Efficient Design of 2'S complement Adder/Subtractor Using QCA.

D.Ajitha [1]
Assistant Professor, Dept of ECE,
S.I.T.A.M..S., Chittoor,
A.P. India.

P.Yugesh Kumar,[2]
SITAMS,Chittoor,

Dr.K.Venkata Ramanaiah[3] ,
Associate Professor, ECE Dept ,
Y S R Engineering college of Yogi Vemana University,
Proddatur.

Dr.V.Sumalatha[4],
Associate Professor,
Dept of ECE,
J.N.T.U.C.E.A
Ananthapur, A.P.

Abstract: **The alternate digital design of CMOS Technology fulfilled with the Quantum Dot Cellular Automata. In this paper, the 2's complement adder design with overflow was implemented as first time in QCA & have the smallest area with the least number of cells, reduced latency and low cost achieved with one inverter reduced full adder using majority logic. The circuit was simulated with QCADesigner and results were included in this paper.**

Key words**: QCA, Majority Gate, Nano digital circuits.**

I. INTRODUCTION:

QCA technology has significant advantages of fast speed, high density and low power consumption. Therefore, it is considered as attractive for the development of digital circuits to meet the requirements of technology scaling[1]. Many digital circuits including memories, shift registers [6] and simple processor have been designed using QCA. Recent researchers have shown much interest in designing efficient binary arithmetic circuits again under this category much more concentrated on efficient adder circuits. Therefore design methods for complex QCA circuits have also been explored to achieve more efficient designs. However, with the development of computer architecture, the binary arithmetic has became the standard number system for electronic computers, and now dominates the modern computing world. While binary computer arithmetic design has been extensively investigated, limited attention has been given to design adder/subtractor in a single circuit. Hence, this paper explores the possibility of implementing the adder/subtractor in a single circuit with QCA technology as a first time.

In this paper efficient 1-bit full adder [10] has taken to implement the above circuit by comparing with previous 1-bit full adder designs [7-9]. A full adder with reduced one inverter is used and implemented with less number of cells. Latency and power consumption is also reduced with the help of optimization process [5]. Therefore efficient two's complement adder is also designed using the efficient 1-bit full adder [10] and reduced all the parameters like area delay and cost compared with CMOS [14].

The paper is organized as follows: In Section 2, QCA basics and design methods to implement gates functionalities is presented. Section 3 comparison 1-bit full adders in QCA implementation. Section 4 explains about the implementing 2's complement adder/subtractor with Simulation results .Finally, conclusions are provided in Section 5.

II QCA basics: QCA is based on electrons confining in dots and each cell has four quantum Dots [2]. The four dots are located in the corners of squares structure as shown in fig 1. The electrons tunnel through neighboring dots to the proper location during the clock transition.

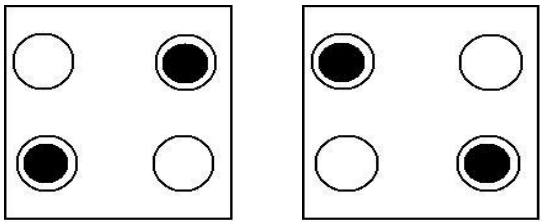


Fig1 Schematic of a QCA cell: (a): Logic '1' (b): Logic '0'.

Several approaches have been suggested for computation with an array of QCA cells. One approach is based on transferring the array to an excited state from a ground state by merely applying input data (without

explicit clocking). The array is expected to settle to a new ground state. However, sometimes the transition may result in a metastable intermediate state. In QCA, the logic states are not stored as voltage levels. Instead, the location of individual electrons determines the binary state.

QCA Clock Zones: To facilitate transfer to a new ground state, another approach based on clocking has been suggested . Clocking (by application of an appropriate voltage to a cell) leads to adjustment of tunneling barriers between quantum dots to transfer of electrons between the dots [4]. Clocking is performed in one of the two ways: zone clocking and continuous clocking. In each zone clocking, each QCA cell is clocked using a four-phase clocking scheme as shown in Fig 2. The four phases correspond to *switch*, *hold*, *release* and *relax*. In the switch phase, cells begin unpolarized and with low potential barriers but the barriers are raised during this phase. In the hold phase, the barriers are held high while in the release phase, the barriers are lowered. In the last phase, namely relax, the barriers remain lowered and keep the cells in an unpolarized state. An alternative to zone clocking, called continuous clocking, involves generation of a potential field by a system of submerged electrodes

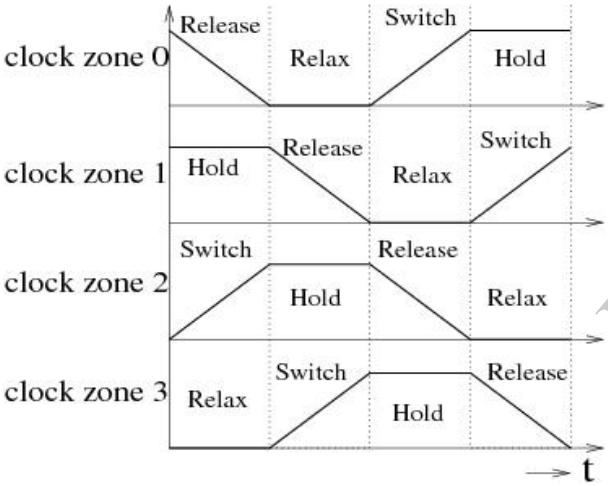


Fig 2:QCA clock zones

A (logic) wire is nothing but series of QCA cells. Two types of crossovers are used to build various circuits they are 1). Coplanar crossover, 2).Multilayer crossover. Based on comparison Multilayer crossovers [9] are used to construct all designs in this paper.

QCA WIRE:

In a QCA wire, the binary signal propagates from input to output because of the Coulombic interactions between cells[3]. This is a result of the system attempting to settle to a ground state. Any cells along the wire that are anti-polarized to the input would be at a higher energy level, and would soon settle to the correct ground state. The propagation in a 90-degree QCA wire is shown in Fig.3. Other than the 90-degree QCA wire, a 45-dgree QCA wire can also be used. In this case, the propagation of the binary signal alternates between the two polarizations. Further, there exists a so-called non-linear QCA wire, in which cells with 90-degree orientation can be placed next to one another, but off center.

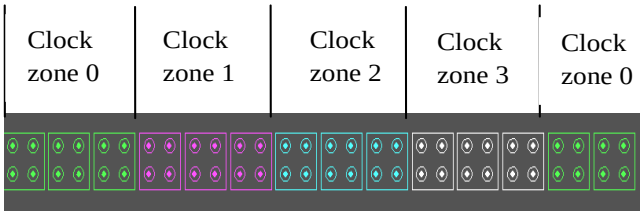


Fig 3: QCA wire

Majority Gate and Not gate:

Fig 4,Fig 5 shows the majority gate and its layout, fig 6,fig 7shows NOT gate and its layout. By using these two gates various Boolean expressions are realized. The majority gate performs a three-input logic function [13].

where M(A,B,C) = A.B+A.C+B.C.

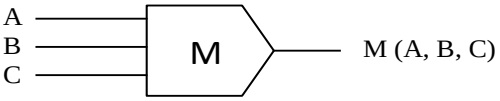


Fig 4: Majority gate.

By fixing the polarization of one input as logic "1" or "0", we can obtain an OR gate and an AND gate respectively. More complex logic circuits can then be constructed from OR and AND gates.

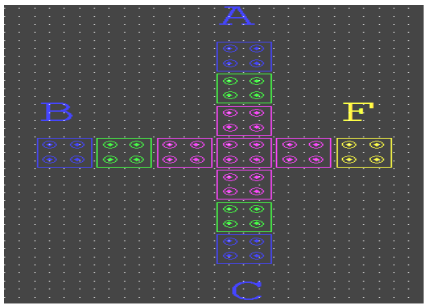


Fig 5: Layout of Majority gate.

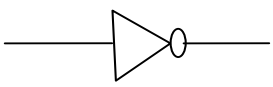


Fig 6: Inverter

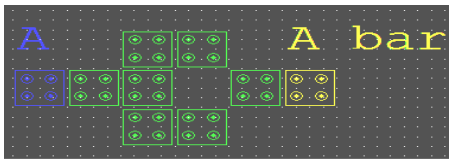


Fig 7: Layout of Inverter

III 1-bit full adders in QCA:

One-bit QCA adder consists of three majority gates and two inverters [8,9]as shown in Fig. 8.we arrange n proposed one-bit adders vertically in a column. The clocking of the cells within the n-bit adder is designed such that the carry will propagate down to the last bit before the sum is calculated, thereby implementing a CLA adder. This CLA adder design requires approximately larger in area. Furthermore, the latency will be more.

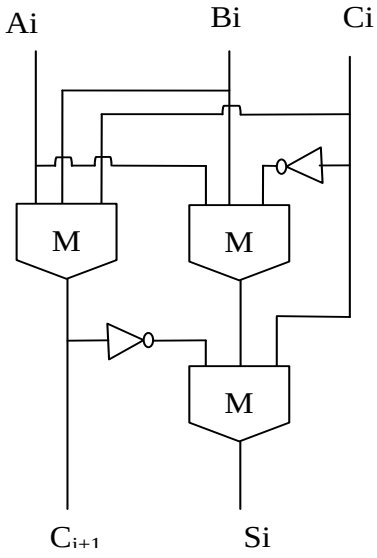


Fig 8.Conventional Full adder

QCA Addition Algorithm:

A one-bit full adder is defined as follows:

Inputs: Operand bits a, b and carry-in cin
Outputs: Sum bit s and carry-out cout

$$s = a.b.c_{in} + \bar{a}.\bar{b}.c_{in} + \bar{a}.b.\bar{c}_{in} + a.\bar{b}.\bar{c}_{in}$$

$$c_{out} = a.b + a.c_{in} + b.c_{in}$$

By using the majority function, the QCA addition algorithm as shown below equations

$$c_{out} = m(a, b, c_{in})$$

$$\bar{c}_{out} = m(\bar{a}.\bar{b}.\bar{c}_{in})$$

$$s = m(\bar{c}_{out}.c_{in}, m(a, b, \bar{c}_{in}))$$

The modified one inverter reduced full adder [10] can be implemented from the definition of the majority gate function for three Boolean variables as shown below

$$c_{i+1} = M(a_i, b_i, c_i);$$

So that it requires only one inverter as shown in Fig.9. Shows the representation for sum in an one-bit adder.

$$s_i = M(\overline{M(a_i, b_i, c_i)}, M(\overline{M(a_i, b_i, c_i)}, b_i, c_i), a_i);$$

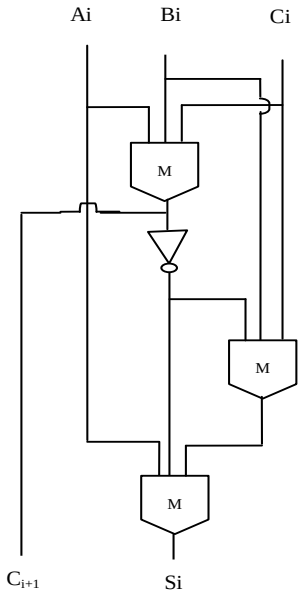


Fig 9: Modified 1- bit full adder

IV 2'S COMPLEMENT ADDER / SUBTRACTOR:

When dealing with 2's complement, any bit pattern that has a sign bit of zero in other words, a ***positive*** number) is just the same as a normal binary number..If, on the other hand, the sign bit is 1,it means, that the corresponding decimal number is ***negative***, and the bit pattern needs to be converted out of 2's complement. In normal subtraction process 2' complement subtraction is preferred over 1's complement due to advantages of 2' complement subtraction. No end around carry is required and also complexity is less in 2' complement subtraction.

OVERFLOW IN 2'S COMPLEMENT ADDITION

The condition for overflow is different if the bit string representation is 2'sComplement. If two numbers **x** and **y** have opposite signs (one is negative, the other is non-negative), then the sum will never overflow. The result will

have the sign either sign of **x** or **y**. Thus, overflow can only occur when **x** and **y** have the same sign.

- One way to detect overflow is to check the sign bit of the sum. If the sign bit of the sum does not match with the sign bit of **x** and **y**, then there is overflow.
- Suppose **x** and **y** both have sign bits with value 1. That means, both representations represent negative numbers. If the sum has sign bit 0, then the result of adding two negative numbers has resulted in a non-negative result, Overflow has occurred.
- Suppose **x** and **y** both have sign bits with value 0. That means, both representations represent non-negative numbers. If the sum has sign bit 1, then the result of adding two non-negative numbers has resulted in a negative result, Overflow has occurred.

In the above two cases if two n-bit numbers of the same sign are added/subtracted it requires n+1 bits to store the result otherwise it leads to wrong answers. In that cases overflow is useful to detect to whether the result is fit into the destination register or not and it clearly shows that to consider outer most carry bit to give the correct final result. So that would suggest that one way to detect overflow is to look at the sign bits of the two most significant bits and compare it to the sum.

However, there is an easier formula, is more though one that obscure. Let the carry out of the full adder adding the least significant bit be called $\mathbf{C_0}$. Then, the carry out of the full adder adding the next least significant bit is $\mathbf{C_1}$. Thus, the carry out of the full adder adding the most significant bits is $\mathbf{C_{k-1}}$. While adding two **k** bit numbers, the overflow can be detected as

$$\text{Overflow} = C_k \text{ XOR } C_{k-1}$$

This is effectively XOR the carry-in and the carry-out of the leftmost full adder. The XOR of the carry-in and carry-out differ if there is either a '1' being carried in, and a '0' being carried out, or if there's a 0 being carried in, and a 1 being carried out. Let's look at each case:

Case 1: '0' carried in, and '1' carried out

If a 0 is carried, then the only way that 1 can be carried out is if $\mathbf{X_{k-1} = 1}$ and $\mathbf{Y_{k-1} = 1}$. In that way, the sum is 0, and the carry out is 1. This is the case when you add two negative numbers, but the result is non-negative.

Case 1: '1' carried in, and '0' carried out

The only way 0 can be carried out if there's a '1' carried in is if $\mathbf{X_{k-1} = 0}$ and $\mathbf{Y_{k-1} = 0}$. In that case, 0 is carried out, and the sum is '1'. This is the case when you add two non-negative numbers and get a negative result.

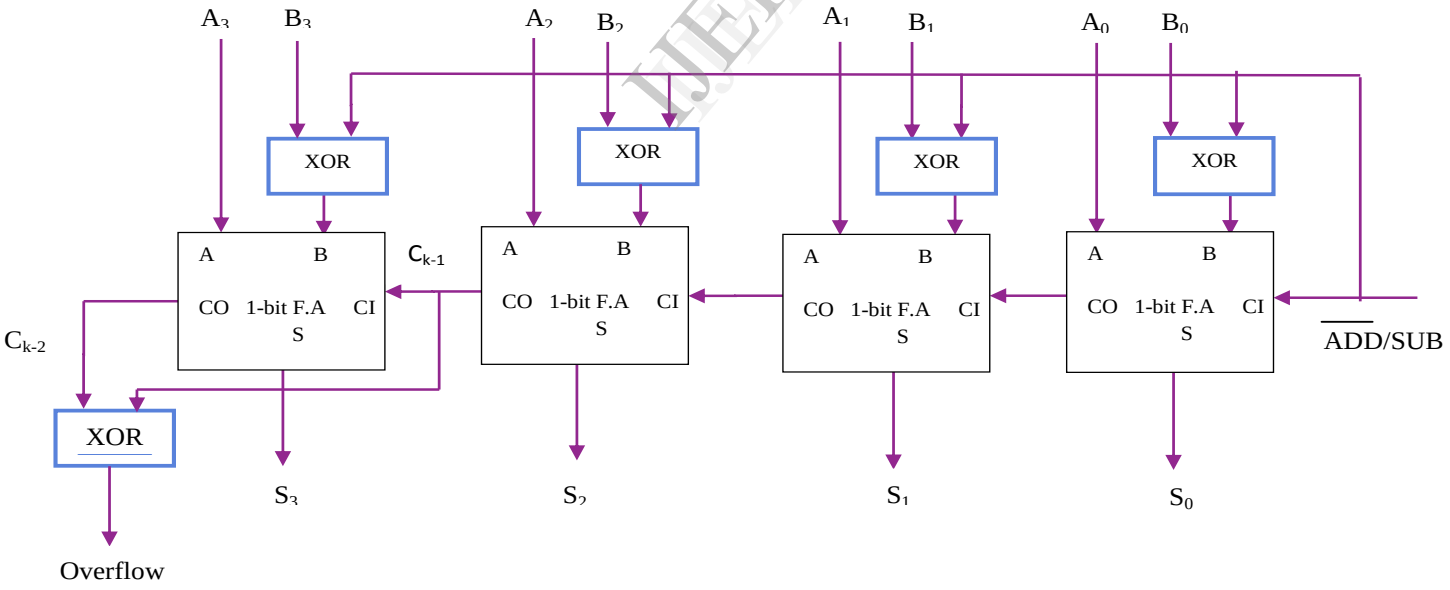


Fig.10. 2'S Complement Adder/ Subtractor

The above fig 10.shows the 2'S Complement Adder/ Subtractor along with the indication of overflow. [15].

If $\overline{ADD}/SUB = 0$, normal binary addition will be done with input carry=0.

If $\overline{ADD}/SUB = 1$,then A+ (1's complement of B) +1 will be done ,as this is the normal way of 2's complement subtraction.

2'S COMPLEMENT ADDER DESIGN

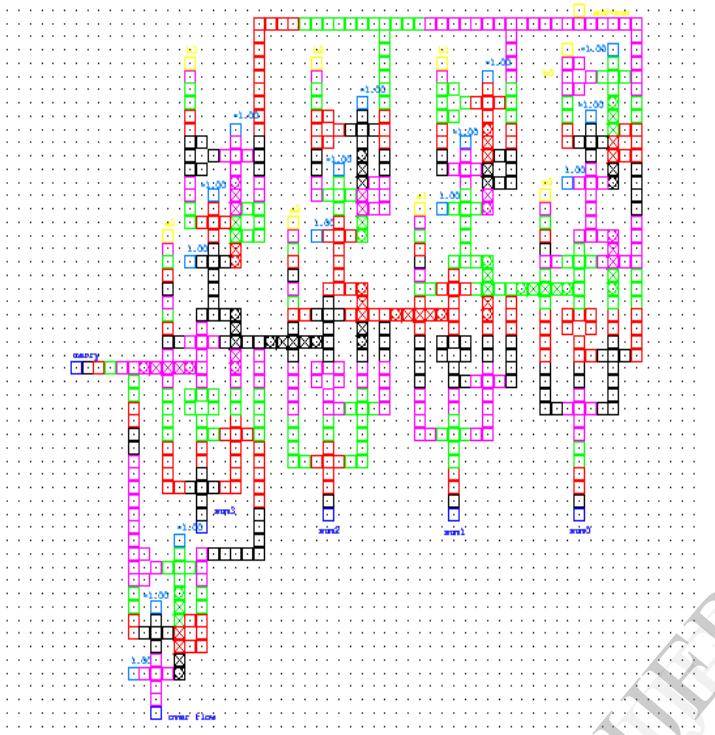
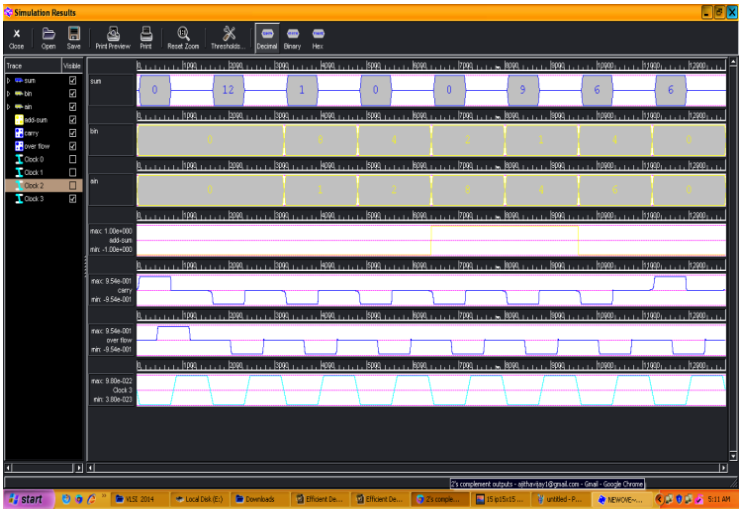


Fig 11: Layout design of 2's complement adder / subtractor in QCA

SIMUALTION WAVEFORMS



V. CONCLUSION:

The 2's complement adder / Subtractor using one inverter adder was constructed in this paper which have less area ,delay and cost consumption compared conventional 1- bit full adder element. Comparing this design with CMOS technology the CMOS requires Area about 207(um^2) .While the area decreases the overall cost also decreases. This is very important constraint while fabricating chips in high performance systems requires optimized circuits with lesser number of components. The overall cost function [5] & [12] is defined as Overall Cost =Area X $latency^2$.Table 1.gives the comparison analysis By using a single circuit both addition and subtraction can be performed efficiently.

2'scomplement adder /subtractor	No.of Cells	Area(um^2)
Using 1-bit full adder in QCA [8,9]	750 cells	1.2
Using 1-bit full adder in QCA [10]	695 cells	1.00

Table.1. comparison of 2's complement adder

REFERENCES:

[1] International Technology Roadmap for Semiconductors (ITRS), http://www.itrs.net, 2007.

[2] C.S. Lent, P.D. Tougaw, W. Porod, and G.H. Bernstein, "*Quantum Cellular Automata,"* Nanotechnology, vol. 4, no. 1 pp. 49-57, Jan. 1993.

[3] A. Orlov et al., *"Experimental Demonstration of a Binary Wire for Quantum-Dot Cellular Automata,"* Applied Physics Letters, vol. 74, no. 19, pp. 2875-2877, May 1999.

[4].Gary H Bernstein, Islam shah Amlani, Alexei O Orlov, Craig S Lent, Gregory L Snider," *Observation of Switching In A Quantum-Dot Cellular Automata Cell",* In Proc. Nanotechnology, 1999, Pp.1-8.

[5] Weiqiang Liu, ,Liang Lu and MaireO'Neil,Earl E. Swartzlander, *" Cost efficient decimal adder design in Quantum-dot cellular automata*" ISCAS 20-23 May 2012 PP 1347-1350.

[6] Vetteth, A. et al., "*Quantum-Dot Cellular Automata Carry-Look-Ahead Adder and Barrel Shifter"*, Proc. IEEE Emerging Telecomm.Technologies Conf. 2004.

[7] Wang, W., Walus, K. and Jullien G.A., *"Quantum-Dot Cellular Automata Adders"*, Proc. Third IEEE Conf. Nanotechnology, pp.461- 464, 2003.

[8] Zhang, R., Walus, K., Wang, W. and Jullien G.A., "*Performance Comparison of Quantum-Dot Cellular Automata Adders*", Proc. IEEE Int'l Symp. Circuits and Systems, vol.3, pp.2522-2526, 2005.

[9] Cho, H. and Swartzlander, E.E. Jr., *"Adder Designs and Analyses for Quantum-Dot Cellular Automata"*, IEEE Trans. on Nanotechnology, vol.6, no.3, 2007.

[10] Vikramkumar Pudi, K. Sridharan*," Low Complexity Design of Ripple Carry And Brent-Kung Adders in QCA",* In Proc.IEEE, 2011, Pp.1-14

[11] Cho, H. and Swartzlander, E.E. Jr., "*Adder and Multiplier Design in Quantum-Dot Cellular Automata*", IEEE Trans. on Computers, vol.58,no.6, 2009.

[12] C.D.Thompson,"*Area –time complexity of VLSI"* in Proc.7th Annual ACM Symp .Theory of Computing,1979,pp.81-88.

[13] R.Lindaman, *"A theorem for deriving Majority logic networks within an augmented Boolean algebra"*.IEEE transactions on electronic computers EC-9(3):338-342, September 1960.

[14] Habsah Abdul Shaer, Md. Mamun, Mohd. Marufuzzaman and H. Husain "*Design of a High Speed Low Power 2's Complement Adder Circuit"* Research Journal of Applied Sciences, Engineering and Technology 5(8): 2556-2564, 2013.

[15] Digital Design ,4th Edition by M.Morris Mano and Michael D.Ciletti.