

Optimizing Circuits for QCA Routing

Juliana Rezende S. B. Alves¹, Pedro Arthur R. L. Silva¹, Geraldo Fontes²,
Ricardo dos S. Ferreira², José Augusto M. Nacif¹

¹Instituto de Ciências Exatas e Tecnológicas, Campus UFV-Florestal

²Departamento de Informática, Campus UFV-Viçosa
Universidade Federal de Viçosa, Brazil

{juliana.baia, pedro.arthur, geraldo.fontes, ricardo, jnacif}@ufv.br

Abstract—Quantum-Dot Cellular Automata (QCA) is a new promising technology with great advantages to integrated circuits like faster speed, low power computation and very small size. QCA technology faces several implementation challenges which are being addressed by recent advances. One of the most difficult challenges related to the process of creating circuits with QCA technology is the routing process. In this paper we redesign some circuits with equivalent logic through simplification of functions in order to analyze the circuit routing, also making a comparison among the circuits.

I. INTRODUCTION

Current industry technologies, like CMOS (Complementary Metal-oxide Semiconductor), for manufacturing of integrated circuits are nearly reaching their physical limits [4] and new technologies are needed to replace CMOS. Some new nanotechnologies are promising alternatives to CMOS.

QCA (Quantum-Dot Cellular Automata) technology [6] uses quantum dots instead of electric current. Although this technology is not yet available in industry, small scale QCA cells have been fully tested and manufactured [5]. QCA consists in bi-stable cells connected by a force field capable of arranging itself to perform logical operations [7]. This technology has been applied to develop logic circuits which have presented low power, small size and high clock rates.

QCA is basically a nanometric square cell with four quantum dots and two electrons. Each quantum dot is close to a vertex of the square. The QCA cell can be polarized, which means that the electrons can move to two diagonally opposed corners of the cell. Because of their Coulomb repulsion they will go to a diagonal opposite in the cell and this will make the cell polarize at -1 or $+1$, which represents the logic 0 and 1, as we can see in Fig. 1. The black dots are the quantum dots containing an electron.

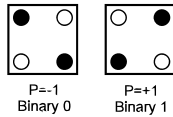


Fig. 1: Polarized QCA cells.

When the cell is polarized, the electrons move to their positions influencing nearby cells by Coulomb interactions.

Consequently, a cell that is polarized at $+1$, influences nearby cells to polarize at $+1$. This is the principle that explains why QCA technology does not need electric current, as the information is passed along cells with Coulomb interactions. The polarization of cells is used to encode binary information, and with the help of physical interaction of the QCA cells, boolean logic is implemented.

Build a QCA circuit with an effective routing is very important and requires time and ability because a QCA circuit follows a specific data flow. Since QCA technology follows specific patterns, one of the challenges is finding a better routing for the circuit because traditional circuits are not optimized for QCA routing.

Before building the circuit it is necessary to analyze the boolean function. There are many different strategies to synthesize and implement a logic function. When we are implementing the logic function we have to analyze the number of cells and gates in the circuit. Placement determines the location of each of these elements and the next step is the routing which adds wires to connect the components.

Connections between different cells are provided by wires that may extend through or past one or more cells to provide connections in the integrated circuit [1]. When we connect the components of the circuit we have to follow design rules and an efficient routing algorithm is needed. To reach an efficient routing we should find a positive trade off between area and speed.

Majority logic can be investigated to improve the synthesis and optimization of the circuits. Recently, a novel logic representation structure has been proposed to optimize Boolean functions called Majority-Inverter Graph (MIG) [2]. This proposal is completely based on majority and inverter operations and may reduce the number of logic levels.

In this work we investigate MIG-based optimization techniques of boolean functions which provide a more effective QCA routing. Besides that, we will use an algorithm for placement and routing for Quantum-Dot Cellular Automata, which is based on the approach proposed in [8], for verify the routing in the circuit. Analyzing the area and speed of the implemented circuits we were able to define the best strategy to build a QCA circuit in order to facilitate routing. We have implemented two logic functions with different logic gates.

This paper is organized as follows: Section II presents the methodology used in this study. The results are presented in section III and the conclusion and future work are shown in section IV.

II. METHODOLOGY

The two logic functions explored in this work are described by Equations 1 and 2.

$$F = x(y + uv) \quad (1)$$

$$F = x \otimes y \otimes z \quad (2)$$

When we build circuits in QCA we have to take care about the synchronization of the inputs and outputs. It's important that the inputs stay synchronized in all the majority gates on the same level, and the placement of inputs is very relevant, as well as routing.

All the circuits shown here have been built using QCADesigner [9] with USE (Universal, Scalable and Efficient) clocking scheme [3] which facilitates synchronization. USE clocking scheme controls the dataflow through its four clock zones. These clock zones are represented by the colors: green, purple, turquoise, and white.

We use MIG-based methods [2] to generate optimized graphs to be used in circuit placement and routing. A MIG is a homogeneous logic network that is an acyclic graph with each node representing the majority function. In a MIG, edges are marked by a regular or complement attribute. MIGs can be optimized in terms of area (size), delay (depth) and power (switching activity).

The MIG Boolean Algebra [2] is defined over the set $(B, M, ', 0, 1)$ where M is the majority operator of three variables and $'$ is the complement operator [2]. There is a set of five primitive transformation rules, referred to Ω , which is an axiomatic system for $(B, M, ', 0, 1)$. Because the length of the exact transformation sequence of a MIG might be impractical for modern computers three transformations have been derived from Ω . These transformations, referred to as Ψ , facilitates the MIG manipulation task.

In this work we apply MIG-depth optimization technique [2] in the circuits described by Equations 1 and 2. This technique is presented in Algorithm 1.

Algorithm 1: MIG-DEPTH OPTIMIZATION PSEUDO-CODE.

Data: $MIG\alpha$

Result: $OptimizedMIG\alpha$

```

1 for (cycles=0; cycles<effort; cycles++) do
2    $\Omega.Ml \rightarrow R(\alpha); \Omega.Dl \rightarrow R(\alpha); \Omega.A(\alpha); \Psi.C(\alpha);$ 
3    $\Omega.A(\alpha); \Psi.C(\alpha);$ 
4    $\Psi.R(\alpha); \Psi.S(\alpha);$ 
5    $\Omega.Ml \rightarrow R(\alpha); \Omega.Dl \rightarrow R(\alpha); \Omega.A(\alpha); \Psi.C(\alpha);$ 
6 end

```

The placement and routing algorithm for QCA, based on the approach proposed in [8], is a tool that automates the realization of a QCA circuit, and also verifies wire crossing and clock synchronization at lower level. This method generates solutions based on the USE clocking scheme [3] represented by a square matrix. An even line of the matrix has direction from left to right and a even column has direction from the bottom up, for odd lines the direction is from right to left and odd columns from top to bottom.

We model each circuit described by Equations 1 and 2 using a majority circuit representation. Figures 2 and 3 present the majority circuits related to the circuit described by Equations 1 and 2, respectively. Each majority gate corresponds to a node in a MIG. It is important for the reader to note that a 3-input majority gate behaves as a 2-input AND gate when the third input is connect to logic level 0 and as a 2-input OR gate when the third input is connected to logic level 1.

Figures 2a and 3a present the original majority circuits described by Equations 1 and 2, respectively. The original majority circuits are used as inputs to the MIG depth optimization algorithm (Algorithm 1), leading to the circuits illustrated in Figures 2b and 3b, respectively. The applied optimization decreases the circuits depths by 1 level on the first circuit and 2 levels on the second circuit.

The next step is apply the placement and routing algorithm described in [8] to both original (Figures 2a and 3a) and depth optimized circuits (Figures 2b and 3b). The resulting circuits are presented in Figures 4 and 5, respectively. Figures 4a and 5a present the layouts after placement and routing of the original circuits, while Figures 4b and 5b present the layout after placement and routing of the depth optimized circuits.

III. RESULTS

This Section presents the area and synthesis time results obtained from the redesigned circuits. Tables I and II show a comparison between the area and synthesis time of the original circuits and their depth optimized versions. Comparing both circuits we can analyze the impact the depth optimization on the generated layout.

In Tables I and II we can see that the depth optimization [2] leads to circuits that occupy less area. For the Equation 1 circuits (Table I) the difference is around 10%. However, the layout generated for optimized majority circuit described by Equation II occupies less than 50% of the original circuit area. On the other hand, the time consumed by the synthesis process varies and does present a well defined trend.

TABLE I: Area and synthesis time for Equation 1.

Function $x(y + uv)$	Area (Squares ^a)	Synthesis time
Original circuit layout	20	8 s
Optimized circuit layout	18	15 s

^aThe squares consider each cell with 18 nm length and 18 nm width. Each square contains 5x5 cells.



Fig. 2: Majority circuits described by Equation 1.

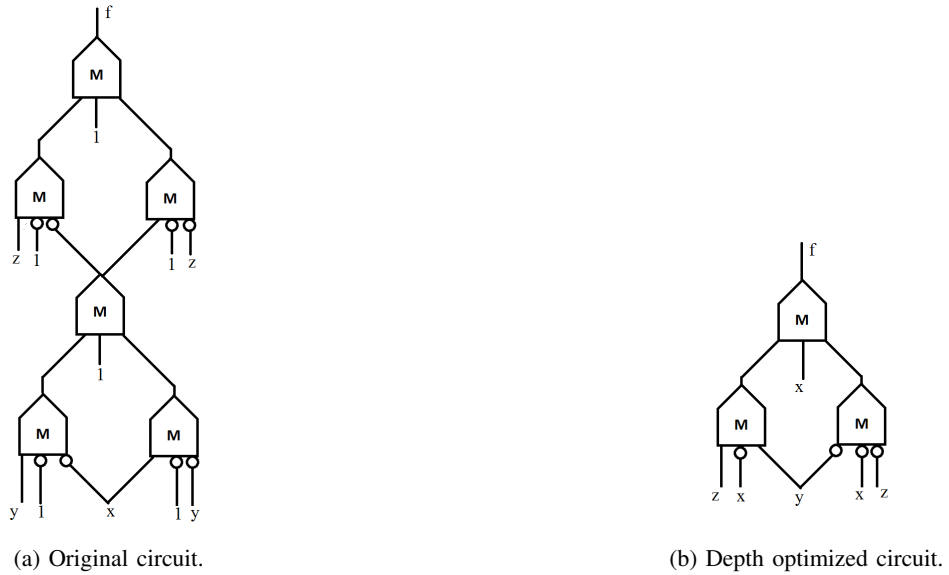


Fig. 3: Majority circuits described by Equation 2.

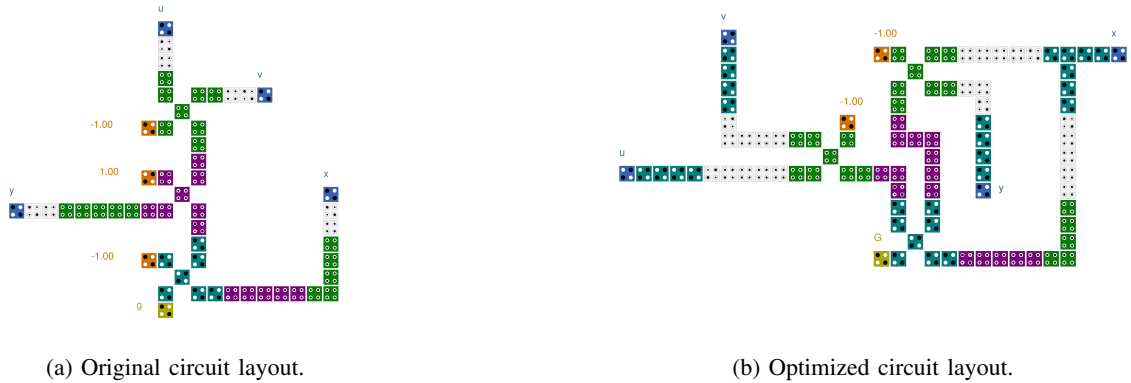


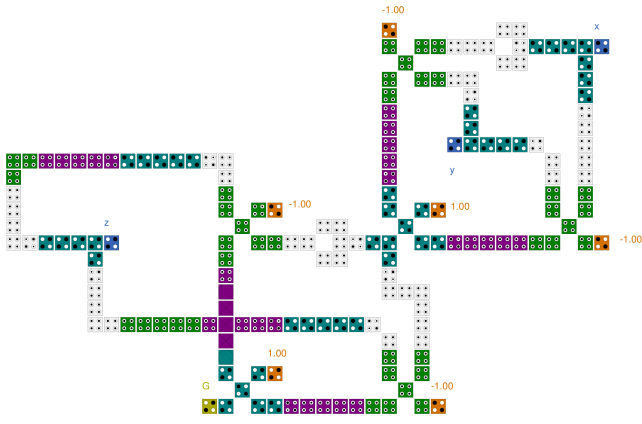
Fig. 4: QCA circuit layout for Equation 1.

IV. CONCLUSIONS AND FUTURE WORK

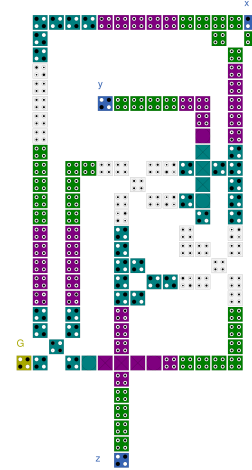
This work compared the layout area generated by a QCA placement and routing tool over two classes of circuits described in majority logic: (1) original circuit; (2) depth optimized circuit.

We showed examples of majority logic depth optimization that decrease area occupied by a QCA layout generated by a placement and routing tool.

As future work, we plan to extend these results to more complex QCA circuits.



(a) Original circuit layout.



(b) Optimized circuit layout.

Fig. 5: QCA circuit layout for Equation 2.

TABLE II: Area and synthesis time for Equation 2.

Function XOR	Area (Squares ^a)	Synthesis time
Original circuit layout	40	30 s
Optimized circuit layout	18	25 s

^aThe squares consider each cell with 18 nm length and 18 nm width. Each square contains 5×5 cells.

ACKNOWLEDGMENTS

We would like to thank CAPES, CNPq, and FAPEMIG for the financial support.

REFERENCES

[1] A. Ali and V. Patel. Integrated circuit routing, Mar. 9 2004. US Patent 6,704,918.
 [2] L. Amarú, P-E. Gaillardon, and G. De Micheli. Majority-inverter graph: A novel data-structure and algorithms for efficient logic optimization. In *Proceedings of the 51st Annual Design Automation Conference, DAC '14*, pages 194:1–194:6, New York, NY, USA, 2014. ACM.

[3] C. A. T. Campos, A. L. Marciano, O. P. V. Neto, and F. S. Torres. Use: A universal, scalable, and efficient clocking scheme for qca. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(3):513–517, March 2016.
 [4] D. Jorgenson. Moore's law and the emergence of the new economy. *Semiconductor Industry Association 2005 annual report: 2020 is closer than you think*, pages 16–20, 2005.
 [5] R. K. Kummamuru, A. O. Orlov, R. Ramasubramaniam, C. S. Lent, G. H. Bernstein, and G. L. Snider. Operation of a quantum-dot cellular automata (qca) shift register and analysis of errors. *Electron Devices, IEEE Transactions on*, 50(9):1906–1913, 2003.
 [6] C. S. Lent and P. D. Tougaw. A device architecture for computing with quantum dots. *Proceedings of the IEEE*, 85(4):541–557, 1997.
 [7] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernstein. Quantum cellular automata. *Nanotechnology*, 4(1):49, 1993.
 [8] A. Trindade, R. Ferreira, J. Nacif, D. Peixoto, and O. V. Neto. A placement and routing algorithm for quantum-dot cellular automata. In *SBCCI - Symposium on Integrated Circuits and Systems Design*, pages 1–6. IEEE, 2016.
 [9] K. Walus, T. J. Dysart, G. A. Jullien, and R. A. Budiman. Qcadesigner: a rapid design and simulation tool for quantum-dot cellular automata. *IEEE Transactions on Nanotechnology*, 3(1):26–31, March 2004.