ISSN 1512-1232

IMPLEMENTATION OF PROGRAMMABLE LOGIC DEVICES IN QUANTUM CELLULAR AUTOMATA TECHNOLOGY

Dr.E.N.Ganesh

Professor ECE Department REC Chennai, INDIA
Email : enganesh50@yahoo.co.in

Abstract

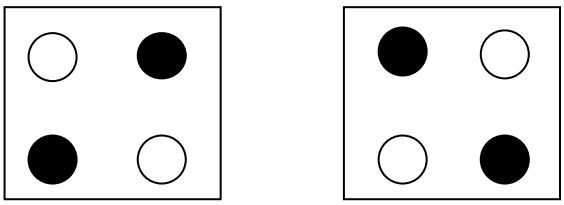
Quantum cellular automata technology is one of the alternative nano-scale computing technologies in future. QCA circuits can be used to study and design programmable logic devices. This paper discusses about QCA based simple Programmable logic devices of AND –OR function circuit, Fixed OR and programmable AND QCA circuit, Programmable AND and OR function circuits and finally simple Programmable logic array structure to implement XOR logic. These structures can be implemented and programmed easily there by giving easy defect detection and building of successful nano – scale circuits. We have designed and simulated PLD QCA devices using QCA designer tool and this study useful for designing complex Field Programmable logic devices at nanoscale.

Keywords*: Quantum cellular automata, Programmable Logic devices, Programmable Logic arrays, Majority logic method, Field programmable logic devices.*

1. Introduction.

1. a 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 modeled 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 QCA cell created with four quantum Dots positioned at the vertices of a square. [2] [3.]. The electrons are quantum mechanical particles; 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 neighboring cells. Fig 1 below shows quantum cells with electrons occupying opposite vertices.



1.a 1.b

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

This interaction forces between the neighboring cells 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 [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

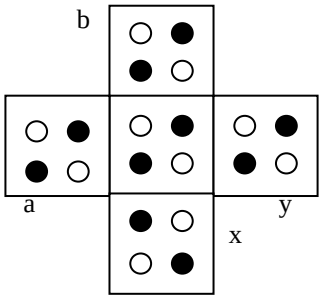
ISSN 1512-1232

implemented with the combination of these logic primitives. The fundamental method for computing is majority gate or majority voter method [1] [4]. Suppose three inputs are given to QCA circuit, then the output of the QCA structure is tabulated in table1.

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

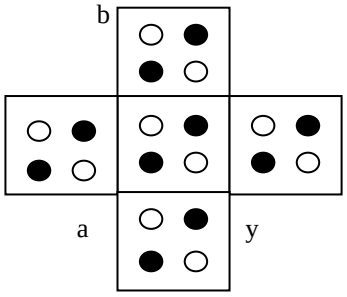
INPUT	OUTPUT MAJORITY VOTING
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

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. 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.



Control I/P
Fig 2.Majority AND gate [3] [6]

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.



Control input
Fig 3 Majority OR gate [6] [8] [10] [11]

ISSN 1512-1232

I b. QCA Clocking.

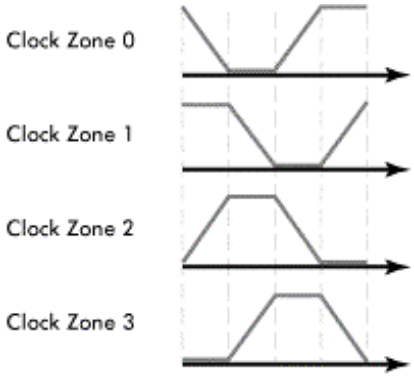


Figure 4 shows clocking scheme of QCA circuits [4][5][12.]

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]. 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 -15]. 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 [14.] 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 4. 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. Thus QCA accomplishes logical operations and data movement via Columbic interaction rather than electric current flow [9].

This paper discusses about Programming logic array structures (PLA) and based on Quantum cellular automata architecture (QCA). Using PLD based on QCA cells allows the defects present in the circuit can be readily detected and PLD structures are reprogrammable. DeHon and Wilson proposed a nano-wire-based sublithographic PLA design [16.]. Likharev and Strukov presented CMOL FPGA [17]. Graunke showed interesting algorithmic approaches to map logic functions to PLAs with defective crossbar switches, though many papers have been published on designing QCA-based circuits, only a few have examined design issues with programmable structures that use QCA cells [18], [19], [20]. However, these designs either are difficult to implement. We proposed a simple PLD structure with programmable select line, AND, OR function. Using select lines we can program the basic PLD structure to work as a logic function or a simple wire. We have implemented simple XOR function using PLA grid structure. Following this introduction, section II describes simple AND – OR Majority function and simulation of PLD structure, section III shows Fixed OR plane QCA simulation, section IV shows Programmable OR and AND function, finally section V shows Programmable logic array QCA structure and its simulation. We have simulated the Programmable logic structures using QCADESIGNER tool [21]

II Simulation of QCA AND – OR PLD.

We havel considered here simple PLD (AND – OR function) structure. Let A, B, C are the inputs for QCA PLD device, two AND majority gates are constructed with control input of 0 (-1) and output of and gates are given to a OR gate to get XNOR function. The main advantage in QCA technology is the signal input and their inversion available due to rotation of input cells and tapping the signal and its inversion depending on the structure orientation as shown in figure 6. Figure 5 shows the logic representation of QCA PLD structure. Here we have shown XNOR

ISSN 1512-1232

implementation, Let o and o1 be the output of Majority AND gates and or1 be the output of OR gate for implementation of XNOR function. Similarly the simple Boolean function (bc + c') can be implemented as shown in figure 6 as or2 its output. We can program the control input (0 or 1) of all the majority gates to get AND or OR gate. We have simulated and designed the simple PLD structure using QCADESIGNER tool [21]. The parameters of interest in this simulation are tabulated in table 2.

Table 2 design parameters for designing Simple PLD structures

PARAMETERS FOR SIMULATION	VALUE
No of samples taken	12,800
Method for simulation	Bi-stable approximation
Radius of effect	65 nm
Relative permittivity	12.9
Clock High	9.8×10^{-23}
Clock Low	3.8×10^{-22}
Clock amplitude factor	2
Layer separation	11.5nm
Maximum separation per sample	100
No of cells used	140
Area in micro meter	0.16 μm^2
Total simulation time	5 s (ns)
No of inputs	3
No of outputs	2
No of Majority functions	5 (2 And, Or, XNOR, Boolean function)
Clock at which output is activated	Clock 3
Temperature at which simulation performed	7K

Equation 1 and 2 gives simple Majority And ,OR function

$$AND = a.b = m(a, b, 0) \quad (1)$$

$$OR = a + b = m(a, b, 1) \quad (2)$$

The majority logic of XNOR and Boolean functions can be written mathematically as

$$Or1 = ab + a'b' \quad (3)$$

$$Or1 = m(m(a, b, 0), m(a', b', 0), 1) \quad (4)$$

M(a,b,0) and M(a',b',0) are AND gates, Or1 as XNOR gate as in equation 4.

$$Or2 = bc + c' \quad (5)$$

$$Or2 = m(c', m(b, c, 0), 1) \quad (6)$$

Equation 6 gives Boolean function of bc + c'

Table 3 gives binary values of XNOR function and Boolean function. It requires minimum of 5 majority function to implement PLD structure. We program the majority gates as OR or AND

ISSN 1512-1232

gate for a simple PLD structure with a separate control line can also be provided for providing control inputs. In QCA technology when the clock is low, the information can be latched, so four clocks are necessary for propagation of information. In the above considered PLD structure, inputs are available at clock 0, clock 1 is used to find o, o1 and o2. Clock 3 is used to find or1 and or2. At clock3 XNOR and Boolean function can be evaluated. Figure 7 shows the simulated waveform of QCA AND – OR structure.

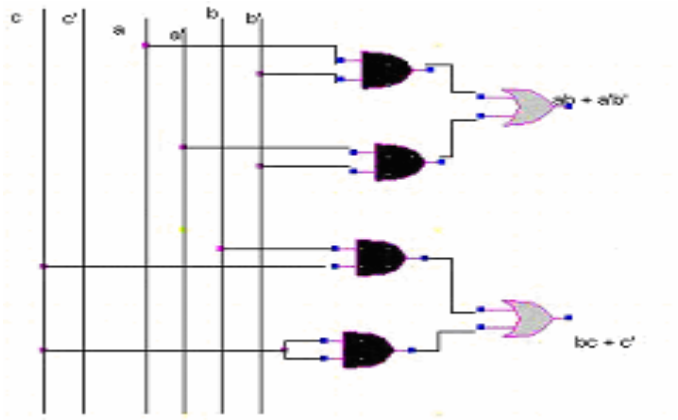


Figure 5 Logic representation of QCA AND – OR PLA structure implementation of XNOR and Boolean function (bc+c').

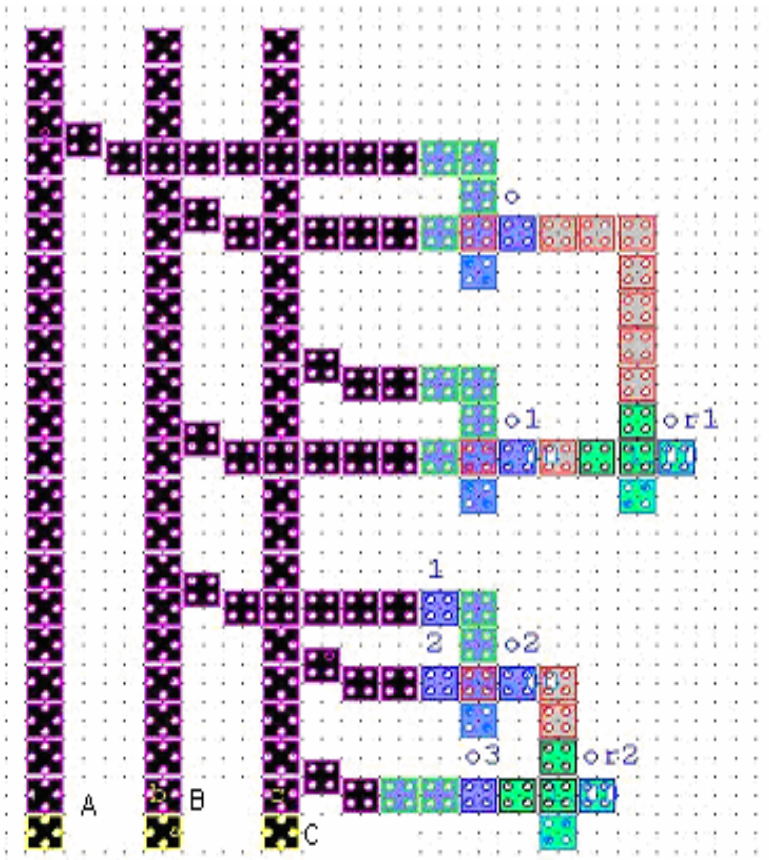


Figure 6 QCA AND –OR PLD implementation

ISSN 1512-1232

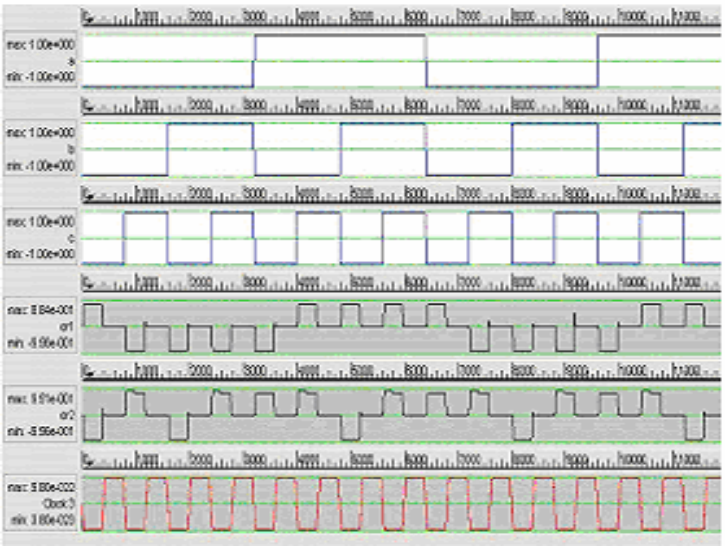


Figure 7 Simulated waveform for QCA AND – OR PLA implementation

III Simulation of Fixed OR gate – Programmable AND gate

Next we will consider about fixed no of OR gates and Programmable AND gate to implement any logic function for the given Inputs. The advantage of this logic PLD is OR gates are fixed (Control input is 1 for OR gate), so OR plane acts as a simple wire logic, there by ensuring easy taping of output from AND planes. By changing the AND plane control input AND planes can be converted to OR plane or vice versa. Control input can be fed to the AND, OR Planes like inputs A and B. Figure 8 shows the logic diagram of Fixed OR gate QCA cell. Figure 9 shows the QCA PLD cell representation. Figure 10 shows the simulated waveform of QCA PLD structure.

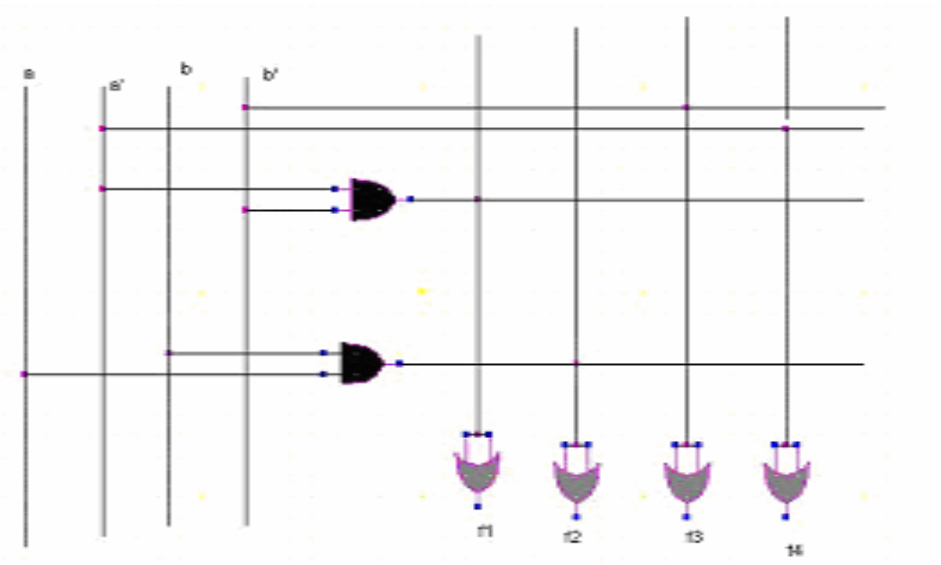


Figure 8 logic representation of Fixed OR gate QCA cell

ISSN 1512-1232

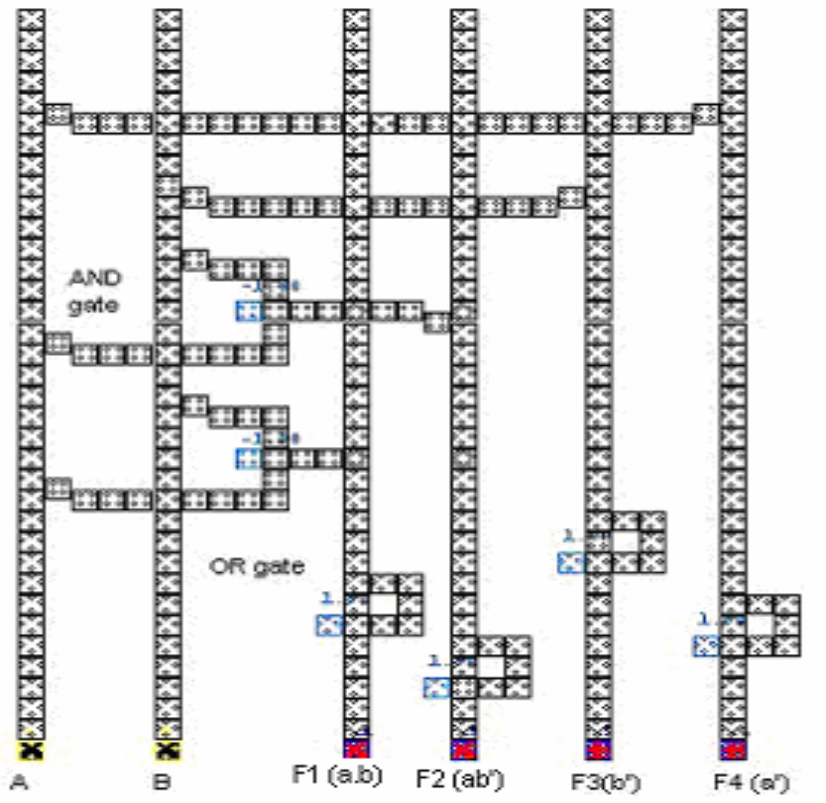


Figure 9 PLD QCA cell of FIXED OR plane QCA cell

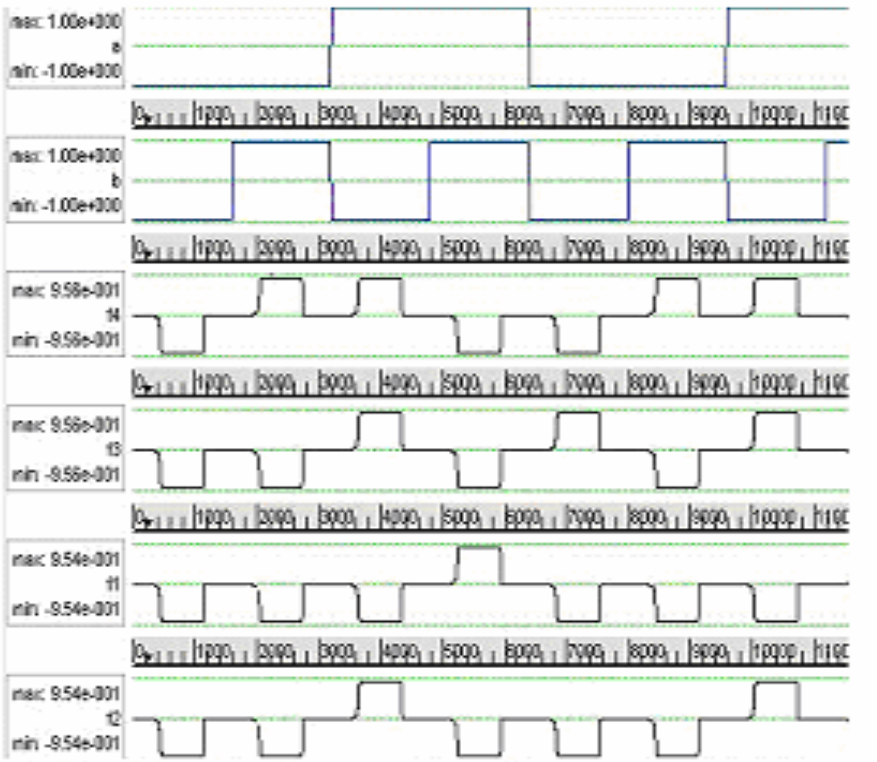


Figure 10 simulated waveforms of. PLD QCA – Fixed OR plane cells.

Control input to AND plane can be changed to 1 so that fixed or and programmable or gates can be constructed. Here we have considered fixed OR and programmable AND planes. Or planes ensure that correct output tapped from programmable AND gates. We have shown the following functions for the circuit considered.

ISSN 1512-1232

$$F1 = a.b = a.b + a.b \quad (7)$$
$$F1 = a.b = m(m(a,b,0),m(a,b,0),1) \quad (8)$$

$$F2 = ab' = a.b' + a.b' \quad (9)$$
$$F2 = ab' = m(m(a,b',0),m(a,b',0),1) \quad (10)$$

$$F3 = b' \quad (11)$$

$$F4 = a' \quad (12)$$

For the design and simulation of the equations above we have considered the same parameters as in table 2. But the area and the speed differ slightly. Figure 10 shows the simulated waveforms for the function considered. Function F1 gives 'ab' function (majority of a, b and 0 (control input of AND gate)), F2 gives 'ab'', F3 gives b' and F4 gives a'. All QCA PLD are clocked circuits, inputs are available at clock 0, at clock 1 programmable AND gate is determined and at clock3 fixed or gate is evaluated. So at clock3 output function F1, F2, F3 and F4 are found. AND gates are named programmable since same gates are used as OR gate. If the control input is simply changed as one of the input, the majority gate can act as simple latch circuit. A separate control wire (like one of the input wire) can be provided for evaluating simple AND gate or Latch circuit. Hence the circuit considered can be programmed to have simple majority AND gate. The no of cells considered in our simulation are 332, area required for our simulation is 1.5 μm^2, speed for simulating circuit is 7 seconds (7 ns),no of inputs considered are two (a and b) and no of outputs are four and operated at temperature 7k.

IV Simulation of Programmable OR and AND QCA.

Figure 11 shows the programmable AND and OR gate QCA structure. Two separate control lines are provided (+1 and -1) along with the inputs as shown in figure 11, Programmable AND gates are provided near to the inputs and b, Programmable OR gates are used to get the output. We have considered here four functions F1, F2, F3 and F4 for this structure. The parameters considered for the design and simulation are same as that of table 2. No of inputs considered here is 4 (a,b and two control lines) and no of output is 4 (F1, F2,F3 and F4), total area considered here is 2.5 μm^2, total no of QCA cells are 682, time required to execute is 9 seconds (9 ns). All inputs are available at clock0, clock1, 2,3 are used to transmit inputs to programmable AND gate, again at clock 0 programmable AND and OR gates are evaluated and finally third cycle of clock 0 is used to get the outputs. The output functions are evaluated according to Majority logic method. Equation 13,14 15 and 16 gives F1 to F4 of Programmable AND and OR structure.

$$F1 = b = 1 \quad (13)$$

$$F2 = a \quad (14)$$

$$F3 = a.b = m(a,b,0) = m(m(a,b,0),m(a,b,0),1) = ab \quad (15)$$

$$F4 = a.b' = m(a,b',0) = m(m(a,b',0),m(a,b',0),1) = a.b' \quad (16)$$

Figure 12 shows the simulated waveform of QCA Programmable AND and OR structure. At clock 0 outputs are available, Programmable AND OR gates are evaluated at clock0. Three cycles of clocking (clock 0 to 3) are required for simulating the circuit.

ISSN 1512-1232

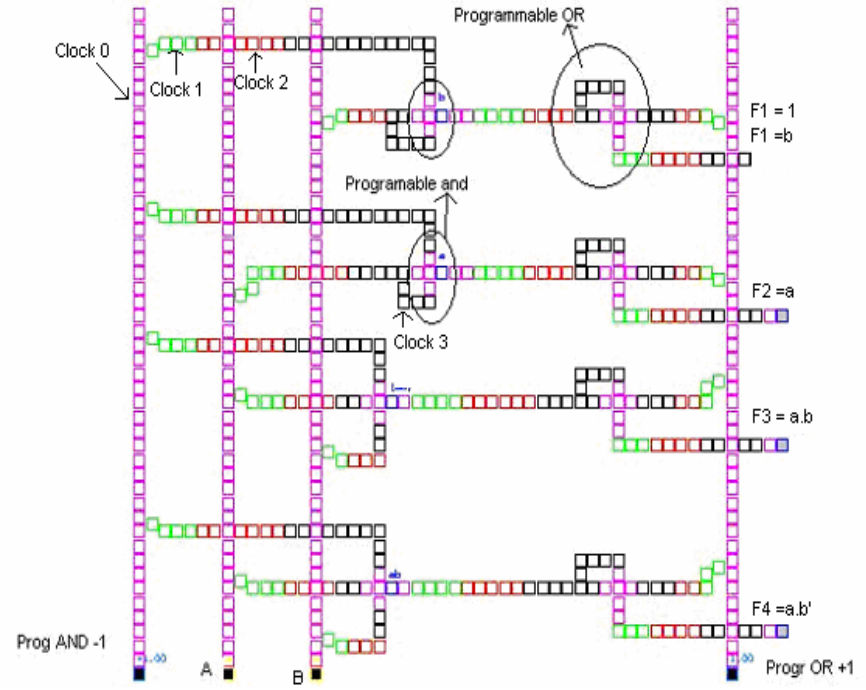


Figure 11 shows simple QCA Programmable OR and AND gate structure.

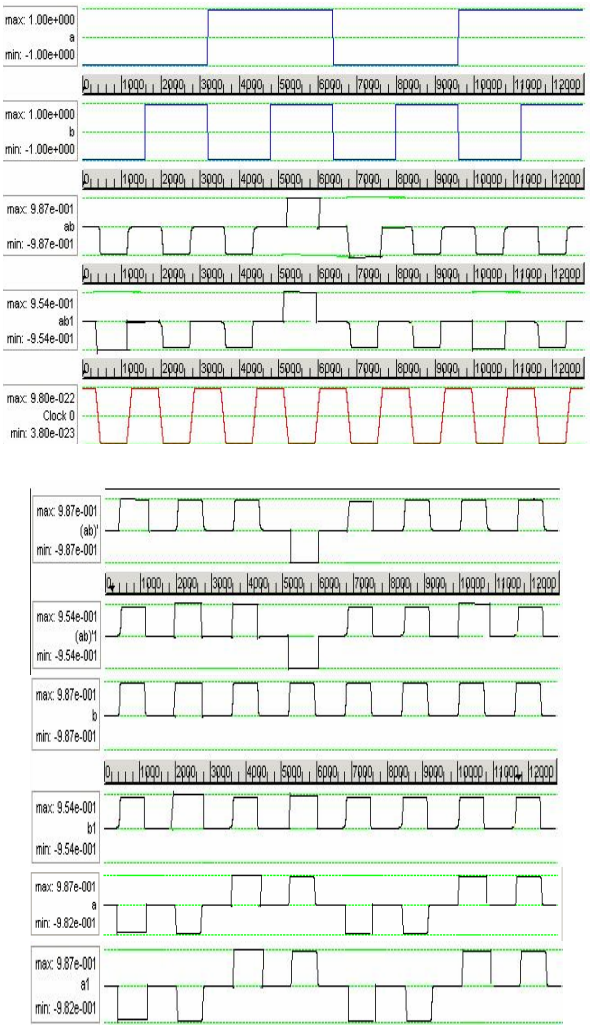


Figure 12 simulated waveform of simple QCA Programmable OR and AND gate structure.

ISSN 1512-1232

V Simulation of PLA QCA .

We use a combination of AND and OR planes made of QCA cells to implement a PLA. AND or OR gates can be implemented by using a single majority gate. Each PLA cell consists of programmable select bit denoted by 'select' bit. We considered here AND plane which consists of AND, OR majority logic gate for programming. If select bit is programmed as one or zero, logic function or a simple wire can be performed. A simple QCA wire is shown in figure 13.

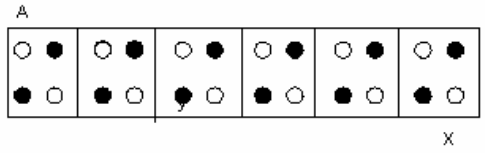


Figure 13 QCA CELL as a wire Output, X = Input A

A QCA cell schematic of AND PLA cell is shown in figure 13 and its logic representation is shown in figure 14. The OR PLA cell is constructed by switching the positions of AND and OR gates. We have considered AND PLA cell programmed as logic gate as well as a wire. The schematic of AND PLA cell consist of one AND, OR gate as shown in figure 15. Let A be the input to the AND gate and -1(0) be the control input to the AND gate, B be the input to the OR gate and +1 (1) be the control input to the OR gate. Each QCA cell has three inputs and one output, the third input to the OR gate is SELECT line which can be programmed as 1 or 0. If the select line is equal to one (+1), the PLA cell act as a wire and if it is zero (-1), PLA cell acts as a AND logic gate. The functionality of the AND PLA cell can be written as

OUTPUT = Input.Input1, if SELECT = 0 (17)

OUTPUT = m(Input, m(Input1,SELECT, +1), -1) (18)

OUTPUT = Input, if SELECT = 1. (19)

OUTPUT = m (Input,1,-1) (20)

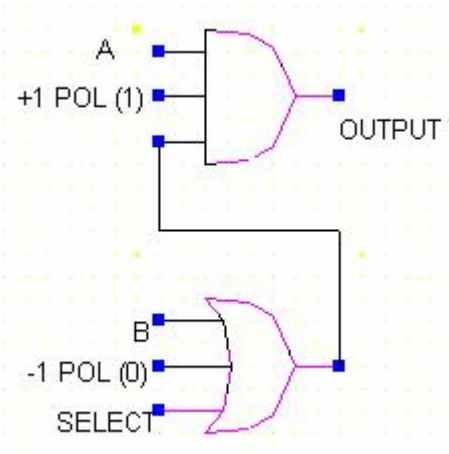


Figure 14 Logic diagram of PLA QCA cell

Figure 15 shows the PLA QCA cell representation simulated using QCA designer tool. PLA QCA cell has 10 QCA cell with two control input and one output. The simulated waveforms for PLA QCA cell is shown in figure 16 and 17. Figure 16 shows PLA QCA cell acts as a wire and Figure 17 shows PLA QCA cell acts as AND gate.

ISSN 1512-1232

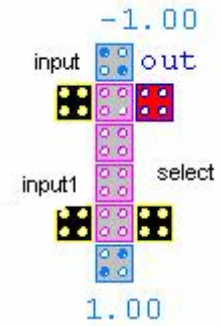


Figure 15 PLA QCA cell

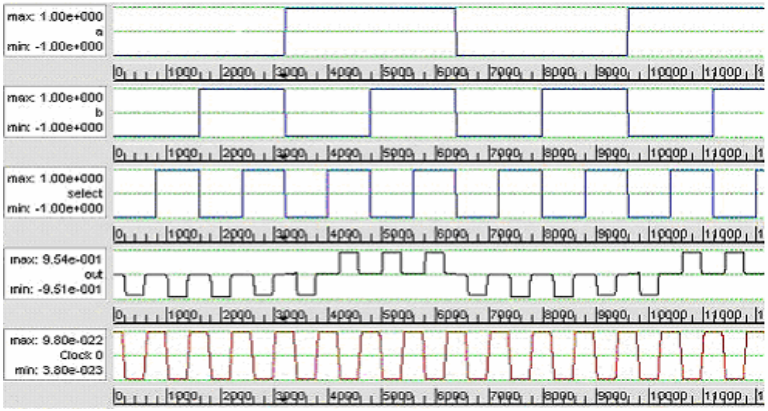


Figure 16 PLA QCA AND Cell simulated waveforms, select = 1 Output = a

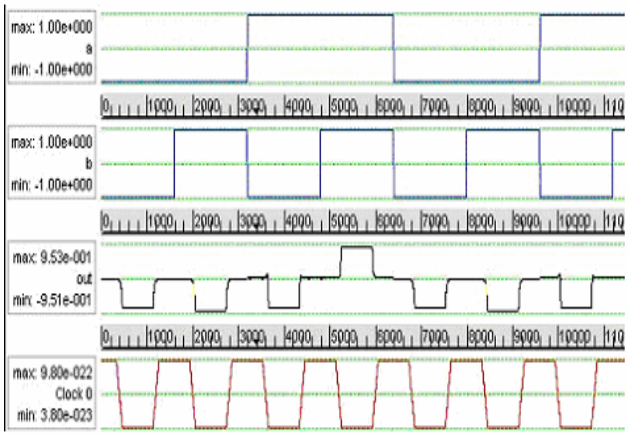


Figure 17 PLA QCA AND cell simulated waveforms, select = 0 Output = a.b.

Table 3 shows the PLA AND cell

Input A	Input 1 B	Select	Control Input OR	Output
0	0	0	1	0
0	1	0	1	0
1	0	0	1	0
1	1	0	1	1
0	0	1	1	0
0	1	1	1	0
1	0	1	1	1
1	1	1	1	1

ISSN 1512-1232

QCA-based majority gate is logically *bidirectional*, i.e., any one of the four boundary QCA cells can be treated as the output while the rest of the three as inputs. To program a cell, we treat the QCA cells corresponding to the select bit as the output of the OR gate. To drive a logical '1' or '0' to the select bit, we set Input A to '0' and Input1 B as to '1' or '0', respectively. The functionality and programming of OR cells are very similar except that '1' corresponds to logic mode and '0' corresponds to wire mode. Fig 18. depicts logic representation of a QCA PLA cell. During normal operation, the inputs come from below, the Input signals A move from left to right, and the Input1 B signals move from top to bottom. By setting a1 = a2 = a3 = 1, b55 = b66 = 0, b1 = X, b2 = X', b3 = Y ,b4 = Y', it is easy to verify that the PLA performs the following two logic functions: b5 = X XOR Y, and b6 = X' OR Y'. Figure 19 shows QCA PLA cell representation.

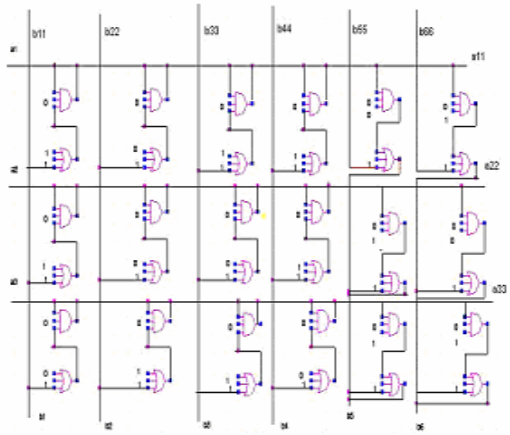


Figure 18 Logic representation of QCA PLA

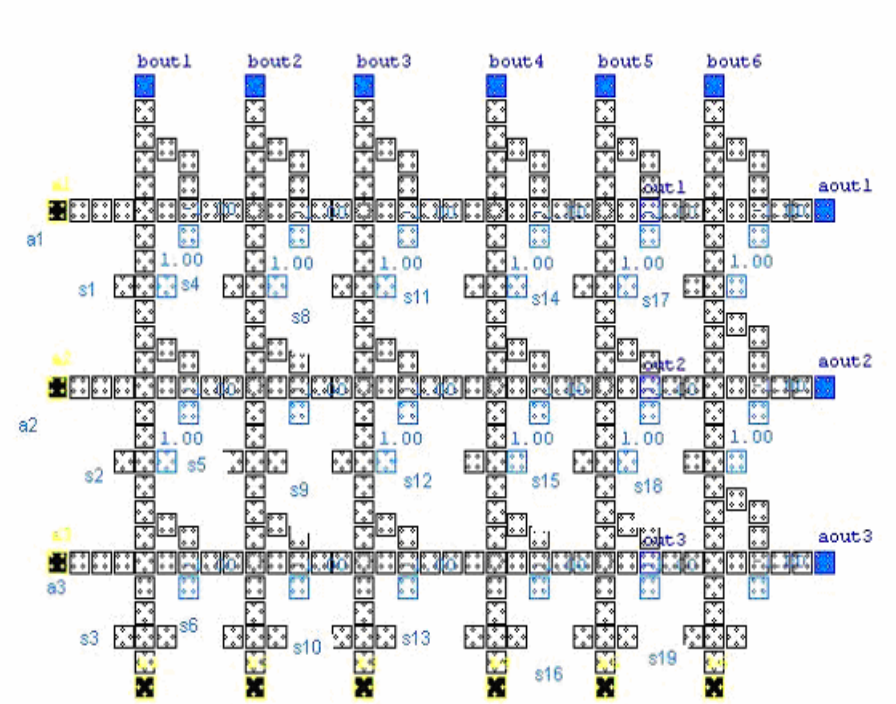


Figure 19 QCA PLA structure with XOR –OR function implementation.

We have adopted two clocking regions for QCA PLA cell, one for QCA cell and other for select and control lines, so that the output and inputs are in one clock region and select is in another clock region. We have adopted another clocking approach, all inputs a1, a2, a3 and their output a11, a22, a33 forms one clocking group, similarly all inputs b1,b2,b3,b4,b5 and b6 and their output bout1 to bout6 forms one clocking group, all majority logic gate evaluation in one clocking group and XOR – OR output cell in the forth clocking phase. This technique gives all AND plane and OR

ISSN 1512-1232

plane terms are evaluated in second clocking phase, inputs are available in first clocking phase, XOR – OR plane group only at third clocking phase and output are available at fourth clocking phase. Figure 20 shows the simulated waveforms for XOR gate. Inputs b1 as one input, b3 as another input and the output at b5 gives b1 XOR b3 i.e. b1b3 XOR b2b4, if the select lines are 1011, 1001, 0110 for the first four gates of QCA PLA structure. The last two planes (columns) of AND, OR cell planes are changed in their structure in order to get the output at a11, a22, a33. Here we have given select lines to AND plane instead of OR plane so that the b5, b6 provides output lines vertically. By changing the select lines to either vertical or horizontal lines, required logic functions can be performed.

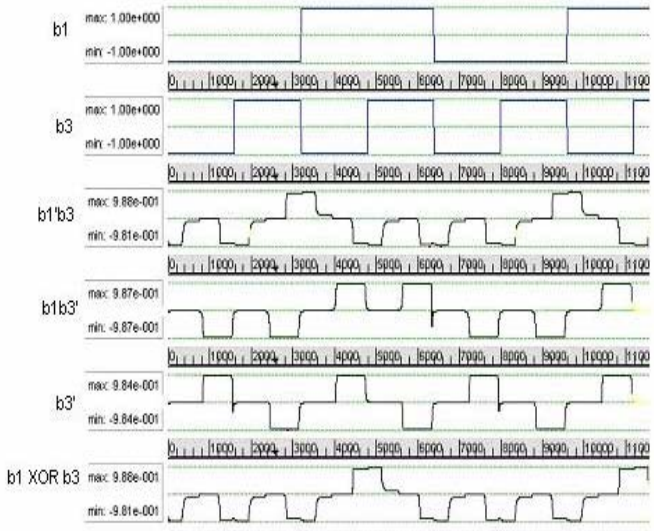


Figure 20 shows the simulated waveform of XOR gate from PLA QCA cell

Assigning a cell into either **logic** or **wire** mode can be done by setting the select bit of the cell to '0' or '1' (see Section 3.1), programming a PLA can thus be achieved by driving the desired select bit values to corresponding PLA cells. The fact that a number of PLA cells, such as those in the same column of the AND plane, share the same input requires that a proper programming sequence be followed. A simple programming sequence is followed here, program row by row for AND column and Column by column OR plane. Initially program one row say a1 of AND plane, the desired select bit value for each cell on this row is applied to corresponding column bit, at the same time the row bit value should not change select bit, after a row is programmed, the electric field for the select bits on this row must be kept adequately positive such that these values would not affect when input values to OR plane is selected, similarly column bits are selected through b1 to b6 and corresponding select bits of cell connected to columns are not affected. The same procedure is followed to OR plane also. If defect occurs due to improper row, select and column, that can be detected by the following procedure

1.Initially AND plane cells are selected, row bits of the AND plane are checked and wire mode logic is used to check in row wise of AND Planes.

2.Similarly all OR planes are checked by selecting column wise, Make wire mode logic and check the column bits.

3.Check row and column bits for individual defects by wire mode logic.

The design parameters in this section is same as that of table2, no of inputs and outputs considered here is 9, total area considered here 900 nm^2, no of qca cells are 458 cells, total simulation time 14 seconds(14ns), Clock 0 is used to get the output. Thus we can program PLA QCA cell for different logic functions.

ISSN 1512-1232

Table 3 gives binary values of XNOR and Boolean function M – Majority logic

a	b	c	a’	b’	c’	m(ab0)	m(a’b’0)	m(bc0)	OR1	OR2
0	0	0	1	1	1	0	1	0	1	1
0	0	1	1	1	0	0	1	0	1	0
0	1	0	1	0	1	0	0	0	0	1
0	1	1	1	0	0	0	0	1	0	1
1	0	0	0	1	1	0	0	0	0	1
1	0	1	0	1	0	0	0	0	0	0
1	1	0	0	0	1	1	0	0	1	1
1	1	1	0	0	0	1	0	1	1	1

VI Conclusion

1.We have constructed and simulated simple QCA AND – OR Plane structures, Fixed OR plane structures, programmable AND and OR plane structures and QCA PLA structures. This study shows complex Programming logic devices can be constructed and can be simulated. Experimental observations are still in research, these simulations may help us to analyze and study complex devices.

2.Programmable logic arrays can be used to construct Field programmable gate arrays that ensures nano FPGA devices for the future.

3.The design parameters considered here shows that little power dissipation that to only due to majority gate evaluation and clocking, hence very low power dissipation for these circuits.

4.We conclude that QCA cells can be used to construct Programmable logic devices there by leading a way for nano circuits.

REFERENCE

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.*

ISSN 1512-1232

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. A. DeHon and M.J. Wilson, "Nano-wire based sublithographic Programmable Logic Arrays," published in *Proc. of Int. Symp. on FPGAs,* (February 22-24, 2004), pp. 123-132
17. D.B. Strukov and K.K. Likharev, "CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices," *Nanotechnology*, 16(2005) 888-900.
18. C.R. Graunke, D.I. Wheeler, D. Tougaw, and J.D. Will, "Implementation of a Crossbar Network Using Quantum-dot Cellular Automata," *IEEE Trans. on Nano,* Vol. 4, No. 4, July 2004.
19. Niemier, M.T. and Kogge, P.M. "The '4-Diamond' Circuit - A Minimally Complex Nano-scale Computational Building Block in QCA," In *Proc. of the IEEE Comp. Soc. Symp. on VLSI*, pp. 3-10, February 2004
20. J. Huang, M. Momenzadeh, M.B. Tahoori, F. Lombardi: "Design and characterization of an and-or-inverter (AOI) gate for QCA implementation," *ACM GLVLSI Gymp. 2004*, 426-429
21. www.qcadesigner.ca

Article received: 2010-07-05