# STUDY AND IMPLEMENTATION OF MUX BASED FPGA IN QCA TECHNOLOGY

E.N.Ganesh[1] / V.Krishnan[2]

*1. Professor, Rajalakshmi Engineering College 2. UG Student, Rajalakshmi Engineering College*

***ABSTRACT***

*This paper presents a simple Multiplexer based Field programmable gate arrays with interconnects based on quantum cellular automata technology. Quantum cellular automata (QCA) technology is a promising nanotechnology of the future. QCA based 4:1 Multiplexer are designed and constructed as a module in a FPGA. Multiplexer based designs are used to implement complex Boolean functions and each module can act as a logic element or simple Multiplexer. We have studied here, NOR based logic to implement Sum function of an adder in a QCA FPGA. Finally we have designed and simulated the MUX based logic elements to construct QCA FPGA. This study can be useful for building complex Configurable logic blocks to design a complete FPGA.*

***Keywords***

*Quantum dot cellular automata, FPGA, Multiplexer*

## 1. QUANTUM DOT CELLULAR AUTOMATA

Quantum-dot Cellular Automata (QCA) is an emerging technology that offers a revolutionary approach to computing at nano-level [1]. Quantum dots are nanostructures created from standard semi conductive materials. These structures are modelled as quantum wells. They exhibit energy effects even at distances several hundred times larger than the material system lattice constant. A dot can be visualized as well. Once electrons are trapped inside the dot, it requires higher energy for electron to escape. The fundamental unit of QCA is a QCA cell created with four quantum Dots positioned at the vertices of a square. [2] [3.]. The electrons are quantum mechanical particles and they are able to tunnel between the dots in a cell. The electrons in the cell that are placed adjacent to each other will interact; as a result the polarization of one cell will be directly affected by the polarization of its neighbouring cells.

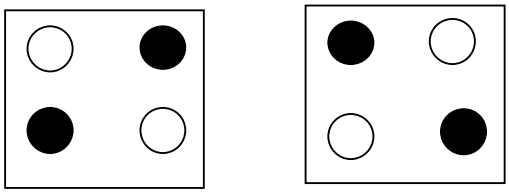

Fig.1 QCA cells with four quantum dots.1.a P = +1 (Binary 1) 1.b P = -1 (Binary0) [1][3][4][5]

DOI : 10.5121/ijdps.2011.2305 

Table 1 – Majority voting scheme [4] [5]

| INPUT | OUTPUT MAJORITY VOTING |
|---|---|
| 000 | 0 |
| 001 | 0 |
| 010 | 0 |
| 011 | 1 |
| 100 | 0 |
| 101 | 1 |
| 110 | 1 |
| 111 | 1 |

This interaction forces between the neighbouring cells which is able to synchronize their polarization. Therefore an array of QCA cells acts as wire and is able to transmit information from one end to another end [6] [7]. To perform logic computing, we require universally a complete logic set, we need a set of Boolean logic gates that can perform AND, OR, NOT and FANOUT [8] operations. The combination of these is considered as universal because any general Boolean function can be implemented with the combination of these logic primitives. The fundamental method for computing is majority gate or majority voter method [4] [10]. Suppose three inputs are given to QCA circuit, then the output of the QCA structure is tabulated in table1.The majority gate produces an output that reflects the majority of the inputs. The majority function is a part of a larger group of functions called threshold functions. Threshold functions works according to inputs that reaches certain threshold before output is asserted. The majority function is most fundamental logic gate in QCA circuits [9]. In order to create an AND gate we simply fix one of the majority gate input to 0 (P = -1). To create OR gate we fix one of inputs to 1 P = +1. The inverter or NOT gate is also simple to implement using QCA. If we place two cells at 45 degrees with respect to each other such that they interact inversely. The output of majority AND gate reflects the majority of the inputs. Suppose input A =1, B = 1, Control input 0(-1), the output is equal to 1.

## 1.1 QCA CLOCKING

Clocking is the requirement for synchronization of information flow in QCA circuits. It requires a clock not only to synchronize and control information flow but clock actually provides power to run the circuit [11] [12] [13].

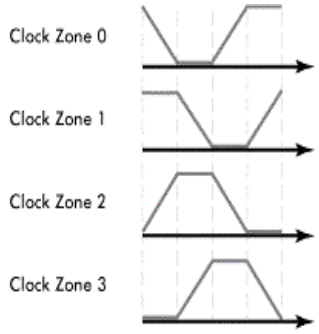


Fig.2 shows clocking scheme of QCA circuits [4][5][9][12.]

The cells are not powered from any other external source apart from the clock. These clocks have been proposed to control the potential barriers between the dots. When the clock signal is high the potential barriers between the dots are low and electrons effectively spread out in the cell and no net polarization exists [14]. As the clock signal is switched low, the potential barriers between the dots are raised high and the electrons are localized such that a polarization is developed based on the interaction of their neighbors [15.] In short when clock is high cell is unlatched and when clock is low cell is latched. In order to pump information down a circuit in a controllable manner four clocking zones are available as shown in Figure 2. Each of clocking signal lagging in phase by 90 degrees with respect to one before, in this way, the cells are latched in series and propagate information in the same direction. So clocking is essential for QCA circuits [17] [18]. Thus QCA accomplishes logical operations and data movement via Columbic interaction rather than electric current flow. This paper discusses about field programming gate arrays using Multiplexer based on Quantum cellular automata architecture (QCA). Field programmable gate array is a flexible architecture for Programmable logic device [21] [22].

FPGAs represent an attractive application of QCA technology [23], their relatively homogeneous structure is well suited to fabrication at the nanoscale level. Furthermore, the general-purpose nature of FPGAs could allow many applications to be implemented using QCA. Previous work on QCA-based FPGAs [11][24] has focused on programmable interconnect. In contrast, this paper focuses on interconnect and Field programmable logic. The goal of this work is to design and evaluate simple FPGA structure using 4:1 Multiplexers. All the circuits are simulated using QCADESIGNER tool [16] [20].Interconnection networks for traditional CMOS FPGAs utilize memory to configure the pass transistors which regulate the flow of information between programmed logic elements [24]. In QCA there are no direct counterparts to Memory or pass transistors, and keep the fabrication complexity to a minimum with the logic elements which are static.

## 2. FPGA –NOR LOGIC ELEMENTS

Routing and logic element together constructs NOR logic based FPGA. We propose simple FPGA as shown in figure 3 to implement sum logic of an adder. Figure 4 shows the QCAFPGA with sum function using NOR logic. Let A and B be the input of 3 X 3 QCA FPGA and Sum and Sum' be the output, first module act as a NOR logic, second module as Not logic using same NOR cells. So the sum and NOR logic function can be taken as output. Figure 5 shows the simulated waveform of QCAFPGA Nor logic for implementation of sum function.

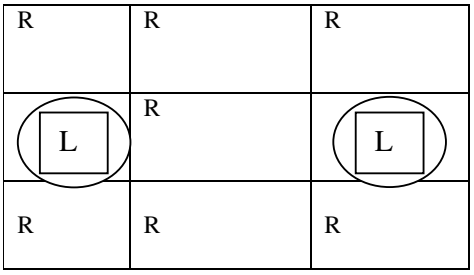


Fig. 3 FPGA of Sum two input elements a and b, R- Routing elements and L- logic elements

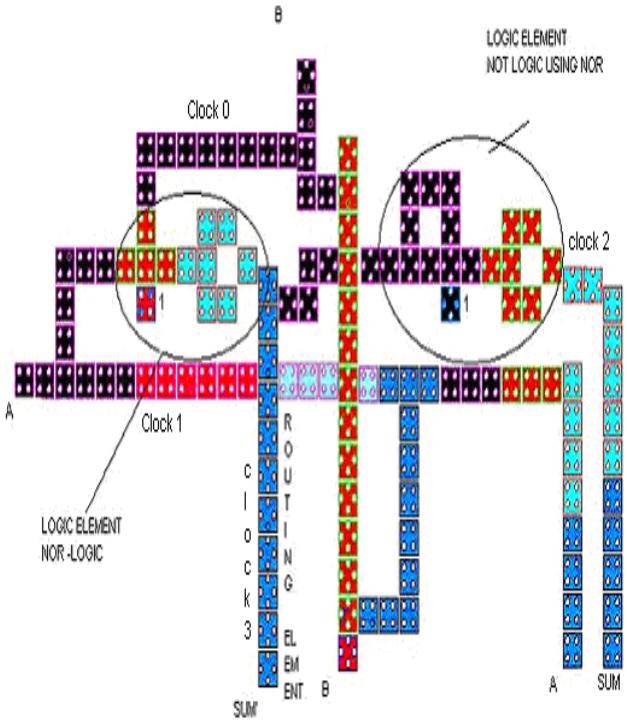


Fig. 4 QCA FPGA with sum function using NOR logic

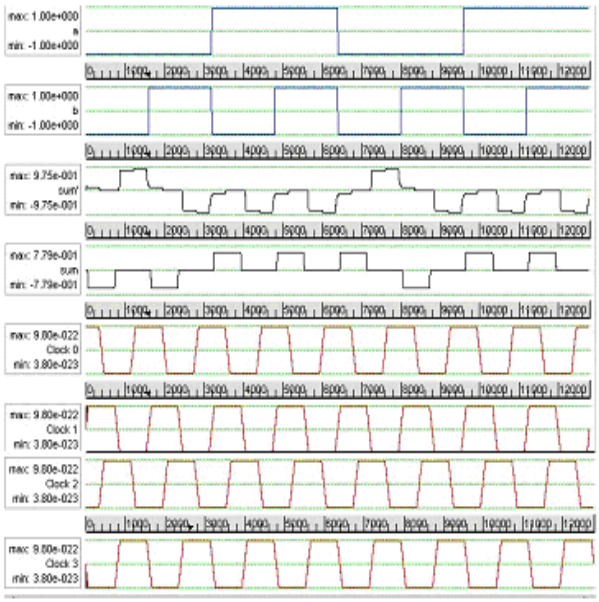


Fig.5 NOR logic implementation of QCA FPGA for sum function.

The output sum and sum' (NOR) is evaluated at clock 3. We conclude that simple FPGA can be constructed using routing and logic elements. This study is helpful for designing FPGA nanocircuits. The only drawback with these FPGA are MUX based FPGA requires the control signals to be stored while fabricating it or separate two routing lines to be fixed for control signals and also the clocking phase to be routed properly. To avoid these limitations, minimum number of logic elements with fixed clocking phase and separate memory for each logic

elements could be implemented. This study can be useful to built configurable logic blocks to construct and design FPGA based on Nanocircuits.

## 2.1. MUX based FPGA

It is considered here Multiplexer based FPGA to construct basic Boolean functions. Each Module has single 4:1 Multiplexer to select and route either horizontal or vertical lines. The overall logic expression for a single module can be written as

$$Y\ (output) = (A.Sx' + B.Sx)\ (S0 + S1)' + (C.Sy + D.Sy')\ (S0 + S1) \quad (3)$$

$$= A.Sx'.S0'S1' + B.Sx.S0'S1' + C.Sy.S0 + C.Sy.S1 + D.Sy'S0 + D.Sy'S1 \quad (4)$$

By connecting the input of this module in different ways, the module can be programmed as a variety of 2 or 3 or 4 input gates. For Example, by connecting A,B,C and S1 to 0, the expression for output y becomes

$$Y = D.Sy'S0 \quad (5)$$

$$Y = m(m(D,Sy',0), S0,0) \quad (6)$$

Equation 5 acts as a 3 input logic AND gate. Similarly, by connecting different inputs to high and low, the module can be designed for more than one logic gate. Using the above logic QCA Multiplexer is constructed using majority logic as shown in figure 6. Four phase clocks are used to construct QCA MUX, it has 6 AND gates and 4 OR gates. Clock 0 is used to set all the inputs and select lines availability, clock 1 is used to evaluate Majority function of AND gates in First and Second MUX, Clock 2 is used for OR gate majority logic in MUX 1 and MUX2 as well as control line selection for third MUX, clock 3 is used to evaluate And gates of MUX 3, again clock 0 is used for OR gate output of MUX3. So the output y is from third MUX. Figure 6 and 7 shows the simulated waveform of QCA 4: 1 MUX.

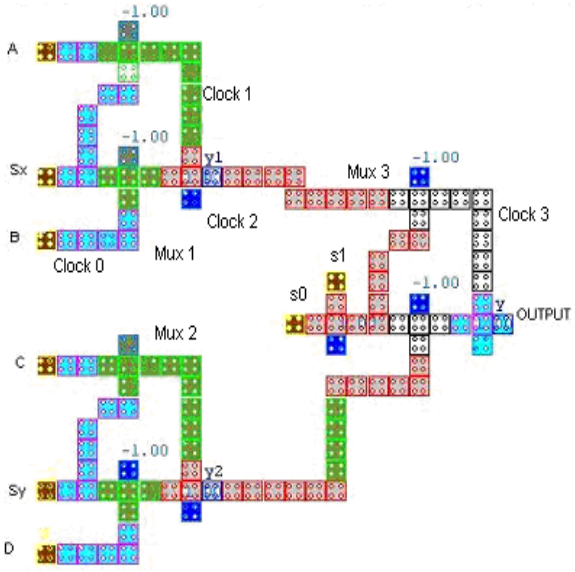


Fig.6 QCA cell 4:1 MUX

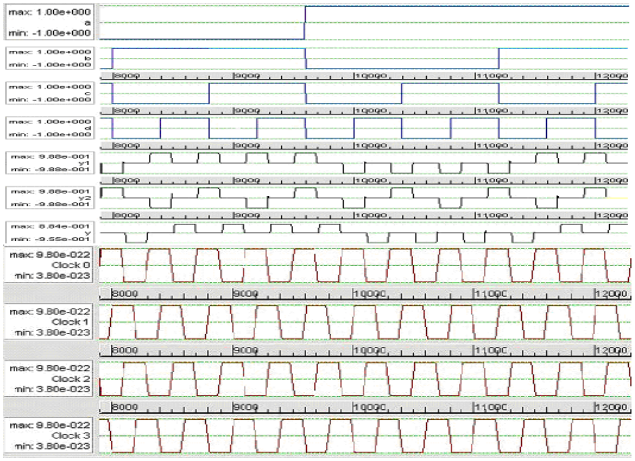


Fig.7 simulated waveform of QCA 4:1 MUX. Output Y is at Clock 0. with Sx = 1 and S0 = 1, Y = b.

Let Y be the output which is same as that of input b by setting select lines Sx and S0 as 1, similarly other functions can be performed using this module. Figure 8 shows the simulated waveform of equation 5 and 6. The output Y is high if all the inputs are high, remaining bits are set low in a QCA MUX diagram.

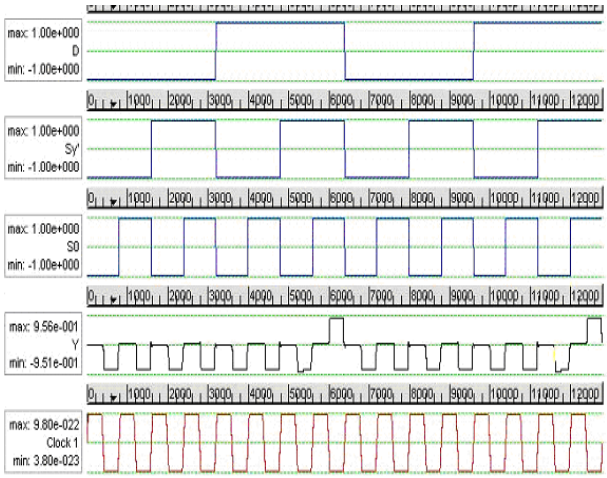


Fig.8 simulated waveform of equation 5 and 6 with setting D, Sy' and S0 as high to get 3 input AND gate.

Thus the single module can be used as logic gate as well as MUX for routing. As shown in figure 9, four such MUX can be used to construct simple QCA FPGA. Figure 9 shows the MUX based 2 X 2 QCA FPGA.

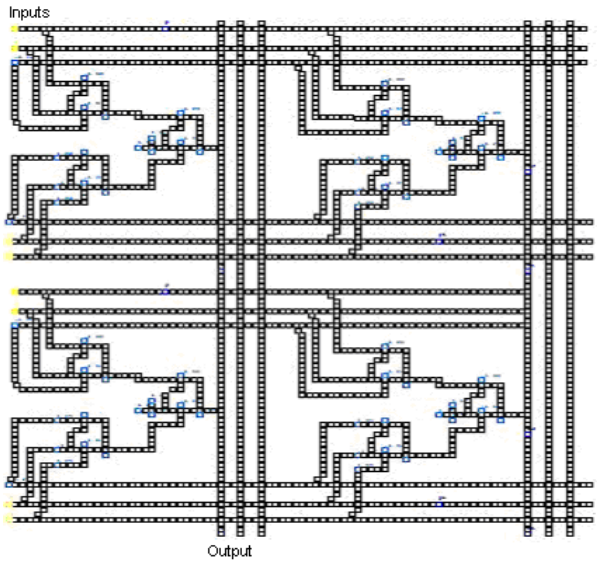


Fig.9 shows the 2 X 2 QCA FPGA using 4 : 1 MUX

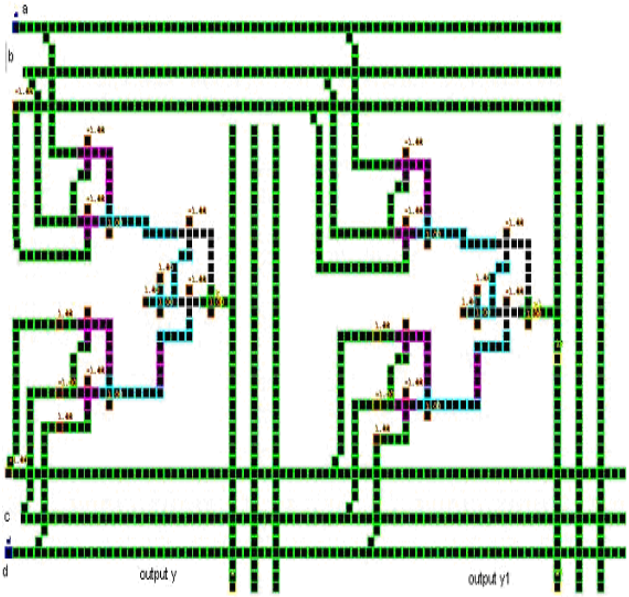


Figure 10 shows the mux 1 and 2 in 2X 2 QCA FPGA.

We have designed 2 X 2 QCA FPGA with horizontal lines for input feeding and vertical lines for getting the output. We have not programmed the interconnect lines. Due to the advantage of QCA technology feeding and crossing through a single line is possible by making one of its lines either a crossed line or tapped line in inverted mode. The programming mode of interconnect is also possible. Let a,b,c,d are the inputs running horizontally to all the MUX in the modules, the output with its required logic function can be tapped from the vertical lines. The horizontal lines are rotated to $45^0$ so that at crossing the horizontal lines won't interface with vertical lines. So it is easy to route input and output separately, even we can interchange the vertical and horizontal lines as input and output. We have shown here input are in horizontal and output taken from vertical lines. By changing the required input lines to high or low the different logic function can be obtained as explained before. Figure 11 shows the simulated waveform of 2 X 2 QCA FPGA. We have designed for getting output y as signal a and output

y1 as signal d at first clock phase. The simulated waveform shows output y resembles as input signal a and output y1 as input signal d waveforms. Each MUX module is designed using Majority logic method; it requires minimum of ten majority gates to implement. The nature of QCA technology is the availability of inverted signal in a wire which can be used for switching without need of designing switching element. We can also design separate switching element to control input lines or lines from the modules to route to proper output. We have simulated QCA FPGA structure to act as SUM as well as simple Multiplexer. In the next section, Circuit design of MUX based logic gates using NOR QCA gates is considered.

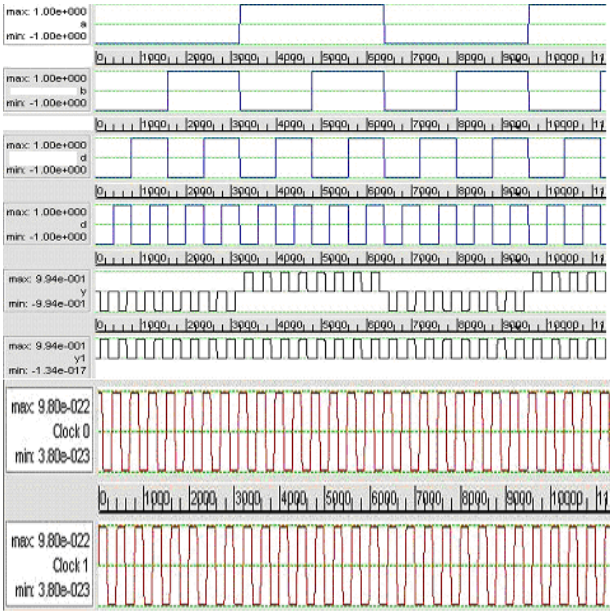


Fig.11 Simulated waveform for 2 X 2 QCA FPGA.

## 3. DESIGN OF QCA MUX IN FPGA MODULE

Equation 3 and 4 gives Mux based Boolean expression in which we select the input and control lines to get required logic. For example setting a,c,d ,sy,s0 and s1 as zero and B,Sx as one we get

$$Y = B.Sx.S0'S1' \qquad (7)$$

In majority logic method, equation 7 can be written as

$$Y = m(m(B,Sx,0),m(S0',S1',0),0) \qquad (8)$$

Equation 8 gives as simple AND logic function, when we change the control input as 1 for the first majority gate we can get OR logic, equation 8 can be written as

$$Y = m\,(m(B,Sx,1),m(S0',S1',0),0) \qquad (9)$$

When we feed AND / OR logic to $45^0$ degree inverted vertical line we can get NAND / NOR logic. Inverted wires can be used to select the universal gate structure; thereby required logic can be obtained. We have simulated all the QCA circuits with the parameters considered in table 2, except no of inputs and outputs, area, no of cells and time for simulation will change according to the circuit considered. For the MUX based module, no of inputs be 8 (4 inputs, 4 control lines) and no of output be 1, 138 QCA cells are considered with area of 0.14 $\mu m^2$, time

for simulating the circuit is 0.8 second ( 0.8 ns). We used bistable method of approximation to polarize the electrons as in [7]. Further QCA FPGA CLB [25] can be designed using the proposed QCA FPGA Module efficiently. Efficiently in terms of we conclude this session by saying mux based design gives easy implementation of FPGA, further if we consider proper switching elements and memory along with the mux, complex FPGA circuits can be designed and simulated.

Table 2 Design parameters to design a module using QCAdesigner tool.

| S.NO | PARAMETERS FOR SIMULATION | VALUE |
|---|---|---|
| 1 | No of samples taken | 12,800 |
| 2 | Method for simulation | Bi-stable approximation |
| 3 | Radius of effect | 65 nm |
| 4 | Relative permittivity | 12.9 |
| 5 | Clock High | $9.8 \times 10^{-23}$ |
| 6 | Clock Low | $3.8 \times 10^{-22}$ |
| 7 | Clock amplitude factor | 2 |
| 8 | Layer separation | 11.5nm |
| 9 | Maximum separation per sample | 100 |
| 10 | No of cells used Per module | 138 |
| 11 | Area in micro meter | 0.14 $\mu m^2$ |
| 12 | Total simulation time | 1 s (ns) |
| 13 | No of inputs | 2(a,b), control inputs (Sx,Sy) |
| 14 | No of outputs | 1 |
| 15 | No of Majority gates | 10 ( 6 AND 4 OR gates) |
| 16 | Clock at which output is activated | Clock 0 |
| 17 | Temperature at which simulation performed | 7K |

## 4. CONCLUSION

In conclusion we have studied programmable interconnect structure for constructing Field Programmable Gate arrays, 2X2 and 3X3 interconnects are studied and how it is routed to a module in a FPGA. We also simulated neither simple NOR based sum logic as logic element in FPGA modules and the results show promising FPGA nanocircuits. We also designed and studied MUX based FPGA module for implementation of logic gates and Boolean functions. This study can be useful for simulating complex configurable logic blocks (with LUT and memory along with MUX) for a complete FPGA design.

## REFERENCES

[1].K.Walus, Wei Wang and Julliaen *et al*, "Majority logic reduction for Quantum Cellular Automata" in *Proc IEEE Nanotechnology conf, vol 3 December 2004*.

[2] K.Walus, Wei Wang and Julliaen *et al*, "Quantum Cellular Automata adders" in *Proc IEEE Nanotechnology conf, vol 3 page461-463December 2004*.

[3] K.Walus, Schulaf and Julliaen et al, "High level Exploration of Quantum Dot Automata" in Proc IEEE Nanotechnology conf, vol 2,page 30- 33 2004

[4] K.Walus, Schulaf and Julliaen et al, "Circuit design based on majority gates for application with Quantum dot cellular automata" in Proc IEEE Nanotechnology conf, vol 4,page 1350- 1354, 2004.

[5] K.Walus, Dimitrov and Julliaen *et al*, "Computer Architecture structure for Quantum Cellular Automata"in *Proc IEEE Nanotechnology conf, vol 3,page 1435 – 1439 2003*

[6] K.Walus, Dysart and Julliaen *et al*, "QCQ Designer A Rapid design and simulation tool for quantuim dot cellular automata" in *IEEE transactions on Nanotechnology conf, vol 3,No – 2 June 2004.*

[7] K.Walus, Dysart and Julliaen *et al*, "Split current Quantum dot cellular automata modeling and simulation " in *IEEE transactions on Nanotechnology conf, vol 3 March 2004.*

[8] A. Vetteth *et al*., "Quantum dot cellular automata carry-look-ahead adder and barrel shifter," in *Proc. IEEE Emerging Telecommunications Tech-nologies Conf*., 2002.

[9] , "RAM design using quantum-dot cellular automata," in *Proc.2003 Nanotechnology Conf*., vol. 2, 2003, pp. 160–163.

[10] C. S. Lent and P. D. Tougaw, "A device architecture for computing with quantum dots," *Proc. IEE*E, vol. 85, no. 4, pp. 541–557, 1997.

[11] W. Porod, "Quantum-dot devices and quantum-dot cellular automata," *Int. J. Bifurcation Chaos*, vol. 7, no. 10, pp. 2199–2218, 1997.

[12] I. Amlani *et al*., "Experimental demonstration of a leadless quantum-dot cellular automata cell," *Appl. Phys. Lett*., vol. 77, no. 5, pp. 738–740,

2000.

[13] A. Orlov *et al*., "Experimental demonstration of clocked single-electron switching in quantum-dot cellular automata," *Appl. Phys. Lett*., vol. 77,

no. 2, pp. 295–297, 2000.

[14] I. Amlani *et al*., "Digital logic using quantum-dot cellular automata," *Scienc*e, vol. 284, pp. 289–291, 1999.

[15] A. Orlov *et al*., "Experimental demonstration of a binary wire for quantum-dot cellular automata," *Appl. Phys. Lett*., vol. 74, no. 19, pp.

2875–2877, 1999.

[16] K. Walus. (2002) ATIPS Laboratory QCADesigner Homepage. ATIPS Laboratory, Univ. Calgary, Calgary, Canada. [Online]. Available:

http://www.atips.ca/projects/qcadesigner

[17] G. Toth, "Correlation and Coherence in Quantum-Dot Cellular Au-tomata," Ph.D. dissertation, Univ. Notre Dame, Notre Dame, IN, 2000.

[18] G. Toth and C. S. Lent, "Quasiadiabatic switching for metal-island quantum-dot cellular automata," *J. Appl. Phys*., vol. 85, no. 5, pp. 2977–2984, 1999.

[19] K. Hennessy and C. S. Lent, "Clocking of molecular quantum-dot cellular automata," *J. Vac. Sci. Technol. B*., vol. 19, no. 5, pp. 1752–1755,

2001.

[20] www.qcadesigner.ca.

[21] S. Brown and Z. Vranesic. *Fundamentals of Digital Logic with VHDL Design*. McGraw-Hill Science/ Engineering/Math, second edition, July 2005.

[22] W. Wolf. *FPGA-Based System Design*. Prentice Hall PTR Upper Saddle River, NJ, USA, 2004.

[23] M. Niemier, A. Rodrigues, and P. Kogge. "A potentially implementable FPGA for quantum dot cellular automata". In *First Workshop on Non-Silicon Computation (NSC-1)*, Boston, MA, 2002. .

[24] M. T. Niemier and P. M. Kogge. The "4-diamond circuit" — a minimally complex nano-scale computational building block in QCA. In *VLSI, 2004. Proceedings. IEEE Computer society Annual Symposium on*, pages 3–10, 2004

[25] M.A.Amiri, M.Mahdavi, S.Mirzakuchaki "QCA Implementation of MUX based FPGA CLB" Proceedings of the *International Conference on* In Nanoscience and Nanotechnology, 2008. ICONN 2008.VOL I, pp. 141-144.