Binary Adder- Subtracter in QCA

Kalahasti. Tanmaya Krishna
Electronics and communication Engineering
Sri Vishnu Engineering College for Women
Bhimavaram, India

***Abstract*: In VLSI fabrication, the chip size has been reduced dramatically. With the current pace of scaling CMOS technology is set to hit a roadblock in the next few years, where it cannot be further scaled down due to several reasons like tunnel currents, quantum effects, sub-threshold leakage, short channel effects, fabrication costs and interconnect delay etc. One of the most promising nanotechnologies which can replace the present transistor based CMOS technology is the Quantum-Dot Cellular Automata. The major advantages of this technology are lesser power dissipation, improved speed and dense structures. In this paper, I propose an adder-subtracter as an extension to the already existing adder, which can perform both addition and subtraction operations, by occupying lesser area and reducing delay considerably.**

Keywords—Adder-Subtractor;Quantum-dot Cellular Automata (QCA).

I. INTRODUCTION

Quantum-dot Cellular Automata (QCA) is one of the most promising nanotechnologies which can replace the present transistor based CMOS technology. QCA has drawn a great deal of attention for the design of efficient logic circuits (in terms of area, delay and power). Special efforts are focused on designing the arithmetic circuits as these are the basic operations of any digital system. This paper is mainly focused on Binary addition.

For this design environment, traditional CMOS design architectures were considered as the reference. Ripple Carry Adder (RCA), Carry Look Ahead Adder (CLA) and Conditional sum adders were presented in [6].The Carry Flow Adder (CFA) shown in [7] was an improved version of RCA in which detrimental wires effects were reduced, thereby reducing the delay. Parallel-prefix architectures including Brent-Kung (BKA), Kogge-Stone, Ladner-Fischer, and Han-carlson adders, were analyzed and implemented in [8].

In [11] an innovative technique is used to implement a 2-bit adder, which occupies low-area and gives reduced delay considerably. Theoretical formulations demonstrated for both CLA and BKA in [10] were exploited here to realize this 2-bit addition slice. Here, the carry is propagated through two subsequent bit-positions with the delay of only one majority gate and the design is followed by clever top level architecture reducing the unnecessary clock phases due to long interconnections. And this adder runs in RCA fashion.

In this brief, an adder-subtracter is proposed to perform both the addition and subtraction operations at high-speed and low-area in QCA. The adder which is proposed in [11] is extended to perform both the operations by using 2's complement method.

The rest of the paper is organized as follows: A brief background of QCA technology and existing adder designs in QCA in section II, Adder-subtracter design is introduced in section III, Simulation and comparison results were presented in section IV, and finally, conclusions are given in section V.

II. BACKGROUND

(a) Basics of QCA

QCA Cell is the fundamental component of QCA Logic. Each QCA cell is made of four quantum dots in which two mobile electrons can be trapped which can tunnel between the dots. Due to the repulsion between the electrons, two electrons always take up the diagonal positions. Binary levels are represented by the positions of the electrons inside the cell unlike the voltage or current levels as in the CMOS. Figure 1 show the structures of quantum cell with two different polarizations.

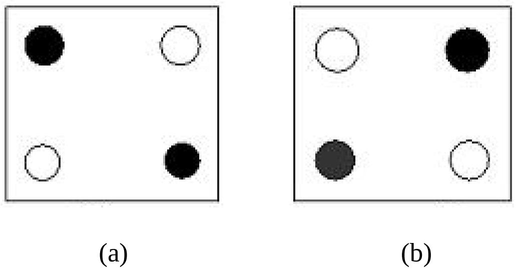

Figure. 1. QCA cells with 4 quantum dots

(a) Cell with polarization P= -1 (Logic '0')

(b) Cell with polarization P= +1 (Logic '1')

In the case of QCA, the clock signal modulates the inter dot barrier. QCA designs rely on a set of 4 clocks, phase shifted with respect to each other. Clock is used to push the information from inputs to the output by modifying the cell tunneling energy. QCA circuits have a four phase clock as against the CMOS which has only two states high and low and all the four phases have a phase shift of 90°, as shown in Fig. 2. Power to the QCA circuit comes from the clock itself unlike external power supply in case of CMOS. To assign the clock, the QCA circuit is divided into four sub arrays and a single potential modulates the inter-dot barriers in all the cells present in given array. Clock feature allows the sub array to perform a certain calculation at which it does not have influence from its successor and it acts as input to the next sub array, adding the feature of pipelining.

The four phases of clock correspond to switch, hold, release, and relax. In Switch phase, the barriers are raised and the QCA cells become polarized according to the state of input drivers. In the Hold phase, the barriers are kept high so that the cells in that sub array retain their values. During the Release phase, the barriers are lowered and the cells are allowed to relax to unpolarized state. In the final relax phase the barriers kept on low and the cells remains unpolarized.

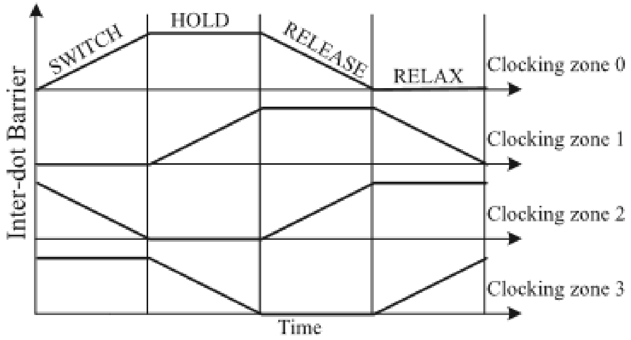

Figure. 2. Clock signals in four clocking zones

Unlike computation mechanisms that involve the transfer of electrons, as in CMOS gates, QCA computation does not involve electron transfer between adjacent QCA cells. Hence power dissipation is very less in circuits designed with QCA cells. Since only few electrons are involved in QCA computations, it is susceptible to thermal issues. Therefore it is important to consider power as an important parameter during the QCA design process. The power loss in clocked QCA circuits are classified as switching power and leakage power. Leakage power loss is independent of input states and occurs when the clock energy is raised or lowered to polarize or depolarize a cell. Switching power dependents on input data and occurs when the cell changes the state. Total power loss can be controlled by varying the rate of change of transitions in the clocking function.

Majority gate and Inverter are the two basic logic devices of QCA. Fig.3 shows the structure of the majority gate (MG) with 3 inputs, driver cell and output cell. Computation starts by driving the driver cell to the lowest energy state. Input cell changes its state by the signal which arrives towards the driver cell. Driver cell always gets the binary value of majority of input signals since it is where the repulsion for the mobile electrons in driver cell is at minimum and output follows the state of driver cell. The logic function for the majority gate is given by M(a,b,c) = ab+bc+ca with a, b, c as boolean variables. In this way two input AND, OR gates can be designed by keeping the third variable as either 0 or 1 respectively.

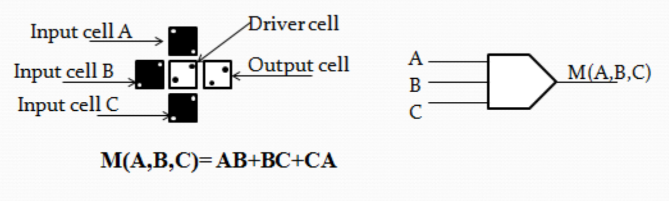

Figure 3. Structure of the Majority Gate, its schematic.

Inverter is the other basic logic device of QCA. If cells are placed diagonally to each other then they will have opposite polarizations. QCA Inverter is designed by this characteristic, such as shown in Fig. 4

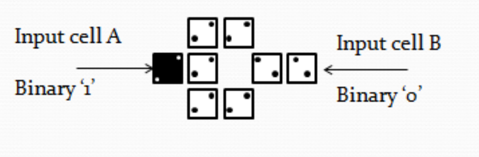

Figure. 4. Structure of the QCA Inverter

(b) Existing adders in QCA

n-bit Ripple carry adder (RCA) is implemented by cascading 'n' no. of 1-bit full adders. There by taking much time to generate the resultant output. For a 1-bit FA, carry in to carry out path consists of 1 MG and carry in to sum bit path consists of 2 MGs plus 1 inverter. Thus, the worst case path delay for both RCA and CFA became (n+2) MGs plus 1 inverter. Layouts of both the RCA (8-bit) and CFA (8-bit) are shown in figure. (5) and figure. (6) respectively.

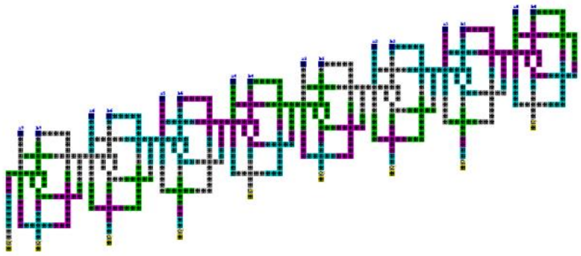

figure.5. Layout of 8-bit RCA

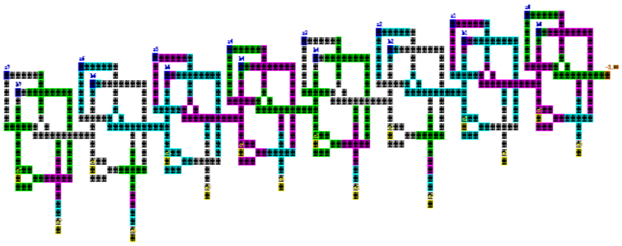

figure. 6. Layout of 8-bit CFA

A Carry Look Ahead adder (CLA) architecture formed by 4-bit slices was also presented in [7]. In CLA, carry generate $g_i = a_i . b_i$ and carry propagate $p_i = a_i + b_i$, computed for each bit are grouped by four. And the computational path delay consists of 7+4 ($\log_4$ n) cascaded MGs and 1 inverter, Where 'n' is the no. of input bits to be processed. Layout of CLA (8-bit) is shown in figure.7

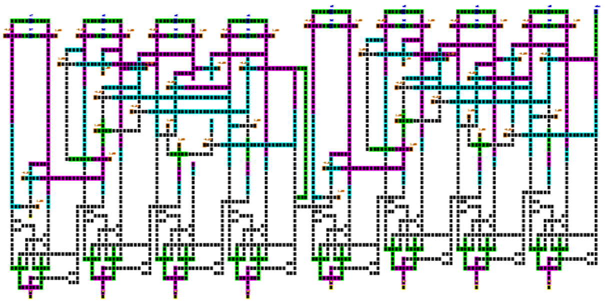

figure. 7. Layout of 8-bit CLA

And parallel-prefix adder, Brent-Kung adder (BKA) in [8] more efficient logic design as it can achieve lower computational path delay compared to previously described architectures. If it is an n-bit adder, then the prefix tree consists of 2*$\log_2$ n -2 stages. And the worst case computational path consists of 4*$\log_2$ n -3 cascaded MGs and 1 inverter. Layout of BKA (8-bit) is shown in figure.8

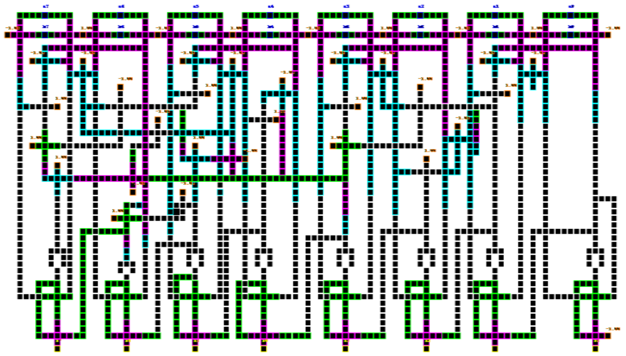

Figure 8. Layout of 8-bit BKA

The Novel architecture proposed in [11] for the implementation of ripple adder is efficient both in terms of area and delay. The carry generated at the least significant bit position is propagated through two subsequent bit positions with the delay of just 1 MG. Thus the adder has a worst case path almost halved with respect to conventional RCA and CFA architectures. The worst case computational path of novel adder consists of (n/2) +3 cascaded MGs and 1 inverter. Layout of novel adder (8-bit) is shown in figure.9

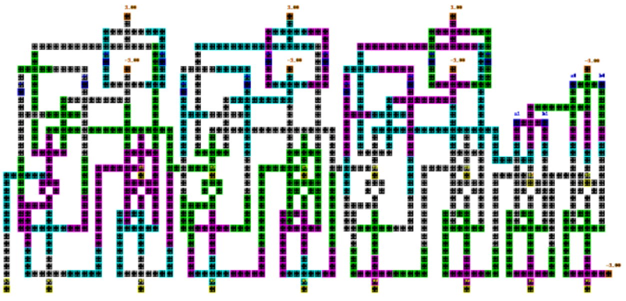

Figure 9. Layout of 8-bit novel ripple adder

III. PROPOSED ARCHITECTURE FOR ADDER-SUBTRACTER

The proposed adder-subtracter design is implemented by extending the novel ripple adder demonstrated in [11]. The adder is modified in such a way that it can perform both addition and subtraction operations depending on the control signal ‘add/subtract’ and the method used here is 2’s complement method of subtraction.

Let the inputs to the n-bit adder-subtracter module be a = $a_n....a_0$ and b = $b_n....b_0$.

In the proposed architecture, one of the two inputs i.e., either ‘a’ or ‘b’ is ‘xor’ed with the control signal input ‘add/subtract’. And also the ‘carry in’ of the least significant FA is fed with this ‘add/subtract’ control signal.

When **add/subtract** = **‘0’** then

Sum bit gives you **S = a + b.**

When **add/subtract** = **‘1’** then

Sum bit gives you **S = a’+b+1 = a - b.**

The modified carry module is shown in figure below.

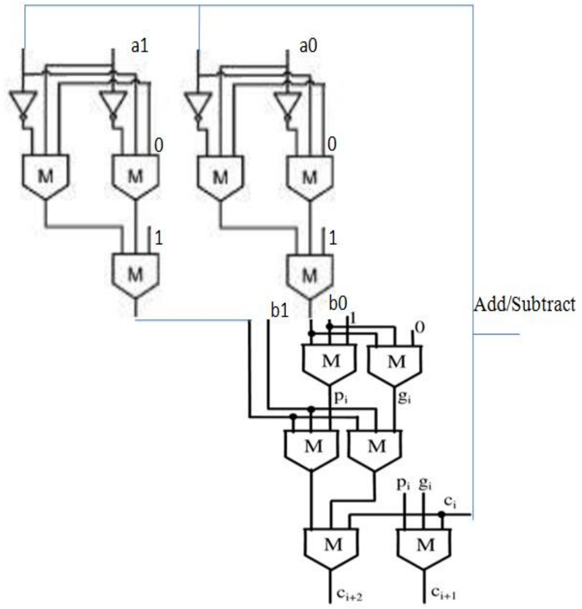

Figure 10. Basic carry module of 2-bit novel ripple adder

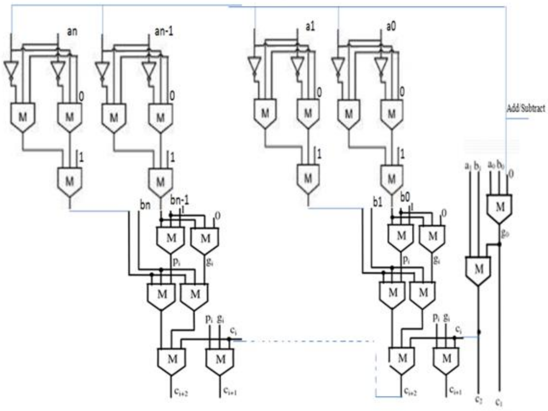

Figure 11. Carry block for n-bit novel ripple adder

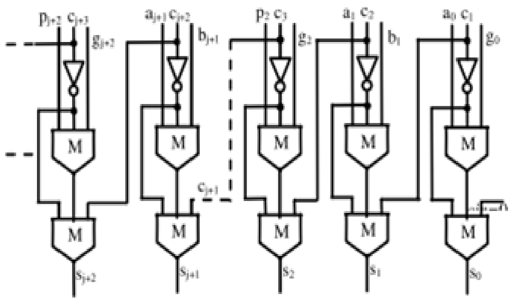

Figure 12. Sum block for n-bit novel ripple adder

IV.RESULTS

All the adder architectures were implemented for 8-bit input (range) using QCA designer tool [5], adopting the same rules and simulation settings used in [6]-[11]. Multilayer Crossing is used in designing the layouts. Simulation engine used is coherence vector engine for the accurate results.

Comparison results for the adders were given in the table below

Adder Name	No. of bits	Area (uu^2)	Clock delay (p sec)	Clock phases	Area-Delay product (uu^2-p sec)
RCA	8	1.67	23/4	11	4.592
CFA	8	1.21	2 2/4	10	3.025
CLA	8	4.11	4	16	16.44
BKA	8	2.11	22/4	10	5.275
Novel ripple adder	8	1.34	2	8	2.68

Table 1: Comparison results of existing adders

Graphical representation of the comparison is shown in figure. 13

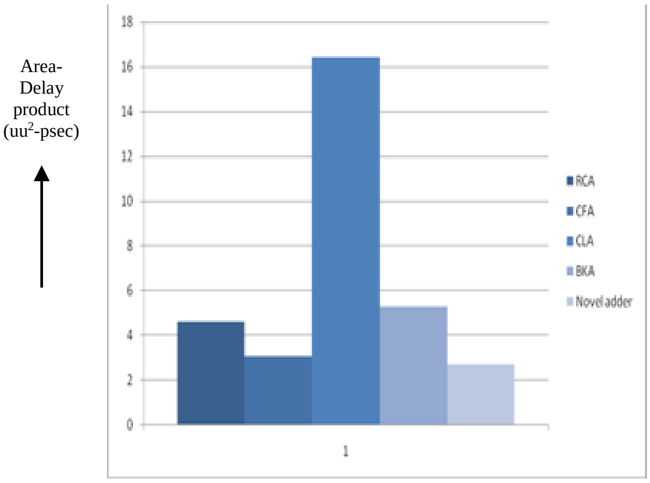

Figure 13. Graphical representation of ADP comparison among existing adders

Layout and Simulation results for the 8-bit adder subtracter were shown in figure (14) and figure (15) respectively.

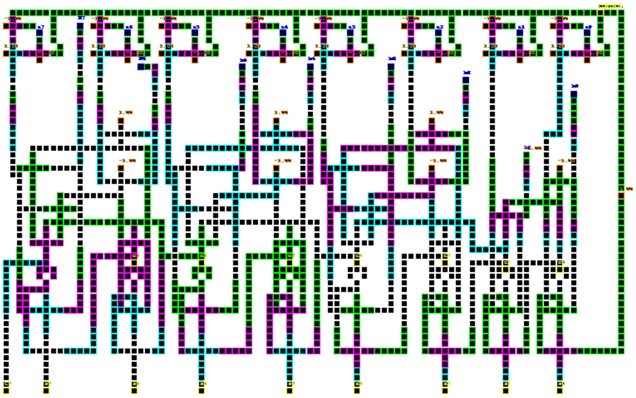

Figure 14. Layout of the 8 bit adder-subtracter in QCA

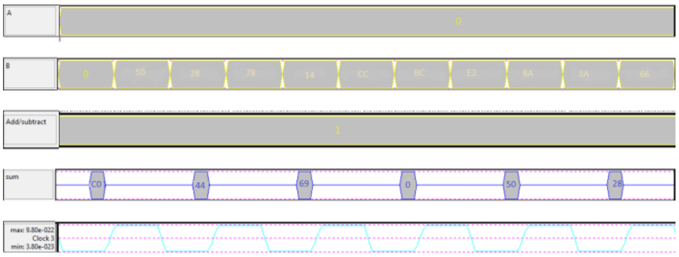

Figure 15. Simulation results for the 8-bit adder-subtarcter.

Carry bit is included in the output sum bus of 9 bits as most significant bit. And the first valid outputted after 3 clock cycles respectively. For the calculation of the least significant carry out, it takes 5 clock phases and it takes 7 clock phases for the carry propagation to the most significant bit and also to compute the sum output.

V.CONCLUSION

The novel ripple adder design is implemented and it achieved both area and speed performances higher than all other existing adders. And this adder is extended to adder-subtracter at a cost of 12 clock phases i.e., it takes 12 clock phases to give the desired output. It occupies an area of 2.17uu^2.

REFERENCES

[1] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernestein, “Quantum cellular automata,” Nanotechnology, vol. 4, no. 1, pp. 49–57, 1993.

[2] M. T. Niemer and P. M. Kogge, “Problems in designing with QCAs: Layout = Timing,” Int. J. Circuit Theory Appl., vol. 29, no. 1, pp. 49–62, 2001.

[3] J. Huang and F. Lombardi, Design and Test of Digital Circuits by Quantum-Dot Cellular Automata. Norwood, MA, USA: Artech House, 2007.

[4] Sang-Ho Shin, Jun-Cheol Jeon and Kee-Young Yoo, "Wire-crossing technique on Quantum-Dot Cellular Automata", NGCIT 2013, ASTL Vol.27, pp. 52-57, 2013.

[5] Konrad Walus, Timothy J. Dysart, Graham A. Jullien, Arief R.Budiman, "QCADesigner: A Rapid Design and Simulation Tool for Quantum-Dot Cellular Automata", IEEE Transactions on Nanotechnology, Volume. 3, no.1,PP.26-31,march 2004.

[6] H. Cho and E. E. Swartzlander, “Adder design and analyses for the Quantum-dot cellular automata,” IEEE Trans.Nanotechnology, vol. 6, no. 3, pp. 374–383, Jun. 2007.

[7] H. Cho and E. E. Swartzlander, “Adder and multiplier design in quantum-dot cellular automata,” IEEE Trans. Compute., vol. 58, no. 6, pp. 721–727, Jun. 2009.

[8] V. Pudi and K. Sridharan, “Low complexity design of ripple carry and Brent–Kung adders in QCA,” IEEE Trans. Nanotechnology., vol. 11, no. 1, pp. 105–119, Jan. 2012.

[9] W. Liu, L. Lu, M. O’Neill, and E. E. Swartzlander, Jr., “Design rules for quantum-dot cellular automata,” in Proc. IEEE Int. Symp. Circuits Syst., May 2011, pp. 2361–2364.

[10] S. Perri and P. Corsonello, “New methodology for the design of efficient binary addition in QCA,” *IEEE Trans. Nanotechnology.*, vol. 11, no. 6, pp. 1192–1200, Nov. 2012.

[11] Stefanie Perri, Pasquale Corsonello, and Giuseppe Cocorullo,"Area-DelayEfficient Binary Adders in QCA",IEEE Trans. on VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, vol. 22, no. 5, may 2014