

Exploration of the Synchronization Constraint in Quantum-dot Cellular Automata

Frank Sill Torres^{1,2} Pedro A. Silva³ Geraldo Fontes³ José Augusto M. Nacif³ Ricardo Santos Ferreira³
Omar Paranaíba Vilela Neto⁴ Jeferson F. Chaves^{4,5} Rolf Drechsler^{1,2}

¹Cyber-Physical Systems, DFKI GmbH, Bremen, Germany

²Group of Computer Architecture, University of Bremen, Germany

³Universidade Federal de Viçosa, Brazil

⁴Universidade Federal de Minas Gerais, Brazil

⁵Centro Federal de Educação Tecnológica de Minas Gerais, Brazil
frasillt@uni-bremen.de, {jnacif,ricardo}@ufv.br, omar@dcc.ufmg.br

Abstract—Quantum-dot Cellular Automata (QCA) is a field-coupled nanotechnology which might enable design with high performance and extraordinary low energy dissipation. Information processing and flow in QCA is controlled by external clocks, which requires a proper synchronization already during circuit design phase. In this paper, we discuss the fundamental differences between local and global synchronicity in QCA circuits. Further, we show that it is possible to relax the global synchronicity constraint and discuss the consequent impact on the design performance. Simulation results indicate that the design size can be reduced by about 70% while the throughput performance declines by similar values.

Index Terms—Quantum-dot Cellular Automata, Synchronicity, Place and Route

I. INTRODUCTION

The Quantum-dot Cellular Automata (QCA) nanotechnology offers a promising alternative to conventional circuit technologies. In QCA, computations and data transfer are carried out via local fields between nanoscale devices, the so called QCA cells, that are arranged in patterned arrays [1]. Further, information is represented in terms of the polarization of the cells. Theoretical and experimental results indicate that QCA-based approaches have the potential to allow for systems with highest processing performance and remarkably low energy dissipation [2]. Consequently, numerous contributions on their physical realization have been made in the past, e. g. molecular Quantum Cellular Automata [3], atomic Quantum Cellular Automata [4] or nanomagnetic logic [5].

QCA apply external clocks in order to prevent metastability and to control the data flow amongst logic elements [6]. These clocks modify the state of QCA cell such that a cell is in reset or can change or not its polarization, and thus, its logic value. Commonly, four clocks, numbered from 1 to 4 and phase-shifted by 90 degree, are applied. For fabrication purposes, cells are usually grouped in a grid of square-shaped clock zones such that all cells within a clock zone are controlled by the same external clock [7]. It is important to note that correct

data flow is only possible between cells controlled by consecutively numbered clocks. That means, cells controlled by clock 1 can solely pass their data to cells controlled by clock 2 etc. and, finally, from clock 4 to clock 1. Consequently, data are passed between cells in a pipeline-like fashion controlled by the external clocks (more details will be given in section II). This behavior led to the common assumption that QCA circuits must not employ only a local but also global pipeline-like behavior, e. g. in [8], [9]. That means, it is assumed that all signal paths arriving at the same logic gates must have equal length and that all signals must always arrive at the respective logic gates in a synchronized manner. This requirement put some limitations on the design automation of QCA circuits and demands some design overhead, as will be discussed in section III.

The intention of this work is to highlight the possibility to design QCA circuits that do not possess a global pipeline-like behavior. Therefore, we introduce the basics of Quantum-dot Cellular Automata (section II), before we discuss the aspect of synchronicity in QCA designs (section III). Based on the conclusions of this discussion we propose modification to an existing QCA placement and routing algorithm (section IV), which is followed by a simulative comparison of QCA designs that are fully synchronized or not (section V). Finally, we draw some conclusions (section VI).

II. QUANTUM-DOT CELLULAR AUTOMATA

This section introduces the nanotechnology Quantum-dot Cellular Automata (QCA) and discusses basic aspects of QCA circuit design.

A. QCA states and logic gates

Quantum-dot Cellular Automata (QCA) are a field-coupled nanotechnology that executes computations fundamentally different from current technologies. In QCA, information is

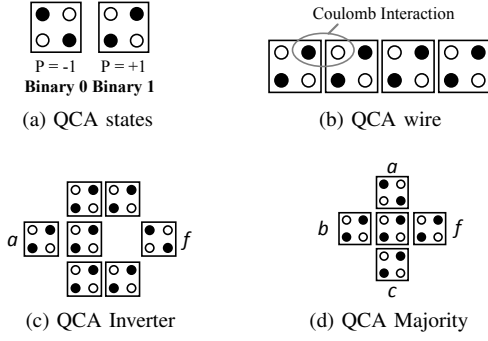


Fig. 1: QCA states and basic cells

stored in terms of the polarization of nanosized cells and can be propagated to adjacent cells using Coulomb forces.

The basic element of QCA is a *cell* that is usually composed of four quantum dots which are able to confine an electric charge [10], [11]. These quantum dots are arranged at the corners of a square, such as depicted in Fig. 1a. Further, each cell contains two free and mobile electrons, which are able to tunnel between adjacent dots, while tunneling to the outside of the cell is prevented by a potential barrier. The electrons within a cell experience mutual repulsion due to Coulomb interaction and, thus, tend to locate at opposite corners of the square. Consequently, an isolated cell may be assume one of the two *cell polarizations* $P = -1$ and $P = +1$ as depicted in Fig. 1a. This allows for an encoding of binary information by identifying $P = -1$ with a binary 0 and $P = +1$ with a binary 1.

Further, the polarizations of neighboring cells influence each other—again by Coulomb interaction. This allows for the design of wires as well as logic gates. For example, Fig. 1b shows a QCA wire where a signal is propagated through several cells from left to right by Coulomb interaction. Further, Fig. 1c depicts an Inverter gate, where, again from left to right, an input signal is copied to two paths, which are then combined diagonally, such that the input value is inverted. Finally, Fig. 1d shows a *majority gate*, where the output is identically to the majority of the input signals. Further classical logic operations such as AND and OR gates can be easily derived from the majority gate by locking one of its inputs to a binary 0 (leading to an AND) or binary 1 (leading to an OR).

B. QCA clocking

In order to execute these and more complex logic operations, a dedicated *clocking* is required which, starting with the initialization of the QCA cells, properly propagates information among the cells and avoids metastable states [12]. To this end, external clocks are employed which regulate the intercellular tunneling barriers within a QCA cell such that the cell can be polarized (i. e., tunneling is prevented) or not (i. e., electrons may tunnel between adjacent quantum dots within the cell). Typically, a clock consists of four phases:

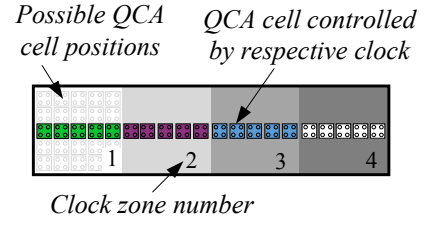


Fig. 2: QCA wire with cells in four clock zones

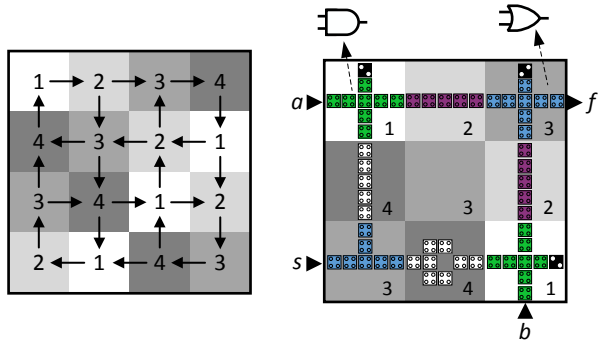
- In the so-called *relax* phase, the cell is depolarized and does not contain any information.
- During the following *switch* phase, the interdot barriers are raised which forces the cell to polarize into one of the two antipodal states (according to the polarization of surrounding cells).
- In the following *hold* phase, the cell keeps its polarization and may act as input for adjacent cells.
- During the final *release* phase, the interdot barriers are lowered again thereby removing the previous polarization of the cell.

Normally, four clocks shifted by 90 degrees are provided in order to enable the propagation of information among cells [6]. Using these clocks, the data flow can be controlled by applying appropriately shifted clock signals such that the cells which shall pass their data are in the *hold* phase at the same time when the cells that shall receive the data are in the *switch* phase. For fabrication purposes, cells are usually grouped in a grid of square-shaped clock zones such that all cells within a clock zone are controlled by the same external clock [7].

Fig. 2 depicts an exemplary QCA wire that has the extension of four clock zones. Moreover, possible locations for further cells are indicated in gray in the most left clock zone. All QCA cells within the same clock zone are controlled by the same clock signal. Note that consecutive clock signals are shifted by one phase. That means, if clock zone 1 is in the *hold* phase then clock zone 2 will be in the *switch* phase, clock zone 3 will be in the *relax* phase and clock zone 4 will be the *release* phase. In this state, cells in clock zone 2 polarize according to the polarization of the adjacent cells in clock zone 1 while cells in clock zone 3 and 4 are without polarization. During the next clock phase, clock zone 2 changes to *hold*, while clock zone 3 is in the *switch* phase. Consequently, data is passed from zone 2 to 3 (and so on).

C. QCA Circuit Design

In order to design a QCA circuit, traditional design solutions for logic synthesis of conventional circuits can be employed for generation of initial netlists. Therefore, already available realizations of typical gates such as Inverter, OR, AND, XOR, etc. can be applied [13]. During the following placement and routing (P&R), these gates must be arranged such that the corresponding clocking is respected, i. e. data are properly passed from one gate to another. To this end, usually a fix



(a) USE grid with clocking zones (b) Multiplexer mapped to USE grid zones

Fig. 3: USE clocking scheme

arrangement of clock zones is imposed on a QCA layout [14], [15].

There have been several proposals for clocking schemes, like 2DDWave [14], tile-based design [7] or USE (*Universal, Scalable and Efficient*) [15]. Without loss of generality, we apply the latter in this work, which is characterized by a high regular architecture and the ability of creating feedback paths, which turns USE very suitable for automation of QCA design.

USE defines a grid of clock zones, which are arranged such that all inner clock zones have two adjacent neighboring clock zones that can provide data and two neighbors that can receive data. The clock zones are numbered from 1 to 4, whereby consecutive numbered zones have clock signals shifted by 90 degree. Fig. 3a depicts the concept of USE (each square is a clock zone that contains 5×5 QCA cells, following the proposal from [15]). Further, the arrows indicate the possible data flow between adjacent clock zones.

Fig. 3b depicts the 2:1 MUX function $f = a\bar{s} + bs$ placed on a USE grid. Using conventional synthesis tools, the gate netlist have been extracted which then has be mapped to the USE grid such that the output of one QCA structure, i. e. gate or wire, is always propagated to the input of a QCA structure containing the next operation/wire. In total, the resulting QCA circuit possesses design costs of 3×3 clock zones which is equal to 15×15 QCA cells and a critical path of 5 clock zones ($s \rightarrow f$).

III. SYNCHRONICITY OF QCA CIRCUITS

In this section, we discuss the difference between global and local synchronicity in QCA designs. Further, we show that—contrary to the state of the art—global synchronicity is not a mandatory constraint in QCA designs.

A. Global and Local Synchronicity

When considering synchronicity in QCA circuits, one has to distinguish between local and global synchronicity. The former means the data flow constraint, discussed in section II, which requires that data can only be transferred between cells located in consecutive numbered clock zones. On the other

hand, global synchronicity refers to the global pipeline-like behavior of QCA circuits.

The example depicted in Fig. 4 and Fig. 5 shall highlight the differences. Note that, for the sake of simplicity but without loss of generality, this example is not applying a USE grid. The circuit shown in Fig. 4 consists of two inputs $In1$ and $In2$ and three random operations $o1$, $o2$ and $o3$, with the first two having one input while the latter operation possesses two inputs. Each of the three depicted cases differs in the position of input $In1$. The curves in Fig. 5 relate to the clock signals of all four zones, the input signals, that change when clock 1 enters in switch phase (falling clock slope), and the data at points A and B, which both contain inputs of operation $o3$.

In all three cases, local synchronicity is guaranteed. That means, all data flow is only between cells in consecutive numbered clock zones. Further, in case 1 global synchronicity is given, i. e. operation $o3$ receives related input signals. This is also true for case 2, even though both $In1$ are connected with different clock zones. However, the distance between both is less than 4 clock zones (see also the indicated red line). Consequently, in the following clock zone 1 all data from $In1$ and $In2$ are synchronized again, because all clock zones with number 1 change into *switch* phase at the same time. In contrast, case 3 misses the global synchronicity, because data of input $In1$ arrive one clock cycle before the related data of input $In2$.

B. Unsynchronized QCA Circuits

A fundamental characteristic of globally synchronized designs is that new data can be applied to the primary inputs of the circuit in each clock cycle. After the first input data passed the circuit, correspondingly new results arrive at the circuit's primary outputs in each clock cycle—resulting in a circuit throughput of 1. Furthermore, a globally synchronized circuit does not require synchronization elements like latches, as, by definition, all related data are always synchronized.

However, in contrast to many related statements in the literature, e. g. in [8], [9], global synchronicity (GS) is not a mandatory constraint in QCA circuits [16]. For example, the circuit depicted in Fig. 4c misses GS, because data from both inputs are not arriving at the same time at operation $o3$. A common solution to this problem would be the relocation of $In1$ or $In2$ such that paths have equal lengths, as e. g. in Fig. 4a. However, this usually comes at high costs in terms of area [17]. Instead, we propose to reduce the frequency with which new input data are applied. That means for the given example, data connected at $In1$ and $In2$ must be kept stable for two clock cycles—leading to a reduced throughput of $1/2$. On the other hand, this approach allows for the reduction of area costs, latency and design complexity. Consequently, there is a trade-off between performance and area costs.

An important parameter of globally unsynchronized QCA circuits is the frequency with which new input data can be sent to the circuit. This frequency depends on the maximum difference between the arrival times of all inputs of a gate of the QCA circuit. That means, it must be assured for all gates

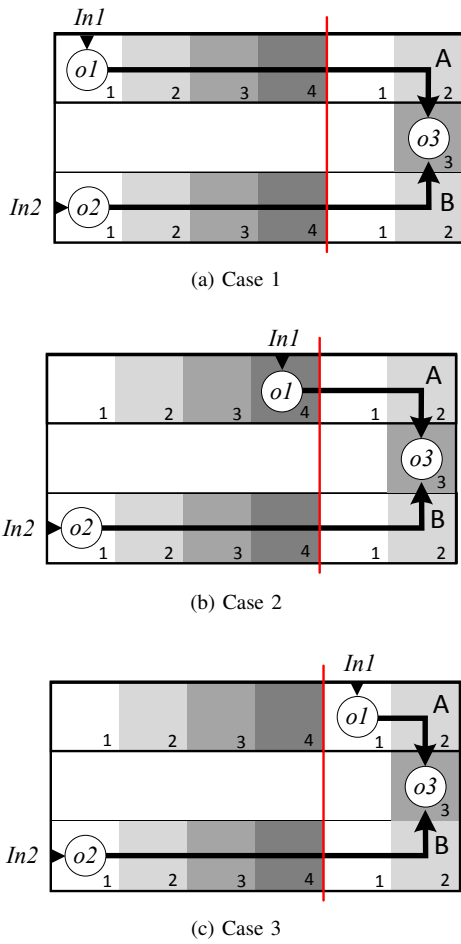


Fig. 4: QCA circuit possessing and missing the global synchronicity. The red line indicates the limit until where $In1$ could be placed such that paths $In1 \rightarrow o3$ and $In2 \rightarrow o3$ are synchronous. For the sake of simplicity, this example is not using USE.

that its inputs are synchronous at least for one clock cycle before new inputs arrive.

Following example shall detail the related analysis. Fig. 6a depicts an exemplary QCA circuit which does not possess global synchronicity. Several of the operations oX have two inputs which have diverging arrival times. In detail, the inputs of $o6$ arrive after 1 and 9 clock phases, the inputs of $o8$ after 10 and 14 clock phases, and the inputs of $o9$ after 12 and 16 clock phases. As each clock cycle last 4 clock phases, the maximum difference in terms of clock cycles results from the ceiling division by 4. The means, in case of $o6$ the difference results to $\lceil 9/4 \rceil - \lceil 1/4 \rceil = 2$. Similar, the maximum difference in terms of clock cycles for $o8$ and $o9$ results to 1. Hence, both inputs $In1$ and $In2$ must not change for two additional clock cycles in order to assure correct operation. Fig. 6b depicts the curves of the clock in clock zones 1, the inputs and the signals at points A and B, both highlighted in Fig. 6a. One can note that only at the third clock cycle, operation $o6$ has

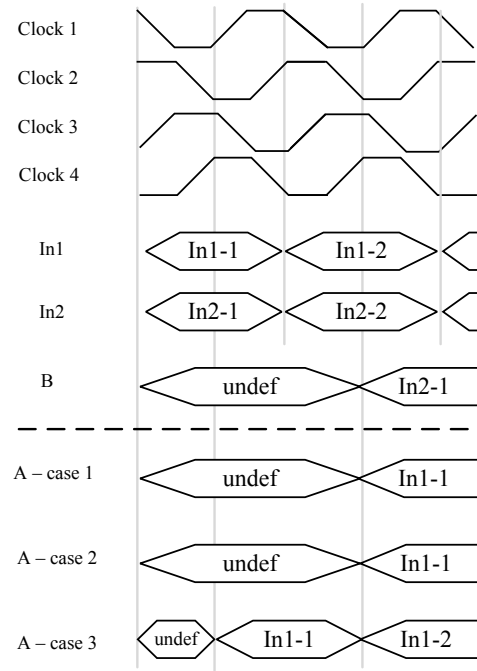


Fig. 5: Curves of clock zone signals, inputs and data at points A and B for all three cases shown in Fig. 4

synchronized inputs, i. e. its both inputs have data (In1-1 and In2-1) that have been sent at the same time.

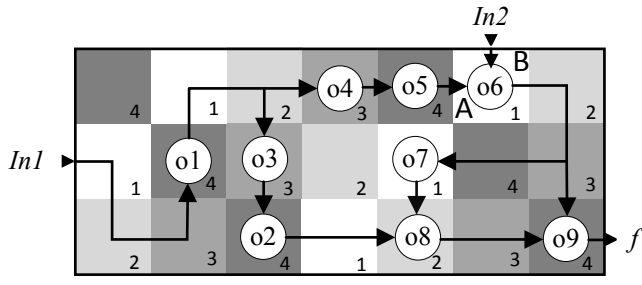
The presented circuit has a latency of 16 clock phases, i. e. 4 clock cycles. If the input frequency is reduced to $1/3$ of the clock frequency then the first correct results will arrive after 6 clock cycles. Next, every three clock cycle new correct outputs will be available.

IV. MODIFIED PLACE AND ROUTE ALGORITHM

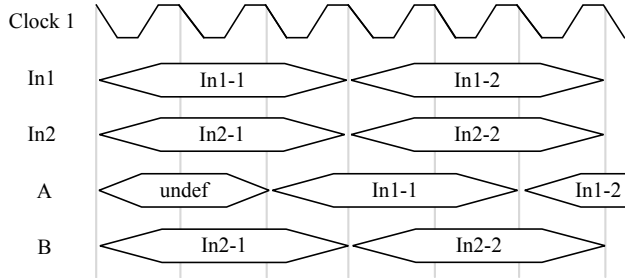
In Electronic Design Automation, the placement and routing (P&R) generates a final layout starting from a gate-level netlist. Here, placement means the location of gates on the grid, while routing refers to the connection of these gates via wires.

In QCA, the P&R is NP-complete leading to high computation costs even for small circuits [18]–[21]. Hence, we propose in [21] a P&R algorithm based on a divide-and-conquer strategy that notably reduces the complexity. The presented P&R algorithm applies the USE, but can be easily adapted for similar clocking schemes. The approach starts with a decomposition of the gate-level graph that is guided by the reconvergent paths. Next, for each partition the corresponding QCA layout is generated. In the final step, the entire circuit is rebuilt by aligning the nodes that overlap partitions, followed by a routing of all inter-subgraphs wires. In case of the latter, the algorithm assures that all interconnections between two graph depth levels are locally and globally synchronized.

The example in Fig. 7 demonstrates the principal steps of the P&R algorithm presented in [21]. First, the circuit is represented as a graph where each node is a gate (see Fig. 7a).



(a) Layout of globally unsynchronized QCA circuit



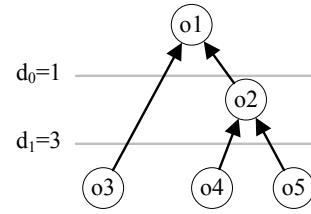
(b) Curves of clock 1, inputs and signals at points A and B

Fig. 6: Unsynchronized QCA circuit

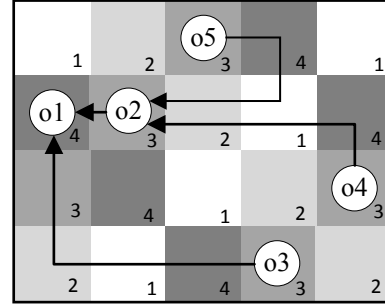
Next, a distance is defined between each level. This is followed by placement of the nodes level by level starting from the primary outputs (see Fig. 7b). In the depicted example, the inter-level distance d_0 between node $o1$, which connects to the output, and $o2$ is 1. Fig. 7c illustrates the corresponding differences of each clock zone to the clock zone containing node $o1$. As one can see, there are only two possible positions for node $o2$ if the distance to $o1$ shall be 1. In this example, the algorithm chose the right neighboring clock zone of $o1$. Further, the distance d_1 between the nodes in level 1 and level 2 has been defined with 3. Note that for the distance between nodes $o1$ and $o3$ this sums up to 4. Hence, the algorithm tries to place the nodes $o4$ and $o5$ such that both have a distance of 3 to node $o2$, while $o3$ is placed such that it has a distance of 4 to node $o1$. In order to improve the results, the proposed P&R algorithm varies the inter-level distances d_0, d_1, \dots, d_{n-1} with $1 \leq d_i < \max(4, n)$, where n means the graph depth. However, it is assured that the distance between nodes in same level to nodes in higher level is always the same, i. e. global synchronicity is given for all tries.

In order to relax the GS constraint, we modified the algorithm such that each edge of the graph possesses its own distance. This enables that nodes in the same level can have a different distance to nodes in higher level. Nevertheless, distances must be chosen such that local synchronicity is assured.

The example depicted in Fig. 8 shall highlight these modifications. Fig. 8a and Fig. 8b show the graph and the respective layout if global synchronicity (GS) is assured. The numbers at the edges in Fig. 8a indicate the distance between the nodes.



(a) Gate-level sub-graph with synchronized inter-level distances



(b) Corresponding layout

$d=3$ 1	$d=2$ 2	$d=5$ 3	$d=4$ 4	1
$o1$ 4	$d=1$ 3	$d=2$ 2	$d=3$ 1	$d=4$ 4
$d=1$ 3	$d=4$ 4	$d=3$ 1	$d=6$ 2	$d=5$ 3
$d=2$ 2	$d=3$ 1	$d=4$ 4	$d=5$ 3	$d=6$ 2

(c) Distances to node node $o1$

Fig. 7: Example for unmodified P&R algorithm assuring global synchronicity

As one can see, these numbers follow the definitions of the inter-level distances, also shown in the figure. Consequently, the distances between nodes $o4, o5$ and $o6$ to node $o1$ is always 8. In contrast, Fig. 8c illustrates the same graph, but with new distances. As one can see, the distance between nodes in same level to a node in a different level is not always the same. For example, the distance between $o4$ and $o1$ reduces to 7, and it changes to 6 between $o5, o6$ and $o1$. Consequently, the maximum delay, i. e. its latency, of the circuit has been reduced from 8 to 7. Furthermore, the layout can be more compact, as Fig. 8d indicates. Here, the grid layout could be reduced from 18 clock zones to 15, while the number of occupied clock zones reduced from 16 to 13.

V. ANALYSIS RESULTS

This section compares the design costs for selected benchmark circuits [22]–[24] with and without global synchronicity. We have also included some circuits generated using the ABC

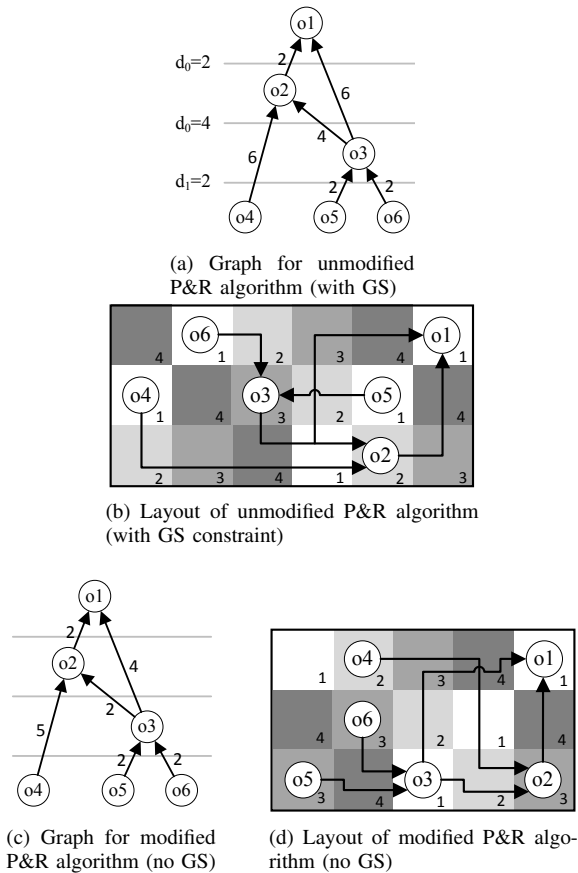


Fig. 8: Example comparing of graphs and resulting layouts for P&R assuring and ignored global synchronicity (GS) constraint. The numbers on the edges indicate the distances between nodes.

synthesis tool [25]. Tab. I lists our results. The columns *Grid area* indicate the complete area of the generated QCA grid, including empty clock zones, while the columns *Occupied clk-zones* refer solely to the clock zones that contain QCA cells. The columns *Latency* report the length of the longest path in terms of clock zones and the column *Throughput* lists the throughput of the designs without global synchronicity in comparison to its fully synchronized counterpart. One can note that disregarding global synchronicity can lead to reduction of occupied clock zones and latency, but not in all cases (e. g. *FA-MAJ* and *BI_r2*). This reduction, though, comes at the costs of a declining throughput. Further, in most cases the reduction of occupied clock zones and area is comparable, with exception of the benchmark *t*. Here, the grid area increases while the number of occupied clock zones is reduced. Thus, the qualification of this result depends on the possibility to use the unoccupied area for further circuits, which is e. g. the case of the presented P&R algorithm.

Fig. 9 compares the reduction of occupied clock zones, latency and throughput if global synchronicity is ignored. Results indicate that disregarding global synchronicity can reduce the occupied area by up to 67% (*clpl*) and in average

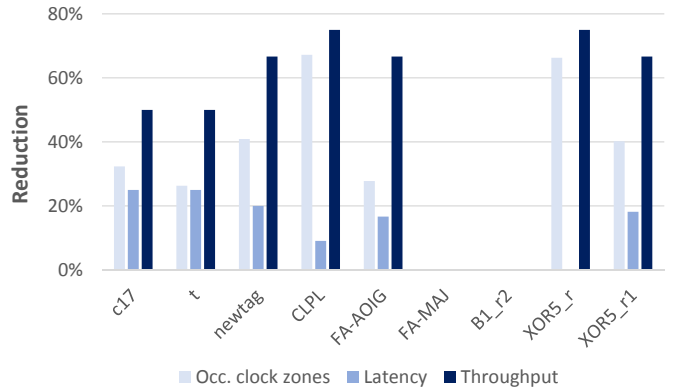


Fig. 9: Reduction of occupied clock zones, latency and throughput if global synchronicity is ignored

by 33%. In case of the latency, reductions of up to 25% (*clpl* and *t*) and in average by 13% can be reported. However, only in about 50% of all benchmarks the latency could be improved. In contrast, the throughput declined by up to 75% (*clpl*), i.e. factor 4, and in average by 50%. In most cases, the value for throughput reduction is comparable to area reduction. However, in two cases, namely *t* and *FA-AOIG*, there is a strong discrepancy between area and throughput reduction. Hence, it is up to the designer to decide which of the parameters he wants to prioritize.

VI. CONCLUSION

Quantum-dot Cellular Automata (QCA) is a promising nanotechnology with remarkable characteristics in terms of performance and energy consumption. QCA apply external clocks for control of information transfer such that circuits can have a pipeline-like behavior. We revealed in this work that, in contrast to what is common believe, this behavior is not a mandatory constraint for QCA circuits. Therefore, we discussed the differences between local and global synchronicity (GS) in QCA circuits. Further, we showed how placement and routing algorithm can be modified in order to allow to consider or not the GS constraint. Simulation results for selected benchmarks indicate that relaxing the GS constraint can lead to area reductions of about 70%, while the throughput reduces in similar range. Further, the latency could be improved by up to 25%. Hence, designers have a further degree of freedom in order to explore the full potential of the QCA technology.

ACKNOWLEDGMENTS

This work was supported in part by the University of Bremen's graduate school SyDe, funded by the German Excellence Initiative, and the Brazilian agencies CNPq, FAPEMIG and CAPES.

REFERENCES

- [1] C. S. Lent and P. D. Tougaw, "A device architecture for computing with quantum dots," *Proceedings of the IEEE*, vol. 85, no. 4, pp. 541–557, Apr 1997.

TABLE I: Simulation Results

Benchmark				Synchronized			Not synchronized			
	Gates	Inputs	Outputs	Grid area	Occupied clk-zones	Latency	Grid area	Occupied clk-zones	Latency	Throughput
c17	12	7	5	56	34	8	35	23	6	0.50
t	15	10	5	49	38	8	56	28	6	0.50
newtag	20	12	8	80	44	10	48	26	8	0.33
CLPL	21	10	11	132	64	11	33	21	10	0.25
FA-AOIG	12	9	3	56	36	12	36	26	10	0.33
FA-MAJ	8	5	3	35	29	10	35	29	10	1.00
B1_r2	16	13	3	60	44	11	60	44	11	1.00
XOR5_r	37	32	5	252	252	24	140	85	24	0.25
XOR5_r1	31	26	5	195	97	22	160	58	18	0.33

^a Number of clock zones, ^b Number of clock zones, ^c Compared to synchronized design.

- [2] J. Timler and C. Lent, "Power gain and dissipation in quantum-dot cellular automata," *Journal of Applied Physics*, vol. 91, no. 2, pp. 823–831, 2002.
- [3] V. Arima, M. Iurlo, L. Zoli, S. Kumar, M. Piacenza, F. Della Sala, F. Marino, G. Maruccio, R. Rinaldi, F. Paolucci, M. Marcaccio, P. G. Cozzi, and A. P. Bramanti, "Toward quantum-dot cellular automata units: thiolated-carbazole linked bisferrocenes," *Nanoscale*, vol. 4, pp. 813–823, 2012.
- [4] T. R. Huff, H. Labidi, M. Rashidi, M. Koleini, R. Achal, M. H. Salomons, and R. A. 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.
- [5] I. Eichwald, A. Bartel, J. Kiermaier, S. Breitreutz, G. Csaba, D. Schmitt-Landsiedel, and M. Becherer, "Nanomagnetic logic: Error-free, directed signal transmission by an inverter chain," *IEEE Transactions on Magnetics*, vol. 48, no. 11, pp. 4332–4335, Nov 2012.
- [6] K. Hennessy and C. Lent, "Clocking of molecular quantum-dot cellular automata," *Journal of Vacuum Science & Technology B*, vol. 19, no. 5, pp. 1752–1755, 2001.
- [7] J. Huang, M. Momenzadeh, L. Schiano, M. Ottavi, and F. Lombardi, "Tile-based qca design using majority-like logic primitives," *J. Emerg. Technol. Comput. Syst.*, vol. 1, no. 3, pp. 163–185, Oct. 2005.
- [8] J. Huang, M. Momenzadeh, and F. Lombardi, "Design of sequential circuits by quantum-dot cellular automata," *Microelectronics Journal*, vol. 38, no. 4, pp. 525 – 537, 2007, special Issue of the 6th International Symposium on Quality Electronic Design (ISQED) March 21-23 San Jose, CA.
- [9] L. A. Lim, A. Ghazali, S. C. T. Yan, and C. C. Fat, "Sequential circuit design using quantum-dot cellular automata (qca)," in *2012 IEEE International Conference on Circuits and Systems (ICCAS)*, Oct 2012, pp. 162–167.
- [10] W. Liu, E. Swartzlander Jr, and M. O'Neill, *Design of semiconductor QCA systems*. Artech House, 2013.
- [11] C. Lent, P. Tougaw, W. Porod, and G. Bernstein, "Quantum cellular automata," *Nanotechnology*, vol. 4, no. 1, p. 49, 1993.
- [12] E. Blair and C. Lent, "An architecture for molecular computing using quantum-dot cellular automata," in *IEEE-NANO 2003*, vol. 1. IEEE, 2003, pp. 402–405.
- [13] D. A. Reis, C. A. T. Campos, T. R. B. S. Soares, O. P. V. Neto, and F. S. Torres, "A methodology for standard cell design for qca," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2016, pp. 2114–2117.
- [14] V. Vankamamidi, M. Ottavi, and F. Lombardi, "Two-dimensional schemes for clocking/timing of qca circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 1, pp. 34–44, Jan 2008.
- [15] C. Campos, A. Marciano, O. Neto, and F. S. Torres, "USE: A universal, scalable, and efficient clocking scheme for QCA," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 35, no. 3, pp. 513–517, 2016.
- [16] F. S. Torres, M. Walter, R. Wille, D. Große, and R. Drechsler, "Synchronization of Clocked Field-Coupled Circuits," in *IEEE Nano*, 2018, pp. 1–4.
- [17] F. S. Torres, R. Wille, P. Niemann, and R. Drechsler, "An energy-aware model for the logic synthesis of quantum-dot cellular automata," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. PP, no. 99, 2018.
- [18] T. Teodosio and L. Sousa, "Qca-1g: A tool for the automatic layout generation of qca combinational circuits," in *Norchip 2007*, Nov 2007, pp. 1–5.
- [19] A. Trindade, R. Ferreira, J. A. M. Nacif, D. Sales, and O. P. V. Neto, "A placement and routing algorithm for quantum-dot cellular automata," in *Proceedings of the 29th Symposium on Integrated Circuits and Systems Design: Chip on the Mountains*, ser. SBCCI '16. Piscataway, NJ, USA: IEEE Press, 2017, pp. 12:1–12:6. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3145862.3145874>
- [20] M. Walter, R. Wille, D. Große, F. S. Torres, and R. Drechsler, "An exact method for design exploration of quantum-dot cellular automata," in *Design, Automation and Test in Europe*, 2018.
- [21] G. Fontes, P. A. R. L. Silva, J. A. M. Nacif, O. P. V. Neto, and R. Ferreira, "Placement and routing by overlapping and merging qca gates," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2018.
- [22] S. Yang, "Logic synthesis and optimization benchmarks," Tech. Rep., Dec. 1989.
- [23] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," in *Circuits and Systems, 1989.*, *IEEE International Symposium on*, May 1989, pp. 1929–1934 vol.3.
- [24] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran," in *Proceedings of IEEE Int'l Symposium Circuits and Systems (ISCAS 85)*. IEEE Press, Piscataway, N.J., 1985, pp. 677–692.
- [25] R. Brayton and A. Mishchenko, "ABC: An academic industrial-strength verification tool," in *CAV*, 2010, pp. 24–40.