POLITECNICO DI TORINO
Repository ISTITUZIONALE

Physical design and testing of Nano Magnetic architectures

Original
Physical design and testing of Nano Magnetic architectures / Turvani, Giovanna; A., Tohti; Bollo, Matteo; Riente, Fabrizio; Vacca, Marco; Graziano, Mariagrazia; Zamboni, Maurizio. - (2014), pp. 1-6. (Intervento presentato al convegno Design Technology of Integrated Systems In Nanoscale Era (DTIS), 2014 9th IEEE International Conference On tenutosi a Santorini nel 6-8 May 2014) [10.1109/DTIS.2014.6850676].

Availability:
This version is available at: 11583/2562941 since:

Publisher:
IEEE - INST ELECTRICAL ELECTRONICS ENGINEERS INC

Published
DOI:10.1109/DTIS.2014.6850676

Terms of use:

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

Publisher copyright

(Article begins on next page)

12 October 2024

Physical Design and Testing of Nano Magnetic Architectures

G. Turvani, A. Tohti, M. Bollo, F. Riente, M. Vacca, M. Graziano, M. Zamboni
Electronics and Telecommunications Department, Politecnico di Torino, Italy

***Abstract*—Nano-Magnetic Logic (NML) is a promising candidate to substitute CMOS technology since it is characterized by very low power consumption and it can combine computation and memory in the same device. Several works analyze this technology at device level; nevertheless a higher level analysis is required to fully understand its potentials. It is actually fundamental to analyze how an architecture of realistic complexity can be really implemented taking into account the physical limits due to technology, and which performance it could consequently reach.**

We present here a physical design and test methodology based on our tool ToPoliNano, which allows analyzing circuits using models specifically targeted for this technology. We developed an automatic engine for placing and routing combinational NML circuits including as constraints realistic rules due to currently available fabrication processes. After the place and route phase, ToPoliNano also allows to perform a circuit logical simulation, detailed at the single nanomagnet level. Furthermore this tool has the ability to analyze and test circuits based on NML, considering the impact that process variations and faults have on the logical behavior of the circuit.

I. Introduction

Quantum dot Cellular Automata (QCA) [1] is one of the most promising technologies studied for the post-CMOS scenario. It has two main implementations, Molecular QCA [2] [3] and NanoMagnets Logic (NML) [4]. While Molecular QCA allows to reach extremely high clock speeds [5], the technological feasibility is still beyond the possibilities offered by modern fabrication processes. We focus therefore our architectural studies on NML technology, where a good number of papers on experimental fabrication is already available in literature [6] [7] [8]. Figure 1.a shows a simple example of an NML circuit. Single domain nanomagnets are used to represent logic values '0' and '1'. Circuits are organized placing magnets on a plane. Information propagation and logic computation are obtained through magnetodynamic interaction among neighbor magnets. A wire (Figure 1.a) can be obtained simply aligning magnets horizontally, while functions like a logic AND (Figure 1.a) can be implemented for example changing the shape of magnets [9]. A particular block, the crosswire (Figure 1.a) [7] allows to cross two wires on the same plane without interferences. The crosswire is required to build any complex circuits, up to now no multilayer structures are possible in NML.

To successfully switch one magnets from one state to the other, a clock mechanism is required. Magnets are forced in an intermediate state thanks to an externally applied magnetic field. This magnetic field is normally generated using a current flowing through wires placed under the magnets plane (Figure 1.a), since it is the only solution experimentally demonstrated up to now [7]. Moreover, thanks to the presence of thermal

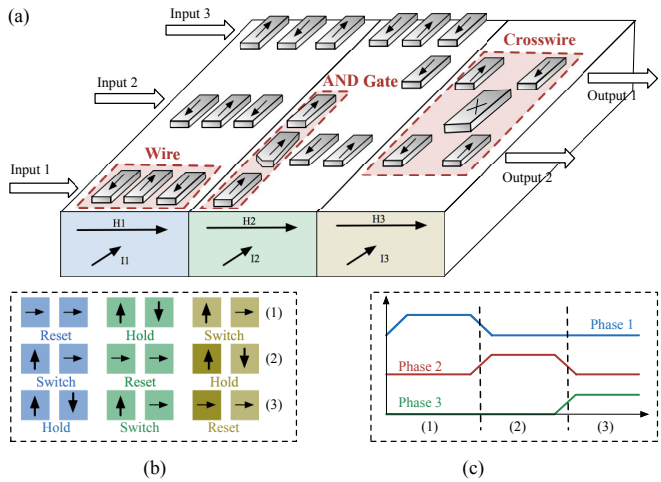

Figure 1: a) Example of NML circuit. Single domain nanomagnets are used to represent the logic values. Logic gates and signals propagation is accomplished through magnetodynamic interaction among neighbor magnets. An external magnetic field generated by a current flowing through a wire buried under the magnets plane is used as a clock signal to assure signals propagation. b) A multi-phase clock system is required to correctly propagate information in presence of noise. Circuits are divided in small areas called clock zones, made by a limited number of magnets. c) Three clock signals with a phase difference of 120° are applied to different clock zones.

noise which limits the number of magnets that can be cascaded [10], a multiphase clock system must be used. Circuits are divided in small areas, called clock zones, including a limited number of magnets (Figure 1.b). One of three clock signals is applied to every clock zone (Figure 1.b) [11]. At every time step, magnets of a clock zone that are switching (SWITCH), see on their left side other magnets in a stable (HOLD) state, that act as inputs. At the same time on their right magnets are in the RESET state and have no influence on signals propagation.

Due to the advancement of the experimental activity on NML technology, many works are available in literature that focus on circuits architectures. For example in [12] a PLA-like structure was proposed. In [13] a more complex structure based on the systolic array principle, an architecture made by array of identical processing elements, was analyzed. While these are good works the circuits studied are very simple. To effectively analyze NML technology we have focused our attention on

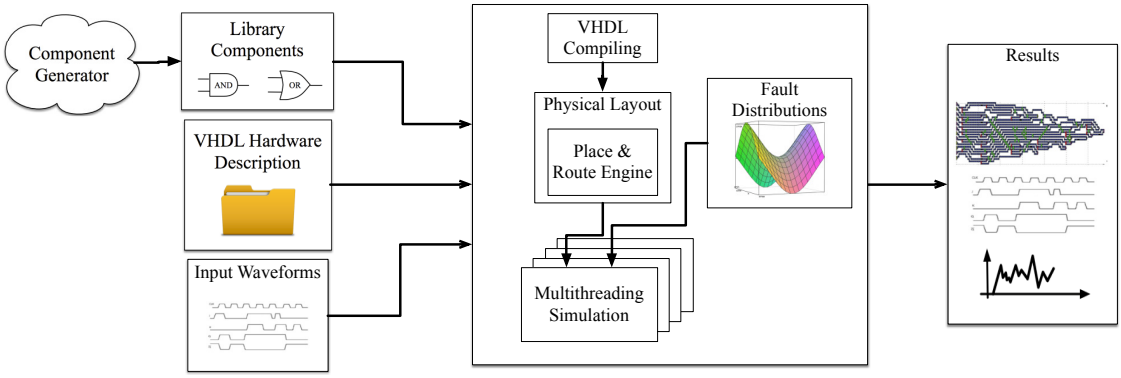

Figure 2: ToPoliNano flow chart. The tool accepts four types of input, a VHDL file describing the circuit, a NML circuit layout stored in the library or manually designed, inputs waveform files for the logical simulation and fault/distances distribution for the fault analysis. Depending on the options selected and the inputs provided ToPoliNano can generate a layout or simply visualize a previously created one. Then ToPoliNano can perform different analysis on the circuit layout, a logical simulation, a fault analysis and an area and power estimation.

more complex and realistic circuits, like microprocessors [14], decoders for telecommunication [15] and a systolic array for biosequences analysis [16]. Due to the complexity of such circuits, finite element micromagnetic simulators cannot be used to verify the circuit functionality. As a consequence we have always used behavioral models based on VHDL language [14] as a way to simulate complex circuits. Recently, we have started the development of a dedicated tool for design and analysis of NML circuits called ToPoliNano [17] [18]. This tools allows to design circuits following the top-down approach of CMOS technology. Circuits are described with VHDL language and the layout is automatically generated [19] and then simulated using a behavioral model [20] due to the complexity of circuits. This tool provides an improved methodology to design and test complex NML circuits, methodology that we present here in this work alongside with several improvements to the place&route algorithm and a new feature that allows to evaluate the impact of fault caused by process variations during the logic simulation.

II. ToPoliNano

ToPoliNano is a software tool created by the VLSI group of Politecnico di Torino developed to help researchers to study emerging technologies. The idea behind the development of this software arises from the fact that nowadays there are no tools able to perform automatic synthesis and layout generation of circuits based on emerging technologies, like NML. ToPoliNano is developed in C++ and it counts more than 100k lines of code. It can be run on different platforms such as Linux, Windows and Mac OS X. Our CAD has been built around the idea of giving to researchers a way to use the same top-down approach used for CMOS technology. ToPoliNano uses therefore as entry point a circuit described with VHDL language. The VHDL code can be loaded from an internal library or can be created and modified with the in-built editor. Starting from the VHDL code the tool is able to generate the circuit layout based on NML technology and to perform on it a logical simulation. ToPoliNano can also provide information and statistics on the occupied area and the power consumption.

As shown in Figure 2, the internal structure of the software can be divided into two sub-parts: the first one is related to the description of the circuit and depend on the Component Generator (CG). The second part is represented by the Place&Route algorithm and the simulator itself which are able to generate the circuit layout, to simulate and to analyze circuits. The main task of the CG is to create the basic blocks (such as: and, or, inverters) that will be used by the Place&Route (P&R) engine to draw the final circuit, starting from a VHDL files. Those components are mapped according to specific rules tailored for NML technology. In this way a Component Library (CL) is created. Inside the component library users can find the fundamental logic gates correspondent to the structural VHDL description and more complex circuits, manually created or previously generated by ToPoliNano.

ToPoliNano flow begin with the analysis of a VHDL file. The code is then parsed in order to identify and extract the basic components that will compose the final circuit and the interconnections among them. At the end of this phase, therefore, the software provides a data structure, implemented through a graph, where each node represents an elementary block and each edge is an interconnection.

Once this first data structure is generated, it will be optimized by the Place&Route engine according to different algorithms that will be described in Section III. The aim of this first optimization process is to reduce the number of crosswires, components that have a big impact on the circuit area. The optimized graph will be further processed by ToPoliNano to give to the user a graphical representation of the final circuit. This new data structure of the circuit is then used as entry point for the third stage of the flow: the logic simulation process. Users can customize different simulation parameters, like the simulation step time, if the fault analysis must be enabled and in this case the number of iterations of the fault analysis simulation. The fault analysis is a newly introduced feature of ToPoliNano that allows to estimate errors in the circuit logic behavior caused by magnet displacements

due to process variations. The fault analysis is briefly described in Section V.

In addition to this ToPoliNano provides also an area and power analysis option. Different circuit parameters can be evaluated, like the total power consumption, the power consumption due to magnets switching, the clock losses, the total circuit area and the percentage of effectively occupied area, the total number of magnets used to generate the final layout and the number of faulty magnets.

III. HDL Graph Optimization: Cross-Wire Minimization Techniques

The main entry point in ToPoliNano is a circuit described using VHDL language. As mentioned in Section II, after the VHDL file parsing, the HDL graph is generated. The HDL graph maps exactly the VHDL description and each node of the graph represents a logic gate. The HDL graph is then elaborated by the Place&Route engine to optimize it according to the technological constraints given by the NML technology. A detailed representation of the Place&Route working flow is depicted in Figure 3.

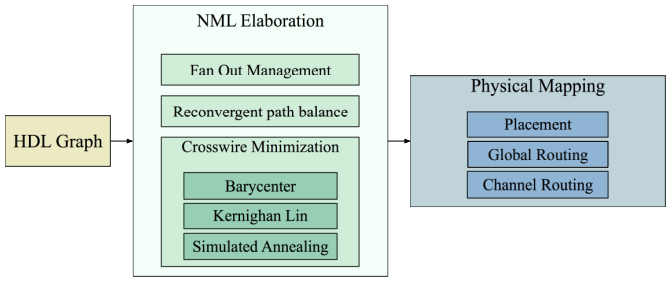

Figure 3: Physical Design flow. The entry point is the HDL graph generated starting from the VHDL input file. The graph is elaborated and optimized throughout subsequent steps (NML Elaboration) and then the circuit layout is created and further optimized (Physical Mapping).

The Place&Route engine can be divided in two ideal parts, the NML graph elaboration and the physical mapping (see Section IV for further details on the physical mapping part). In the NML graph elaboration part a first optimization is required to adapt the initial graph to the requirements of NML and QCA technology more in general. Referring to Figure 3, the Fan-Out Management [19] algorithm generally reduces the number of output edges from one node, in order to respect the fan-out limitation of NML technology. Generally in NML technology no more than 2 or 3 edges are allowed as outputs for every node, due to the limited number of magnets that can be cascaded inside a clock zone [10]. Then, the "Reconvergent Path Balance" [19] algorithm is applied in order to balance the structure of the graph and equalize the length of input magnetic wires of each magnetic gate. This step is required to correctly synchronize magnetic signals and avoid errors in the signals propagation.

Since NML is a planar technology, whenever there is a crossing between two interconnection wires a magnetic crosswire block must be used. A crosswire is a block available in this technology [7] that allows to cross two magnetic wires on the same plane without interferences. The necessity of using crosswires has a noticeable impact on circuit, greatly increasing the global area [15]. As a consequence a reduction of the total number of crossing is mandatory. The most efficient method to reduce the number of crosswire is rearranging nodes inside the graph. We inspired to three main algorithms for crosswire minimization as a starting point, the Barycenter, the Kernighan Lin and the Simulated Annealing. Then we redefined them and found a new implementation adapted and optimized to NML technological constraints. The detailed description of the three new algorithms and their comparison can be found in [19]. The *Modified-Kernighan-Lin* and *Modified-Simulated-Annealing* algorithm generally provide slightly better performance, but their execution time is much higher. As a consequence, for the technology we are aiming to in this paper, we focused our recent efforts on improving the *Modified-Barycenter* algorithm, which is a far simpler and faster algorithm.

The *Modified-Barycenter* algorithm reduces the number of crosswires swapping the position of each node inside the graph, trying to place each node directly above the node connected to it. As a novelty we adopted two techniques to optimize the barycenter: first the Breadth-First Search algorithm, a classic technique used in graphs analysis, is applied to reorder the graph and then the new version of the algorithm is implemented as *Modified-Barycenter*.

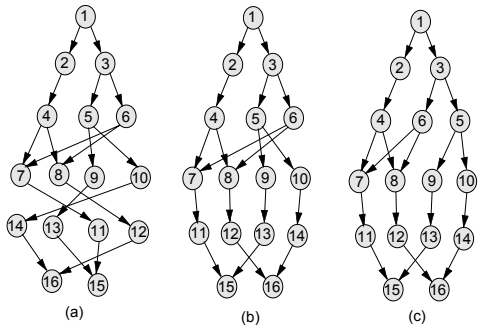

Figure 4: a) Initial graph before wire cross minimization techniques are implemented. b) Graph after Breadth-First Search sorting is executed. c) Graph after the optimized barycenter method is applied.

Figure 4 shows an example of graph during the various optimization steps. The graph of Figure 4.a is obtained from the HDL graph after the application of a new *Modified-Fan-Out Management* algorithm and of the "Modified-Reconvergent Path Balance" algorithms are applied. Nodes are then reordered thanks a Breadth-First Search algorithm (Figure 4.b). It is worth noticing that a first cross-wires reduction is already obtained in this phase. Then an optimized version of the Barycenter that we have developed is finally applied to the graph (Figure 4.c). The difference between the original Barycenter (already a modified version with respect to the one known in literature) and the improved one used here resides in the entry point. In the first version the algorithm is applied line by line from the input nodes of the first layer to the outputs nodes of the last layer. The optimized method starts instead from the layer where there is the largest number of cross-wires.

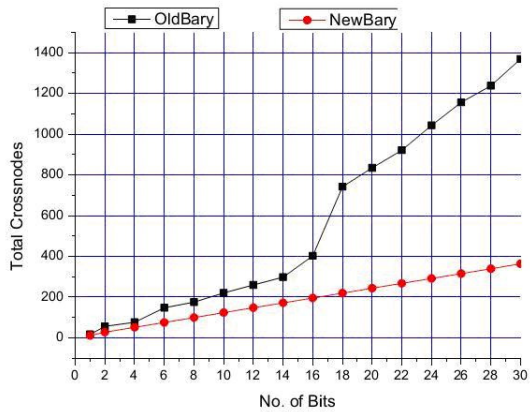

Figure 5: Comparison in terms of number of crosswires between the old barycenter technique and the improved one combined with the Breadth-First Search algorithm. The test circuit is a ripple carry adder with a variable number of bits.

In Figure 5, a comparison in terms of number of crossings is shown between the original Barycenter algorithm (the one we proposed in [18] [**?**], [17] to be clear) and the improved one presented here and combined with the Breadth-First Search algorithm. The test circuit is an N-bit ripple carry adder, where the number of bits varies from 1 to 30 bits. Thanks to this combined approach we obtain an improvement of more than 300%, which is a remarkable result.

IV. Physical Mapping

The second main part of the Place&Route engine is the physical mapping (Figure 3), where the graph is converted into the final representation of the magnetic circuit. During this phase, three operations are executed sequentially: Placement, Global Routing and Channel Routing.

Placement. In this phase each node of the graph (Figure 6.a) is translated into the corresponding NML logic gate (Figure 6.b). No optimizations are performed: blocks are placed following the node positions given by the New-Barycenter method. The logic gate library includes magnetic wires, inverters, logic and and logic or [9], crosswires [7] and couplers. Couplers are fan-out blocks which simply split a wire in two. After the blocks are created for all the nodes in the graph, they are placed next to each other respecting the minimum distance achievable at the technological level. Blocks are placed in alternate columns. Every column represents a clock zone and each of them contains a number of blocks equal to the number of nodes of the corresponding layer. The void columns are used for interconnection wires and are filled in the channel routing phase. When this operation is completed, blocks belonging to the same column are shifted to the center in order to start the wires routing and the area optimization.

Global Routing. In this phase blocks are moved into their final position with the aim of maximize the circuit compaction, reducing the length of interconnection wire and the total area. In the first version of the algorithm [19], the relative position of blocks inside a clock zone was fixed and equal to 1 magnets. Blocks were shifted as a whole inside a clock zone, trying

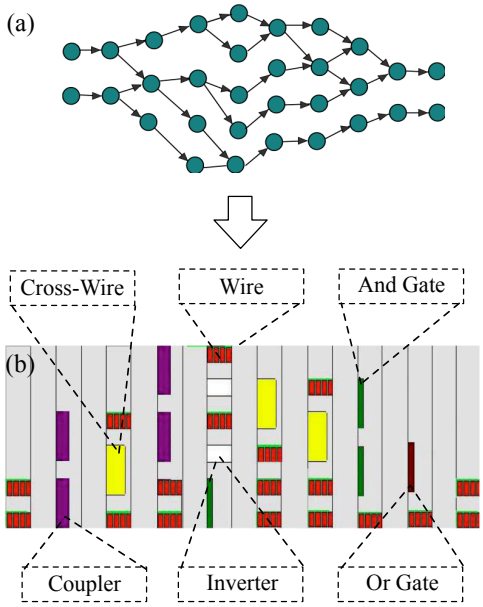

Figure 6: (a) Optimized graph. (b) Each node of the graph is mapped to a NML logic gate. The gate set available is made by wires, inverters, couplers, and, or, crosswires.

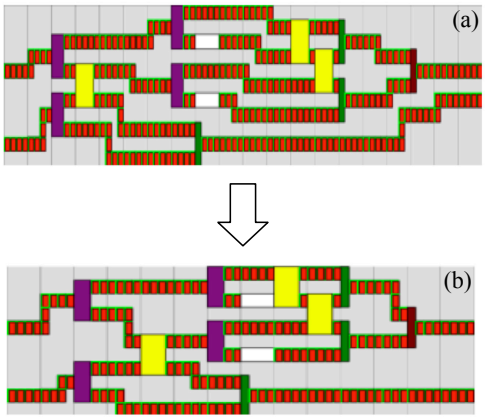

Figure 7: Example of half adder obtained after the physical mapping. (a) Half adder obtained with the original placement algorithm. (b) Half adder obtained with the improved placement algorithm. The area is greatly reduced.

to globally minimize the length of interconnection wires. We present here as an absolute novelty a magnet-placement algorithm notably improved with respect to the previous version discussed in [18]. In this new placement algorithm every block inside a clock zone can be moved independently. Thanks to this solution, it is now possible to reach a better alignment between output pins of a block and the input pins of the blocks in the next clock zone. This algorithm is applied to the whole circuit except to those zones where the density of cross wires is very high. In those areas no optimizations are performed and they are left untouched. The reason behind this choice resides on the fact that areas with an high density of crosswires are

already very compact, even without optimization. We proved that applying this modified version of the algorithm also to those clock zones reduces the circuit compactness and is not convenient.

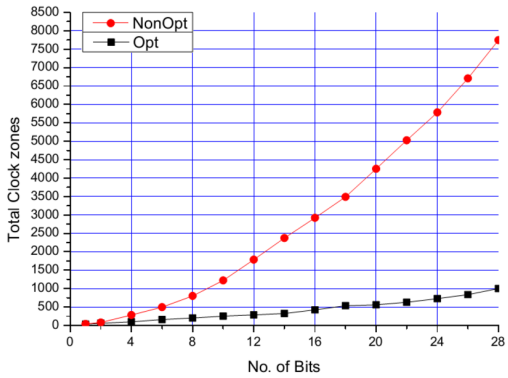

Figure 8: Comparison between the old placement and the new placement algorithm. The test circuit is a ripple carry adder with a variable number of bits. The improvement is around 800%.

Figure 7 shows the comparison between an half adder obtained with the original placement algorithm (Figure 7.a) and an half adder obtained with the improved algorithm (Figure 7.b). The area is greatly reduced with the new algorithm. The improved algorithm makes interconnection wires more straight optimizing therefore the interconnections area. Figure 8 shows a comparison in terms of total number of clock zones between the two algorithms. The test circuit is a ripple carry adder with a variable number of bits. The improvement depends on the number of bits, but is around 800% in case of the 28 bit adder, which is an astonishing result.

Channel Routing. After the placement of blocks obtained in the previous step, the routing phase takes place. Magnetic blocks included in neighbor rows are connected, starting from the first row. This phase does not provide any optimization, wires are simply placed evaluating the distance between input and output pins on neighbor rows. In this phase no further optimization is adopted, as interconnection wires already have the minimum possible length thanks to the results obtained in the global routing phase. The example of layout shown in Figure 9 (a) depicts an 8 bits 1-to-4 demultiplexer generated ToPoliNano, starting from a proper VHDL description. The resulting layout is optimized by the new set of algorithms included in ToPoliNano in order to obtain the smallest possible area.

V. Simulation and Testing

Once the physical mapping stage generated the final circuit, the layout is presented to the user in graphical form in the main ToPoliNano window. The circuit can be observed at any zoom level, starting to the whole circuit to the detail of a single magnet. This represents the first form of test available in ToPoliNano, a visual inspection that allows the user to easily understand if something went wrong during the whole process. As a second and more accurate test, it is possible to simulate the circuit using a behavioral simulation. However, the simulation engine requires a new data structure.

Starting from this planar and extremely regular representation a new data structure is loaded into the main memory. This new matrix-like representation features a much finer granularity respect to the previous graph-based structure. Now, indeed, each node of the matrix represents a single magnet. The simulation is based on a switch-level algorithm, where every magnet is considered a tri-state device. For every clock zone the new state of a magnet is evaluated considering the magnetization value of its neighbors. The simulation principle is shown in Figure 10. The simulation matrix is visited column by column starting from the beginning to the end of the clock zone evaluating the state of every magnets. While a physical-based simulation is not possible with so complex circuits, we have based the simulation algorithm on the results of low level micromagnetic simulations, thus providing the most accurate result within the limits of a logical simulation. More details on the simulation algorithm are not reported here, but they can be found in [20].

The user can choose to start two different kind of simulations. The fist one, the ideal one, has the only purpose to verify the correctness of the logical behavior of the generated circuit. The second one instead, is a newly introduced type of simulation which has the goal of running a MonteCarlo-like simulation that can take into accounts defect derived from the process variations. To achieve this, a new simulation algorithm has been implement. This newly introduced algorithm visits the simulation matrix in the same way as the ideal one does, but it uses a more refined method to calculate the magnetization of devices. Magnetic interaction between two magnets strongly depends on their distance [21]. As a consequence, this variable should be considered during the simulation phase. We have therefore introduced in the simulation algorithm an option to evaluate the effect of a variation of distances among magnets. Basing our work on the low level simulations of [21], we have defined a probability that, if distances among magnets is bigger than a certain value, the magnets state is not evaluated correctly. While this algorithm is not as accurate like a low level simulation can be, it has the advantage that it allows to estimate faults due to process variations on very big circuits. Moreover different distributions of distances can be selected testing their effect on the circuit behavior.

VI. Conclusions

We have proposed a design and test methodology for NML circuits entirely based on a top-down approach as happens in CMOS. It is based on ToPoliNano, the tool that we have specifically developed for emerging technologies analysis. ToPoliNano allows to describe circuits using VHDL language as in CMOS and it automatically creates the low level layout. Circuits can then be analyzed through logical simulations, a fault analysis and parameters like area and power consumption can be also evaluated. We have also presented here our recent improvements to the place&route algorithms that greatly reduce circuits area.

As a future work we are studying a floorplan generation engine to effectively tackle complex circuits. We are also working to extend place&route algorithms to sequential circuits, which

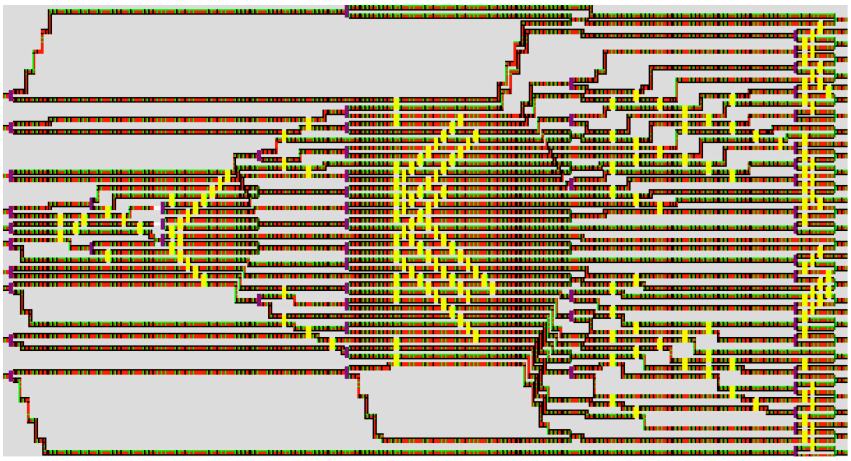

Figure 9: An 8 bits 1-to-4 Demultiplexer layout generated by ToPolinano.

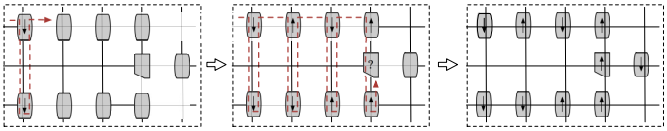

Figure 10: Simulation algorithm working principle. The simulation matrix is visited column by column from the beginning to end of the clock zone, evaluating the state of each magnet.

are particularly difficult to handle with intrinsically pipelined technology, like NML. At the same time we are polishing the code so that we can finally release the tool to the community, to extend the benefits of ToPoliNano to anyone working in this field.

References

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

[2] U. Lu and C. Lent, "Theoretical Study of Molecular Quantum-Dot Cellular Automata," *Journal of Computational Electronics - Springer*, vol. 4, pp. 115–118, 2005.

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

[4] W. Porod, "Magnetic Logic Devices Based on Field-Coupled Nanomagnets," *Nano & Giga*, 2007.

[5] A. Pulimeno, M. Graziano, A. Saginario, V. Cauda, D. Demarchi, and G. Piccinini, "Bis-ferrocene molecular QCA wire: ab-initio simulations of fabrication driven fault tolerance," *IEEE Transaction on Nanotechnology*, vol. 12, no. 4, pp. 498–507, May 2013.

[6] N. Rizos, M. Omar, P. Lugli, G. Csaba, M. Becherer, and D. Schmitt-Landsiedel, "Clocking Schemes for Field Coupled Devices from Magnetic Multilayers," in *International Workshop on Computational Electronics*. Beijin, China: IEEE, 2009, pp. 1–4.

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

[8] D. Karunaratne and S. Bhanja, "Study of single layer and multilayer nano-magnetic logic architectures ," *Journal Of Applied Physics*, no. 111, 2012.

[9] M. Niemier, E. Varga, G. Bernstein, W. Porod, M. Alam, A. Dingler, A. Orlov, and X. Hu, "Shape Engineering for Controlled Switching With Nanomagnet Logic," *IEEE Transactions on Nanotechnology*, vol. 11, no. 2, pp. 220–230, Mar. 2012.

[10] 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.

[11] 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.

[12] M. Crocker, M. Niemier, and X. Hu, "A Reconfigurable PLA Architecture for Nanomagnet Logic," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 8, no. 1, Feb. 2012.

[13] X. Ju, M. Niemier, M. Becherer, M. P. W. Porod, P. Lugli, and G. Csaba, "Systolic Pattern Matching Hardware With Out-of-Plane Nanomagnet Logic Devices," *IEEE T. on Nanotechnology*, vol. 12, no. 3, May 2013.

[14] M. Vacca, M. Graziano, and M. Zamboni, "Nanomagnetic Logic Microprocessor: Hierarchical Power Model," *IEEE Transactions on VLSI Systems*, vol. 21, no. 8, pp. 1410–1420, Aug. 2012.

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

[16] J. Wang, M. Vacca, M. Graziano, and M. Zamboni, "Biosequences analysis on NanoMagnet Logic," *International Conference on IC Design and Technology*, pp. 131–134, May 2013.

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

[18] S. Frache, D. Chiabrando, M. Graziano, F. Riente, G. Turvani, and M. Zamboni, "ToPoliNano: Nanoarchitectures Design Made Real," *IEEE International Symposium on Nanoscale Architectures (NANOARCH)*, pp. 160–167, 2012.

[19] M. Vacca, S. Frache, M. Graziano, L. Crescenzo, F. Cairo, and M. Zamboni, "Automatic Place&Route of Nano-magnetic Logic circuits," Jul. 2013, pp. 58–63.

[20] M. Vacca, S. Frache, M. Graziano, and M. Zamboni, "ToPoliNano: A synthesis and simulation tool for NML circuits ," *IEEE International Conference on Nanotechnology*, pp. 1–6, Aug. 2012.

[21] 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.