Design and Implementation of Full Adders using QCA

V.Subashini
R.S. Koteeshwari
E.G.S. Pillay Engineering College,Nagapattinam

ABSTRACT

A approximate adder implemented with quantum-dot cellular automata (QCA) isdescribed.The most basic arithmetic operation is the addition of two binary digits, i.e. bits. In many fields the adder plays an important role but in most of the field the accuracy is not in concern. So we proposed a novel approximate adder of quantum dot cell automata (QCA).The proposed adder is used to reduce the circuit complexity and time delay with low error rate.The circuit complexity reduction is achieved by reducing the majority gate in the adder circuit. The operation of QCA circuits is simulated and verified using QCA Designer bistable vector simulation.

INTRODUCTION

Adders are commonly found in the critical path of many building blocks of microprocessors and digital signal processing chips. Adders are essential not only for addition, but also for subtraction, multiplication, and division. Addition is one of the fundamental arithmetic operations.A combinational circuit that performs addition of two bits is called a Half Adder. One that performs addition of three bits is called a Full Adder.In this paper we propose a full adder design with a minimum error rate compared with the normal full adder.

The performance of a traditional arithmetic circuit has limits.In order to further enhance performance, approximatearithmetic designs scarify accuracy to reduce energyconsumption and reduce costs related to manufacturing, verification and testing. These approximate arithmeticdesigns can be accepted in error-tolerance applications orapplied in applications related to human senses such as sight andhearing.

The paradigm of approximate computing is specific toselect hardware implementations of DSP blocks. It is shownin [4] that an embedded reduced instruction set computingprocessor consumes morepercentage of the energy in supplying dataand instructions, and less percentage of the energy while performingarithmetic only. Therefore, using approximate arithmetic insuch a scenario will not provide much energy benefit whenconsidering the complete processor. Programmable processorsare designed for general-purpose applications with noapplication-specific specialization. Therefore, there may notbe many applications that will be able to tolerate errors dueto approximate computing.

As the CMOS technologies approach its fundamental physical limit, there has been extensive research in recent years in development of nanotechnology for future generation IC. As an alternative to CMOS-VLSI, researchers have proposed an approach to computing with quantum dots, the quantum cellular automata(QCA).

Quantum dot cellular automata emerged as a new paradigm, beyond current switches to encode binary information. QCA encodes binary information in the charge configuration within a cell. Coulomb interaction between cells is sufficient to accomplish the computation in QCA arrays-thus no interconnect wires are needed between cell. No current flows out of the cell so that low power dissipation is possible.

The basic elements of QCA are QCA cell, Majority gate and Invertor[3].These is important elements. In QCA cell each cell is having four quantum dots and is having two free electrons. The locations of the electrons determine the binary states.Fig.1 shows the QCA cell diagram.

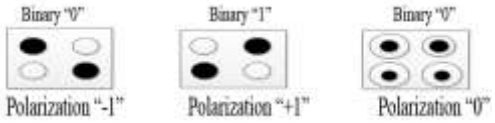

Fig.1 QCA Cell Polarization.

Adjacent QCA cells interact in an attempt to settle to a ground state determined by the current state of the inputs[1]. This is most clear in the case of the QCA wire shown in Figure 2.

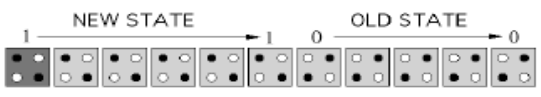

Fig 2.QCA wire

The polarization of the input cell is propagated down the wire, as 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 majority gate and inverter are shown in Fig. 3 and Fig. 4 respectively[2]. The majority gate is the fundamental QCA logic gate. The output cell will polarize to the majority polarization of the input cells. The majority gate performs a three-input logic function. Assuming the inputs are A, B and C, the logic function of the majority gate is

m(A, B,C) = A . B + B . C + A . C

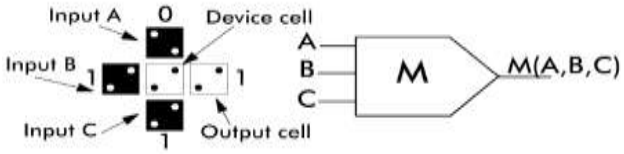

Fig 3. A QCA majority gate

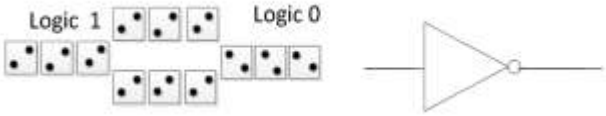

Fig.4. Invertor

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.

In conventional digital VLSI design, it is assumed that a circuit/system should function perfectly to provide accurate results. In non-digital world, ideal operations are seldom needed e.g. “analog computation” that provides “good enough” results instead of totally accurate results may in fact be acceptable. For many digital systems, the data already contained errors e.g. in a communication system errors may occur everywhere. In communication system, the analog signal out coming from the outer world must first be sampled before being transformed to digital data at the frontend of the system. The digital data are then processed and transmitted in a noisy channel before convert back to an analog signal at the back end of the system. Errors may take place anywhere during this process. Due to the advance in transistor size scaling, factors such as noise and process variations which are previously not important are becoming important in today’s digital IC design.

The architectures commonly employed in traditionalCMOS designs are considered a first referencefor the new design environment. Ripple-carry (RCA), carrylook-ahead (CLA), and conditional sum adders were presentedin [11]. The carry-flow adder (CFA) shown in [12]was mainly an improved RCA in which detrimental wireseffects were mitigated. In our design we proposed an approximate adder with improved RCA compared with the other methods.

In this paper, we propose an Approximate Adder (AA), which can configure the accuracy of results. The main contributions of our work are the following.

- We proposed anConventional one bit QCA full adder from the design of normal one bit QCA full adder to reduce the circuit complexity and the area constrain. The normal full adder design contain five majority gate and 3 inverters but our proposedConventional one bit QCA full adder design contain three majority gate and 2 inverters.
- We proposed an Approximate Adder1, Adder2, Adder3 and Adder4, (AA1, AA2, AA3, AA4) from the design of one bit QCA full adder to reduce the circuit complexity and the area constrain. The Approximate Adder1, Adder2 and Adder3 (AA1, AA2, AA3) contain two majority gate and 1 inverter but the Approximate Adder4(AA4) contain one majority gate and 1 inverter.

QCA Adders

A one-bit full adder design was first proposed by the University of Notre Dame . It consists of five majority gates and three inverters [6], [7] shown in fig: 1. Then One-bit QCA adder is designed to reduce the hardware complexity.

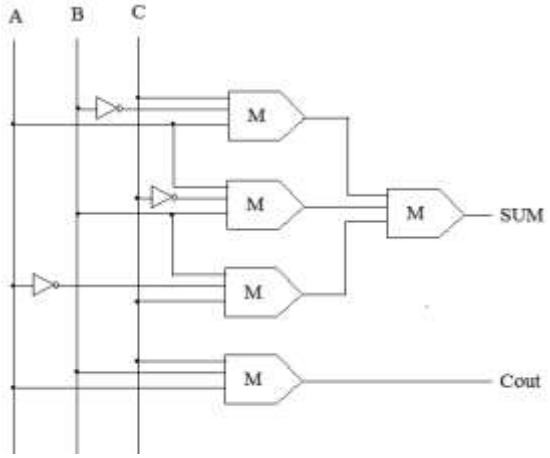

FIGURE 1: One-bit QCA full adder

PROPOSED APPROXIMATE ADDER

Our main objective is to develop an efficient design of QCA based Approximate Adder. In this section, an approximate adders are developed from the one-bit QCA adder designs.

One-bit QCA adder:

One-bit QCA adder consists of three majority gates and two inverters [2] shown in fig:2. It results in reduced hardware compared to the original full adder. The function of the sum and the carry are given by:

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

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

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

It consists of three majority gates and three inverters.

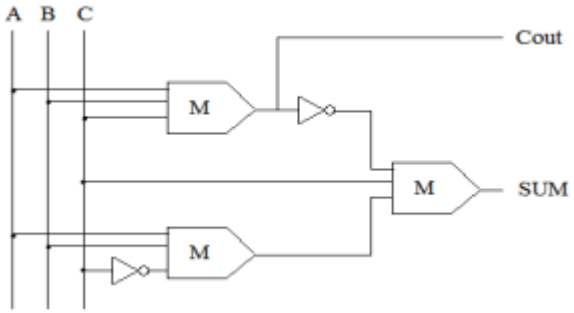

FIGURE 2: Conventionalone-bit QCA full adder

APPROXIMATE ADDER (AA1)

Fig 3 shows the Approximate Adder (AA1). In this design, the sum output is achieved by the one majority gate with an input of A, B and $\overline{C}$. Then the carry is generated from the one majority gate with an input of A, B and C. The majority gate count for this design is 2. The function of the sum and the carry are given by:

C_0 = m(a, b, c) ;

Sum = m(a, b, $\overline{c}$) ;

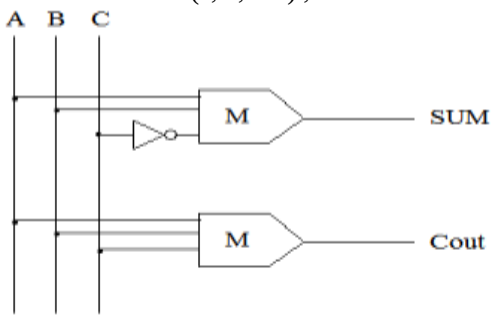

FIGURE : 3 APPROXIMATE ADDER (AA1)

APPROXIMATE ADDER (AA2)

Fig 4 shows the Approximate Adder (AA2). In this design, the sum output is achieved by the one majority gate with an input of A, $\overline{B}$ and C. Then the carry is generated from the one majority gate with an input of A, B and C. The majority gate count for this design is 2. The function of the sum and the carry are given by:

C_0 = m(a, b, c) ;

Sum = m(a, $\overline{b}$, c)) ;

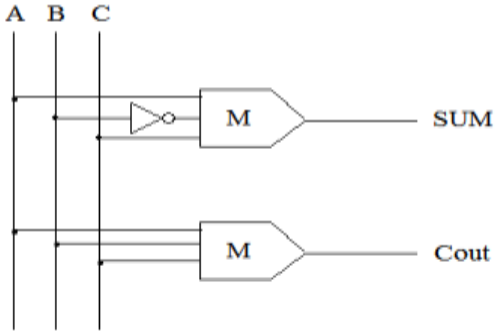

FIGURE : 4 APPROXIMATE ADDER (AA2)

APPROXIMATE ADDER (AA3)

Fig 5 shows the Approximate Adder (AA3). In this design, the sum output is achieved by the one majority gate with an input of $\overline{A}$, B and C. Then the carry is generated from the one majority gate with an input of A, B and C. The majority gate count for this design is 2. The function of the sum and the carry are given by:

C_0 = m(a, b, c) ;

Sum = m($\overline{a}$, b, c)) ;

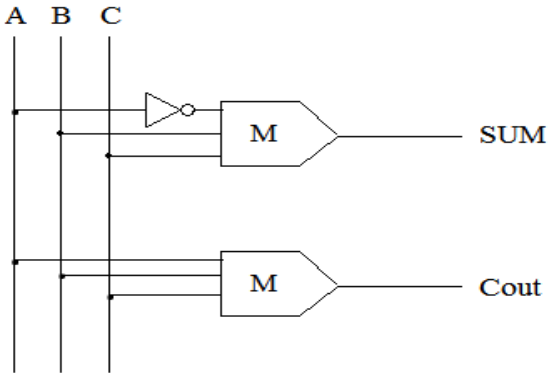

FIGURE : 5 APPROXIMATE ADDER (AA3)

APPROXIMATE ADDER (AA4)

Fig 6 shows the Approximate Adder (AA4). In this design, the carry output is achieved by the one majority gate with an input of A, B and C. Then the

sum is obtained from the inverted output of carry. The majority gate count for this design is 1. The function of the sum and the carry are given by:

$C_0 = m(a, b, c)$;

Sum = $\overline{C_0}$;

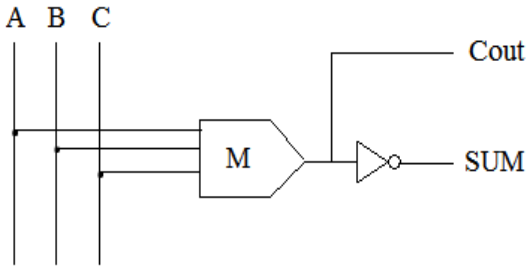

FIGURE : 6 APPROXIMATE ADDER (AA4)

For these approximate designs, a metric must be used to assess the approximation with

respect to the correct (exact) result; the so-called error distance has been proposed in [5] as figure of merit for inexact computing. For a given input, the *error distance* (ED) is defined as the arithmetic distance between an inexact output *a* and the correct output ED(a,b) = | a – b | = | $\sum_i a[i] * 2^i - \sum_j b[j] * 2^j$ |, a s : (1)

where*i*and *j* are the indices for the bits in *a* and *b*, respectively. For example, the two erroneous values “01” and “00” have an ED of 1 and 2 with the correct (exact) value “10”.

The full adder is the basic building block in the ripple carry adder, and most other adder circuits. A ripple carry adder is a digital circuit that produces the arithmetic sum of two binary numbers. It can be constructed with full adders connected in cascaded with the carry output from each full adder connected to the carry input of the next full adder in the chain.Figure 7 shows the 4bit approximate adder (RCA) which is designed from the approximation adder 4(AA 4). Layout shown in Fig 13.

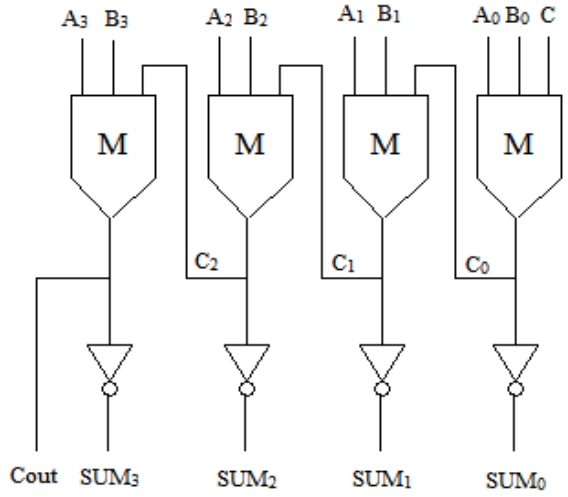

FIGURE : 7 4-bit APPROXIMATE ADDER

TABLE 2 Comparison table for various types of adder

Types of adder	No. of Majority gates	No. of Inverters
Normal FA	5	3
Proposed One bit QCA FA	3	2
AA 1	2	1
AA 2	2	1
AA 3	2	1
AA 4	1	1

From the above table, The normal full adder design contain five majority gate and 3 inverters then the proposed one bit QCA full adder design contain three majority gate and 2 inverters then the Approximate Adder1, Adder2 and Adder3 (AA1, AA2, AA3) contain two majority gate and 1 inverter but the Approximate Adder4(AA4) contain one majority gate and 1 inverter. From this table we came to know about that using the Approximate Adder4 instead of normal full adder we can reduce the circuit complexity.

QCA IMPLEMENTATION ANDFUNCTIONAL VERIFICATION

All the designs were verified using QCADesigner version 2.0.3 [8]. In the bistable approximation, weused the following parameters: cell size = 18 nm,number of samples = 182 800, convergence tolerance = 0.001000, radius of effect = 41 nm, relativepermittivity=12.9, clock high=9.8e−22, clock low=3.8e − 23, clock amplitude factor = 2.000, layerseparation = 11.5000 nm, and maximum iterationsper sample = 1000. In our QCA layouts. Fig. 15shows the simulation results of the Conventional one-bit QCA full adderQCAlayout shown in Fig. 8,9,10,11,12 and 13, which has been verifiedusing the QCADesigner tool. QCA cells compared topreviously reported designs.

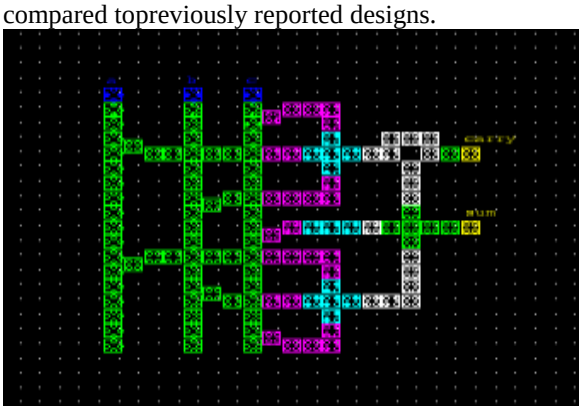

FIGURE :8 QCA Layout Of Conventional one-bit QCA full adder

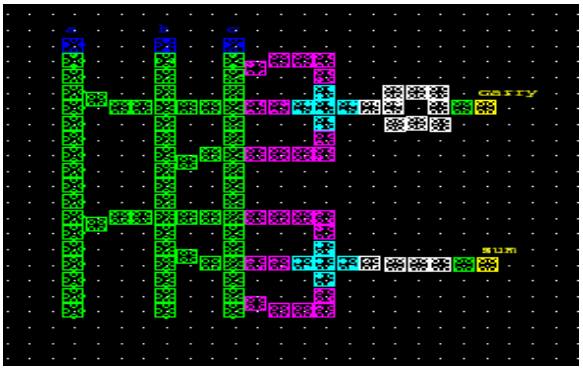

FIGURE :9 QCA Layout Of Approximate Adder1 (AA1)

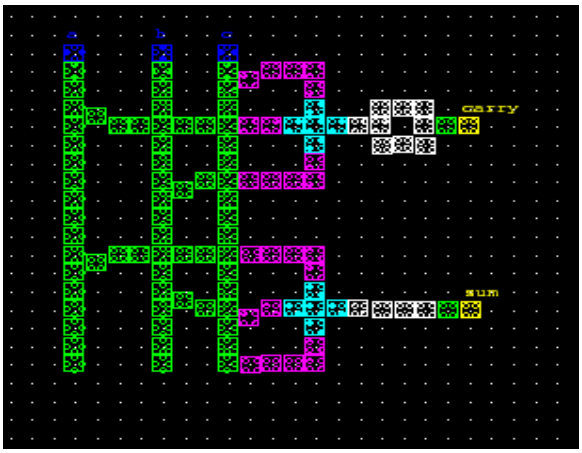

FIGURE :10 QCA Layout Of Approximate Adder2 (AA2)

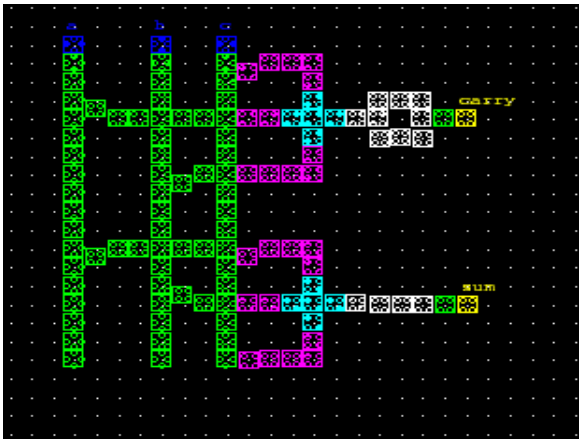

FIGURE :11 QCA Layout Of Approximate Adder3 (AA3)

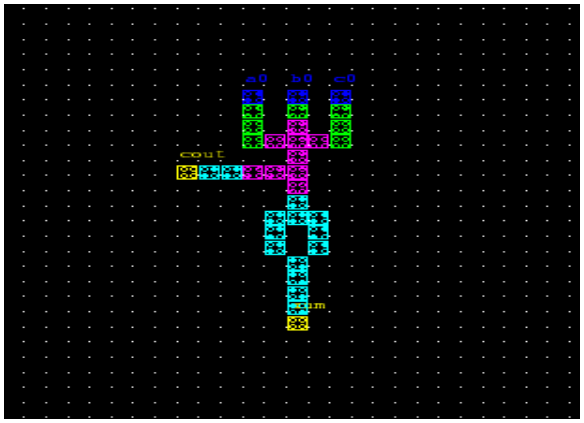

FIGURE :12 QCA Layout Of Approximate Adder4(AA4)

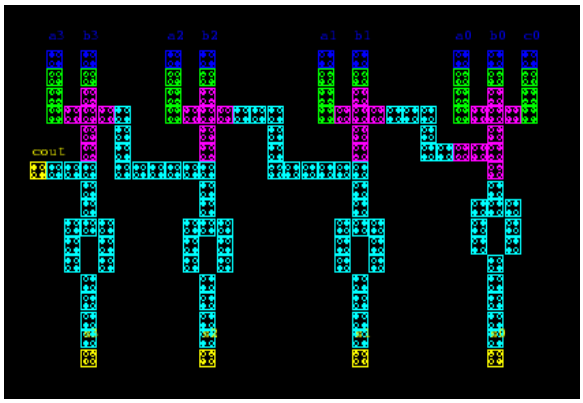

FIGURE : 13 QCA Layout Of 4-bit APPROXIMATE ADDER

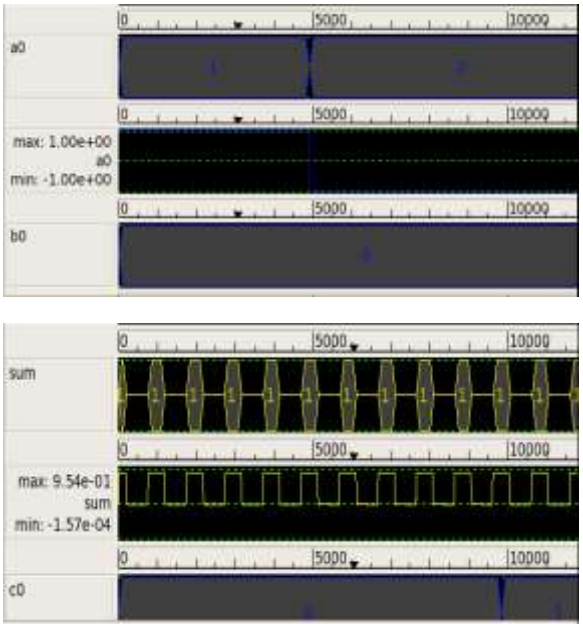

FIGURE : 14 Input of AA4 using QCADesigner

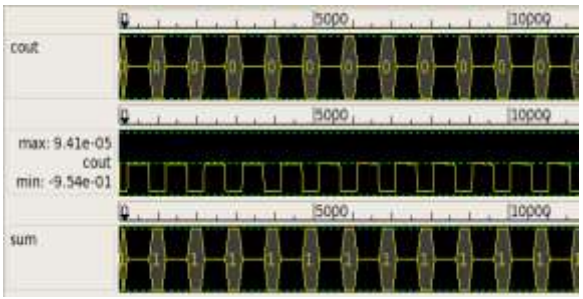

FIGURE : 15 output of AA4 using QCADesigner

CONCLUSION

In this paper, Approximate Adder Using QCA has been presented that reduces the number of majority gates compared to the conventional full adder .from all the above approximations AA4 will give effective results in terms of area and error rate.The proposed Approximate Adder produces the accurate output rather than the exact output with low error rate. When the errors introduced by these approximations were reflected at a high level like signal processing algorithms..the impact on output quality was very little. A decrease in the number of majority cells helped in reducing overall area when number of bits increases. The proposed Approximate adder has been design and simulated using the QCA Designer tool for the four-bit adder.

REFERENCES

1. Anoopvetteth, konradwalus, vassils.Dimitrov, graham a. Jullien, " Quantum-DotCellular Automata Carry-LookaheadAdderAnd Barrel Shifter", ATIPS Laboratory.
2. Wei wang, konradwalus and g. A. Jullien, "Quantum-Dot Cellular Automata Adders", 461 - 464 vol.2 2003 Third IEEE Conference on Nanotechnology. IEEENANO.
3. Kodam. Latha, m. Nanda maharshi, "Design Of Adders Using QCA" vol. 6, issue 4, pp. 1750-1759, International Journal of Advances in Engineering & Technology, Sept. 2013.
4. W. Dally, J. Balfour, D. Black-Shaffer, J. Chen, R. Harting, V. Parikh, J. Park, and D. Sheffield, "Efficient embedded computing," *Computer*, vol. 41, no. 7, pp. 27–32, Jul. 2008.
5. Zhixi Yang, Ajaypat Jain, Jinghang Liang, Jie Han and Fabrizio Lombardi, "Approximate XOR/XNOR-based Adders for Inexact Computing", 690 – 693 2013 13th IEEE Conference on IEEE NANO.
6. P.D. Tougaw and C. S. Lent, "Logical devices implemented usingquantum cellular automata," *J. Appl. Phys.*, American Institute ofPhysics, vol. 75, pp. 1818-1824, 1994.
7. A. Vetteth, K. Walus, V. S. Dimitrov, and G. A. Jullien, "Quatum-dotcellular automata carry-look-ahead adder and barrel shifter," *in Proc*.IEEE Emerging Telecommunications Technologies Conference, Dallas,TX, Sept. 2002.
8. HimanshuThapliyal, NagarajanRanganathan," Reversible Logic-BasedConcurrently Testable Latches forMolecular QCA", IEEE transactions onnanotechnology, vol. 9, no. 1, january 2010.
9. H. Cho and E. E. Swartzlander, "Adder design and analyses forquantum-dot cellular automata," *IEEE Trans. Nanotechnol.*, vol. 6, no. 3,pp. 374–383, May 2007.
10. H. Cho and E. E. Swartzlander, "Adder and multiplier design inquantum-dot cellular automata," *IEEE Trans. Comput.*, vol. 58, no. 6,pp. 721–727, Jun. 2009.