

Highly Accurated QCA Binary Adder with Area and Time Optimization

^{*1}A.RamMohanRao and ^{#2}T.BabuRao

^{*}M.Tech, NOVA College of Engg & Technology, VLSI-SD, Jangareddygudem, AP-India

[#]Assistant Professor, ECE Dept, NOVA College of Engg & Technology, Jangareddygudem, AP-India

¹abburiramohan@gmail.com

Abstract— As transistors decrease in size more and more of them can be accommodated in a single die, thus increasing chip computational capabilities. However, transistors cannot get much smaller than their current size. The quantum-dot cellular automata (QCA) approach represents one of the possible solutions in overcoming this physical limit, even though the design of logic modules in QCA is not always straightforward. In this brief, we propose a new adder that outperforms all state-of-the-art competitors and achieves the best area-delay tradeoff. The above advantages are obtained by using an overall area similar to the cheaper designs known in literature. The 64-bit version of the novel adder spans over 18.72 μm^2 of active area and shows a delay of only nine clock cycles, that is just 36 clock phases. . Area and time optimizations are the main criterians in this paper. More efficient area and time optimized QCA adder involve many more applications in real world like DSP, BIOMEDICALS.

Keywords- adders, majority gate (MG), quantum dot cellular automata(QCA), Carry-Look ahead Adder (CLA), Ripple Carry Adder(RCA)

I. INTRODUCTION

Current silicon transistor technology faces challenging problems, such as high power consumption and difficulties in feature size reduction. Nanotechnology is an alternative to these problems. The Quantum dot cellular automata (QCA) are one of the attractive alternatives [1]. Since QCAs were introduced in 1993 by lent et al, and experimentally verified in 1997. QCA is expected to achieve high device density, extremely low power consumption and very high switching speed. QCA structures are constructed as an array of quantum cells with in which every cell has an electrostatic interaction with its neighboring cells [2]. QCA applies a new form of computation, where polarization rather than the traditional current, contains the digital information. In this trend, instead of interconnecting wires, the cells transfer the information throughout the circuit [4].

This paper describes the design of serial adder in QCA. It is designed using a full adder, 2:1 decoder and a D-latch. The

paper is organized as follows; the background of QCA technology is explained. We also provided the QCA clocking and described the QCA Designer tool. And also the design and implementation of full adder, Simulation results followed by conclusion are presented in the last section.

II. BACKGROUND

A. QCA basics

QCA technology is based on the interaction of bi-stable QCA cells constructed from four quantum dots. The cell is charged with two free electrons, which are able to tunnel between adjacent dots. These electrons tend to occupy antipodal sites as a result of their mutual electrostatic repulsion. Thus, there exist two equivalent energetically minimal arrangements of the two electrons in the QCA cell, as shown in Fig. 1. These two arrangements are denoted as cell polarization $P = +1$ and $P = -1$. By using cell polarization $P = +1$ to represent logic "1" and $P = -1$ to represent logic "0," binary information is encoded in the charge configuration of the QCA cell [2][5].

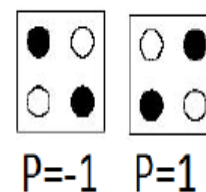


Fig. 1 QCA cell polarization

B. QCA logic devices The fundamental QCA logic primitives include a QCA wire, QCA inverter, and QCA majority gate[4]-[6], as described below. QCA Wire: In a QCA wire, the binary signal propagates from input to output because of the electrostatic interactions between cells. The propagation in a 90° QCA wire is shown in Fig. 2. Other than the 90° QCA wire, a 45° QCA wire can also be used. In this case, the propagation of the binary signal alternates between the two polarizations [4].

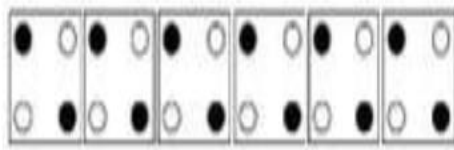


Fig. 2.1 QCA wire (900)

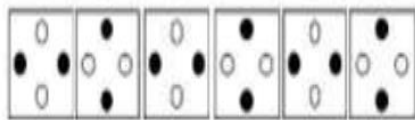


Fig. 2.2 QCA wire (450)

QCA Inverter: The QCA cells can be used to form the primitive logic gates. The simplest structure is the inverter shown. Fig. 3, which is usually formed by placing the cells with only their corners touching. The electrostatic interaction is inverted, because the quantum-dots corresponding to different polarizations are misaligned between the cells [3].

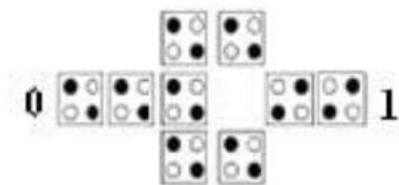


Fig. 3. QCA inverter

QCA Majority Gate: The QCA majority gate performs a three-input logic function. A layout of a QCA majority gate is shown in Fig. 4.1. Assuming the inputs are a , b and c , the logic function of the majority gate is $M(a, b, c) = ab+bc+ac$. (1)

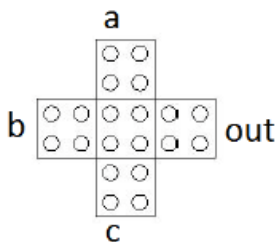


Fig. 4.1 QCA majority gate

The tendency of the majority device cell to move to a ground state ensures that it takes on the polarization of the majority of its neighbours. The device cell will tend to follow the majority polarization because it represents the lowest energy state [3]. By fixing the polarization of one input to the QCA majority gate as logic "1" or logic "0" an AND gate or OR gate will be obtained, respectively, as shown in Fig. 4.2.

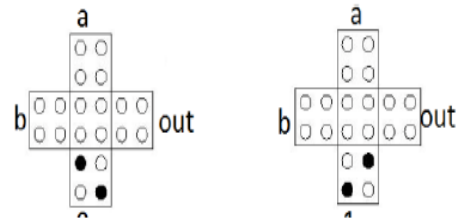


Fig. 4.2. 2-input AND and 2-input OR gates

III. CLOCKING

The QCA circuits require a clock, not only to synchronize and control information flow but also to provide the power to run the circuit since there is no external source for powering cells. With the use of four phases clocking scheme in controlling cells, QCA processes and forwards information within cells in an arranged timing scheme. Cells can be grouped into zones so that the field influencing all the cells in the zones will be the same. A zone cycles through 4 phases. In the Switch phase, the tunneling barriers in a zone are raised. While this occurs, the electrons within the cell can be influenced by the Columbic charges of neighboring zones. Zones in the Hold phase have a high tunneling barrier and will not change state, but influence other adjacent. Lastly, the Release and Relax decrease the tunneling barrier so that the zone will not influence other zones. These zones can be of irregular shape, but their size must be within certain limits imposed by fabrication and dissipation concerns. Proper placement of these zones is critical to design efficiency. This clocking method makes the design of QCA different from CMOS circuits. [8]. The Fig. 5. Shows the four available clock signals. Each signal is phase shifted by 90° degrees. When the clock signal is low the cells are latched. When the clock signal is high the cells are relaxed and have no polarization. In between the cells are either latching or relaxing when the clock is decreasing /increasing respectively.

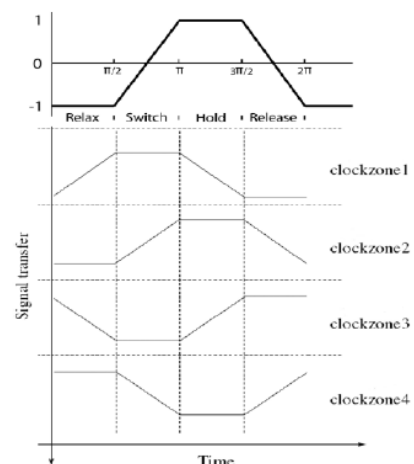


Fig. 5 four phases of the clock

IV. QCA DESIGNER TOOL

QCA logic and circuit designers require a rapid and accurate simulation and design layout tool to determine the functionality of QCA circuits. QCA Designer gives the designer the ability to quickly layout a QCA design by providing an extensive set of CAD tools. As well, several simulation engines facilitate rapid and accurate simulation. It is the first publicly available design and simulation tool for QCA. Developed at the ATIPS Laboratory, at the University of Calgary, QCA Designer currently supports three different simulation engines, and many of the CAD features required for complex circuit design. [9], [10].

V. QCA IMPLEMENTATION A. Design of a Serial Adder The serial adder is a combination of full adder, a 2:1 multiplexer and a D-flip-flop. The design of the individual part is shown independently first and later on the combination of all the three will result in serial adder [11]. The schematic of the serial adder is shown in Fig.6.

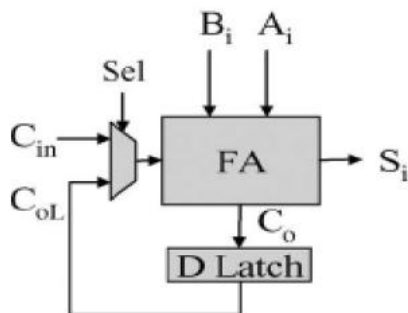


Fig.6 Schematic of a serial adder

B. Design of a Full Adder

The schematic of a full adder is shown in Fig.7.

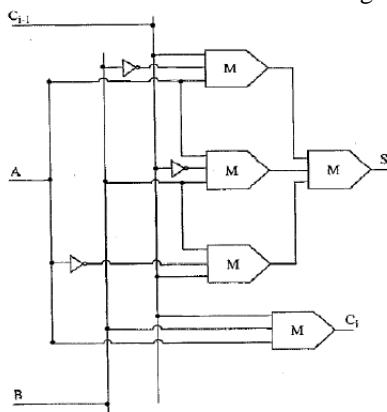


Fig.7 Schematic of a full adder

University of Notre Dame first proposed the design of a QCA full adder. The full adder is created from reduced majority logic. Reduction of sum of product logic to majority

will almost always lead to smaller layouts [13]. The full adder is a good example of a system where the majority circuit uses less logic gates than the best sum of products decomposition [12].

V. SAT-BASED LATTICE SYNTHESIS

This section shows how the problem of synthesizing one level of the 3*3 lattice with Shannon gates can be reduced to a SAT instance. We assume the reader's familiarity with the basics of Boolean satisfiability as presented, for example, in [14, 15].

Overview of the algorithm:

Synthesis of the 3*3 lattice is performed level by level. On each level, we solve the SAT problem, create the layout of that level, and proceed to the next level. Synthesis terminates when a level is reached on which all cofactors are constants. To formulate the SAT problem for one level of the lattice, we consider all the nodes on the level following immediately after the given one. The given level is called the upper level; the level following immediately after the given one is called the lower level. We formulate the set of requirements representing all possible routings of cofactors from the upper level to the lower level. Only non-constant cofactors are considered for routing. The constraints generated for all the nodes of the lower level are added to the set of all constraints representing the SAT instance.

Let us consider node N on the lower level. In the 3*3 lattice, there are six cofactors (c_1, \dots, c_6) that can potentially be routed to this node. These cofactors come from the nodes on the left (L), in the center (C), and on the right (R) nodes, with respect to the node N (Fig 3).

A group of SAT variables (x_{N1}, x_{Nk}) is associated with each node N of the lower level. These variables represent mutually exclusive possibilities of joining in node N the cofactors coming from the upper level. Suppose the cofactors ($c_1 \dots c_6$) are different non-constant Boolean functions. The group of six variables ($x_{N1} \dots x_{N6}$) is used to represent the possibilities that node N receives only one cofactor. Another group of six variables ($x_{N7} \dots x_{N12}$) is used to represent the possibility that N receives a pair of cofactors combined using join-vertex operation [16, 18, 23]. The join-vertex operation is defined for cofactors coming from different nodes and having opposite polarity. This is why there are only six pairs, namely (c_1, c_4), (c_2, c_3), (c_3, c_6), (c_4, c_5), (c_1, c_6), and (c_2, c_5).

Proposed adder architecture:

To introduce the novel architecture proposed for implementing ripple adders in QCA, let consider two n-bit addends $A = a_{n-1}, \dots, a_0$ and $B = b_{n-1}, \dots, b_0$ and suppose that for the i th bit position (with $i = n - 1, \dots, 0$) the auxiliary propagate and generate signals, namely $p_i = a_i + b_i$ and $g_i = a_i \cdot b_i$, are computed. c_i being the carry produced at

the generic $(i-1)$ th bit position, the carry signal c_{i+2} , furnished at the $(i+1)$ th bit position, can be computed using the conventional CLA logic reported in (2). The latter can be rewritten as given in (3), by exploiting Theorems 1 and 2 demonstrated in [15]. In this way, the RCA action, needed to propagate the carry c_i through the two subsequent bit positions, requires only one MG. Conversely, conventional circuits operating in the RCA fashion, namely the RCA and the CFA, require two cascaded MGs to perform the same operation. In other words, an RCA adder designed as proposed has a worst case path almost halved with respect to the conventional RCA and CFA. Exploited in the design of the novel 2-bit module shown in Fig. 1 that also shows the computation of the carry $c_{i+1} = M(p_i g_i c_i)$. The proposed n -bit adder is then implemented by cascading $n/2$ 2-bit modules as shown in Fig. 6(a). Having assumed that the carry-in of the adder is $c_{in} = 0$, the signal p_0 is not required and the 2-bit module used at the least significant bit position is simplified. The sum bits are finally computed as shown in Fig. 6(b).

It must be noted that the time critical addition is performed when a carry is generated at the least significant bit position (i.e., $g_0 = 1$) and then it is propagated through the subsequent bit positions to the most significant one. In this case, the first 2-bit module computes c_2 , contributing to the worst case computational path with two cascaded MGs. The subsequent 2-bit modules contribute with only one MG each, thus introducing a total number of cascaded MGs equal to $(n - 2)/2$. Considering that further two MGs and one inverter are required to compute the sum bits, the worst case path of the novel adder consists of $(n/2) + 3$ MGs and one inverter.

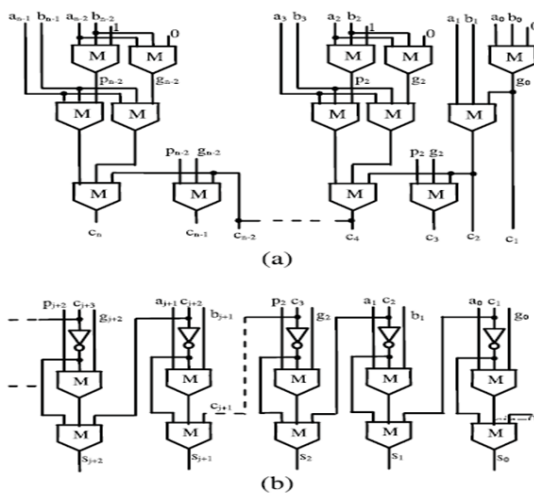


Fig 8: a novel n - bit adder (a) carry chain and (b) sum block

VI. CONCLUSION AND FUTURE WORK

The simulation results showed that the proposed algorithm performs better with the total transmission energy metric than

the maximum number of hops metric. The proposed algorithm provides energy efficient path for data transmission and maximizes the lifetime of entire network. As the performance of the proposed algorithm is analyzed between two metrics in future with some modifications in design considerations the performance of the proposed algorithm can be compared with other energy efficient algorithm. We have used very small network of 5 nodes, as number of nodes increases the complexity will increase. We can increase the number of nodes and analyze the performance.

REFERENCES

- [1] P. Farm and E. Dubrova, Technology Mapping for Chemically Assembled Electronic Nanotechnology, Proc. IWLS '02.
- [2] S. Goldstein and M. Budiu, Nanofabrics: Spatial computing using molecular electronics, Proc. 28th Annual International Symposium on Computer Architecture, (Gothenborg, Sweden), June 2001.
- [3] H. Hasegawa, A. Ito, Ch. Jiang, and T. Muranaka, Atomic Assisted Selective MBE Growth of InGaAs Linear and Hexagonal Nanowire Networks For Novel Quantum Circuits, Proc. 4th Intern Workshop on Novel Index Surfaces (NIS'01), Sept 16-20, Apset France.
- [4] S.P. Khatri, R.K. Brayton, A.L. Sangiovanni-Vincentelli, Cross-Talk Noise Immune VLSI Design Using Regular Layout Fabrics, Kluwer Academic Publishers, Boston, Hardbound, ISBN 0-7923, 7407-X, June 2001, 144 pp
- [5] C. Lent, P. Tougaw, W. Porod, and G. Bernstein, Quantum cellular automata, Nanotechnology, 4: pp. 49-57, 1993.
- [6] C.S. Lent, and P.D. Tougaw, A device architecture for computing with quantum dots, Proceedings of the IEEE, 85, 4, pp. 541 - 547, 1997.
- [7] P. Lindgren, R. Drechsler, B. Becker, Synthesis of Pseudo-Kronecker Lattice Diagrams. Proc. of Intl. Workshop of Applications of Reed-Muller Expantions to Circuit Synthesis, 1999, Victoria, B. C., Canada, pp. 197 - 204.
- [8] J. P. Marques-Silva, K. A. Sakallah, GRASP: A Search Algorithm for Propositional Satisfiability. IEEE Trans. Comp. Vol. 48, No. 5, May 1999, pp. 506-521.
- [9] M. W. Moskewicz et al., Chaff: Engineering an Efficient SAT Solver. Proc. DAC'01, pp. 530-535.
- [10] A. Mukherjee, R. Sudhakar, M. Marek-Sadowska, S. I. Long, Wave Steering in YADDs: A Novel Non-Iterative Synthesis and Layout Technique. Proc. DAC'99, pp. 466-471.
- [11] M. T. Niemier, A. F. Rodrigues, P.M. Kogge, A Potentially Implementable FPGA for Quantum Dot Cellular Automata, Workshop on Non-Silicon Computation (NSC), in conjunction with Int. Symp. On High Performance Computer Architecture (HPCA), February 3, 2003
- [12] M. Perkowski, and E. Pierzchala, New Canonical Forms for Four-Valued Logic, Report, Electrical Engineering Department, PSU . 1993.
- [13] E. Pierzchala, and M. Perkowski, Patent #5,959,871, September 28, 1999. Programmable Analog Array Circuit.
- [14] M. Perkowski, E. Pierzchala, and R. Drechsler, Ternary and Quaternary Lattice Diagrams for Linearly-Independent Logic, Multiple-Valued Logic and Analog Synthesis, Proc. ISIC-97, Singapur, 10-12 Sept.1997.
- [15] M. Perkowski, L. Jozwiak, R. Drechsler, and B. J. Falkowski, Ordered and Shared, Linearly Independent, Variable-Pair Decision Diagrams, Proc. First International Conference on Information, Communications and Signal Processing, ICICS'97, Singapur, 9-12 Sept. 1997. Session 1C1: Spectral Techniques and Decision Diagrams.
- [16] M. Perkowski, E. Pierzchala, and R. Drechsler, Layout-Driven Synthesis for Submicron Technology: Mapping Expansions to Regular Lattices, Proc. First International Conference on Information, Communications and Signal Processing, ICICS'97, Singapur, 9-12 Sept. 1997. Session 1C1: Spectral Techniques and Decision Diagrams.

- [17] M. Perkowski, M. Chrzanowska-Jeske, and Y. Xu, Lattice Diagrams Using Reed-Muller Logic. Proc. of Intl. Workshop of Applications of Reed-Muller Expansions to Circuit Synthesis, 1997, Oxford Univ., U.K., pp. 85 - 102.
- [18] M. A. Perkowski, M. Chrzanowska-Jeske, and Yang Xu, Multi-Level Programmable Arrays for Sub-Micron Technology based on Symmetries, Proc. ICCIMA'98 Conference, pp. 707-720, February 1998, Australia, published by World Scientific.
- [19] M. Perkowski, A. Al-Rabadi, P. Kerntopf, A. Mishchenko and M. Chrzanowska-Jeske, Three-Dimensional Realization of Multiple-Valued Functions using Reversible Logic, Invited Talk, Proc. Workshop on Post-Binary Ultra-Large Scale Integration Systems (ULSI), , pp. 47 - 53, May 21, 2001, Warsaw, Poland.
- [20] E. Pierzchala, M. A. Perkowski, S. Grygiel, A Field Programmable Analog Array for Continuous, Fuzzy and Multi-Valued Logic Applications, Proc. ISMVL'94, pp. 148 - 155, Boston, MA, May 25-27, 1994.
- [21] E. Sentovich, et al., SIS: A System for Sequential Circuit Synthesis, Tech. Rep. UCB/ERI, M92/41, ERL, Dept. of EECS, Univ. of California, Berkeley, 1992.
- [22] A. Singh, G. Parthasarathy, M. Marek-Sadowska, Interconnect Resource-Aware Placement for Hierarchical FPGAs. Proc. ICCAD '01, pp. 132-137.
- [23] F. Somenzi. CUDD Package, Rel. 2.3.1, <http://vlsi.Colorado.EDU/~fabio/CUDD/cuddIntro.html>