Cell Number Optimization for Quantum Cellular Automata based on AND-OR-INVERTER

Zohre Beiki[a], Satar Mirzakuchaki[b], Mohsen Soryani[a], Naser Mozayani[a]
[a] Computer Engineering Department, Iran University of Science and Technology, Tehran, Iran
[b] Electrical Engineering Department, Iran University of Science and Technology, Tehran, Iran
z_beiki@comp.iust.ac.ir, (m_khuchaki, soryani, mozayani)@iust.ac.ir

Abstract- Majority and inverter gates together make a universal set of Boolean primitives in Quantum-dot Cellular Automata (QCA) circuits. However, an experimental evaluation has shown that MV is not efficiently used during technology mapping by existing logic-synthesis tools. In this paper, we propose an approach, based on Genetic Algorithm, which reduces the area size of QCA circuits. Simulation results show that the proposed method is able to reduce area in QCA circuits design.

Key words: majority gate, AOI gate, Genetic Algorithm, QCA, area reduction.

I. Introduction

QCA is a new nanotechnology that attempts to create general computational functionality by controlling the position of single-electrons. It is anticipated that these technologies can achieve a density of 10^{12} devices/cm^2 and operate at terahertz frequencies [1]. QCA is a possible candidate to replace CMOS IC design [3].

GA is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions for optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions for optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. In a GA, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions [2]. As an example, in 1999, Carlos et.al used a genetic algorithm to design digital in logic circuit with gate counts constraint. This paper we present a novel QCA chromosome. This chromosome uses the AND-OR-INVERTER (AOI) gate, introduced in [7], as well as Majority gates and inverter gates.

We have simulated a 3 input logic circuit using the proposed chromosome and compared the results with results of another GA optimized design. A performance improvement of about 30% was achieved.

The rest of this paper is organized as follows: in section 2 some related background materials are presented. The proposed chromosome is introduced in section 3 and in section 4 simulation results are shown. Finally section 5 concludes the paper.

II. background materials

A: QCA Basics

QCA technology is based on the interaction of bi-stable QCA cells. A QCA cell is a square nanostructure of electron wells which confine free electrons. Each cell has quantum dots. These quantum dots are to hold a single electron per dot. At each of the four corners of the cell, a dot is located and two electrons are injected into the cell [3]. Due to Columbic repulsion, the two electrons reside in opposite corners. This results in two possible polarizations as seen in Fig. 1. This basic cell has made it possible to realize QCA-based storage elements, wires, and logic gates [4].

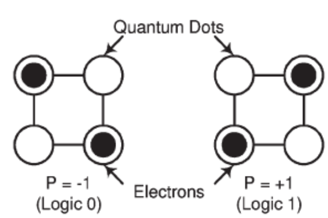

Figure.1 Two polarized QCA cell [4]

B: Majority Voter (MV) and Inverter (INV) gates

QCA implementation of logic design based on Majority and inverter gates consists of interconnecting MVs and INVs. The QCA MV gate is a device which implements a majority function. The device cell always assumes a majority polarization. The reason for this action is that it is in the polarization state in which the Columbic repulsion between electrons in the input cells is minimized. The Logic function of the majority gate is:

M (A, B, C) =AB+AC+BC

Where A, B and C are the inputs. By fixing the polarization of one input as logic '1' we can obtain an OR gate and by fixing the polarization of one input as logic '0' we can obtain an AND gate, respectively. In a QCA inverter, cells oriented at 45° with respect to each other take on opposing polarizations [5]. Fig. 2 shows QCA circuit for MV and INV gates.

C: And Or Inverter (AOI) gate:

The AOI gate, which is introduced in [7], has seven cells. Five of these cells act as inputs, one as a device cell and the last cell is the output cell.
Assuming A, B, C, D and E as inputs, the logic function of the AOI gate is:
AOI (A, B, C, D, E) = DE + (D+E) (A'C' + A'B + BC')
Fig. 3 shows the QCA AOI gate.

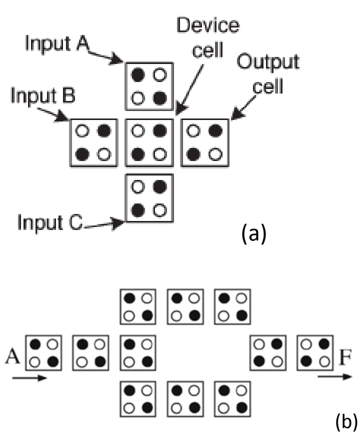

Figure 2: a) QCA majority gate b) inverter gate

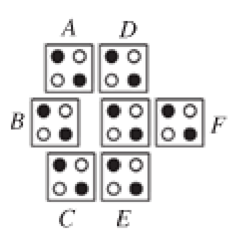

Figure 3: QCA AOI gate

D: chromosome structure introduction and fitness function

In [8], a chromosome is introduced which follows a tree structure. In this chromosome, internal nodes are either MV or INV gates and external nodes (leafs) are circuit inputs or constants. This chromosome has been used in [7-9] for their related designs. The chromosome's structure, implementing the function F (A, B, C) =

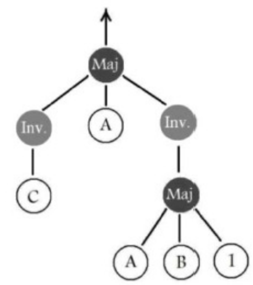

Figure4: A chromosome for M (A,C',M(A,B,1)')

A'B'C' + AC', is shown in Fig. 4.
To introduce the fitness function, we define it based on the similarity of the chromosome to the expected logical function. Also as known, the chromosome is preferred when it has fewer nodes. Suppose n is the number of input variables, F is the Boolean function, and C is the chromosome. The fitness function is defined as [8]:

$$Fit(C_i) = \frac{N(F,C_i)}{2^n} \quad (1)$$

Where N (F,C_i) is the number of identical minterms between chromosome C and function F. In a condition in which a chromosome has the same minterms as the function F, the fitness function presented in
(1) Will have its maximum value. In that case, a different fitness function is used in order to include the number of nodes used in the chromosome.

$$Fit(C_i) = Fit(C_i) + \frac{1}{Nodes\,(C_i)}$$

(2)
Where Nodes(C_i) denotes the number of nodes in C_i.

E: Mutation and Crossover

Mutation, in normal states, just changes one or more genomes in a population which include other genomes from problem space. But the process has a complication level in our structure. As an example, suppose we are to replace the inverter node with the majority or AOI node. Since the number of inputs in the structure of inverter gate is different from the number of inputs of the majority or

AOI gate, this process becomes infeasible. To solve this complication, we use a new method of mutation which works properly in our situation. In the new method, a random genome and its sub-tree is generated. Then the new genome replaces a genome (including its sub-tree) that has the worst fitness in the population. This replacement will be done in some specific probability in generations. This new gnome can be a combination of both MV and INV gates, together with AOI gate. For crossover, a random node and its sub-tree in one chromosome is exchanged by a random node and its sub-tree in another chromosome [9].

III. Proposed chromosome

Here we present our proposed chromosome's structure. In this structure, we use AOI gate in addition to the majority gate and the inverter gate. An example of the proposed chromosomes which implements the function F (A, B, C) = A'B'C' + AB'+ ABC', is shown in Fig. 5. Simulation results for implementation of different logic function using this chromosome show that it is possible to achieve considerable decrease in used area, in comparison with using older methods. As known, used area is one of the most important parameters in QCA circuit design. The proposed chromosome's structure, which has two crossover points, is presented in Fig. 6.
Figure 7 shows the result of mutation on the chromosome of Figure 5.

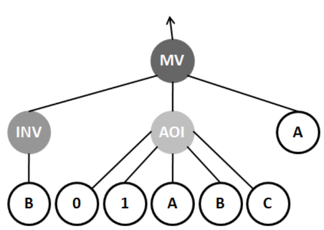

Figure5: Our proposed chromosome for
F (A, B, C) = A'B'C' + AB'+ ABC'

IV. Simulation Results

After simulating our proposed algorithm and the resulted circuit, we compare our simulation results with simulation results of presented methods in older studies. The circuits, which were used in comparisons, are 13 standard three-input functions, which are introduced in [10]. Through implementation of these 13 functions we can realize all possible three-input functions. The simulation results are presented in Table 1.
As it can be figured out we have about 29.8% of performance improvement in used area. The achievement concludes that using gates of only two types in QCA circuit design is not wise and we have to use gates of all three MV, INV and AOI types, based on the needs of the target circuits.

V. CONCLUSION

In QCA designs, low area usage has a high importance. In this paper, we introduced a chromosome structure which uses AOI gate, in addition to the majority and inverter gates. Implementation results show that over 29.8% performance improvement in used area is achieved. In future studies, we can work on decreasing the level count and/or the required clock number for implementation of different functions.

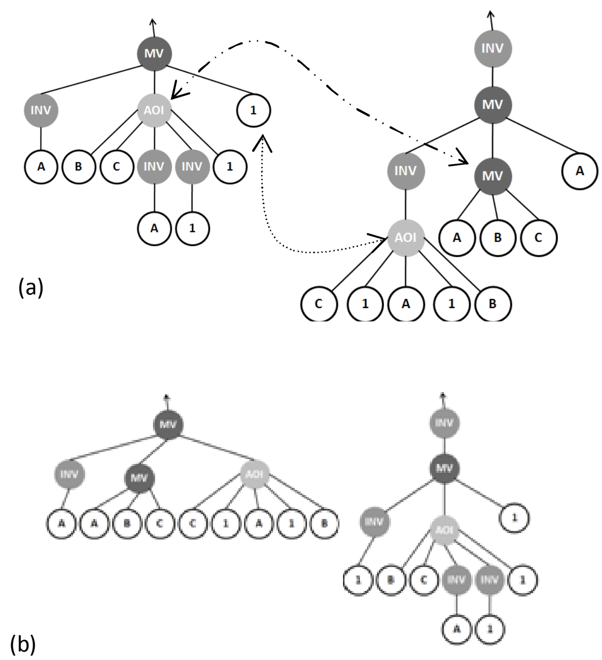

Figure6: Crossover; a) Before Crossover b) After Crossover

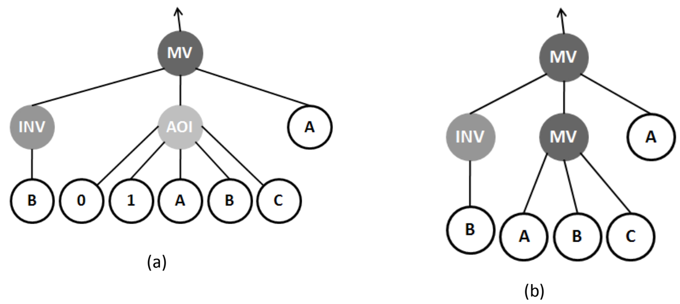

Figure7: Mutation; a) Before Mutation b) After Mutation

Table 1: Simulation Results

Functions		Previous Algorithm[6]			Proposed Algorithm				
		Number of MV	Number of INV	Effective Area (A_{MV})	Number of AOI	Number of MV	Number of INV	Effective Area (A_{MV})	Improvement %
1	F= AB′C	2	0	2	1	0	0	1.771	11.45
2	F=AB	1	0	1	0	1	0	1	0
3	F=A′BC+A′B′C′	3	2	6.334	1	1	1	4.438	29.93
4	F=A′BC+AB′C′	5	3	10	2	1	0	4.54	54.5
5	F=A′B+ BC′	2	0	2	1	0	0	1.771	11.45
6	F=AB′ +A′BC	4	2	7.334	1	1	0	2.771	62.2
7	F=A′BC+ABC′+A′B′C′	4	3	9.001	2	1	1	6.209	31.01
8	F=A	1	0	1	0	1	0	1	0
9	F=AB+AC+BC	1	0	1	0	1	0	1	0
10	F=A′B+B′C	3	1	4.667	1	1	0	2.771	40.6
11	F=A′B+BC+AB′C′	5	3	10	1	2	1	5.43	45.6
12	F=AB+A′B′	3	2	6.334	1	1	0	2.771	56.2
13	F=ABC′+A′B′C′+AB′C+A′BC	3	3	8.001	1	1	1	4.438	44.53
Average		2.84	1.46	**5.28**	0.92	0.92	0.3	**3.07**	**29.8**

Reference

[1] M. Askari, M. Taghizadeh, KH. Fardad, "Digital Design Quantum-Dot Cellular automata (A Nanotechnology Method)", International Conference on Computer and Communication Engineering, 2008, page(s): 952-955.

[2] Mitchell Melanie, an Introduction to Genetic Algorithms, a Bradford Book the MIT Press, Cambridge, Massachusetts, London, England, fifth printing 1999.

[3] M.R. Azghadi, O.Kavehei and K.Navi, "A Novel Design for Quantum-dot Cellular Automata Cells and Full Adders", Journal of Applied sciences, 2007, 7(22), page(s): 3460-3468.

[4] C.S. Lent, P.D. Tougaw, W. Porod, G.H. Bernstein, "Quantum Cellular Automata", Journal of Nanotechnology, 1993, 75(3), page(s): 49-57.

[5] K. Walus and G.A. Jullien, "Design Tools for an Emerging SoC Technology: Quantum-Dot Cellular Automata", Proceedings of IEEE 2006, 94(6), page(s): 1225-1244.

[6] M.R. Bonyadi, "A New Hybrid Genetic-Based Reduction Method in Nanoelectronic Circuits", World Applied Sciences Journal, 2010, 9(6), Page(s): 666- 673.

[7] M. Momenzadeh, J. Huang, M.B. Tahoori, F. Lombardi, "Characterization, Test, and Logic Synthesis of And-Or-Inverter (AOI) Gate Design for QCA Implementation", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2005, 24(12), page(s): 1881-1893.

[8] M.R. Bonyadi1 ,S.M.R. Azghadi, N.M. Rad, K. Navi, E. Afjei, "Logic Optimization for Majority Gate-Based Nanoelectronic Circuits Based on Genetic Algorithm", International Conference on Electrical Engineering, 2007, Page(s): 1 – 5.

[9] M. Houshmand, S. H. Khayat, R. Rezaei, "Genetic Algorithm based Logic Optimization for Multi-Output Majority Gate-Based Nano-electronic Circuits", IEEE International

Conference on Intelligent Computing and Intelligent Systems, 2009, Page(s): 584 – 588.

[10] R.Zhang, K.Walus, W.Wang, G.A.Jullien, “A Method of Majority Logic Reduction for Quantum Cellular Automata”, IEEE Transaction on Nanotechnology, 2004, 3(4), page(s): 443-450.