BCD Computing Structures in Quantum- Dot Cellular Automata

Maryam Taghizadeh[1], Mehdi Askari[2], Khossro Fardad[3]
[1,3] *Department of computer and Electrical Engineering, Islamic Azad University, Behbahan, Iran*
[2] *Department of computer and Electrical Engineering, Islamic Azad University, Omeidyeh, Iran*
Email (taghizadehmail@gmail.com, mehdiaskari58@yahoo.com, khossro.fardad@gmail.com)

Abstract

This paper proposes a detailed design analysis of BCD computing circuit for quantum-dot cellular automata (QCA). QCA is attracting a lot of attentions due to its very small sizes and low power consumption. The primary device, a quantum-dot cell, can be used to make gates, wires, and memories as such it is the basic building block of nanotechnology circuits. Because of its innovation, the current literature shows design several circuit structures. This paper proposes a modular design of a BCD adder in QCA and shows analyses based upon those designs. The designs are analyzed according to complexity, and area. Using QCADesigner, the layout is simulated. .

I. INTRODUCTION

Quantum-dot cellular automata (QCA) is one of the six promising technologies for nano-scale computing listed in the International Technology Roadmap for Semiconductors (ITRS) 2004 [1]. QCA, first proposed in 1993 by Lent, is a paradigm for low-power, high speed, extremely small sizes that could be realized in a variety of material systems. Until now several circuit designs using QCA technology have been proposed [2, 3]. Recent papers show that QCA can achieve high density [4], fast switching speed [5].

In the QCA research area, there are two suggestions. One is a physical design and the other is an algorithmic design [4]. Further, no general design guidelines have been proposed so far. A straightforward extension of a simple functional design pattern may fail. This makes designing a large scale circuits using QCA technology an extremely time-consuming process. High level designs focus on the logical and algorithmic design other than the physical design. Even though the actual QCA circuit designs need to manage considerable physical interactions which are possibly undesirable and disruptive. A very slight polarization of a cell induces a much larger polarization of its neighbour. The neighbour also feedback a larger polarization to the cell even before the neighbour's polarization is saturated. Such synergic effect amplifies not only the polarization of a signal, but also that of a noise which propagates through the sneak noise path. The algorithmic approach is also an important aspect in large systems.

This paper presents practical BCD adder design. The design follows the conventional design approaches. The architectural designs are based on modular structures for design reuse. Also the design follows two crossover options; coplanar crossings and multilayer crossovers. The design and layout analyzed according to the complexity, and area.

The paper is organized as follows. In section II, the background of QCA technology is presented. Section III and IV show the design approaches and the architectural design of the BCD adder in QCA, respectively. Analyses of simulation results and conclusions are presented in Section V and VI.

II. QCA BACKGROUND

All printed A quantum cell can be viewed as a set of four charge containers or dots, positioned at the corners of a square. Within this cell, two extra electrons are available, and by raising and lowering the potential barriers with the clock, an electron can localize on a dot. The electrons are forced to the corner positions by Coulomb repulsion. Thus, two different polarizations are available: p=1 and p=-1 as shown in Fig.1. Respectively, these polarizations provide a logical one and a logical zero. Once the cell polarization is set, the adjacent cell is affected by that cell and has the same polarization due to Columbic repulsion. In this way, signals are propagated. Also there are special purpose rotated cells that are used for wire crossings.

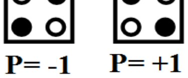

Figure 1. Two Polarized QCA Cells.

978-1-4244-1692-9/08/$25.00 ©2008 IEEE

A. Majority Gate

The fundamental QCA logical circuit is the three-input majority gate. This majority gate performs a three-input logic function of the majority gate is M (a, b, c) = ab + bc + ac. The gate symbols and their layouts are shown in Fig.2. AND and OR gates are implemented by setting one input to a constant.

$$a.b = M(a,b,0)$$

$$a + b = M(a,b,1)$$

If one input is set to 0, then the output is the AND of the other two inputs. If one input is set to 1, then the output is the OR of the other two inputs. With ANDs, Ors, and inverters, any logic circuit can be implemented. Several circuits are introduced in [4, 6].

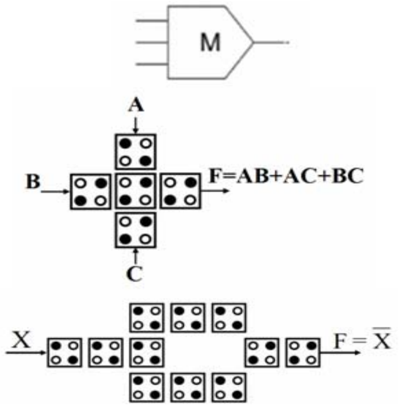

Figure 2. QCA Inverter and Majority Gate

B. QCA Clock

A QCA computation is performed by controlling the tunneling with a four phase "clock" signal. The clocking of QCA can be accomplished by controlling the potential barriers between adjacent quantum-dots [7, 8]. The clock used in QCA consists of four phases: hold, release, relax, and switch. It is considered that the lag between adjacent phases is 90°. Rather, it can be said that the clock changes phase when the potential barriers that affect a group of QCA cells (referred to as a clocking zone) are raised or lowered or remain raised or lowered.

During the switch phase, the interdot barrier is gradually raised, and the QCA cell settles down to one of the two ground polarization states as influenced by its neighbors. During the hold phase, the interdot barrier is held high, suppressing electron tunneling and maintaining the current ground polarization state of the QCA cell. During the release and relax phases, the interdot barriers are lowered, and the excess electrons gain mobility. In these two phases, a QCA cell remains unpolarized. Overall, the polarization of a QCA cell is determined when it is in its switch phase by the polarizations of its neighbors that are in switch and hold phases. The unpolarized neighbors in release and relax phases have no effect on determining the state of the QCA cell [9].

If QCA cells are lined up side by side and clocked appropriately, they act as a wire, propagating a signal down its length [10]. There is a 90 phase shift from one clock zone to the next. The different clocking zones are indicated in our layouts by the unlike shades of gray background of the cells as shown in Fig. 3.

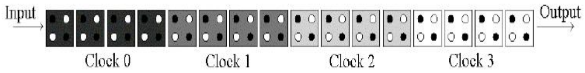

Figure 3. Different Clocking Zones

The cells in each clock zone behave like a single latch. This can simplify pipeline design. Signal feedback is possible but not recommended because extra wires are needed to keep the previous state value. Straight-forward design is better in QCA. Based on these QCA characteristics, a pipeline design with no feedback is chosen for the design of the BCD adder.

III. CROSSOVER AND MULTILAYER DESIGN

In QCA, there are two crossover options. These are coplanar crossings and multilayer crossovers. It has been believed that single layer designs are possible with QCA because of the ability to create co-planner crossovers [10]. Coplanar crossings require using two cell types (regular and rotated). The regular cell and the rotated cell do not interact with each other when they properly aligned, so rotated cells can be used for coplanar wire crossing. They have very little mutual interaction.

In the coplanar crossing, rotated cells are used when two wires cross. In a coplanar crossing, there is a possibility of a loose binding of the signal which causes a discontinuity of the signal propagation and there is the possibility of back-propagation from the far side constant input. So putting enough clock zones between the regular cells across the rotated cells is required. So the complex circuits using coplanar crossover are very likely fail [11]. As well, the crossover is extremely sensitive to fabrication errors and depends on the ability to fabricate two types of QCA cells. Therefore is presented a solution to this problem [10].

Multilayer crossovers use more than one layer of cells like multiple metal layers in a conventional IC. The multilayer wire crossings are shown in Fig. 4. On the other hand, a multilayer crossover is quite straightforward from the design viewpoint and the signal connection is steadier. The implementation process is less well understood than that for coplanar crossing.

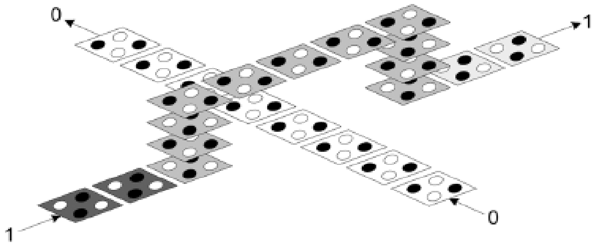

Figure 4. Multilayer Wire Crossing Layout

The four-bit adder designs in this paper don't use crossover but we only use the multilayer for correction of the adder result.

IV. BCD ADDER DESIGN

This section describes the architectural design of BCD adder. In computing and electronic systems, binary coded decimal (BCD) is an encoding for decimal numbers in which each digit is represented by its own binary sequence. It drawbacks are the increased complexity of circuits needed to implement mathematical operations and an inefficient encoding, its is still widely used in financial, commercial, and industrial applications. To perform addition in BCD, we can first add up in binary format, and then perform the conversion to BCD afterwards. This conversion involves adding 6 to each group of four digits that has a value of greater than 9.

In this paper, we have used two four bit binary adders and a conversion. The binary adder is ripple carry adder (RCA). A synchronous QCA full adder consists of three MV gates, two invertors and 5 different clock zones. The following equations are used for a half adder and full adder.

Half adder:

$$C = M(a, b, 0)$$

$$S = M(c', M(1, a, b), 0)$$

Full adder:

$$S_i = a_i b_i c_i + a_i b_i' c_i' + a_i' b_i c_i' + a_i' b_i' c_i$$

$$= M(M'(a_i, b_{i,} c_i), M(a_i, b_i, c_i'), c_i)$$

$$C_{i+1} = a_i b_i + b_i c_i + a_i c_i$$

$$= M(a_i, b_i, c_i)$$

Using these equations, the gate level design is implemented as shown in Fig. 5. Ripple carry adders are made by series connection of full adders. The carry input for each full adder is the carry output from the adjacent lower full adder.

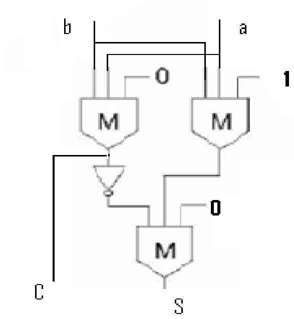

Figure 5. Half Adder Schematic

A full adder layout shows in Fig. 6[1]. The four bit adder is simply done by making a connection between the carry-out of the first bit full adder and the carry-in of the second bit full adder and adding more delays for synchronization. Since the carry-in value of the second bit full adder will be delayed by 5 clock zones, the other two inputs of the second bit full adder should be delayed by 5 clock zones. In addition, output of the first digit full adder should also be delayed by 5 clock zones. So extra wire channels are added to the ripple carry adder input and output sides for synchronization. An n bit adder has a delay of n clock cycles. A four bit ripple carry adder layout is shown in Fig.7.

A four-bit BCD adder consists of two four bit ripple carry adders and a block for the correction of the result. The following equation is used for the conversion of the BCD result.

$$x = M(M(M(Z_8, Z_4, 0), M(Z_8, Z_2, 0), 1), C_{out}, 1)$$

The four-bit BCD adder layout is shown in Fig. 8.

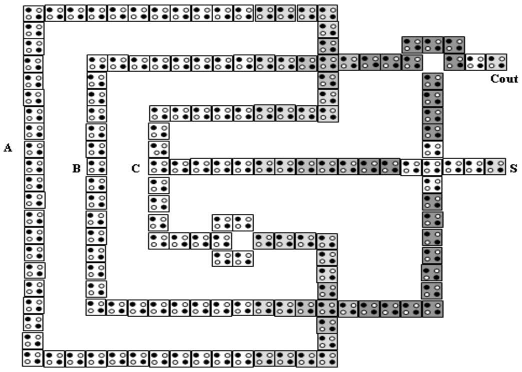

Figure 6. QCA Full Adder Layout.

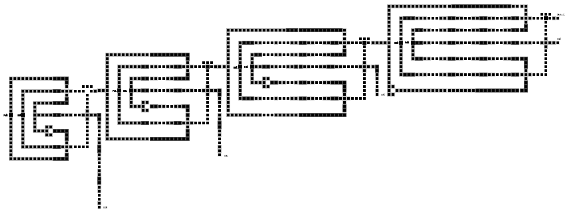

Figure 7. A Four-bit Ripple Carry Adder

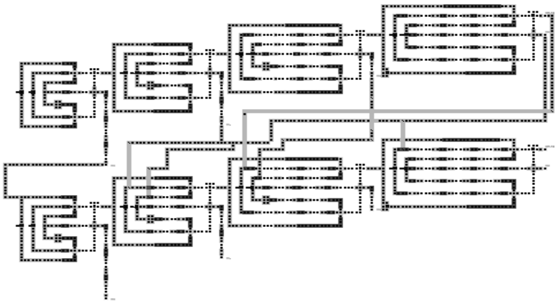

Figure 8. A Four- bit BCD Adder Layout.

V. SIMULATION

With QCADesigner, the circuit functionality is verified. This tool allows users to do a custom layout and then verify QCA circuit functionality by simulations. It includes two different simulations engines such as a bistable approximation and a coherence vector. We have used bistable approximation for simulation. The input and output waveforms are shown in Fig. 9. The area in this design is 5.8 um^2.

VI. CONCLUSION

This paper presents a QCA-based four-bit BCD adder structure. We have examined crossover and multilayer designs and haven't used crossover design but we have used just the multilayer design in the correction result. The binary adder implemented in this study is ripple carry adder (RCA). We have simulated a four-bit BCD adder with QCADesigner. The results give us an area of 5.8 *um^2*. Finally, for a fast design in QCA, complexity constraints are very critical issues and the design needs to use architectural techniques to improve the speed considering these limitations.

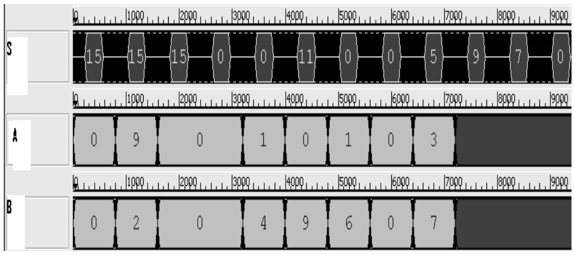

Figure 9. Simulation Result.

REFERENCES

[1] M. Choi, M. Choi, Z. Patitz and N. Park, "Efficient and Robust Delay-Insensitive QCA (Quantum-Dot Cellular Automata) Design" IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems, 2006.

[2] R. Zhang, K. Walus, W. Wang and G. A. Jullien, "A Method of Majority logic Reduction for Quantum Cellular Automata" *IEEE Trans. Nanotechnol*, Vol. 3, No. 4, pp. 443-450, Dec. 2004.

[3] W. Porod, "Quantum-dot devices and quantum-dot cellular automata" *Inter. J. Bifurcation and Chaos,* Vol. 7, No. 10, pp. 2199-2218, 1997.

[4] H. Cho and E.E. Swartzlander, "Adder Designs and Analyses for Quantum-Dot Cellular Automata" *IEEE Trans. On Nanotechnology*, Vol.6, No. 3, May 2007.

[5] J. M. Seminario, "A molecular device operating at terahertz frequencies: Theoretical simulations" *IEEE Trans. Nanotechnol.,* Vol. 3, No. 1, pp. 215-218, Mar. 2004.

[6] K. Walus, G. Schulhof, G. A. Jullien, R. Zhang and W. Wang, "Circuit design based on majority gates for applications with quantum-dot cellular automata" in *conf. Rec. 38th Asilomar Conf. Signals, Systems and Computers*, Vol. 2, pp. 1354-1357 ,2004.

[7] A.O. Orlov, I. Amlani, G.H. Bernstein, C.S. Lent, G.L. Snider, "Realization of a Functional Cell for Quantum-Dot Cellular Automata," *Science*, Vol. 277, pp. 928-930, 1997.

[8] K. Hennessy and C. S. Lent, "Clocking of molecular quantum-dot cellular automata" *J. Vac. Sci. Technol. B,* 19(5): 1752-1755, 2001.

[9] K. Kim, K. Wu and R. Karri, "The Robust QCA Adder Designs Using Composable QCA Building Blocks" *IEEE Trans. On Computer-Aided Design of Integrated Circuits and System,* Vol. 26, No. 1, pp. 176-183, January 2007.

[10] K. Walus and G. A. Jullien, "QCA Co-Planner Wire-Crossing and Multi-Layer Networks" *ATIPS,*

[11] K. Walus, G. Schulhof and G. A. Jullien, "High Level Exploration of Quantum-Dot Cellular Automata (QCA)" in conf.Rec. 38th Asilomar Conf.Signals, Systems and Computers, Vol. 1, pp.30-33, 2004.