Methodology for Automated Design of Quantum-dot Cellular Automata Circuits

Orestis Liolis, *Member, IEEE*, Vassilios A. Mardiris, Ioannis G. Karafyllidis, Sorin Cotofana, *Fellow, IEEE*, and Georgios Ch. Sirakoulis, *Member, IEEE*,

Quantum-dot Cellular Automata (QCA) provide very high scale integration potential, very high switching frequency, and have extremely low power demands, which make the QCA technology quite attractive for the design and implementation of large-scale, high-performance nanoelectronic circuits. However, state-of-the-art QCA circuit designs were not derived by following a set of universal design rules, as is the case of CMOS circuits, and, as a result, it is either impossible or very difficult to combine QCA circuit blocks in effective large-scale circuits. In this paper, we introduce a novel automated design methodology, which builds upon a QCA specific universal design rules set. The proposed methodology assumes the availability of a generic QCA crossbar architecture and provides the means to customize it in order to implement any given logic function. The programming principles and the flow of the proposed automated design tool for crossbar QCA circuits are described analytically and we apply the proposed automated design method for the design of both combinatorial and sequential circuits. The obtained designs demonstrate that the proposed method is functional, easy to use, and provides the desired QCA circuit design unification.

***Index Terms*—Quantum-dot Cellular Automata (QCA), Design methodology, Crossbar architecture, Nanoelectronics.**

I. Introduction

QUANTUM-DOT Cellular Automata (QCA) have been proposed in 1993 by Lent *et al.* [1] and QCA technology potentially provides an avenue beyond Moore's Law and von Neumann architecture electronics. In QCA technology, the logic states are not represented by voltage levels like in the VLSI/CMOS technology, but defined by the Quantum dots that are occupied by the individual electrons within a cell. Due to their great potential many QCA circuits have been proposed [2], [3], [4], [5], [6] and novel fabrication techniques developed [7], [8], [9], [10], [11], which, even though they need to be further developed, are providing a realistic roadmap for future QCA based nanoelectronics. Moreover, a programmable QCA crossbar architecture was introduced in [12], which provides designers the means to obtain robust and efficient QCA circuit designs. In this architecture, programmable logic gates are formed at crossbar cross-points, which function can be determined via the programming lines located at the crossbar top and bottom. Given that, at every and every cross-point one of the universal set of Boolean gates {OR, AND or NOT} can be instantiated, the architecture provides support for the implementation of any digital circuit. We, note that the crossbar architecture is considered as one of the most promising solutions for nanoelectronic circuits [13], because of its fabrication simplicity and the inherent redundancy, which supports defect tolerance [14], [15], [16], [17], [18]. However, state-of-the-art QCA circuit designs were not derived by following a set of universal design rules, as is the case of CMOS circuits, and, as a result, it is either impossible or very difficult to combine QCA circuit blocks in effective large scale circuits.

In this paper, we address this problem and propose an automated methodology for the design of combinatorial and sequential circuits by making use of a programmable QCA crossbar architecture [12]. The proposed methodology aims to tackle one major QCA circuit design issues and provide a design automation that enables compatibility between different QCA circuits. We note that even if the combination of state-of-the-art QCA circuits can be feasible in some specific cases, the interconnection circuit overhead is usually overwhelming, because it can be even larger than the circuits themselves. These compatibility issues result from the lack of universal design rules. The proposed methodology is utilizing the fundamental design rules of the programmable QCA crossbar architecture. In addition, it introduces the universal QCA structural blocks that can be used to design any combinatorial logic circuit. Moreover, the presented methodology is successfully handling the clock zone partitioning to resolve any corresponding signal timing and robustness issues.

In order to design sequential logic QCA circuits, the proposed methodology enhances the aforementioned set of QCA structural blocks with a memory element block. As a result, the design of a memory cell on the programmable QCA crossbar architecture is considered as a prerequisite for the further development of the introduced design methodology [19]. This memory cell provides the means for creating 2^n-bit memories and, at the same time, provides *effective programmability*. This means that the same QCA circuit in programmable crossbar architecture can be either used as a memory cell or as a processing unit. Such a design perspective is possible by exploiting the features of the programmable QCA crossbar architecture, to be analyzed in the next sections. Thus, in the proposed methodology, the memory element block, along with the combinatorial logic QCA blocks are employed to design any sequential logic circuit, while both memory element blocks and combinatorial logic blocks are implemented into the same crossbar.

The research work was supported by the Hellenic Foundation for Research and Innovation (HFRI) and the General Secretariat for Research and Technology (GSRT), under the HFRI PhD Fellowship grant (Project Number 81951).

Orestis Liolis, Ioannis G. Karafyllidis, and Georgios Ch. Sirakoulis are with Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi GR–67100, Greece

Vassilios A. Mardiris is with International Hellenic University, Kavala GR–65404, Greece

Sorin Cotofana is with the Electrical Engineering, Mathematics and Computer Science Faculty, Delft University of Technology, Delft, The Netherlands

The proposed methodology as clearly already stated focus on automated QCA design and not on the proposed clocking schemes. As a result, it shouldnt be compared with several clocking schemes that have already been introduced earlier in the literature [20], [21], [22], having also in mind that the usage of a fixed distribution clocking scheme has several advantages for the design and fabrication of a QCA circuits. As a general comment, every clocking scheme has its drawbacks and, as a result, the majority of the proposed circuits in the literature do not follow any specific scheme. Furthermore, even though these schemes are trying to tackle the clock signal distribution problem, they cant be generally used for all design problems. Just to name some of the open issues that merit further investigation are the quantum-dot cells manual placement, the random location of I/O quantum-dot cells in the circuits and the overhead of the combination. These issues can be overwhelming. Nevertheless, and for sake of clarity, it should be mentioned that in the proposed design methodology, the clock signal distribution is not random, as it is explicitly stated in Section II, while the clock zones for each block are defined properly, and the clock zones sequence is cascadable.

Apart of the methodology, we also present an automated QCA circuit design software tool that automatically generates the QCA circuit layout corresponding to a given user specified logic function. Up to our best knowledge, no similar tool with the similar abilities exists. To further demonstrate the capabilities of the proposed methodology and the corresponding tool, the design of various combinatorial and sequential QCA circuits is delivered together with the corresponding simulation results obtained by QCADesigner [23] based simulations for a default cell size of $18\,\mathrm{nm} \times 18\,\mathrm{nm}$.

The structure of the paper is as follows: Section II introduces the proposed methodology for automated combinatorial QCA circuit design and Section III presents its utilization for the design of two QCA circuit examples. Section IV describes the software tool for automated QCA circuits design and Section V, extends the proposed methodology for the automated design of QCA sequential circuits. In Section VI, two sequential circuits are presented and paper conclusions are drawn in Section VII.

II. Automated QCA Combinatorial Circuit Design

In this section, we introduce a novel methodology for automated combinatorial QCA circuits design. As mentioned in the introduction, the lack of a universal QCA circuit design methodology results in compatibility issues between reported QCA circuits and this is the very problem our methodology is aiming to overcome. Namely, we propose a universal design methodology that, given a combinatorial function F and the generic programmable crossbar of quantum-dot cells proposed in [12], can create a QCA circuit instance able to evaluate F. The programmable QCA crossbar architecture stability has been verified in [12]. This verification has been made theoretically and with the most widespread and reliable simulation tools found in the literature. In this section, only the basic design rules of the architecture will be presented. The programmable QCA crossbar architecture consists of an array of quantum-dot cells (see Fig.1) and a set of rules that can be employed to map any digital circuit onto the crossbar. These rules define how to handle circuit inputs and outputs, how to form logic gates at the crossbar cross points, and how to (re)configure the logic gate operation even during circuit operation. In particular, cross-shape majority gate [24] is one of the very first and most used logic gates in digital design in QCA technology. Using the majority gate, the `OR` and `AND` logic gates can be implemented. More specifically, if one of the three inputs is fixed polarized at -1 (i.e. logic $'0'$)the majority gate is operating as an `AND` gate, and if one of the three inputs is fixed polarized at $+1$ (i.e. logic $'1'$) the majority gate is operating as an `OR` gate. This fixed polarization input is the programming cell. Namely, by polarizing this input either at $+1$ or -1, the same cells topology operates either as `OR` or `AND` gate, respectively. In addition, in [12] a cross-shape inverter has been proposed. These cross-shape logic gates that can be formed at the cross points of the programmable crossbar are shown in Fig.1. As Fig.1 depicts, the programming cells that are used to define the operation of the majority gates are located to the top and to the bottom of the circuit, the inputs are located to the left and the outputs are located to the right.

The straightforward information flow, the well-defined I/O interface, the fixed position of the quantum-dot cells, and the programmability feature make the programmable QCA crossbar architecture the best candidate architecture for an automated QCA design methodology, enabling both scalability and productivity in QCA circuits design.

Even though the programmable QCA crossbar architecture defines a universal design rules set, it does not provide a generic design methodology that can automatically generate the QCA circuit for any design case. Namely, there are still many design problems that need be solved manually, e.g., circuit clocking, logic gate positioning, programming lines distribution. Our design methodology is aiming to provide an efficient and generic solution to these problems.

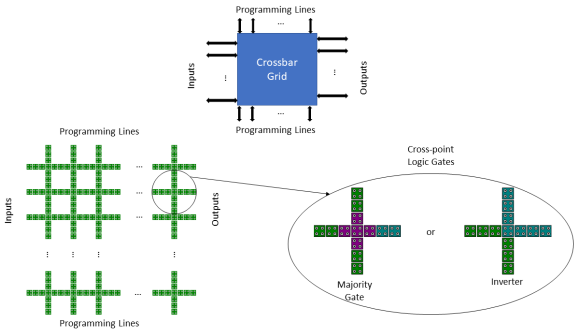

Fig. 1: Programmable QCA crossbar architecture.

The first step towards the development of a universal QCA circuit design methodology is the definition of the circuit information flow. In the proposed design methodology, the information propagates from the left side towards the right side of the circuit, which is achievable by the appropriate handling of the clock zone partitioning. Note that clock zone partitioning is one of the most important QCA circuit design phase, and adiabatic switching [25], [26] is currently considered to

be the best clocking technique able to provide stability and information flow control within QCA circuits. In adiabatic switching, the electrons of every cell are pushed to either neutral state or one of the two possible logic states. In the latter case, the prevailing logic state depends on the polarization of the neighboring cells. This adjustment of electron motion is achieved through applied electric fields controlled by four-phase clock signals, *Switch*, *Hold*, *Release*, and *Relax*, which have a relative phase difference of 90°. The quantum-dot cells in a clock zone are all controlled by the same clock and, as such, the information propagates from one clock zone cells to their neighboring cells of the next clock zone.

To evaluate a given combinatorial function F by means of the QCA technology we rely on fundamental QCA blocks that are able to perform basic Boolean algebra operations, i.e., `NOT`, `AND`, `OR`. Thus, the first design step consists of rewriting F in terms of Boolean algebra operations such that its QCA implementation can be done by means of fundamental QCA blocks only. Subsequently, the selected QCA blocks are to be instantiated within the crossbar space.

Firstly, for blocks placement, we have to take into consideration the logic operations hierarchy. For example, $F = (A \cdot B) + (C \cdot D)$ is implemented in two logic levels, as indicated in Fig. 2, with two `AND` gates in level 1 and one `OR` gate in level 2 with level 1 outputs being level 2 inputs.

The programming line within the crossbar architecture [12] are located at array top and bottom, such that the upper (lower) block makes use of the top (bottom) programming lines. Consequently, each circuit level can accommodate at most two blocks because extra blocks don't have any available programming lines to utilize. In order to overcome this problem, intermediate levels (sub-levels) need to be added. For example, in order to implement $F = (A \cdot B) + (C \cdot D) + (E \cdot F)$ the 3 `AND`-gates are placed on the two level 1 sub-levels and the `OR`-gate in level 2, as depicted in Fig. 3. The number of sub-levels of level i (NSL_i) is defined as

$$NSL_i \geq \frac{NB_i}{2}, \tag{1}$$

where NB_i is the number of blocks at level i. In this particular example, $NB_1 = 3$ and $NB_2 = 1$.

The proposed methodology can be also utilized to design multi output QCA circuits, case in which each and every function F_i has to be implemented separately, as suggested in Fig. 4. Though, since one function is placed bellow the other, we can utilize both top and bottom programming lines in every function. Namely, in the example of Fig. 4 F_1 is utilizing the top programming lines and F_2 is utilizing the bottom.

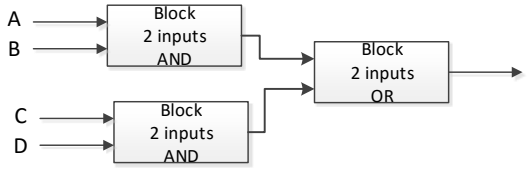

Fig. 2: Diagram of function $(A \cdot B) + (C \cdot D)$.

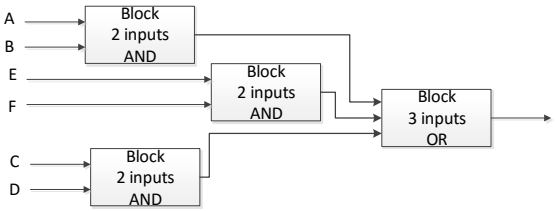

Fig. 3: Diagram of function $(A \cdot B) + (C \cdot D) + (E \cdot F)$.

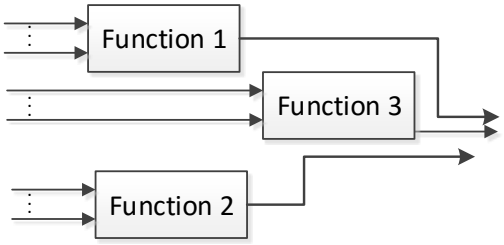

Fig. 4: Diagram of the implementation of more than one function.

QCA blocks can be classified in two classes: (i) blocks that implement basic Boolean algebra operations and (ii) interconnect facilitators. Type (i) blocks can be further divided into subcategories depending on their input cardinality. Figs. 5 - 7 present examples of the proposed type (i) blocks of the proposed methodology, i.e., Fig. 5 depicts a 2-input block, while a 3-input block and a 4-input block is presented in Fig. 6 and Fig. 7, respectively. We note that all circuits are designed with QCADesigner design tool [23].

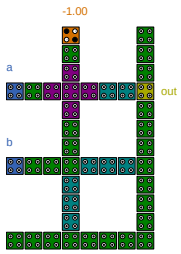

Fig. 5: 2-input block evaluating $a \cdot b$.

The second category contains blocks that can be utilized for signal crossing and branching. The four crossing cases and the two branching cases are presented in Fig. 8 and Fig. 9, respectively.

We note that blocks belonging to the same subcategory are designed such that they exhibit the exact same delay, namely, all 4-input blocks introduce a 7 clock zones delay, all branching blocks a 3 clock zones delay, and so on. This uniform delay block design policy eases the handling of synchronization constraints at the circuit level. However,

This article has been accepted for publication in IEEE Open Journal of Nanotechnology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJNANO.2022.3223413

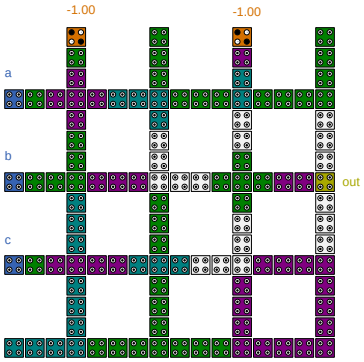

Fig. 6: 3-input block evaluating $a \cdot b \cdot c$.

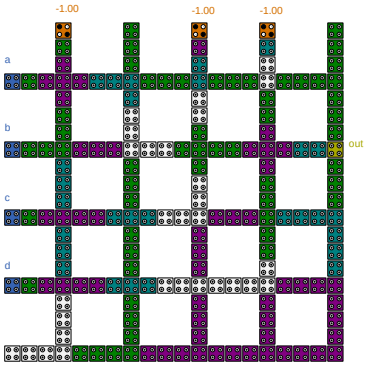

Fig. 7: 4-input block evaluating $a \cdot b \cdot c \cdot d$.

even though the earlier described blocks exhibit fixed and known delays, in QCA circuits the signal propagation between adjacent blocks is performed by binary wires, which induce a wire length dependent delay overhead. Taking this into consideration, QCA blocks placement and binary wires routing are crucial parts of any design technique following QCA operation principles, which seeks the realization of stable and functional QCA circuits. The fact that blocks placed in the upper/lower half of the circuit are utilizing the top/bottom programming lines allows for the realization of different information flows into the upper and the lower parts of the circuit, which are both converging to the right center of the circuit, where the circuit output is located. Moreover, circuit partition into levels and sub-levels enables wire length minimization such that wire induce delay becomes manageable.

The systematic block delays policy combined with interconnection wire length minimization allow the methodology to properly address robustness issues also. Namely, since

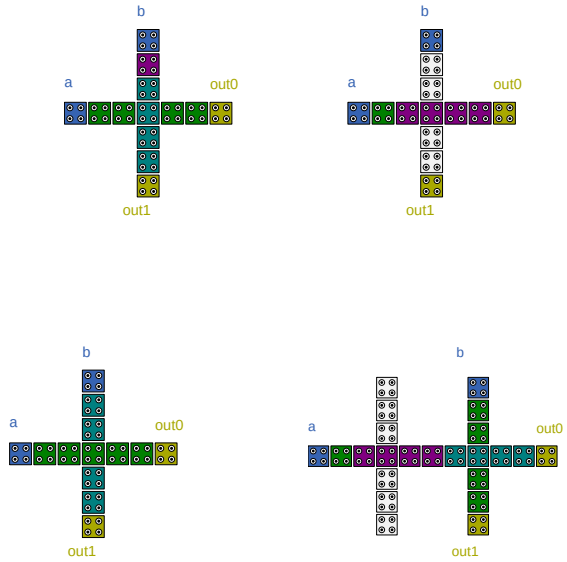

Fig. 8: The blocks that implement the 4 different crossing cases.

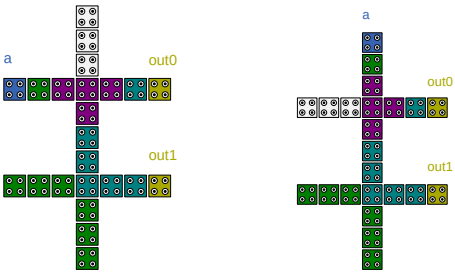

Fig. 9: The blocks that implement the 2 different branching cases.

interconnection wires are as small as possible, it is easier to avoid kinks, i.e., occasions where a quantum-dot cell has a different polarization than the expected one. The maximum length of a QCA binary wire [27] in a specific clock zone is given by

$$N \leq e^{\frac{E_k}{k_b T}}, \tag{2}$$

where E_k is the kink energy, k_b Boltzmann's constant, and T the temperature. The kink energy between two Quantum dots is calculated as

$$E_k^{i,j} = \frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q_i q_j}{|r_i - r_j|}. \tag{3}$$

Thus, to achieve stability any QCA circuit has to be divided into as many clock zones as required while fulfilling Eq. (2). On the other hand one compromise should be thought as the more clock zones are utilized the larger the circuit delay. The proposed methodology is handling all the above issues and the obtained circuits have the smallest possible delay.

To summarize, the proposed methodology encompasses the following steps:

- Transform the to be implemented logic function expression such that it can be implemented by the predefined basic QCA blocks.
- Partition the circuit into levels and sub-levels.
- Place the blocks onto the crossbar grid, while utilizing the top and the bottom programming lines, as earlier described.
- Connect the placed blocks, while taking into consideration clocking and synchronization constraints.

III. Design Methodology Application

In this section, we present two example designs, a $2:1$ and a $4:1$ multiplexer, derived by means of the methodology introduced in the previous section.

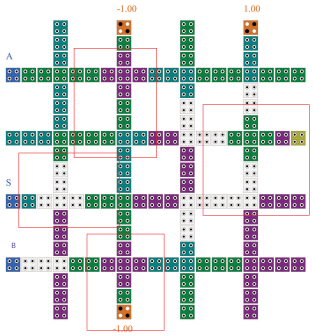

Fig. 10: QCA $2:1$ multiplexer.

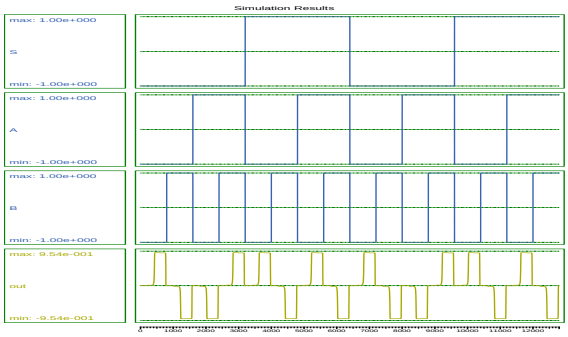

Fig. 11: Fig. 10 circuit simulation results.

The logic function that described the output of a $2:1$ multiplexer is $A \cdot S' + B \cdot S$. Fig. 10 depicts the QCA circuit obtained by following the proposed QCA circuit design methodology. The basic QCA blocks are located inside the red boxes and as one can observe in the Figure the circuit is divided into three levels. Initially the S signal is branched in order to be utilized as input for both 2-input AND gates in the second circuit level. The supplementary input for the upper gate is A, while B is the second input for the lower gate. These two QCA blocks are horizontally mirrored because the programming lines of the first one are located at the top, and the programming lines of the second one at the bottom. The outputs of these two AND gates are inputs of the 2-input OR gate located in the third and final circuit level. The implementation makes use of 136 Quantum-dot cells that occupy $0.16\mu m^2$ and has a 7 clock zones delay. In Fig. 11, the simulation results that prove the circuit functionality are presented. For the simulation of the $2:1$ multiplexer as well for all the other circuits that are presented in the following sections, we made use of the QCADesigner [23]. All the simulations were preformed with QCADesigner coherence vector simulation engine default parameters and default cell size, namely $18\,\mathrm{nm} \times 18\,\mathrm{nm}$.

Likewise, Fig. 12 and Fig. 13 present the $4:1$ multiplexer design and simulation results, respectively. As the $4:1$ multiplexer output behaviour is described by $A \cdot S1' \cdot S0' + B \cdot S1' \cdot S0 + C \cdot S1 \cdot S0' + D \cdot S1 \cdot S0$ its implementation requires four 3-input AND gate blocks and one 4-input OR gate block. The implementation requires 2 levels, while the first level that includes the 4 AND gates, has been implemented in 2 sub-levels. The resulting circuit consists of $1,080$ Quantum-dot cells that occupy $1.06\,\mu\mathrm{m}^2$ and exhibits a 19 clock zones delay.

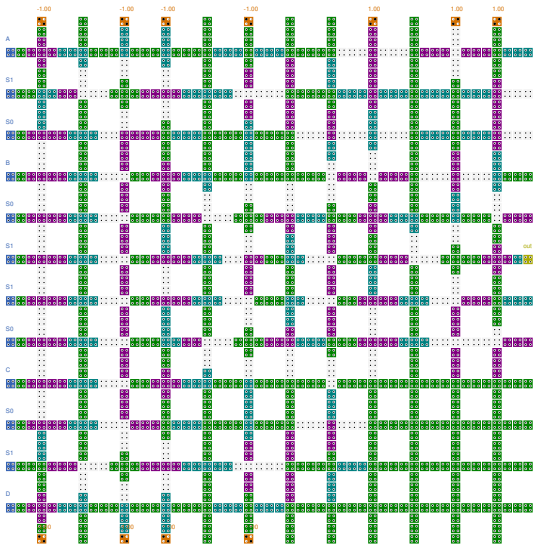

Fig. 12: QCA $4:1$ multiplexer.

IV. Automatic QCA Layout Generation

Based on the circuit design methodology introduced in Section II we developed in C++ a QCA circuit design automation tool. The user provides as input the logic function that she/he wants to implement and the tool automatically generates the layout of its QCA implementation in a QCADesigner compatible format in file with *.qca* extension. The circuit synthesis operation comprises the following five steps:

1) ***QCA blocks selection:*** The logic function F is analyzed and the necessary QCA blocks for its circuit level implementation are chosen.
2) ***QCA blocks position definition:*** The QCA circuit is divided into levels and each level is further divided into sub-levels, if needed, based on the methodology

This article has been accepted for publication in IEEE Open Journal of Nanotechnology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJNANO.2022.3223413

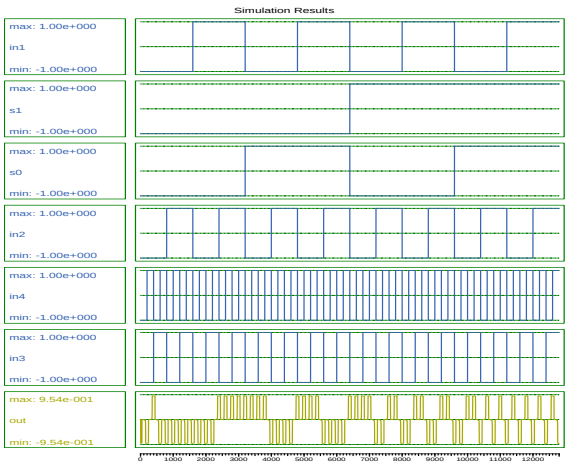

Fig. 13: Fig. 12 circuit simulation results.

discussed in Section II. Then, every block is placed at the corresponding level following the feedforward logical structure of F.

3) ***Wire placing and routing:*** Appropriate binary wires are instantiated in order to connect the outputs of prior blocks to the inputs of following blocks, from F's primary inputs towards its primary output.
4) ***QCA circuit clocking:*** The circuit is divided into clock zones from the left to the right, taking into consideration stability and the other previously discussed constrains.
5) ***Quantum-dot cells placing:*** The Quantum-dot cells are placed at their appropriate crossbar positions and clock zones, according to the positions determined during the previous steps.

Even though the tool relies on a Command Prompt User Interface (UI), the simplicity of the requested actions makes the tool user friendly. As mentioned before, the only required user action is to specify the to be implemented logic function. Fig. 14 depicts the QCA circuit layout for the evaluation of $a{\cdot}b{\cdot}(c{+}d)$ logic function, produced by the tool, without human interference. The correctness of the circuit is verified by means of QCADesigner simulations, which results are presented in Fig. 15.

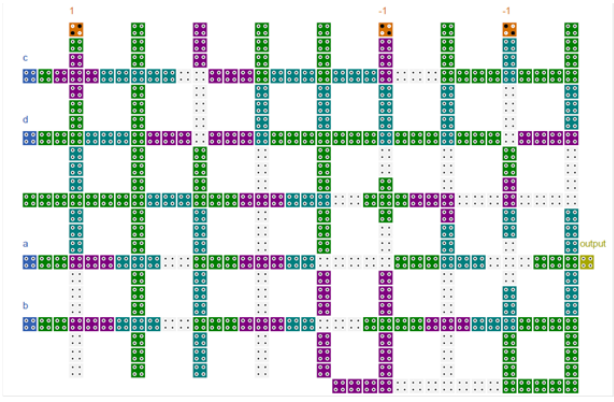

Fig. 14: QCA implementation of $F = a \cdot b \cdot (c + d)$.

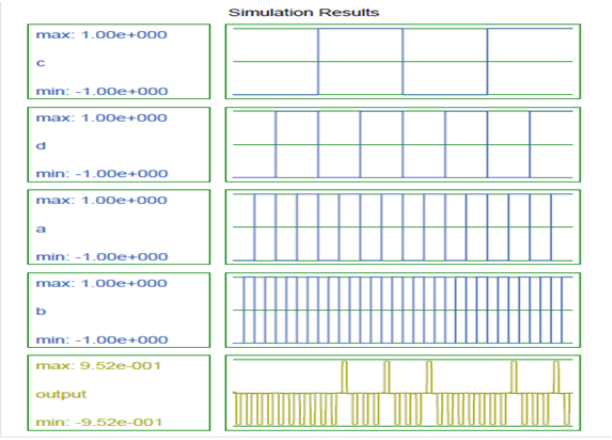

Fig. 15: Fig. 14 circuit simulation results.

V. Sequential QCA Circuit Design Methodology

The methodology of Section II can be utilized to derive the QCA layout of the circuit that implements any given combinatorial logic function. In this section, we introduce the required modifications that enable its utilization for the QCA implementation of sequential logic circuits.

Even though many memory designs have been proposed [28], [29], [30], [31], [32], [33], [34] the majority of them are not compatible with our targeted crossbar architecture. Thus, for the extension of the circuit design methodology to sequential logic we make use of the QCA memory cell presented in [19], which combines the basic advantages of the QCA technology with the capabilities of the programmable crossbar architecture. The memory cell circuit that it utilized by the proposed methodology is presented in Fig. 16.

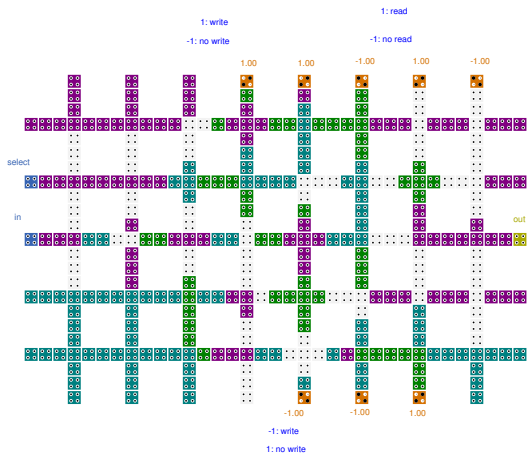

Fig. 16: Crossbar mapped QCA RAM cell in writing mode.

In this implementation the memory cell operations, i.e., read and write, are controlled by the programming lines located at the top and the bottom of the QCA crossbar. More specifically, in the read operation the polarization of the programming Quantum-dot cells value should alternate, i.e., if the polarization of the first top programming cell is -1 the polarization of the second should be $+1$ and vice-versa. The same pattern applies to the fourth and fifth top

and to the first two bottom programming cells. On the other hand, in write mode all the top/bottom programming cells have identical polarization, but the top cell polarization value is different than the polarization of the bottom cells.

The obvious advantage of this implementation is programmability, as the same circuit layout can be utilized in many applications by just changing the polarization of the programming Quantum-dot cells. This makes this RAM implementation promising, since the same circuit can be utilized for different applications with different storage and performance requirements. This RAM structure flexibility and adaptability make it quite attractive for QCA implementations. Last but not least, this approach provides the possibility to implement a given size memory on a prefabricated QCA crossbar, while other RAM QCA technology implementations require from scratch fabrication, which is a great advantage in view of the challenging nature of the QCA circuits fabrication process.

To extend the methodology from combinational to sequential circuit we extend the basic block with the QCA memory block presented in Fig. 16. The memory cell is always in writing mode and the select signal determines the to be stored data value. Fig. 17 presents the generic structure of a QCA circuits designing with memory, which combinational part can be generated by the approach introduced in Section II and the storage part by means of the previously discussed RAM block.

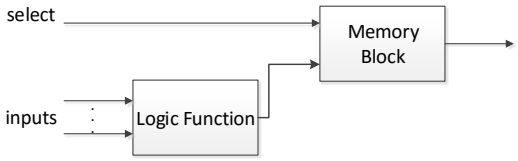

Fig. 17: Generic QCA circuit with memory.

VI. Automated Design Methodology For Sequential Logic Applications

To further clarify the implementation to digital circuits with memory elements on the generic QCA cell crossbar let we assume that the logic function that is computed and stored in Fig. 17 is $a + b + (c + d)$. Fig. 18 presents the QCA circuit obtained after the application of the proposed methodology. In the Figure, the red boxes delimitate the memory block and the two logic blocks and the blue boxes the two crossing blocks. Fig. 19 depicts the corresponding simulation result that demonstrate proper circuit functionality.

We also considered the design of a classic sequential electronic device, i.e., a 4-bit shift register, which has a serial data input and the second one that triggers data shifting. The QCA 4-bit right shift register circuit created by the proposed methodology is presented in Fig. 20, where the four memory blocks that have been used in the design are located inside the red frames. Fig. 21 presents simulation result and by comparing the four output waveforms produced by QCADesigner with Table I data, one can conclude that the QCA circuit behaves as expected.

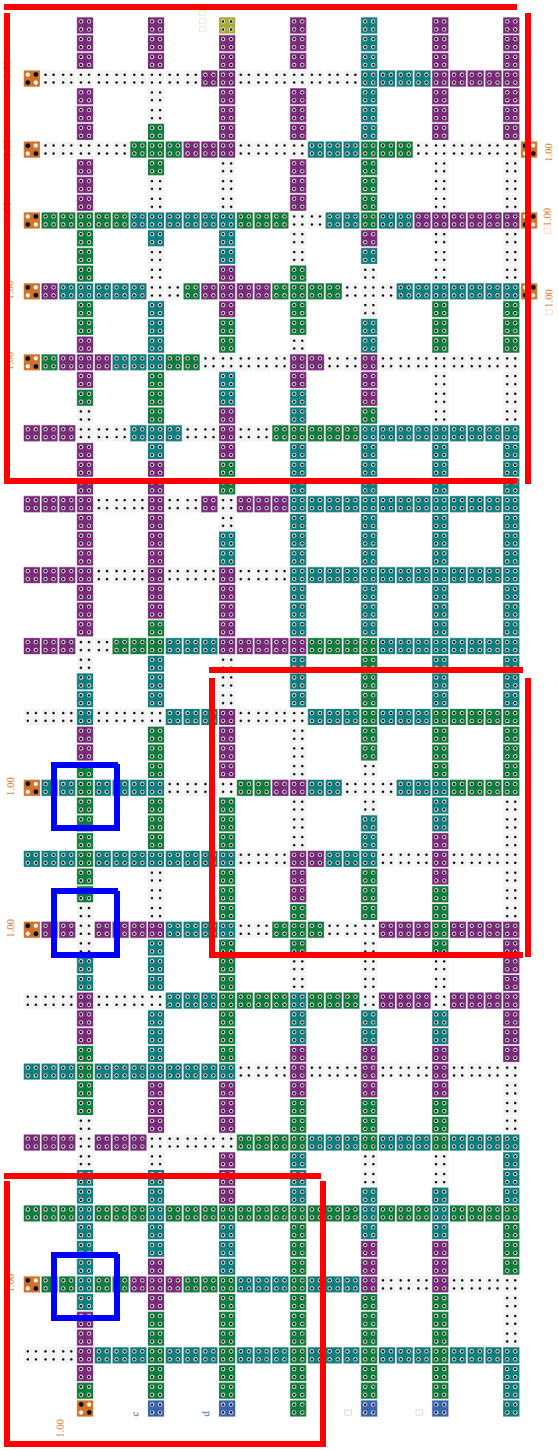

Fig. 18: $a + b + (c + d)$ QCA circuit with memory element.

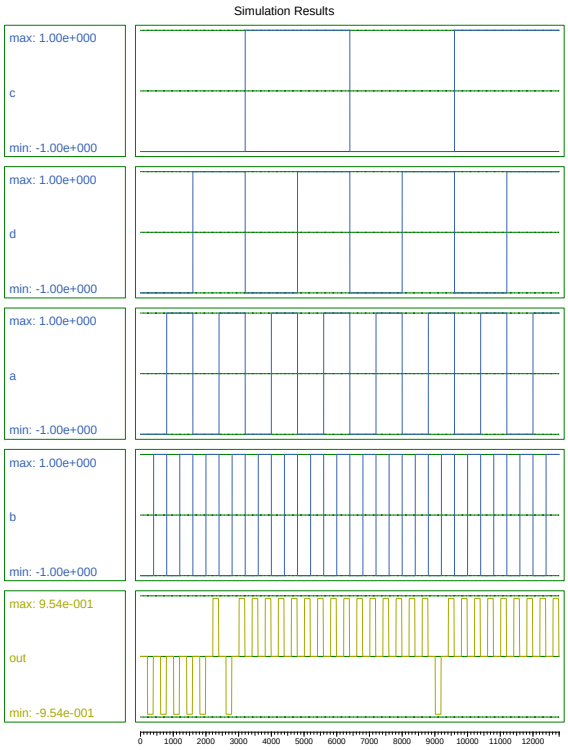

Fig. 19: $a+b+(c+d)$ with memory element simulation results.

VII. Conclusions

This paper addressed one of the QCA technology major issues, the lack of universal design methodologies and architectures, and by implication the unavailability of software design tools that can facilitate the design of large (with thousands quantum-dot cells) QCA circuits. We introduced an automated design methodology that makes use of a generic programmable QCA cell crossbar architecture to derive the implementation of any Boolean logic function. We utilized our proposal for the design of several Boolean logic circuits, which correct behavior was verified by means of the QCADesigner design and simulation tool. Moreover, we extended the methodology for the design of sequential circuits and utilized it for the QCA design of a 4-bit shift register. Furthermore, a software designing tool based on the proposed automated methodology was presented, which automatically generates the QCA circuit layout corresponding to a given user specified logic function.

Even though the proposed methodology constitutes the best solution to deal with the well-known QCA design challenges, future research issues could be considered aiming to continuous improvement of the proposed methodology towards even more fabrication friendly solutions. In particular, the proposed methodology should be considered as the first step towards the resolution of an important drawback of QCA technology once the fundamentals of automated design are established. The combination of the methodology with other clocking schemes could be the second step to that direction since this would make the clock signal distribution even more straightforward.

TABLE I: Example of 4-bit right shift register operation.

Shift Enable	1	1	0	0	0	0	0	1	0	1
Input data	1	0	X	X	X	X	X	0	X	1
Out 0	1	0	0	0	0	0	0	0	0	1
Out 1	0	1	1	1	1	1	1	0	0	0
Out 2	0	0	0	0	0	0	0	1	1	0
Out 3	0	0	0	0	0	0	0	0	0	1

References

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

[2] S. Seyedi and N. Navimipour, "An optimized design of full adder based on nanoscale quantum-dot cellular automata." *Optik*, vol. 158, pp. 243–256, 2018.

[3] S. Heikalabad, M. Asfestani, and M. Hosseinzadeh, "A full adder structure without cross-wiring in quantum-dot cellular automata with energy dissipation analysis." *The Journal of Supercomputing*, vol. 74, no. 5, pp. 1994–2005, 2018.

[4] S. Babaie, A. Sadoghifar, and A. Bahar, "Design of an efficient multilayer arithmetic logic unit in quantum-dot cellular automata (qca)." *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 6, pp. 963–967, 2018.

[5] N. Safoev and J. Jeon, "A novel controllable inverter and adder/subtractor in quantum-dot cellular automata using cell interaction based xor gate." *Microelectronic Engineering*, vol. 222, p. 111197, 2020.

[6] Y. Adelnia and A. Rezai, "A novel adder circuit design in quantum-dot cellular automata technology." *International Journal of Theoretical Physics*, vol. 58, no. 1, pp. 184–200, 2019.

[7] R. Wolkow, L. Livadaru, J. Pitters, M. Taucerg, P. Piva, M. Salomons, M. Cloutier, and B. Martins, "Silicon atomic quantum dots enable beyond-cmos electronics," *Chapter Field-Coupled Nanocomputing*, vol. Volume 8280 of the series Lecture Notes in Computer Science, pp. 33–58, June 2014.

[8] R. Achal, M. Rashidi, J. Croshaw, D. Churchill, M. Taucer, T. Huff, M. Cloutier, J. Pitters, and R. Wolkow, "Lithography for robust and editable atomic-scale silicon devices and memories," *Nature communications*, vol. 9, no. 1, pp. 1–8, 2018.

[9] T. Huff, H. Labidi, M. Rashidi, M. Koleini, R. Achal, M. Salomons, and R. Wolkow, "Atomic white-out: Enabling atomic circuitry through mechanically induced bonding of single hydrogen atoms to a silicon surface," *ACS nano*, vol. 11, no. 9, pp. 8636–8642, 2017.

[10] C. Lent, "Bypassing the transistor paradigm," *Science*, vol. 288, no. 5471, pp. 1597–1599, 2000.

[11] E. Blair, S. Corcelli, and C. Lent, "Electric-field-driven electron-transfer in mixed-valence molecules," *The Journal of Chemical Physics*, vol. 145, no. 1, p. 014307, 2016.

[12] V. Kalogeiton, D. Papadopoulos, O. Liolis, V. Mardiris, G. Sirakoulis, and I. Karafyllidis, "Programmable crossbar quantum-dot cellular automata circuits," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 36, no. 8, pp. 1367–1380, 2017.

[13] J. Heath, P. Kuekes, G. Snider, and R. Williams, "A defecttolerant computer architecture: Opportunities for nanotechnology," *Science*, vol. 280, pp. 1716–1721, 1998.

[14] G. Snider, P. Kuekes, and R. S. Williams, "Cmos-like logic in defective, nanoscale crossbars," *Nanotechnology*, vol. 15, no. 8, p. 881, 2004.

[15] Y. Chen, D. Jung, G.Y.and Ohlberg, D. Li, X.and Stewart, J. Jeppesen, K. Nielsen, J. Stoddart, and R. Williams, "Nanoscale molecular-switch crossbar circuits," *Nanotechnology*, vol. 14, no. 4, p. 462, 2003.

[16] A. DeHon, "Array-based architecture for fet-based, nanoscale electronics," *IEEE Trans. Nanotechnol.*, vol. 2, no. 1, pp. 23–32, Mar. 2003.

[17] I. Vourkas and G. Sirakoulis, "A novel design and modeling paradigm for memristor-based crossbar circuits," *Nanotechnology, IEEE Transactions on*, vol. 11, no. 6, pp. 1151–1159, 2012.

[18] C. Graunke, D. Wheeler, D. Tougaw, and J. Will, "Implementation of a crossbar network using quantum-dot cellular automata," *Nanotechnology, IEEE Transactions on*, vol. 4, no. 4, pp. 435–440, 2005.

[19] O. Liolis, G. Sirakoulis, V. Mardiris, and I. Karafyllidis, "Quantum-dot cellular automata ram design using crossbar architecture," *14th IEEE / ACM International Symposium on Nanoscale Architectures*, 2018.

[20] 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, 2007.

This article has been accepted for publication in IEEE Open Journal of Nanotechnology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJNANO.2022.3223413

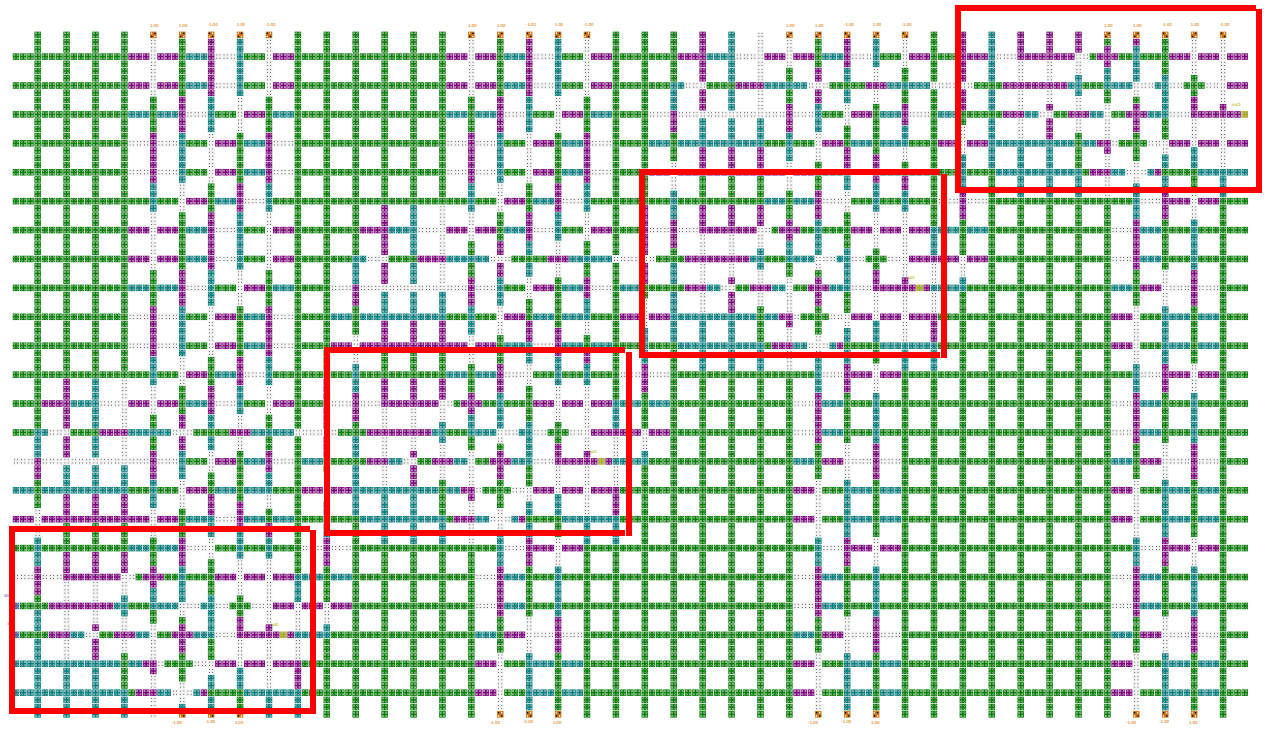

Fig. 20: 4-bit shift register QCA circuit.

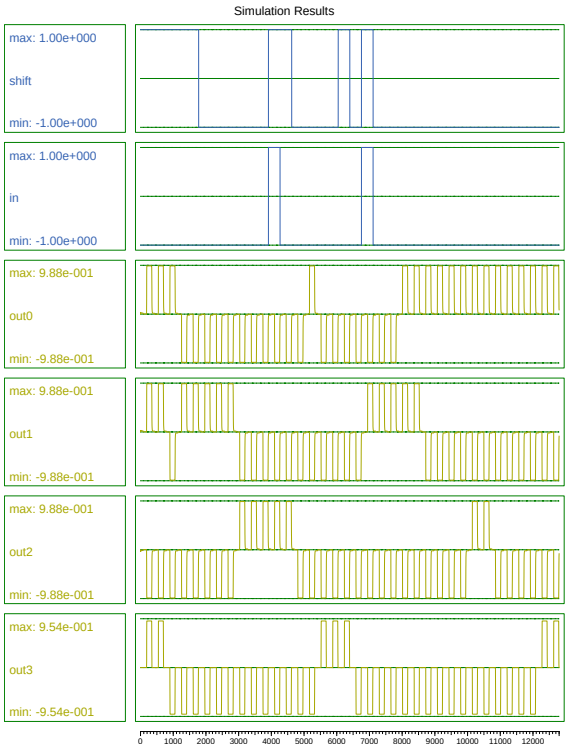

Fig. 21: Shift register simulation results.

[21] A. Campos, C.A.T.and Marciano, O. Neto, and F. Torres, "Use: a universal, scalable, and efficient clocking scheme for qca." *IEEE Transactions on computer-aided design of integrated circuits and systems*, vol. 35, no. 3, pp. 513–517, 2015.

[22] M. Goswami, A. Mondal, M. Mahalat, B. Sen, and B. Sikdar, "An efficient clocking scheme for quantum-dot cellular automata." *International Journal of Electronics Letters*, vol. 8, no. 1, pp. 83–96, 2020.

[23] K. Walus, T. J. Dysart, G. A. Jullien, and R. A. Budiman, "Qcadesigner: a rapid design and simulation tool for quantum-dot cellular automata," *Nanotechnology, IEEE Transactions on*, vol. 3, no. 1, pp. 26 – 31, 2004.

[24] P. Tougaw and C. Lent, "Logical devices implemented using quantum cellular automata." *Journal of Applied Physics*, vol. 75, no. 3, pp. 1818–1825, 1994.

[25] I. Amlani, R. Orlov, A.and Kummamuru, C. Bernstein, G.and Lent, and G. Snider, "Experimental demonstration of a leadless quantum-dot cellular automata cell," *Applied Physics Letters*, vol. 77, no. 5, pp. 738–740, 2000.

[26] A. Orlov, I. Amlani, G. Bernstein, C. Lent, and G. Snider, "Realization of a functional cell for quantum-dot cellular automata," *Science*, vol. 277, no. 5328, pp. 928–930, 1997.

[27] C. Lent, P. Tougaw, and W. Porod, "Quantum cellular automata: The physics of computing with arrays of quantum-dot molecules," *PhysComp '94: Proceedings of the Workshop on Physics and Computing*, 1994.

[28] T. Lantz, *A QCA implementation of a look-up table for an FPGA*. Department of Electrical Engineering - College o Engineering Rochester Institute of Technology, 2006.

[29] K. Walus, A. Vetteth, G. Jullien, and V. Dimitrov, "Ram design using quantum-dot cellular automata," *Nanotech*, vol. 2, 2003.

[30] M. Dehkordi, A. Shamsabadi, B. Ghahfarokhi, and A. Vafaei, "Novel ram cell designs based on inherent capabilities of quantum-dot cellular automata," *Microelectronics Journal*, vol. 42, pp. 701 – 708, 2011.

[31] D. Agrawal and B. Ghosh, "Quantum dot cellular automata memories," *International Journal of Computer Applications*, vol. 46, no. 5, pp. 75 – 87, May 2012.

[32] S. Angizi, S. Sarmadi, S. Sayedsalehi, and K. Navi, "Design and evaluation of new majority gate-based ram cell in quantum-dot cellular automata," *Microelectronics Journal*, vol. 46, pp. 43 – 51, 2015.

[33] A. Sadoghifar and S. Rasouli Heikalabad, "A content-addressable memory structure using quantum cells in nanotechnology with energy dissipation analysis," *Physica B: Condensed Matter*, vol. 537, pp. 202 – 206, 2018.

[34] A. Majeed, E. AlKaldy, and S. Albermany, "An energy-efficient ram cell based on novel majority gate in qca technology," *SN Applied Sciences*, vol. 1, no. 11, p. 1354, 2019.

Orestis Liolis received the Dipl. Eng., M.Sc., and Ph.D. degrees in Electrical and Computer engineering from the Democritus University of Thrace (DUTh), Xanthi, Greece, in 2013, 2015, and 2021, respectively. In 2017, General Secretariat for Research and Technology (GSRT) in Greece and Hellenic Foundation for Research and Innovation (HFRI) have granted him a scholarship in order to complete his Ph.D. thesis. His current research interests include microelectronic and nanoelectronic circuits and applications, quantum circuits and systems, and cellular automata theory and applications.

Vasilios Mardiris received the Dipl. Eng., M.Sc., and Ph.D. degrees in Electrical and Computer Engineering from the Democritus University of Thrace (DUTh), Greece, in 1995, 2005, and 2012, respectively. He joined Technological Educational Institute of Kavala in 2006 as a faculty member where he is currently Professor in the Management Science and Technology Department, International Hellenic University, Greece. In 2000, he received an award from Intel and IBM for the Low Power Design project LPGD ESPRIT #IV 25256. His current research emphasis is on modeling and simulation of nanoelectronics, quantum cellular automata, parallel architectures and computer arithmetic, bioinformatics and computer networks.

Ioannis G. Karafyllidis received the Dipl. Eng. and Ph.D. degrees in electrical engineering from the Aristotle University of Thessaloniki, Greece. In 1992 he joined the Department of Electrical and Computer Engineering, Democritus University of Thrace , Greece, as a faculty member, where he is currently a Professor. His current research emphasis is on quantum computing, modeling and simulation of nanoelectronic devices and circuits, and biological networks modeling. He is a Fellow of the Institute of Nanotechnology, a Founding Member of the American Academy of Nanomedicine and a member of the Technical Chamber of Greece (TEE).

Sorin Cotofana Fellow, IEEE) received the M.Sc. degree in computer science from the Politechnica University of Bucharest, Bucharest, Romania, in 1984, and the Ph.D. degree in electrical engineering from Delft University of Technology, Delft, The Netherlands, in 1998.,He is currently with the Electrical Engineering, Mathematics, and Computer Science Faculty, Delft University of Technology. He has authored or coauthored more than 250 articles in peer-reviewed international journal and conferences. His current research interests include the design and implementation of dependable/reliable systems out of unpredictable/unreliable components, aging assessment/prediction and lifetime reliability aware resource management, and unconventional computation paradigms and computation with emerging nano-devices.,Dr. Cotofana is a Fellow of IEEE Circuits and System Society (CASS) and IEEE Computer Society and an HiPEAC Member. He was a CASS Distinguished Lecturer from 2019 to 2021 and a CASS BoG Member from 2020 to 2022. He received 12 international conferences best paper awards, e.g., 2012 IEEE Conference on Nanotechnology and 2012 ACM/IEEE International Symposium on Nanoscale Architectures. He was the Chair of the Giga-Nano IEEE CASS Technical Committee from 2013 to 2015 and the IEEE Nano Council CASS Representative from 2013 to 2014. He has been actively involved as a Reviewer, Technical Program Committee (TPC) Member, TPC (track), and general (co)-chair in the organization of numerous international conferences. He served as an Associate Editor for IEEE Transactions on Circuits and Systems I: Regular Papers from 2009 to 2011, IEEE Transactions on Nanotechnology from 2008 to 2014, a Senior Editorial Board Member for the IEEE Journal on Emerging and Selected Topics in Circuits and Systems from 2016 to 2017. He is currently an Associate Editor of IEEE Transactions on Computers and the Editor-in-Chief of IEEE Transactions on Nanotechnology.

Georgios Ch. Sirakoulis (M'95) Georgios Ch. Sirakoulis (Member, IEEE) received the Dipl. Eng. and Ph.D. degrees in electrical and computer engineering from the Democritus University of Thrace (DUTh), Greece, in 1996 and 2001, respectively. He is currently an Professor with the Department of Electrical and Computer Engineering, DUTh. He has authored or coauthored more than 320 technical papers, he is a co-editor of seven books, co-author of 30 book chapters, and the guest editor of 15 Special Issues. His current research emphasis is on complex electronic systems, future and emergent electronic devices, circuits, models and architectures, unconventional computing, memristors, cellular automata, quantum cellular automata, bioinspired computation or biocomputation and bioengineering, and modeling and simulation. He is Chair Elected of the Giga-Nano IEEE CASS Technical Committee from 2020 to 2022, the IEEE Nano Council CASS Representative from 2022 to 2023, Vice Chair of IEEE Task Force on Unconventional Computing from 2017, and Vice-President elected for Publications from 2023 to 2024. He has been actively involved as a Reviewer, TPC Member, TPC (track), and general (co)-chair in the organization of many international conferences. He is an Associate Editor for the IEEE Transactions on Nanotechnology, IEEE Transactions on Computers, IEEE Nanotechnology Magazine, Microelectronics Journal, Integration, the VLSI Journal, Journal of Cellular Automata, International Journal of Unconventional Computing, Parallel Processing Letters, Electronics, and International Journal of Parallel, Emergent and Distributed Systems.