Ropper: A Placement and Routing Framework for Field-Coupled Nanotechnologies

Ruan Evangelista Formigoni*
ruan.formigoni@ufv.br
Universidade Federal de Viçosa
Florestal, Brazil

Ricardo S. Ferreira†
ricardo@ufv.br
Universidade Federal de Viçosa
Viçosa, Brazil

José Augusto M. Nacif‡
jnacif@ufv.br
Universidade Federal de Viçosa
Florestal, Brazil

ABSTRACT

Field-Coupled Nanocomputing technologies are the subject of extensive research to overcome current CMOS limitations. These technologies include nanomagnetic and quantum structures, each with its design and synchronization challenges. In this scenario clocking schemes are used to ensure circuit synchronization and avoid signal disruptions at the cost of some area overhead. Unfortunately, a nanocomputing technology is limited to a small subset of clocking schemes due to its number of clocking phases and signal propagation system, thus, leading to complex design challenges when tackling the placement and routing problem resulting in technology dependant solutions. Our work consists on presenting a novel framework developed by our team that solves these design challenges when using distinct schemes, therefore, avoiding the need to design pre-defined routing algorithms for each one. The framework offers a technology independent solution and provides interfaces for the implementation of efficient and scalable placement strategies, moreover, it has full integration with reference state-of-the-art optimization and synthesis tools.

CCS CONCEPTS

• **Hardware → Physical design (EDA)**.

KEYWORDS

placement, routing, clocking, scheme, framework, synthesis, generic

1 INTRODUCTION

The complementary Metal-oxide-semiconductor (CMOS) is the current technology used by the industry to mass produce integrated circuits. The scaling theory of CMOS transistors [12] has been of critical importance to fabricate chips with an increasingly higher number of these switching devices. Unfortunately, the miniaturization of this fundamental component cannot continue indefinitely as atomic limitations [9, 16] bound it. A novel computing paradigm known as Field-Coupled Nanocomputing (FCN) [1] has been extensively studied to overcome this ever approaching barrier. Many technologies have been proposed for this novel method of performing computation [1]. Two important examples of these technologies are the Quantum-Dot Cellular Automata (QCA) [22, 23] and the Nanomagnetic Logic (NML) [11].

FCN technologies show great promise as they offer low power consumption, no static power dissipation and high operation frequencies [20]. They process information by cascading signals through its building blocks, those which are arranged as arrays to build wires and logic elements [7, 18]. When cascading information through these technologies' building blocks, there is a need to synchronize the signal propagation using a clocking system. It is common to organize clocking systems in clocking schemes to solve synchronization issues, thus orchestrating the signal propagation in the circuit [8]. Also, clocking schemes prevent signal disruptions due to technology-related limitations. These advantages come at the cost of some area overhead. Each FCN technology has a distinct way of handling signal propagation. Moreover, the number of clocking phases may vary from one another, resulting in different clocking scheme designs, and thus making the design process of FCN circuits a challenging technology-bounded problem.

The main challenges in FCN P&R include the variety in clocking scheme designs and the need to develop unique algorithms to each one independently. Moreover, each solution needs to implement its own set of procedures for area minimization. Well-known synthesis and optimization tools are of crucial importance to aid the process of working with the circuit graph and making use of sophisticated partition techniques.

In this paper, we present Ropper, a novel open-source electronic design automation (EDA) framework for FCN technologies. Our tool handles the problem of using distinct clocking scheme designs by abstracting technology-bounded characteristics. It also introduces well-defined interfaces for generic placement and routing (P&R) strategies, allowing for trivial implementations of state-of-the-art algorithms. Moreover, Ropper is fully integrated with well-known synthesis and optimization tools [6, 28]. Reference state-of-the-art P&R frameworks support either QCA or NML technologies. Ropper provides support to clocking schemes currently available in the literature. As case studies, we demonstrate the use of two clocking schemes in Ropper. Moreover, Ropper is the only FCN P&R framework that integrates to reference synthesis and optimization tools and offers an open-source implementation[1].

The rest of the paper is divided as follows. In Section 2 we review the main concepts of the QCA and NML technologies as well as two reference clocking schemes. Section 3 presents related work. Section 4 discusses an in-depth overview of the structure of our framework, our main design choices, integration with external libraries, and extensibility features. Finally, in Section 6, we discuss our next goals and planned functionalities.

[1] Available at: https://gitlab.com/formigoni/ropper

2 A REVIEW OF FIELD-COUPLED TECHNOLOGIES

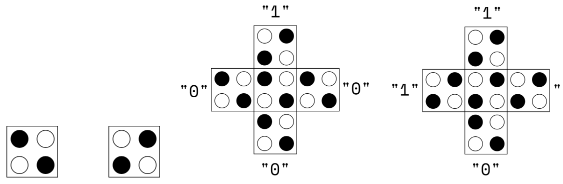
In this Section, we present key concepts about QCA and NML nanotechnologies. We also present clock scheme strategies for nanotechnologies.

2.1 The Quantum-dot Cellular Automata Technology

The QCA technology represents binary information utilizing a set of regular quadrilateral cells. Each of these cells envelops four quantum dots, and two mobile electrons, which can occupy a distinct quantum dot. These quadrilateral structures represent the binary value 0 when assuming an inverted diagonal configuration as shown in Figure 1a, the binary value 1 is represented by a regular diagonal configuration as shown in Figure 1b.

The binary information is propagated when a cell interacts with each other to change the disposition of its' neighbors free electrons. The QCA technology utilizes a four-phase clock cycle, i.e. **switch**, **hold**, **release** and **relax** [2, 17, 21]. By using this clocking system, the signal propagates from cell to cell allowing for circuit synchronization and scalable designs.

The fundamental logic element of the QCA technology is the majority gate. This element is composed of five QCA cells of which three serve as inputs. The middle cell replicates the signal that most appear in the inputs and transfers the resulting signal to a fifth cell which serves as an output. This behavior is also called the majority vote since the gate replicates the signal that appears most. Two majority gate logic examples are shown in Figures 1c and 1b.



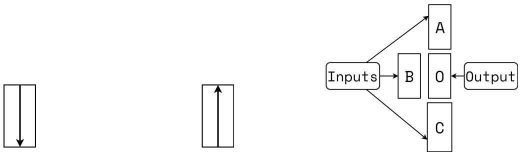
(a) (b) (c) (d)

Figure 1: (a) QCA Binary Logic 0. (b) QCA Binary Logic 1. (c) Majority vote yields the binary result of 0. (d) Majority vote yields the result of 1.

2.2 The Nanomagnetic Logic Technology

The NML technology utilizes nanomagnets that alternate its magnetizations between the upwards and downwards directions, e.g., the binary logic 0 shown in Figure 2a and the binary logic 1 as shown in Figure 2b. The NML technology, also known as the magnetic QCA, also performs majority logic using the majority gate structure from Figure 2c. The magnet polarization and NML signal propagation changes by applying an external magnetic field [3, 15] or exploring the spin Hall effect [4] to avoid the power cost associated with generating the magnetic field [25].

Unfortunately, due to thermal noise, the signal propagation is bound to a maximum of five consecutive magnets [10]; after this limit, the resulting signal can have an incorrect value. The signal disruption caused by this limitation is avoided by the usage of a clocking scheme and a 3-phase clocking system [31], as opposed to the 4-phase clocking system in the QCA technology. These phases are known as **reset**, **hold** and **switch**.

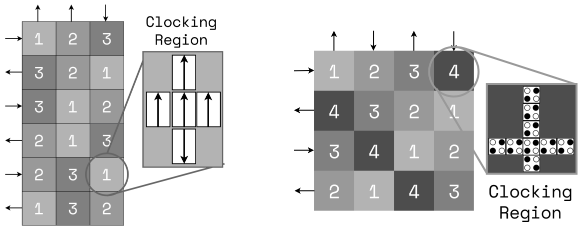


(a) (b) (c)

Figure 2: (a) Downwards magnet polarization representing the binary logic 0. (b) Upwards magnet polarization representing the binary logic 1. (c) A majority logic gate which utilizes A, B, and C as inputs.

2.3 Clocking Schemes

Clocking schemes are patterns which define the signal flow direction and delay. Routing circuits through them mean that we can amortize issues such as signal disruption and synchronization at the cost of some area overhead.



(a) (b)

Figure 3: (a) The basic building block of the Bidirectional Alternating Nanomagnetic clocking scheme. (b) The basic building block of the Universal, Scalable and Efficient clocking scheme.

Figure 3 shows both the clocking schemes that we use in our work, the Bidirectional Alternating Nanomagnetic Clocking Scheme (BANCS) [14] (figure 3a) and the Universal, Scalable and Efficient Clocking Scheme (USE) [8] (figure 3b). In these clocking schemes, each number denotes a region. A region of a clocking scheme is a regular quadrilateral which, has a technology bound size and can hold a certain amount of logical elements. A clocking scheme with regions of dimensions 3x3 can hold up to nine magnets in the BANCS clocking scheme, as shown by the highlight in Figure 3a. Furthermore, a clocking scheme with regions of dimensions 5x5 can hold up to twenty-five quantum-dots as shown by the highlight in Figure 3b.

3 RELATED WORK

P&R for FCN circuits is a complex problem, especially when using clocking schemes. In this Section, we discuss the reference frameworks and their approach to offer feasible solutions. For the NML

Table 1: Technology and Clocking Scheme Support

P&R	Clocking Scheme Support	Open-Source	Extendable
Ropper	Unrestricted	Yes	Yes
ToPoliNano	Not Available	No	Not Available
Fiction	Unrestricted	Yes	Yes

technology the *ToPoliNano* [30] tool provides design and simulation modules. It uses a logic synthesizer to perform the translation of VHDL files into a specific logic gates set, a parser to represent the VHDL as a graph stored in memory, a P&R module to structure the circuits' layout and a simulation module to perform power, data and area analysis. Unfortunately, the tool is not open-source, moreover, it does not offer support for the usage of a clocking scheme and its designed specifically for the NML technology.

Previous work on QCA P&R uses graph partitioning strategies [24, 26, 27], thus, achieving heuristically acceptable solutions. Unfortunately, most of these solutions do not offer open-source implementations. Moreover, the implementations are technology-bounded and difficult to adapt to other FCN paradigms.

The *Fiction* framework [33, 34] is the current open-source state-of-the-art reference for dealing with the FCN P&R design issues in QCA. The authors define a set of three design constraints to be satisfied to realize a successful transposition of the circuit's graph onto a clocking scheme pattern. The first constraint determines that only a single logic gate must be assigned to a clocking region. These regions usually have small dimensions (generally 5x5 for USE and 3x3 for BANCS), which are very close to the dimensions of the majority gate itself. The second constraint dictates that wires must connect logic gates with logical connections or be in adjacent clocking zones. The last constraint refers to the clocking scheme structure, where an identifier is assigned for each clocking phase to control the sequential flow of the signals through the pattern. For the generation of valid scalable P&R solutions, the authors work with solving the orthogonal graph drawing (OGD) [5, 13] problem which has many similarities with the P&R problem. Unfortunately, the OGD algorithm limits *Fiction* flexibility. As a consequence, it is not possible to adjust the number of wires and/or gates allowed for each cell, increasing the area overhead.

Ropper solves the issue of working with distinct clocking scheme designs through a mixed strategy of the breadth-first search and dynamic programming algorithms. Since our framework is not bounded by a specific algorithm, the wire limit in regions with and without assigned logic gates can be changed trivially. Thus, leading to greater area compaction in clocking schemes with moderate sized regions, Figure 4 illustrates this by showing a wire and a majority gate in the same clocking region. Moreover, due to our generic approach, it is possible to deal with tree phase clocking schemes such as BANCS, allowing for the P&R of NML circuits as well.

Table 1 summarizes current technologies and clocking schemes support for the tools presented in this section. The metric used to determine a framework support of a technology is its hability to handle its supported clocking schemes.

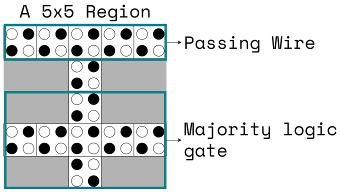


Figure 4: A wire and a majority gate within the same region.

As a summary of this section, Ropper provides dynamically adjustable wire limits for regions with and without an assigned logic gates leading to greater area compaction. Our framework also allows for the usage of various clocking scheme designs due to its generic approach for solving the routing problem, in addition, supporting the P&R process for circuits of the NML technology.

4 ROPPER FRAMEWORK STRUCTURE

In this Section, we present the main characteristics of the Ropper framework.

4.1 Features

Ropper is independent of the clocking scheme, the P&R strategy, and the FCN technology. This level of flexibility is due to configurable technology related parameters. These parameters allow, for example, the customization of the maximum number of wires routed through a region with and without logic gates. Our framework solves clocking schemes issues by using a combination of a Breadth-First Search and a Dynamic Programming strategy. Ropper provides a well-defined interface for creating user-defined strategies to solve the P&R design problem.

Our previous research inspired Ropper [19, 29], where we presented a novel placement and routing algorithm for QCA and also defined a divide-and-conquer methodology to explore all of the solution space of a circuit graph representation. By refining our methods, we have generalized our approach to encompass not only QCA but also other FCN technologies such as NML. Our framework integrates the cirkit [28] tool and its submodules: lorina [28] and mockturtle [28]. The first is a logic network parsing library and the latter, a library for logic optimization and synthesis. Also, our framework has access to cirkit's integration with Berkeley's ABC tool [6], providing its state-of-the-art algorithms. The processes of placement and routing are co-related, raising the challenge of separating the algorithms so that they do not need to interact with each other, thus, improving modularity and extendability.

Our approach decouples the steps above and defines separate interfaces for both of them. In its current version, our framework provides routing through dynamic programming and placement through the implementation of a user-defined strategy. We demonstrate in Section 5 the experimental results of the framework when using the Simulated Annealing algorithm [32] for area minimization, which takes advantage of the multiple inheritance technique available in C++. This technique allows for implementing a concrete class with access to the algorithm's interface and the strategy's dataset. Therefore, using current placement information to perturb and calculate the cost of the solutions during the whole process.

4.2 Routing Graphs Through Divergent Clocking Schemes Patterns

In this Section, we describe how our framework performs the routing of MIGs through distinct clocking schemes. Figure 5 shows a diagram representation of the framework (some structures have been omitted for conciseness).

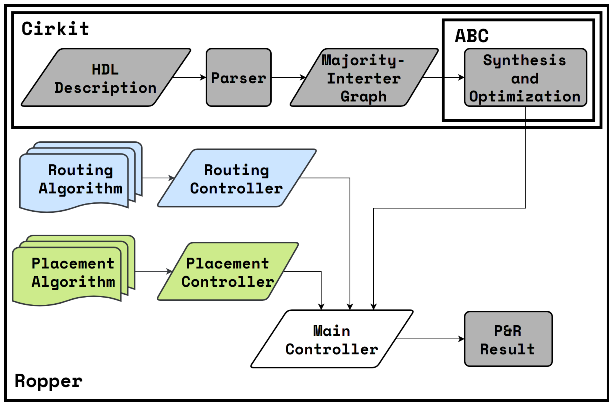


Figure 5: Ropper and external tools modules.

The main controller is used to control both the routing and placement processes. Through this main controller, it is possible to configure the maximum number of wires and/or logic gates that can occupy each cell, i.e. configure the routing controller (marked in blue). The standard routing algorithm used in our framework is a combination of a breadth-first search and dynamic programming (DP) techniques. Our implementation follows generic programming methodologies, therefore, allowing for any algorithm to be used in its place as long as it implements the necessary operations required by the controller.

The breadth-first search sequentially follows the clocking phases. In the USE clocking scheme, this sequence ranges from one to four, i.e. 1,2,3,4,1,2,3,4 and so on. In the BANCS clocking scheme there are clocking phase repetitions through its pattern. Moreover, the used sequence ranges from one to three. In this scenario, changing from a region with a specific clocking phase identifier to another region with the same identifier does not implicate in a distance increase. This is due to the distance being calculated with respect to distinct sequential clocking phases and not the regions themselves. As demonstrated so far, the process of calculating the distance from a point to another consists in finding the minimum path between them; as a direct consequence of this statement, all the paths were already pre-calculated during the distance operations. The dynamic programming algorithm allows for this to happen for all the intermediate regions as well. The DP struture is created along with the breadth first search, caching all the intermediate paths posterior access with constant complexity.

4.3 Creating Controllers

From Figure 5 both the routing and the placement controllers can be configured to behave in different ways to allow more flexibility when tackling a specific problem. Focusing on the placement controller, it is possible to change its behavior by inheritance techniques to create more sophisticated controllers used by the main controller. The level of flexibility ranges from managing a simple brute-force approach to orchestrating a graph partitioning multithreaded approach. The main controller has access to the logic network, the field, and the placements' information. It greatly increases the granularity of the possible implementations at the cost of some added complexity for its final form.

4.4 The Placement Problem

To serve as a base for future work and as a simple example to show our frameworks' capabilities, we implemented a strategy that performs the P&R process sequentially with relation to the graphs' levels. A possible more sophisticated approach can be based on [26] and [24] which uses a global placement strategy and advanced graph partitioning techniques.

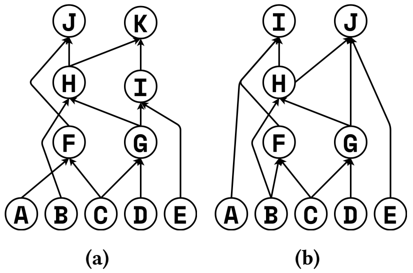
Algorithm 1: One of the P&R Strategies that Can be Used in Our Framework

```
Input  : $G$ - A Majority-Inverter Graph
Output : $S$ - The P&R Result
begin
    PlacementController ← $G$;
    Depth ← $MaximumLevelOf(G)$;
    while PlacementController → $Level()$ ≠ Depth do
        try:
            PlacementController → $Next()$;
            RoutingController → $Route()$;
        catch PlacementFailure
            PlacementController → $Blacklist()$;
        catch RoutingFailure
            PlacementController → $Blacklist()$;
            RoutingController → $Unroute()$;
        catch BaseLevelFailure
            return FALSE;
    end
    return TRUE;
end
```

The implemented strategy is shown by Algorithm 1. It performs sequential method calls for the placement and the routing controllers. Additionally, a blacklist method is used to set a cost that represents infinity to states that lead to unfeasible solutions. To reiterate, the entire placement process could be done before the routing, but for a more didactic and intuitive example, we chose not to do so. The process is successfully completed when the last level of the graph is placed and routed, or all the states from the first level are blacklisted, i.e. the process of P&R was unsuccessful.

5 PUTTING IT ALL TOGETHER

To demonstrate Ropper, we use the benchmark circuit C17 from iscas [35], which is represented by the graphs from Figure 6. Initially, we use the Lorina parsing library to read a Verilog file and translate its structure into the majority-inverter graph (MIG), which is represented by Figure 6a. The graph stored as a MIG is a direct representation of the and-inverter graph (AIG) format using fixed values in the majority logic gates to perform *or* and *and* operations, therefore, not taking advantage of the majority-logic design.



(a) (b)

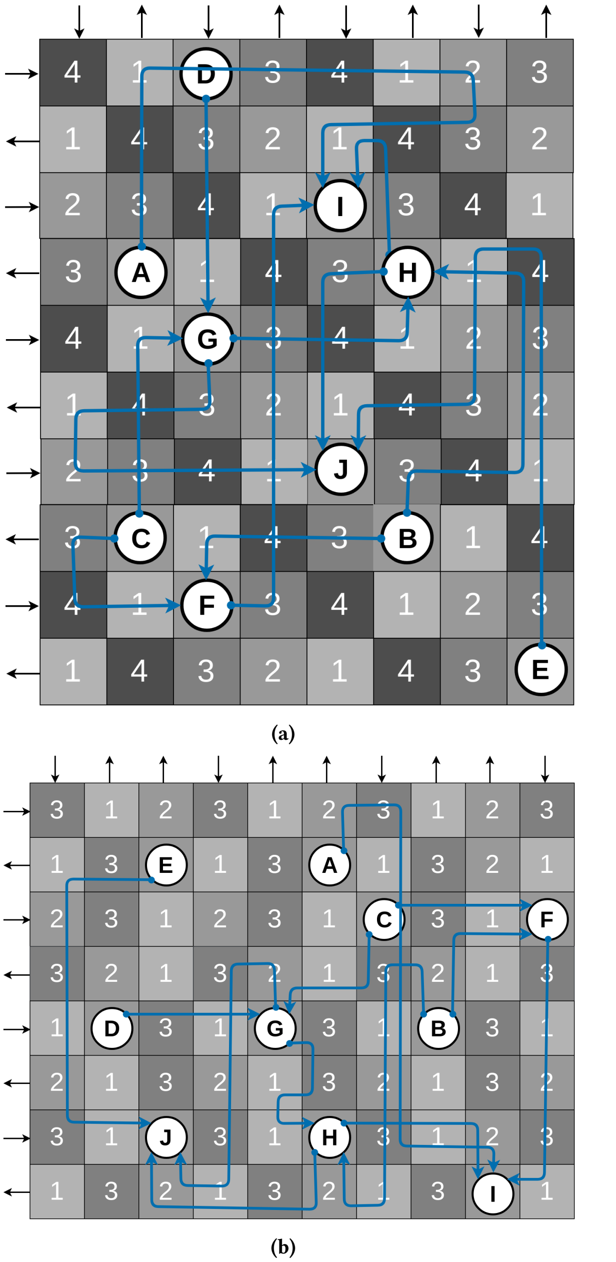
Figure 6: (a) Unoptimized majority-inverter graph. (b) Optimized majority-inverter graph.

To change this scenario we use the mockturtle library to resynthesize the MIG in a more efficient version shown in Figure 6b. We have reduced the number of logic gates by one at the cost of raising the number of edges from ten to eleven. Each level in the graph of this example has a chosen weight of four for the USE clocking scheme and a weight of three for the BANCS clocking scheme. This choice was based on the number of phases for each clocking scheme, with the goal of optimizing the process of finding available paths in each pattern. The distance between two nodes is measured in graph levels. This measurement is used to determine the distance, measured in regions, that a node should be apart from its fanin/fanout. This process can be summarized by Equation 1.

$$FD = LD * W \tag{1}$$

Where *FD* is the field distance, *LD* is the distance between the nodes' current levels (Level Distance) and W is the weight of the levels (Here we assume all the graph levels have the same weight). The P&R process in this paper is performed for both the USE and the BANCS schemes to demonstrate how to use different clocking scheme patterns in our framework.

The parameters used for testing consisted of allowing up to three wires the cross the same region and also allow one wire to cross a region with an assigned logic gate. The resulting area of the circuit is directly affected by technology dependent parameters. Possible future discoveries might lead to a relaxation of them. The resulting P&R is represented by Figure 7. Figure 7a represents the output for the USE clocking scheme and Figure 7b represents the output for the BANCS clocking scheme.



4	1	D	3	4	1	2	3
1	4	3	2	1	4	3	2
2	3	4	1	I	3	4	1
3	A	1	4	3	H	1	4
4	1	G	3	4	1	2	3
1	4	3	2	1	4	3	2
2	3	4	1	J	3	4	1
3	C	1	4	3	B	1	4
4	1	F	3	4	1	2	3
1	4	3	2	1	4	3	E

(a)

3	1	2	3	1	2	3	1	2	3
1	3	E	1	3	A	1	3	2	1
2	3	1	2	3	1	C	3	1	F
3	2	1	3	2	1	3	2	1	3
1	D	3	1	G	3	1	B	3	1
2	1	3	2	1	3	2	1	3	2
3	1	J	3	1	H	3	1	2	3
1	3	2	1	3	2	1	3	I	1

(b)

Figure 7: Final P&R for the circuit C17. (a) Placement and routing in the USE clocking scheme. (b) Placement and routing in the BANCS clocking scheme.

6 CONCLUSION AND FUTURE WORK

In our work we developed an extensible framework to solve the P&R problem when using distinct clocking schemes, allowing for customizable parameters and technology independent behavior. Moreover, we used two clocking scheme designs to verify our methodology and finally, presented the resulting layouts of the P&R processes.

Our framework is an ongoing effort to offer efficient and generic solutions for FCN design flows. As future work directions, we plan to develop more scalable P&R algorithms for FCN circuits..

REFERENCES

[1] Neal G. Anderson and Sanjukta Bhanja. 2014. *Field-Coupled Nanocomputing* (1 ed.). Springer-Verlag Berlin Heidelberg.

[2] Mehdi Askari and Maryam Taghizadeh. 2011. Logic Circuit Design in Nano-Scale using Quantum-Dot Cellular Automata. *European Journal of Scientific Research* 48 (01 2011), 516–526.

[3] Jayasimha Atulasimha and Supriyo Bandyopadhyay. 2010. Bennett clocking of nanomagnetic logic using multiferroic single-domain nanomagnets. *Applied Physics Letters* 97 (10 2010), 173105–173105. https://doi.org/10.1063/1.3506690

[4] Debanjan Bhowmik, Long You, and Sayeef Salahuddin. 2013. Spin Hall effect clocking of nanomagnetic logic without magnetic field. *Nature nanotechnology* 9 (11 2013). https://doi.org/10.1038/nnano.2013.241

[5] Therese C. Biedl. 1996. Improved Orthogonal Drawings of 3-graphs. In *Proceedings of the 8th Canadian Conference on Computational Geometry*. Carleton University Press, 295–299.

[6] Robert Brayton and Alan Mishchenko. 2010. ABC: An Academic Industrial-Strength Verification Tool. In *Computer Aided Verification*, Tayssir Touili, Byron Cook, and Paul Jackson (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 24–40. https://doi.org/10.1007/978-3-642-14295-6_5

[7] S. Breitkreutz, J. Kiermaier, I. Eichwald, X. Ju, G. Csaba, D. Schmitt-Landsiedel, and M. Becherer. 2012. Majority Gate for Nanomagnetic Logic With Perpendicular Magnetic Anisotropy. *IEEE Transactions on Magnetics* 48, 11 (Nov 2012), 4336–4339. https://doi.org/10.1109/TMAG.2012.2197184

[8] Caio Araujo T Campos, Abner L Marciano, Omar P Vilela Neto, and Frank Sill Torres. 2016. Use: a universal, scalable, and efficient clocking scheme for QCA. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35, 3 (2016), 513–517.

[9] T. Chen. 2006. Overcoming research challenges for CMOS scaling: industry directions. In *2006 8th International Conference on Solid-State and Integrated Circuit Technology Proceedings*. 4–7. https://doi.org/10.1109/ICSICT.2006.306040

[10] G. Csaba and W. Porod. 2010. Behavior of Nanomagnet Logic in the presence of thermal noise. In *2010 14th International Workshop on Computational Electronics*. 1–4. https://doi.org/10.1109/IWCE.2010.5677954

[11] GyÃűrgy Csaba, Wolfgang Porod, and ÃĄrpÃąd I. Csurgay. 2003. A computing architecture composed of Field-coupled single domain nanomagnets clocked by magnetic Field. In *International Journal of Circuit Theory and Applications*, Vol. 31. 67–82. https://doi.org/10.1002/cta.226

[12] R. H. Dennard, F. H. Gaensslen, L. Kuhn, and H. N. Yu. 1972. Design of micron MOS switching devices. In *1972 International Electron Devices Meeting*. 168–170. https://doi.org/10.1109/IEDM.1972.249198

[13] Markus Eiglsperger, Sandor P. Fekete, and Gunnar Klau. 1999. Orthogonal Graph Drawing. *Drawing Graphs: Methods and Models, 121-171 (2001)* 2025 (01 1999), 121–171. https://doi.org/10.1007/3-540-44969-8_6

[14] Ruan Evangelista Formigoni, Omar Paranaiba Vilela Neto, and José Augusto Nacif. 2018. BANCS: Bidirectional Alternating Nanomagnetic Clocking Scheme. In *2018 31st Symposium on Integrated Circuits and Systems Design (SBCCI)*. 1–6. https://doi.org/10.1109/SBCCI.2018.8533251

[15] M. Graziano, M. Vacca, A. Chiolerio, and M. Zamboni. 2011. An NCL-HDL Snake-Clock-Based Magnetic QCA Architecture. *IEEE Transactions on Nanotechnology* 10, 5 (Sep. 2011), 1141–1149. https://doi.org/10.1109/TNANO.2011.2118229

[16] Nor Zaidi Haron and Said Hamdioui. 2008. Why is CMOS scaling coming to an END?. In *3rd International Design and Test Workshop*. https://doi.org/10.1109/IDT.2008.4802475

[17] Kevin Hennessy and Craig S. Lent. 2001. Clocking of Molecular Quantum-Dot Cellular Automata. *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures* 19 (09 2001), 1752–1755. https://doi.org/10.1116/1.1394729

[18] A. Imre, G. Csaba, L. Ji, A. Orlov, G. H. Bernstein, and W. Porod. 2006. Majority Logic Gate for Magnetic Quantum-Dot Cellular Automata. *Science* 311, 5758 (2006), 205–208. https://doi.org/10.1126/science.1120506

[19] Geraldo Soares Fontes Junior. 2018. *Explorando o EspaÃğo de SoluÃğÃčo no Posicionamento e Roteamento de CÃľlulas QCA no Esquema de Clock USE*. Master's Thesis. Universidade Federal de ViÃğosa.

[20] Kyosun Kim, Kaijie Wu, and Ramesh Karri. 2006. Quantum-Dot Cellular Automata Design Guideline. In *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E89-A. 1607–1614. https://doi.org/10.1093/ietfec/e89-a.6.1607

[21] Kyosun Kim, Kaijie Wu, and Ramesh Karri. 2007. The Robust QCA Adder Designs Using Composable QCA Building Blocks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 26 (2007), 176–183.

[22] Craig S. Lent, Beth Isaksen, and Marya Lieberman. 2003. Molecular Quantum-Dot Cellular Automata. *Journal of the American Chemical Society* 125, 4 (2003), 1056–1063. https://doi.org/10.1021/ja026856g PMID: 12537505.

[23] Craig S. Lent and P. Douglas Tougaw. 1997. A Device Architecture for Computing with Quantum Dots. In *Proceedings of the IEEE*, Vol. 85. 541 – 557. https://doi.org/10.1109/5.573740

[24] Jean Nguyen, Ramprasad Ravichandran, Sung Lim, and Michael Thaddeus Niemier. 2019. Global Placement for Quantum-dot Cellular Automata Based Circuits. (04 2019).

[25] M. T. Niemier, X. S. Hu, M. Alam, G. Bernstein, W. Porod, M. Putney, and J. DeAngelis. 2007. Clocking structures and power analysis for nanomagnet-based logic devices. In *International Symposium on Low Power Electronics and Design (ISLPED), 2007 ACM/IEEE*. 26–31. https://doi.org/10.1145/1283780.1283787

[26] R. Ravichandran and M. Niemier and. 2005. Partitioning and placement for buildable QCA circuits. In *Proceedings of the ASP-DAC 2005. Asia and South Pacific Design Automation Conference, 2005.*, Vol. 1. 424–427 Vol. 1. https://doi.org/10.1109/ASPDAC.2005.1466200

[27] Ramprasad Ravichandran, Nihal Ladiwala, Jean Nguyen, Mike Niemier, and Sung Kyu Lim. 2004. Automatic Cell Placement for Quantum-dot Cellular Automata. In *Proceedings of the 14th ACM Great Lakes Symposium on VLSI (GLSVLSI '04)*. ACM, New York, NY, USA, 332–337. https://doi.org/10.1145/988952.989033

[28] Mathias Soeken, Heinz Riener, Winston Haaswijk, and Giovanni De Micheli. 2018. The EPFL logic synthesis libraries. arXiv:1805.05121.

[29] Alyson Trindade, Ricardo Ferreira, JosÃľ Augusto M. Nacif, Douglas Sales, and Omar P. Vilela Neto. 2016. Placement and Routing Algorithm for Quantum-dot Cellular Automata. In *2016 29th Symposium on Integrated Circuits and Systems Design (SBCCI)*. 1 – 6. https://doi.org/10.1109/SBCCI.2016.7724048

[30] Marco Vacca, Stefano Frache, Mariagrazia Graziano, Fabrizio Riente, Giovanna Turvani, Massimo Ruo Roch, and Maurizio Zamboni. 2014. ToPoliNano: Nano-Magnet Logic Circuits Design and Simulation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8280 LNCS (01 2014), 274–306. https://doi.org/10.1007/978-3-662-45908-9_12

[31] Marco Vacca, Mariagrazia Graziano, Alessandro Chiolerio, Andrea Lamberti, Marco Laurenti, Davide Balma, Emanuele Enrico, F Celegato, P Tiberto, Luca Boarino, and Maurizio Zamboni. 2014. *Electric Clock for NanoMagnet Logic Circuits*. Springer Berlin Heidelberg, 73–110. https://doi.org/10.1007/978-3-662-43722-3_5

[32] Peter J. M. van Laarhoven and Emile H. L. Aarts. 1987. *Simulated annealing*. Springer Netherlands, Dordrecht, 7–15. https://doi.org/10.1007/978-94-015-7744-1_2

[33] Marcel Walter, Robert Wille, Frank Sill Torres, Daniel Große, and Rolf Drechsler. 2019. fiction: An Open Source Framework for the Design of Field-coupled Nanocomputing Circuits. arXiv:1905.02477

[34] Marcel Walter, Robert Wille, Frank Sill Torres, Daniel GroÃ§e, and Rolf Drechsler. 2019. Scalable Design for Field-coupled Nanocomputing Circuits. In *ASPDAC '19 Proceedings of the 24th Asia and South Pacific Design Automation Conference*. 197–202. https://doi.org/10.1145/3287624.3287705

[35] Saeyang Yang. 1991. *Logic Synthesis and Optimization Benchmarks User Guide* (3 ed.). Microelectronics Center of North Carolina, P. O. Box 12889, Research Triangle Park, NC 27709.