Journal of Computer Science 7 (7): 1072-1079, 2011
ISSN 1549-3636
© 2011 Science Publications

Design and Analysis of Adders using Nanotechnology Based Quantum dot Cellular Automata

[1]S. Karthigai Lakshmi and [2]G. Athisha
[1]Department of ECE, SONA College of Technology Salem, India
[2]Department of ECE, PSNA CET, Dindigul, India

Abstract: Problem statement: The area and complexity are the major issues in circuit design. Here, we propose different types of adder designs based on Quantum dot Cellular Automata (QCA) that reduces number of QCA cells and area compare to previous designs. The quantum dot cellular automata is a novel computing paradigm in nanotechnology that can implement digital circuits with faster speed, smaller size and low power consumption. By taking the advantages of QCA we are able to design interesting computational architectures. The QCA cell is a basic building block of nanotechnology that can be used to make gates, wires and memories. The basic logic circuits used in this technology are the inverter and the Majority Gate (MG), using this other logical circuits can be designed. **Approach:** In this paper, the adders such as half, full and serial bit were designed and analyzed. These structures were designed with minimum number of cells by using cell minimization techniques. The techniques are (1) using two cells inverter and (2) suitable arrangement of cells without overlapping of neighboring cells. The proposed method can be used to minimize area and complexity. **Results:** These circuits were designed by majority gate and implemented by QCA cells. Then, they simulated using QCA Designer. The simulated results were verified according to the truth table. **Conclusion:** The performance analyses of those circuits are compared according to complexity, area and number of clock cycles and are also compared with previous designs.

Key words: Quantum Dot Cellular Automata (QCA), Majority Gate (MG), Serial Bit Adder (SBA), neighboring cells, quantum cells, digital logic, binary signal, circuit design

INTRODUCTION

Quantum Dot Cellular Automata (QCA) is a transistorless computation paradigm that addresses the issues of device density and interconnection. The basic quantum dot cell is charged with two excess electrons and performs computation on coulomb interactions of electrons. Although QCA is still in research stages and research efforts are also in progress to determine the fault-tolerance of the QCA devices. In many ways, QCA circuit can be directly translated from conventional designs with addition of special clocking structures. This provides engineers with relatively easy transition from working with transistor technologies to designing with QCA (Walus *et al*., 2003; Cho and Swartzlander, 2007).

QCA structures are constructed as an array of quantum cells with in which every cell has an electrostatic interaction with its neighboring cells. 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 (Zhang *et al*., 2004). The basic operators used in the QCA are the three input majority gate and inverter. Any QCA circuit can be built using only majority gates and inverters.

This study proposes the design of three different types of Adders. These are Half Adder (HA), Full Adder (FA) and Serial Bit Adder (SBA). These circuits are designed based on the basic logical devices and implemented by cell minimization techniques. These techniques are (1) using 2 cells inverter and (2) suitable arrangement of cells without overlapping of neighbouring cells. Hence the proposed method can be used to minimize the area and complexity.

QCA designer: QCA logic and circuit designers require a rapid and accurate simulation and design layout tool to determine the functionality of QCA circuits. QCADesigner 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

Corresponding Author: S. Karthigai Lakshmi, Department of ECE, SONA College of Technology Salem, India of Calgary, QCADesigner currently supports three different simulation engines and many of the CAD features required for complex circuit design (Walus *et al*., 2004a; 2004b).

Included in the current version of QCADesigner are three different simulation engines. The first is a digital logic simulator, which considers cells to be either null or fully polarized. The second is a nonlinear approximation engine, which uses the nonlinear cell-to-cell response function to iteratively determine the stable state of the cells within a design. The third uses a two-state Hamiltonian to form an approximation of the full quantum mechanical model of such a system. Each of the three engines has a different and important set of benefits and drawbacks. Additionally, each simulation engine can perform an exhaustive verification of the system or a set of user-selected vectors (Walus *et al*., 2004b).

MATERIALS AND METHODS

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= +1and 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 (Wang *et al*., 2003; Walus *et al*., 2003; Cho and Swartzlander, 2007).

B.QCA Logic Devices: The fundamental QCA logic primitives include a QCA wire, QCA inverter and QCA majority gate (Cho and Swartzlander, 2007; Walus *et al*., 2004b; Reza and Vafaei 2008), as described below.

QCAWire: In aQCA 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.

QCA Inverter: The QCA cells can be used to form the primitive logic gates. The simplest structure is the inverter shown. Figure 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.

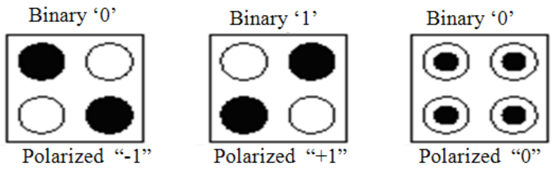


Fig. 1: QCA cell polarization

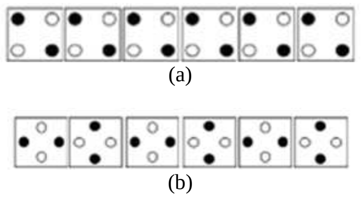


Fig. 2: (a) QCA wire (90°); (b) QCA wire (45°)

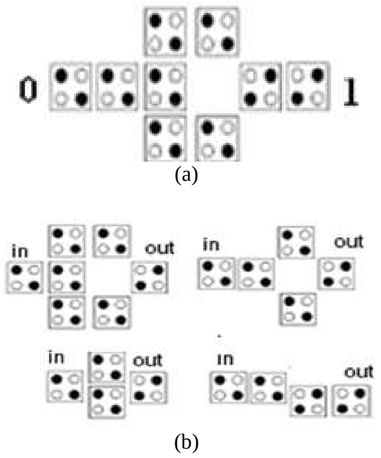


Fig. 3: (a) QCA Inverter; (b) Several inverter types

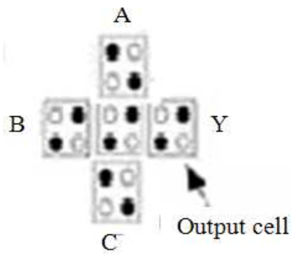
QCA majority gate: The QCA majority gate performs a three-input logic function. Assuming the inputs is A, B and C, the logic function of the majority gate is:

$$M(A, B, C) = AB+BC+CA \tag{1}$$

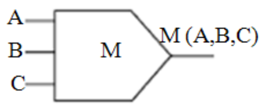
A layout of a QCA majority gate is shown in Fig. 4. The tendency of the majority device cell to move to a ground state ensures that it takes on the polarization of the majority of its neighbors. The device cell will tend to follow the majority polarization because it represents the lowest energy state. 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 follows:

$$M(A,B,0) = AB \tag{2a}$$

$$M(A,B,1) = A+B \tag{2b}$$

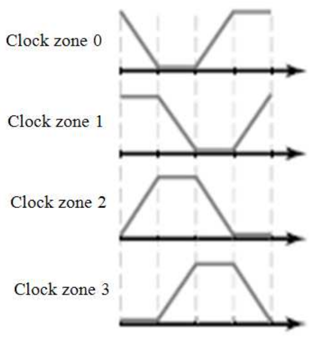


(a)

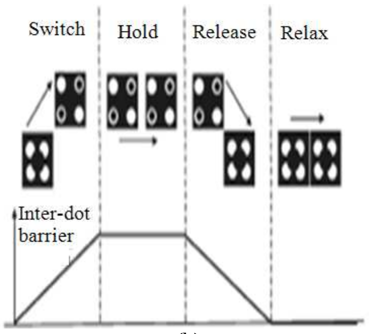


(b)

Fig. 4: (a) QCA majority gate; (b) Majority gate Symbol



(a)



(b)

Fig. 5: The four phases of the QCA clock

Thus, we can base all QCA logic circuits on three-input majority gates. In order to achieve efficient QCA design, majority gate-based design techniques are required. The study of majority gates mainly focuses on the threshold logic (Cho and Swartzlander, 2007).

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(serial add, shifter). With the use of four phase 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 an influence other adjacent zones. 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 (Walus *et al*., 2003; Cho and Swartzlander, 2007; Kim *et al*., 2007).

The Fig. 5 shows the four available clock signals. Each signal is phase shifted by 90°. 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.

Crossover design: In QCA, there are two crossover options (Cho and Swartzlander, 2007; Walus *et al*., 2004a). There are Coplanar Crossings (CC) and Multilayer Crossovers (MC). Coplanar crossings use only one layer, but require using two cell types (regular and rotated). The regular cell and the rotated cell do not interact with each other when they are properly aligned, so rotated cells can be used for coplanar wire crossings. Published information suggests that coplanar crossings may be very sensitive to misalignment. In the coplanar crossing, rotated cells are used when two wires cross. By choosing the connection point from rotated cells, either an original or an inverse of the input is available. 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. On the other hand, a multilayer crossover is quite straightforward from the design perspective and the signal connection is steadier. The implementation process is less well understood than that for coplanar crossings. The multilayer crossovers use more than one layer of cells like multiple metal layers in a conventional IC.

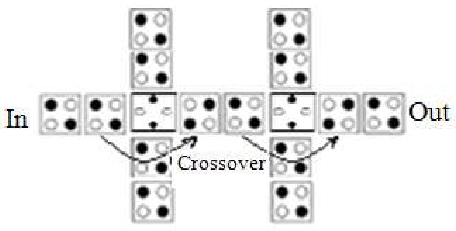


Fig. 6: Layout of coplanar crossings

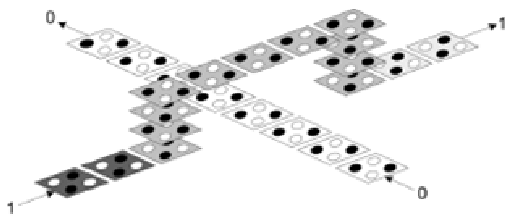


Fig. 7: Layout of coplanar crossings

An example layouts of coplanar and multilayer wire crossings are shown in Fig. 6-7.

Majority gate design: Digital computers perform various arithmetic operations. The most basic operation is the addition . The addition operation is achieved by majority logic that can reduce the overall number of gates required to create the adder.

Half-adder: The half adder is a combinational circuit that performs addition of two bits. It is designed conventionally by EXOR and AND gates. When two inputs A and B are added, the Sum and Carry outputs are produced according to the truth table.

The logic function for half-Adder is:

Sum = A’B+AB’
Carry = AB

The majority gate expression for above equation is:

Sum = M (M (A, B’, 0), M (A', B, 0), 1)
Carry = M ((A, B, 0)

Full-adder: The full adder circuit is implemented by digital logic gates. When three inputs A, B and C are added, the Sum and Carry outputs are produced according to the truth table. We have designed the full adder based on QCA addition Algorithm given in (Zhang *et al*., 2004; Wang *et al*., 2003).

QCA Addition Algorithm:
Majority logic of carry:

Cout = AB+BC+AC
= M (M (B, M(A,C,1) ,0) , M(A,C,0) , 1)
= M(A,B,C)

Majority logic of sum:

Sum= ABC+A'B'C+A'BC'+AB'C'

Direct implementation:

Sum = M(M(A,M(M(B,C,0), M(B',C',0) ,1, 0),

Majority gates:

M (A' M (M (B',C,0),1) ,0),1)
Reduction Technique:
Sum = ABCin+A'B'Cin+A'BC'in+AB'C'in

= (A.B + A'.B') Cin + (A'.B + A.B') C'in
= [A.B + A '.B ' + A.C ' in + A '.C ' in + B.C ' in + B ' C ' in] Cin +
(A '.B + A.B ')C ' in
= [(A '.B ' + A '.C ' in + B '.C ' in) + (A.B + A C ' in + B C ' in)] Cin +
(A '.B + A.B ')C ' in
= [(A '.B ' + A '.C ' in + B '.C ' in) + (A.B + A C ' in + B C ' in)] Cin +
(A.C 'in + B.C 'in) +(A '.Cin + B '.C 'in)
= [(A '.B ' + A '.C ' in + B '.C ' in) Cin + (A.B + A C ' in + B C ' in)] Cin
+ (AB + A.C 'in + B.C 'in) (A '.C 'in + B '.C 'in + A 'B ')
=M (A, ' B ',C ‘in) .Cin + M(A,B,C ' in). Cin + M(A ',B ',C ' in)
M(A,B,C ' in)
= M [M (A ',B ',C ' in), Cin, M(A,B,C ' in)]
Sum =M [C ' out, Cin, M (A, B, C ' in)] 3 majority gates

This reduction technique is used to reduce the number of majority gates from 11-3.

QCA implementation of adders: One bit full adder has been implemented in two different methods (Zhang *et al*., 2004; Wang *et al*., 2003). The first method was conventional (Direct implementation) and consumed a lot of hardware. The second method is (Majority gate reduction) simple and have less hardware requirement. Here we apply the Majority logic method for constructing QCA adders. The proposed adders are implemented with QCA cells and number of cells has reduced by cell minimization techniques. Hence our implementation further reduces the area and complexity.

Table 1: Comparision of adders

	Previous structure		Proposed structure		
QCA adders	Complexity	Area	Complexity	Area	Clock
Half adder	105 cells	300×360 nm^2	77 cells	297×280 nm^2	1
Full adder (using CC)	145 cells	439×367 nm^2	122 cells	381×300 nm^2	1
Full adder (using MC)	137 cells	435×300 nm^2	98 cells	360×280 nm^2	1
Full adder using half adder	218 cells	652×440 nm^2	192 cells	650×320 nm^2	2
Serial bit adder	158 cells	500×382 nm^2	140 cells	460×320 nm^2	1

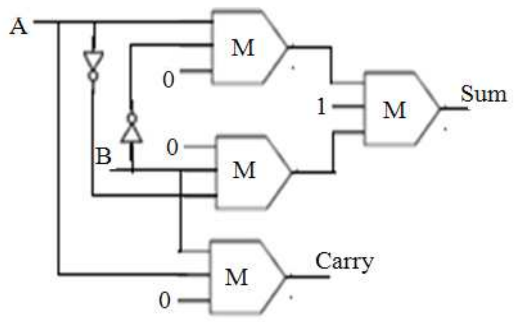


Fig. 8: Half adder schematic

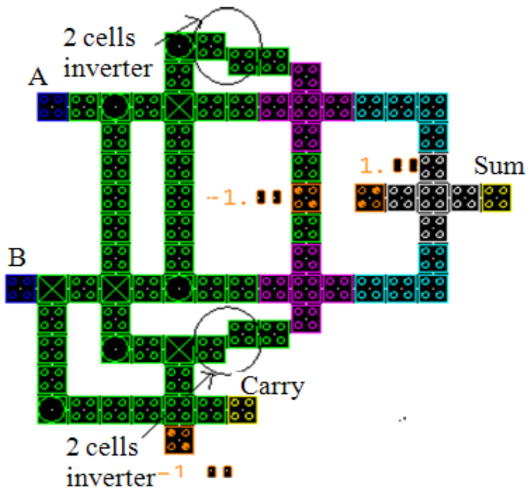


Fig. 9: Layout of half adder

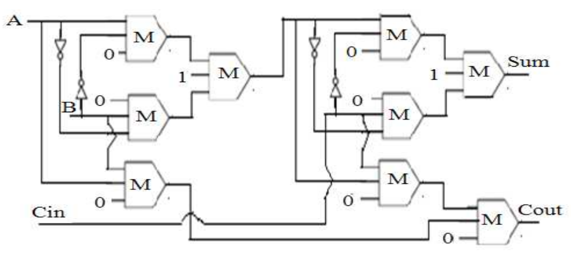


Fig. 10: Full adder schematic using half adders

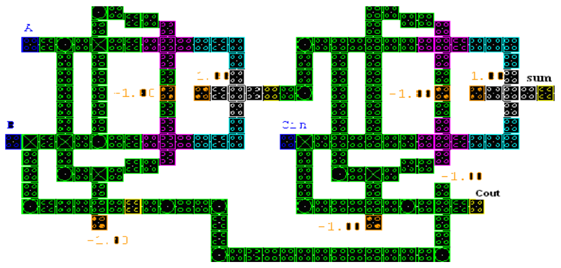


Fig. 11: Layout of Full adder using half adders

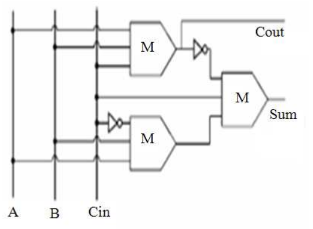


Fig. 12: Full adder schematic

The half adder is designed with 4 majority gates and 2 inverters as shown in Fig. 8. A one bit full adder circuit is constructed by the two half adder circuits and an OR gate. The full adder is designed with 9 majority gates and 4 inverters as shown in Fig. 10.

QCA implementation of the half adder and the full adder is shown in Fig. 9 and 11. The total number of cells required to implement a half adder is 77, with an area of 83160 nm^2 which is much lesser than the previous implementations. The previous implementation has 105 cells with an area of 108000nm^2. A one bit full adder QCA implementation in Fig. 11 requires 192 cells, with an area of 208000 nm^2.

The full adder is designed with 3 majority gates and 2 inverters as shown in Fig. 12. The majority gate design is implemented with two different crossovers as shown in Fig. 13 and 14. The QCA implementation of Fig. 13 required 192 cells, with an area of 114300 nm^2 and this also required less number of cells than previous implementations. The same full adder circuit is implemented with multilayer crossover as shown in Fig. 14. This also required less number of cells than previous designs which is mentioned in the comparison Table 1.

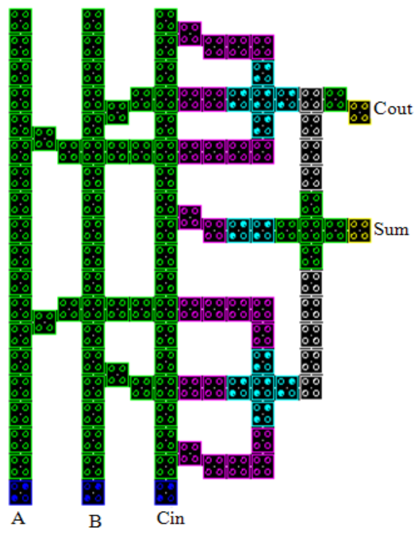


Fig. 13: Layout of full adder (Coplanar crossing)

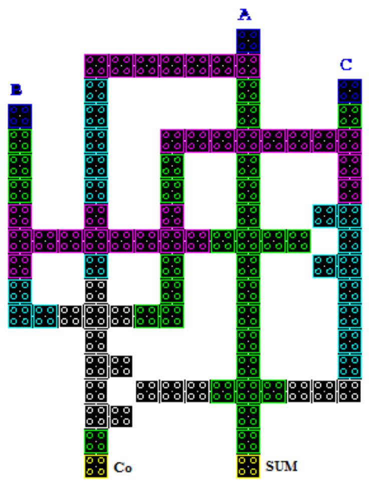


Fig. 14: Layout of full adder (Multilayer crossing)

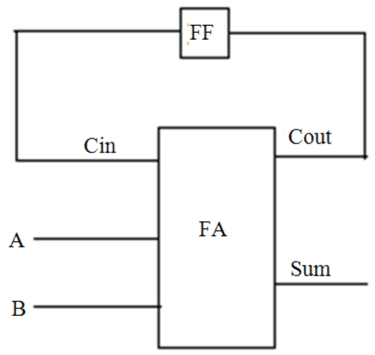


Fig. 15: serial bit adder schematic

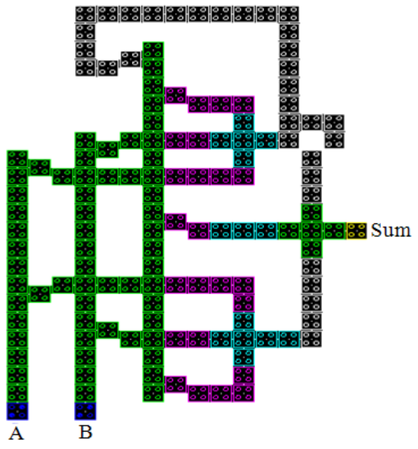


Fig. 16: Layout of serial bit adder

This design can be used to create a bit-serial adder by simply feeding the carry back into the adder (Walus *et al*., 2003). Bits from A and B are entered serially into the circuit, LSB first. The full adder adds the two bits, Ai and Bi, with the carry Ci which is saved from the previous bit calculation and produces the partial results Si+1 and Ci+1. Si+1 is sent to the output while Ci+1 is stored into the FF to be used in the next add cycle.

The serial bit adder is designed with 3 majority gates and 2 inverters. The schematic for this adder is shown in Fig. 15. The corresponding QCA implementation is shown in Fig. 16. The QCA implementation requires 140cells, with an area of 147200 nm^2.

RESULTS

With QCADesigner ver.2.0.3, the circuit functionality is verified. The following parameters are used for a bistable approximation: Cell size 20nm, Number of samples 12800, Convergence tolerance 0.001000, Radius of effect 65nm, Relative permittivity 12.9, Clock high 9.8e-22J, Clock low3.8e-23J, Clock amplitude factor2, Layer separation 11.5nm, Maximum Iterations per sample 10000.

The simulated waveform of half adder is shown in Fig. 17. The circuit has four clocking zones. Initially clock 0 is used to get the inputs A and B. Clock 1 is used to route inputs for majority gate logic, clock 2 is used for finding majority logic and clock 3 is used to compute output. The output is available at clock 0 again. Clock 1 to 3 considered here is a sequence of setup for hold, relax and release phase, to control the flow of information in QCA circuits.Similar to half adder, the full adder and bit serial adder also required 4 clock zones , used to produce required output. The simulated results of full adder and serial bit adder are shown in Fig. 18-20.

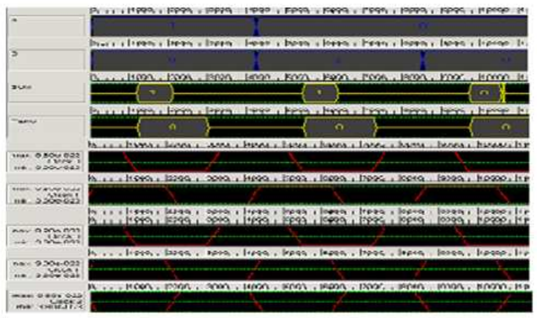


Fig. 17: Simulation Result of Half adder

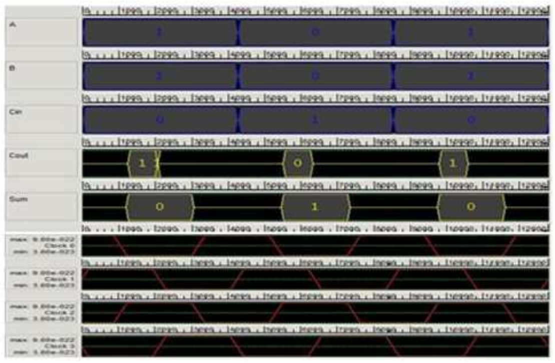


Fig. 18: Simulation result of full adder

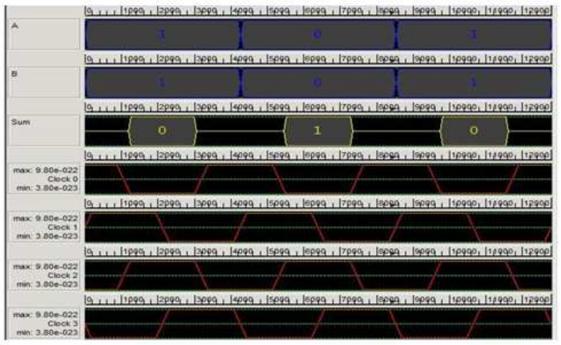


Fig. 19: Simulation result of bit serial adder

DISCUSSION

The proposed design is a solution for minimum number of QCA cells. This could be achieved by cell minimization techniques. From the Table 1 we observed that the proposed layouts are significantly smaller than the previous designs. The same full adder majority gate circuit is implemented by QCA cells using two different crossovers. The fuller results are compared with existing methods and are tabulated. The serial bit adder is implemented by full adder using coplanar crossings. The area and complexity of this adder is given in Table 1.

CONCLUSION

The different types of adder circuits have been designed and tested using QCADesigner software. The operation of these circuits has been verified according to the truth table. The performance analyses of those circuits are compared according to the complexity, area and number of clock cycles. The proposed layouts are significantly smaller than the circuits using CMOS technology and it reduces the area as well as complexity required for the circuit than the previous QCA circuits. The designed QCA circuits can be used to construct arithmetic and logical units and microprocessors. In future this can be extended to build nanocomputers.

REFERENCES

Cho, H. and E.E. Swartzlander, 2007. Adder design and analyses for quantum-dot cellular automata. IEEE Trans. Nano., 6: 374-383. DOI: 10.1109/TNANO.2007.894839

Kim, K., K. Wu and R. Karri, 2007. The robust QCA adder designs using composable QCA building blocks. IEEE Trans. Comput.-Aided Design Integrated Circ. Syst., 26: 176-183. DOI: 10.1109/TCAD.2006.883921

Reza, H. and A. Vafaei 2008. Quantum-dot cellular automata serial adder design exploiting Null conversion logic. World Applied Sci. J., 4: 655-660. ISSN: 1818-4952.

Wang, W., K. Walus and G.A. Jullien. 2003. Quantum-dot cellular automata adders. Proceedings of the 3rd IEEE Conference Nanotechnology, Aug. 12-14, IEEE, pp: 461-464. DOI: 10.1109/NANO.2003.1231818

Walus, K., G.A. Jullien and V. Dimitrov. 2003. Computer arithmetic structures for quantum cellular automata. Proceedings of the 37th Asilomar Conference on Signals, Systems and Computers, Nov. 9-12, pp: 1435-1439. DOI: 10.1109/ACSSC.2003.1292223

Walus, K., G. Schulhof, G.A. Jullien, R. Zhang and W. Wang, 2004a. Circuit design based on majority gates for applications with quantum-dot cellular automata. Proceedings of the Record 38th Asilomar Conference Signals, Systems and Computers, Nov. 7-10, IEEE, pp: 1354-1357. DOI: 10.1109/ACSSC.2004.1399374

Walus, K., T. Dysart, G. Jullien and R. Budiman, 2004b. QCADesigner: A rapid design and simulation tool for quantum-dot cellular automata. IEEE Trans. Nanotechnol., 3: 26-29. DOI: 10.1109/TNANO.2003820815

Zhang, R., K. Walus, W. Wang and G.A. Jullien, 2004. A method of majority logic reduction for quantum cellular automata. IEEE Trans. Nanotechnol., 3: 443-450. DOI: 10.1109/TNANO.2004.834177