Wire Crossing Constrained QCA Circuit Design using Bilayer Logic Decomposition

A. Roohi[1], H. Thapliyal[2], and R. F. DeMara[1]
Department of Electrical and Computer Science, University of Central Florida, Orlando, FL 32816, USA
Department of Electrical & Computer Engineering, University of Kentucky, Lexington, KY 40506, USA

Quantum-dot Cellular Automata (QCA) seek potential benefits over CMOS devices such as low power consumption, small dimensions, and high speed operation. Two prominent QCA concerns of wire crossing complexity and circuit robustness are addressed by developing a three-step *Bilayer Logic Decomposition (BLD)* methodology to design QCA-based logic circuits. The partitioning of QCA computing operations into logic layers realizes considerable improvements in complexity, area, and modularity metrics. Moreover, since larger circuits are divided into two increasingly disjoint sub-planes, verification of the functionality of the design becomes compartmentalized. Design capability of the proposed approach is illustrated and analysed by implementing an area-efficient Full Comparator (FC) based on a novel logic realization. The resulting 1-bit FC achieves 32% improvement in complexity metrics in comparison with the previous optimal QCA-based FC. The related waveforms used in verification of the BLD-generated FC which are obtained by the QCADesigner simulation tool are discussed as a motivating example of the BLD methodology.

Introduction: Power consumption, switching speed, and physical scaling challenges of CMOS technology have motivated research on nanoscale Quantum dot Cellular Automata (QCA) as a transistor-less computing method based around logic networks of QCA cells. QCA cells store binary logic states based on the position of individual electrons determined by Coulomb interaction [1]. The fundamental logic elements realized by these cells are the inverter gates (Fig.1 (a)) and the majority gate, with the Original Majority Gate (OMG) and the Rotated Majority Gate (RMG) as shown in Fig.1 (b) and Fig.1 (c), respectively, along with a QCA wire. The schematic of a QCA wire depicts the consecutive clocking zones which are indicated by different color regions in Fig.1 (d).

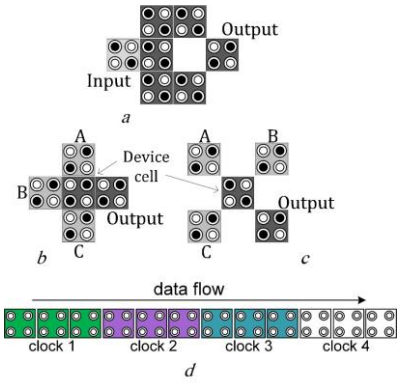


Fig. 1 *Fundamental QCA components, a)* QCA inverter, *b)* OMG, *c)* RMG, *d)* QCA wire

BLD Methodology: Robustness and area-efficiency of logic designs are significant issues in all computing architectures, and of particular impact in QCA circuitry. These are addressed by developing a three-step *Bilayer Logic Decomposition (BLD)* methodology in pursuit of the following objectives:

- **area consumption:** a primary contributor of area consumption in QCA architectures is wire crossing which can be significantly reduced by designing a decoupled multi-layer structure.
- **reliability:** facing cells between two adjacent layers have opposite polarizations [2]. Hence, the cells provide a robust configuration by increasing tolerance to misalignment and displacement defects.
- **modularity:** a bilayer design increases multifunction capability. Several novel architectures for QCA gates have been suggested based on this idea [3-4].
- **testing:** BLD designs are pre-compartmentalized for more tractable verification compared to conventional single-layer designs. Since the designer is provided with each layer's outputs as a specific Boolean logic function, it is possible to verify each layer independently and rapidly using existing design tool flows.

Accordingly, the 3-step BLD methodology is developed, as depicted in Fig. 2. The design flow proceeds as follows:

Step 1) The function *F* to be realized is expressed in simplified Sum of Products (SOP) form.

Step 2) Function *F* is mapped to a binary graph in order to partition the Boolean operations to create two approximately equal gate-count sub-functions. Each sub-function forms a bipartite graph in which each node represents either a Boolean sub-function or logic atom as an input variable.

Step 3) QCA cell realization is designed for each bipartite region within a distinct layer of a bilayer QCA structural layout.

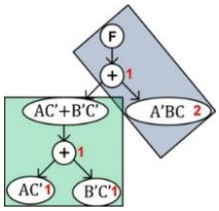
Fig. 2 begins with Step 1) whereby $F = \bar{A}BC + A\bar{C} + \bar{B}\bar{C}$ is specified. In Step 2) the procedure to create binary tree isomorphic to *F* is performed. The weight of *F* is defined by the number of variables which is used to determine exit criteria for partitioning. The *gate pool* is defined as the set containing all existing fundamental elements desired to be implemented in the design; in our technique and this example, the gate pool is comprised of 2-input and 3-input *AND*/*OR* gates. Creation of the gate pool in a binary tree structure facilitates hierarchical design and reuse. For instance, in [3], 5-input *AND* and *OR* are proposed in a dual layer structure which develops the gate pool. Hence, $W_F >$

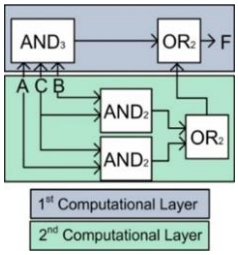
Input: Boolean expression *F*
Output: BLD for F suitable for QCA

Example: $F = \bar{A}BC + A\bar{C} + \bar{B}\bar{C}$

1. F = Create SOP according to the input function if required

2. Create binary tree isomorphic to F
Start
`create` B-graph (F)
`create` node comprise F
W_F = `weight` (F)
// weight F is defined by the number of variables in F
if $W_F > 3$
// 3 is determined by capability of gate pool
`divide` F into F_1 and F_2 while $W_{F1} \approx W_{F2}$
// F_1 denotes right side and F_2 denotes the left side of the tree
// + symbol has priority to × (.) in division operation.
`add` node comprise divider-symbol (+ (add) or . (multiply))
`create` B-graph (F_1)
`create` B-graph (F_2)
return
Stop





3. Map built binary tree to design
if $W_{F1} > W_{F2}$
`create` left (F) & root.child (F) at 1-computational layer
else
`create` right (F) & root.child (F) at 2-computational layer

Fig. 2 *BLD methodology for QCA circuits.*

3 can be changed to $W_F > 5$, which means the tree is permitted to implement a function with maximum of 5 inputs. Step 3) traverses the binary tree to utilize the components in the library to realize the circuit for each of the two layers.

Experimental Results: We use BLD to develop a new approach to realizing a 1-bit FC based on an XOR and an RMG. The logical functions of the comparator can be expressed as:

$$\begin{aligned} O_{A=B} &= AB + \bar{A}\bar{B} = (\overline{A \oplus B}) \\ O_{A<B} &= \bar{A}B = (\bar{A}B + A\bar{B}).B = (A \oplus \mathrm{B}).B \\ O_{A>B} &= A\bar{B} = (\bar{A}B + A\bar{B}).A = (A \oplus \mathrm{B}).A \end{aligned} \quad (1)$$

where A and B are the inputs and $O_{A=B}$, $O_{B>A}$ and $O_{A>B}$, as these correspond to the three possible comparative states of two numbers, are the outputs. As mentioned in the previous section, by adding extra cells in the upper layer of a design we can obtain additional reconfiguration capability. Hence, our proposed design has been implemented using the RMG. If an extra input cell is added on the upper layer above the device cell, then a modified RMG (m-RMG) is constructed. The structure of the QCA implementation of m-RMG which functions as $ABC + \bar{D}(A + B + C)$ is shown in Fig. 3(a). We can utilize this function to implement an efficient 2-input XOR by setting the input cells C and D to -1 and AB, respectively, where -1 denotes a Boolean logic zero condition.

Proof. The Boolean function for XOR is given by:

$$\begin{aligned} & ABC + \bar{D}(A + B + C), set\ C = 0, D = RMG(A, B, 0) = A.B \\ & = AB0 + \overline{AB}(A + B + 0) \\ & = (\bar{A} + \bar{B})(A + B) = \bar{A}B + A\bar{B} = (A \oplus \mathrm{B}) \end{aligned} \quad (2)$$

The cell layout of the proposed XOR is shown in Fig. 3(b). The lower computational layer has three inputs (A, B, C=0) and the desired output. The *AND* operation of A and B is generated using an RMG in the upper computational layer and then it is transmitted downward to the drive cell. All the cells except output of the *AND* have inverting influences on the polarization of the device cell. Therefore, the output of the *AND* gate has an inverting effect on the output. By leveraging equation (1) and the proposed XOR, the design of a dense 1-bit FC is readily obtained. The schematic and cell layout and simulation results of the proposed structure are shown in the Fig. 4. Validation of the functionality of the proposed designs are carried out by leveraging QCADesigner tool [10].

It is worth mentioning that if an n-bit comparator is required, the output being $O_{A=B}$ of the first stage is fed to the input cell C of the next stage. The comparison between the proposed FC and the previous ones are summarized in Table 1 where n stands for the number of bits. As shown in Table 1, our FC design has resulted in significant improvements. It is shown that our design is associated with 32% reduction in cell count in area complexity and delay when compared to the previous designs, and compares favourably to the most cost-efficient previous design [4].

Conclusion: This letter develops and illustrates the 3-step BLD methodology to design 2-computational layer QCA circuits with improved complexity, area, and modularity while also facilitating circuit verification. The capability of the proposed method was analysed by implementing an area-efficient 1-bit FC based on new equations obtained due the developed BLD approach. Future work involves investigation of semi-automated BLD script interfacing to the QCADesigner toolchain.

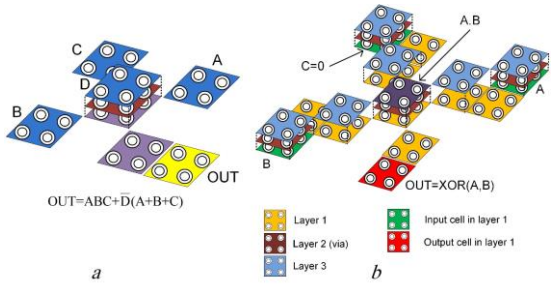


Fig. 3 *Cell layout of*
a m-RMG
b proposed XOR

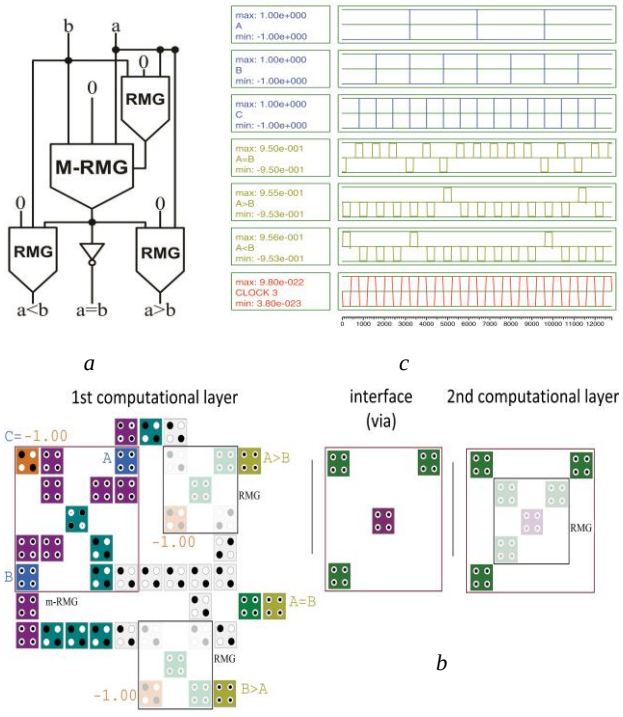


Fig. 4 *The proposed 1-bit QCA full comparator*
a schematic
b cell layout
c simulation results when C=0

Table 1: *Performance metric comparison of Full Comparator designs.*

FC Circuit Design	*n*	Area (μm^2)	Complexity	Delay Clock Phases
Design in [5]	1	0.109	100 cells	5
Design in [6]	1	0.135	97 cells	4
Design in [7]	1	0.289	221 cells	36
Design in [8]	1	n.a.	262 cells	9
Design in [4]	1	0.038	79 cells	4
Design in [9]	2	0.15	155 cells	4
BLD-FC herein	2	0.14	112 cells	8
BLD-FC herein	*1*	*0.04*	*54 cells*	*4*

References

1 Lent, C. S., and Tougaw, P. D.: 'A device architecture for computing with quantum dots', *Proceedings of the IEEE*, 1997, **85**, pp. 541-557.
2 Gin, A., Tougaw, P. D., and Williams, S.: 'An alternative geometry for quantum-dot cellular automata', *J Appl phys*, 1999, **85**, pp. 8281-8286.

3 Sayedsalehi, S., Moaiyeri, M. H., and Navi, K.: 'Design of Efficient and Testable n-Input Logic Gates in Quantum-Dot Cellular Automata', *J Comput Theor Nanos*, 2013, **10**, pp. 2347-2353.

4 Hayati, M., and Rezaei, A.: 'Design and optimization of full comparator based on quantum-dot cellular automata', *ETRI Journal*, 2012, **34**, pp. 284-287.

5 Ke-ming, Q., and Yin-shui, X.: 'Quantum-dots cellular automata comparator', in 2007 7th Int. Conf. on ASIC, 2007, pp. 1297-1300.

6 Xia, Y., and Qiu, K.: 'Comparator Design Based on Quantum-Dot Cellular Automata', *Journal of Electronics & Information Technology*, 2009, **31**, pp.1517-1520.

7 Lampreht, B., Stepancic, L., Vizec, I., Zankar, B., Mraz, M., Bajec, I. L. and, Pecar, P.: 'Quantum-dot cellular automata serial comparator', in Digital System Design Architectures, Methods and Tools, 2008, September, DSD'08. 11th EUROMICRO Conf. on IEEE, 2008, pp. 447-452.

8 Ying, S. Y., Pei, T. Y., and Xiao, L. R.: 'Efficient design of QCA optimal universal logic gate ULG. 2 and its application', in Computer Application and System Modeling (ICCASM), October, Int. Conf. on, IEEE, 2010, **1**, V1-392 - V1-396.

9 Perri, S., Corsonello, P., and Cocorullo, G.: 'Design of Efficient Binary Comparators in Quantum-Dot Cellular Automata', *IEEE Trans. on Nanotechnology*, 2014, **13,** pp.192-202.

10 Walus, K., Dysart, T. J., Jullien, G. A. and Budiman, R. A.: 'QCADesigner: A rapid design and simulation tool for quantum-dot cellular automata', *IEEE Trans. on Nanotechnology*, 2004, **3**, pp. 26-31.