87E-03	18
48	1513594	4918	4.55E-02	8.78E-03	19.3
64	2502933	6259	7.36E-02	1.45E-02	19.7

Table VII: Majority voter based RCAs layout comparison using 4 magnets per clock zone and a maximum of 2 magnets for vertical connections. All the layouts have been obtained by combining BFS and Barycenter methods during the graph processing phase

Nbits	# Magnets	# Clock Zones	Bound. Area [mm^2]	Abs. Area [mm^2]	% Occ
2	1572	52	3.84E-05	9.12E-06	23.7
4	6021	129	1.59E-04	3.49E-05	22
6	10267	173	3.35E-04	2.26E-05	6.74
8	28513	409	1.12E-03	1.65E-04	14.7
10	41813	498	1.84E-03	2.43E-04	13.2
16	116827	932	4.56E-03	6.78E-04	14.9
28	394527	1955	1.54E-02	2.29E-03	14.9
32	676969	3022	2.97E-02	3.93E-03	13.2
48	2213451	6832	1.4E-01	1.28E-02	9.2
64	4207667	10009	2.54E-01	2.44E-02	9.6

Table VIII: Majority voter based RCAs layout comparison using 4 magnets per clock zone and a maximum of 2 magnets for vertical connections. All the layouts have been obtained by combining BFS and Kernighan Lin methods during the graph processing phase

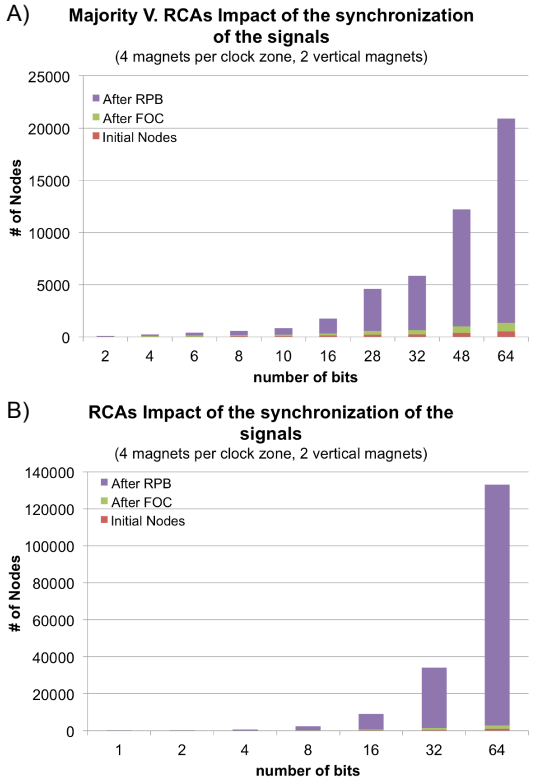

Figure 14: Number of nodes after fan out control and reconvergent path balance routines for: A) majority voter based RCAs; B) normal RCAs

increasing the size of the circuit. The same behavior can be observed by considering the majority gate based RCAs (Fig. 14.A). All these additional nodes have a remarkable impact on the final area of the circuit. For instance, in the 32-bit RCA built using standard full adders, after the execution of the fan out control routine, 898 nodes are present within the iNML graph. However, after the equalization of the reconvergent paths, the number of nodes reaches 32852; this means that the number of nodes is increased by 6291%.

B. ISCAS 85

The ISCAS '85 benchmark consists of a set of combinationational circuits provided by Bryan [24] at the International Symposium on Circuits And System in 1985. Tab. IX reports

Circuit Name	Function	# gates
c17	six NAND gates	6
c432	Priority Encoder	160
c499	ECAT	202
c880	ALU and Control	383
c1355	ECAT	546
c1908	ECAT	880
c2670	ALU and Control	1193
c3540	ALU and Control	1669
c5315	ALU and Selector	2307
c6288	16-bit Multiplier	2406
c7552	ALU and Selector	3512

Table IX: ISCAS '85 benchmark circuits characteristics

Circuit	#CW Init	#CW BFS	#CW Bary	#CW KL	% Reduc. BFS	% Reduc. Bary	% Reduc. KL
c17	20	6	5	6	70	16.66	0
c432	4489	2650	1909	1564	40.96	27.96	40.98
c499	45148	18220	14350	18579	59.64	21.24	-5.31
c880	29012	8124	7014	8785	72	13.66	-8.14
c1355	39993	16852	12340	17469	57.86	26.77	-3.66
c1908	22466	7247	5730	7560	67.74	20.82	-4.32
c2670	80451	21556	16069	23495	73.2	25.45	-8.99
c3540	82291	26202	24593	28662	68.16	6.14	-9.24
c5315	415823	110806	80223	125770	73.35	27.6	-2.02
c6288	678982	153579	139182	158381	77.38	9.38	-3.13
c7552	390525	121481	99847	122092	68.89	18.81	-0.5

Table X: Comparison between cross wire reduction algorithms using ISCAS 85 netlist

the function implemented by each circuit. These benchmarks have been widely used by the research community in order to compare results in the area of test generation. However, the circuit netlists are provided in structural Verilog format. In order to be compatible with ToPoliNano, all the circuits have been re-mapped into VHDL by using a synthesizer. This flow exploits Synopsys in order to re-map the Verilog circuits into VHDL netlists that are made only by ANDs, ORs and INVERTERs. Even in this case, the benchmarks have been run by applying the two sequences of operations introduced for the RCAs. First, a fictitious position is assigned to each node, then the BFS is executed and followed by the Barycenter method (or the Kerninghan-Lin).

Table X reports the comparison between cross wire reduction algorithms. For these circuits, the average reduction in term of intersections obtained by using BFS is significantly increased to 64%. Even in this case, a further cross wires reduction is achieved by applying both Barycenter and KL algorithms (Fig. 15). The average improvement is limited to 20% and 0.3% for Barycenter and KL respectively. However, the overall crossings reduction obtained by combining BFS plus Barycenter or BFS plus KL ranges from 72% to 65%. As an example, the cross wire reduction obtained with the chain BFS plus

Barycenter, on the c1908 netlist, is 584% lower compared to the results presented by the authors in [37]. The execution

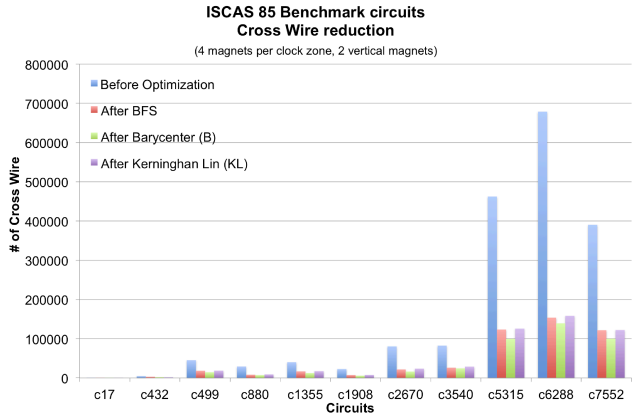

Figure 15: Comparison between cross wire minimization algorithm by considering the ISCAS 85 benchmark

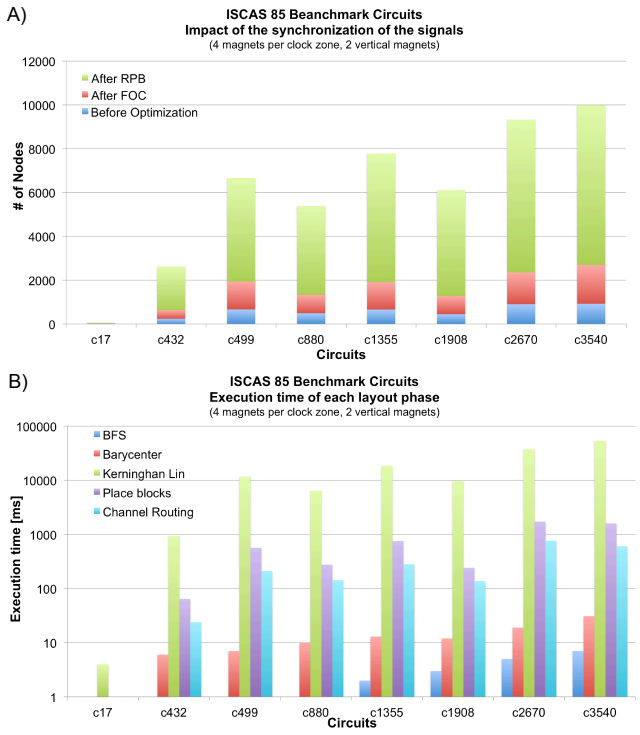

Figure 16: A) Number of nodes after fan out control and reconvergent path balance routines; B) Execution time of the main steps of the design flow by considering the ISCAS 85 benchmark

time required by the main steps of the layout process are summarized in Fig. 16.B. As expected, the KL is the most time consuming compared to the other phases. Fig. 16.A reports the number of nodes introduced by the first steps of the graph elaboration phase. Even in this case, the disadvantage of the clock zone layout and the intrinsic pipelining of technology is evident. Most of the nodes have been introduced to balance reconvergent paths. Here, the increase in the number of nodes is limited to 660% if compared with an average of 3566% obtained with the RCAs netlist.

V. Conclusion

With this paper we have presented the complete flow of ToPoliNano, our CAD Tool for iNML technology. With respect to previous articles already available in literature that analyze specific aspects of the tool; here, we have introduced some novelties like the graphical editor MagCad, the partially and fully hierarchical approaches to layout and a complete benchmarking performed with the ISCAS 85 circuits.
Layout results have been widely discussed, a complete analysis of generic RCAs and ISCAS 85 circuits is provided to the reader. Here, it can be highlighted that, in general, the occupation percentage of the bounding box area is remarkably optimized by applying the BFS plus Barycenter algorithms. Moreover, it can be noticed that, the intrinsic limitation layout = timing, which characterize the iNML technology, is mirrored into the obtained result: the reconvergent paths balance algorithm, which synchronize signals, defines an considerable increase of the total area.
This tool represent a key point for the study of the iNML technology, further improvements will we implemented in the next months. We also aim to publicly release a version of ToPoliNano by the end of the year.

References

[1] C. Lent, P. Tougaw, W. Porod, and G. Bernstein, "Quantum cellular automata," *Nanotechnology*, vol. 4, no. 1, p. 49, 1993.

[2] A. Chaudhary, D. Z. Chen, X. S. Hu, M. T. Niemier, R. Ravichandran, and K. Whitton, "Fabricatable interconnect and molecular qca circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 11, pp. 1978–1991, Nov 2007.

[3] M. L. C. L. Y. Lu, "Molecular electronics - from structure to circuit dynamics," in *Sixth IEEE Conference on Nanotechnology*. Cincinnati-Ohio, USA: IEEE, 2006, pp. 62–65.

[4] R. Wang, A. Pulimeno, M. Roch, G. Turvani, G. Piccinini, and M. Graziano, "Effect of a clock system on bis-ferrocene molecular qca," *IEEE Transactions on Nanotechnology*, vol. PP, no. 99, pp. 1–1, 2016.

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

[6] M. Niemier and al., "Nanomagnet logic: progress toward system-level integration," *J. Phys.: Condens. Matter*, vol. 23, p. 34, Nov. 2011.

[7] M. Vacca, M. Graziano, L. D. Crescenzo, A. Chiolerio, A. Lamberti, D. Balma, G. Canavese, F. Celegato, E. Enrico, P. Tiberto, L. Boarino, and M. Zamboni, "Magnetoelastic clock system for nanomagnet logic," *Ieee Transaction On Nanotechnology*, vol. 13, no. 5, September 2014.

[8] D. Pala, G. Causapruno, M. Vacca, F. Riente, G. Turvani, M. Graziano, and M. Zamboni, "Logic-in-memory architecture made real," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2015, pp. 1542–1545.

[9] M. Cofano, G. Santoro, M. Vacca, D. Pala, G. Causapruno, F. Cairo, F. Riente, G. Turvani, M. R. Roch, M. Graziano, and M. Zamboni, "Logic-in-memory: A nano magnet logic implementation," in *2015 IEEE Computer Society Annual Symp. on VLSI*, July 2015, pp. 286–291.

[10] M. Alam, M. Siddiq, G. Bernstein, M. Niemier, W. Porod, and X. Hu, "On-chip Clocking for Nanomagnet Logic Devices," *IEEE Transaction on Nanotechnology*, 2009.

[11] A. Chiolerio, P. Allia, and M. Graziano, "Magnetic dipolar coupling and collective effects for binary information codification in cost-effective logic devices," *Journal of Magnetism and Magnetic Materials*, no. 324, pp. 3006–3012, 2012.

[12] A. Papp, M. Niemier, A. Csurgay, M. Becherer, S. Breitkreutz, J. Kiermaier, I. Eichwald, X. Hu, X. Ju, W. Porod, and G. Csaba, "Threshold gate-based circuits from nanomagnetic logic," *Nanotechnology, IEEE Transactions on*, vol. 13, no. 5, pp. 990–996, Sept 2014.

[13] G. Turvani, F. Riente, M. Graziano, and M. Zamboni, "A quantitative approach to testing in quantum dot cellular automata: Nanomagnet logic case," in *Ph.D. Research in Microelectronics and Electronics (PRIME), 2014 10th Conference on*, June 2014, pp. 1–4.

[14] G. Csaba, M. Becherer, and W. Porod, "Development of cad tools for nanomagnetic logic devices," *International Journal of Circuit Theory and Applications*, vol. 41, no. 6, pp. 634–645, 2013.

[15] G. Causapruno, F. Riente, G. Turvani, M. Vacca, M. R. Roch, M. Zamboni, and M. Graziano, "Reconfigurable systolic array: From architecture to physical design for nml," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. PP, no. 99, pp. 1–10, 2016.

[16] M. Graziano, M. Vacca, A. Chiolerio, and M. Zamboni, "A NCL-HDL Snake-Clock Based Magnetic QCA Architecture," *IEEE Transaction on Nanotechnology*, vol. 10, no. 5, pp. 1141–1149, Sep. 2011.

[17] V. Vankamamidi, M. Ottavi, and F. Lombardi, "Two-dimensional schemes for clocking/timing of qca circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 1, pp. 34–44, Jan 2008.

[18] G. Csaba and W. Porod, "Behavior of Nanomagnet Logic in the Presence of Thermal Noise," in *International Workshop on Computational Electronics*. Pisa, Italy: IEEE, 2010, pp. 1–4.

[19] M. Alam, J.DeAngelis, M. Putney, X. Hu, W. Porod, M. Niemier, and G. Bernstein, "Clock Scheme for Nanomagnet QCA," in *Inter. Conf. on Nanotechnology*. Hong Kong: IEEE, 2007, pp. 403–408.

[20] M. Graziano, A. Chiolerio, and M. Zamboni, "A Technology Aware Magnetic QCA NCL-HDL Architecture." Genova, Italy: IEEE, 2009, pp. 763–766.

[21] J. Das, S. Alam, and S. Bhanja, "Ultra-low power hybrid cmos-magnetic logic architecture," *Trans. on Computer And Systems*, 2011.

[22] G. Turvani, A. Tohti, M. Bollo, F. Riente, M. Vacca, M. Graziano, and M. Zamboni, "Physical design and testing of nano magnetic architectures," in *Design Technology of Integrated Systems In Nanoscale Era, 2014 9th IEEE International Conference On*, May 2014, pp. 1–6.

[23] G. Csaba, W. Porod, and A. Csurgay, "A computing architecture composed of field-coupled single domain nanomagnets clocked by magnetic field," *international Journal Of Circuits Theory And Applications*, vol. 31, pp. 67–82, 2003.

[24] D. Bryan, "ISCAS '85 benchmark circuits and netlist format," in *International Symposium on Circuits and Systems*. Kyoto: IEEE, 1985.

[25] G. Turvani, F. Riente, F. Cairo, M. Vacca, U. Garlando, M. Zamboni, and M. Graziano, "Efficient and reliable fault analysis methodology for nanomagnetic circuits," *International Journal of Circuit Theory and Applications*, pp. n/a–n/a, 2016.

[26] M. Donahue and D. Porter, "Oommf users guide, version 1.0." National Institute of Standards and Technology, Tech. Rep., 1999.

[27] A. Vansteenkiste, J. Leliaert, M. Dvornik, M. Helsen, F. Garcia-Sanchez, and B. V. Waeyenberge, "The design and verification of mumax3," *AIP Advances*, vol. 4, no. 10, 2014.

[28] M. Graziano, M. Vacca, D. Blua, and M. Zamboni, "Asynchrony in Quantum-Dot Cellular Automata Nanocomputation: Elixir or Poison?" *IEEE Design & Test of Computers*, vol. 28, no. 5, pp. 72–83, Sep. 2011.

[29] M. Vacca, M. Graziano, and M. Zamboni, "Asynchronous Solutions for Nano-Magnetic Logic Circuits," *ACM J. on Emerging Tech. in Comp. Systems*, vol. 7, no. 4, December 2011.

[30] M. Awais, M. Vacca, M. Graziano, and G. Masera, "Quantum dot Cellular Automata Check Node Implementation for LDPC Decoders," *IEEE Tran. on Nanotechnology*, vol. 12, no. 3, pp. 368–377, 2013.

[31] F. Cairo, M. Vacca, M. Graziano, and M. Zamboni, "Domain magnet logic (dml): A new approach to magnetic circuits," in *IEEE International Conference on Nanotechnology*, 2014.

[32] M. Vacca, M. Graziano, and M. Zamboni, "Majority Voter Full Characterization for Nanomagnet Logic Circuits," *IEEE T. on Nanotechnology*, vol. 11, no. 5, pp. 940–947, Sep. 2012.

[33] R. Ravichandran and al., "Partitioning and placement for buildable QCA circuits ," *DAC*, vol. 1, 2005.

[34] M. Niemier and P. Kogge, "Problems in designing with QCAs: Layout = Timing," *Int. J. Circ. Theor. Appl*, 2001.

[35] K. Sugiyama, S. Tagawa, and M. Toda, "Methods for visual understanding of hierarchical system structures," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 2, pp. 109–125, Feb 1981.

[36] A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu, *VLSI Physical Design: From Graph Partitioning to Timing Closure*, 1st ed. Springer Publishing Company, Incorporated, 2011.

[37] W. Chung and al., "Node duplication and routing algorithms for quantum-dot cellular automata circuits ," *IEE Proc. on Circ., Dev. and Sys.*, vol. 153, no. 5, 2006.

[38] M. Vacca, S. Frache, M. Graziano, F. Riente, G. Turvani, M. R. Roch, and M. Zamboni, "ToPoliNano: NanoMagnet Logic Circuits Design and Simulation," *In: Anderson, N.G., Bhanja, S. (eds.), Field-Coupled Nanocomputing: Paradigms, Progress, and Perspectives. LNCS, Springer, Heidelberg.*, vol. vol. 8280, 2014.

[39] R. Zhang, P. Gupta, and N. K. Jha, "Majority and minority network synthesis with application to qca-, set-, and tpl-based nanotechnologies," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 7, pp. 1233–1245, July 2007.

Fabrizio Riente received his M.Sc. Degree with honors (Magna Cum Laude) in Electronic Engineering in 2012 and the Ph.D. degree in 2016 from the Politecnico di Torino. He is currently Postdoctoral Research Associate at the Technical University of Munich. His primary research interests are device modeling, circuit design for nano-computing, with particular interest on magnetic QCA. His interests cover also the development of EDA tool for beyond-CMOS technologies, with the main focus on the physical design.

Giovanna Turvani received the M.Sc. degree with honors (Magna Cum Laude) in Electronic Engineering in 2012 and the Ph.D. degree in 2016 from the Politecnico di Torino. She is currently Postdoctoral Research Associate at the Technical University of Munich. Her interests include CAD Tools development for non-CMOS nanocomputing, architectural design for nanomagnetic computing and device modeling.

Marco Vacca received the Ph.D. degree in Electronics and Communications engineering from the Politecnico di Torino, Turin, Italy, in 2013. He is now an Assistant Professor at the Politecnico di Torino. His research interests include magnetic and molecular devices and other beyond-CMOS technologies. He is also an expert of innovative and unconventional computer architectures.

Massimo Ruo Roch achieved the Electronics Engineering degree in 1965 and the Ph.D. degree in 1993 from the Politecnico di Torino. Since 2002, he has been a full-time Researcher at the Politecnico di Torino. His research interests include digital design of application specific computing architectures, high speed telecommunications, and digital television. Recent activities include design and modeling of MPSoCs, embedded systems for bio applications, and cloud-based systems for e-learning.

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

Maurizio Zamboni received the Electronics Eng. and the Ph.D. degrees in 1983 and in 1988 from the Politecnico di Torino, respectively, where he is currently 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.