ISSN: 0974-5823 Vol. 7 No. 4 April, 2022

International Journal of Mechanical Engineering

Logic Functions Analysis Based on Quantum Cellular Automata

Anusha Bamini.A.M[1], Chitra.R[2], Vijayalakshmi Senniappan[3], Oswalt Manoj.S[4], Rohini.M[5]

[1]Assistant Professor, Department of Computer Science and Engineering

Karunya Institute of Technology and Sciences, India

[2]Associate Professor, Department of Computer Science and Engineering

Karunya Institute of Technology and Sciences, India

[3]Professor & Head, Department of Information technology , Park college of engineering and technology

Kaniyur,Coimbatore 641659, India

[4]Assistant Professor, Department of Computer Science and Business System,

Sri Krishna College of Engineering and Technology, Coimbatore

[5]Assistant Professor, Department of Computer Science and Engineering,

Sri Krishna College of Engineering and Technology, Coimbatore

ABSTRACT- Quantum Dot-Cellular automata (QCA) is an evolving technique and it can be deliberated as a better substitute solution for CMOS technology who is having short channel effects. QCA is a transistor loose technology, and the records is shipped formulated at the rate of the electron and by means of Columbic repulsion principle. The QCA nano-technology is fascinated by many features such as high device density, faster-switching speed and deficient power consumption. The logic functions are considered as the rudiment of all applications of QCA. In QCA, the logic gates and features are devised with the help of three logic styles, namely Majority gate, Nand-Nor-Inverter (NNI) and And-Or-Inverter (AOI). In this paper, the logic gates and functions have been designed and analyzed by using three types of logics. The implementation of the logic functions is executed using QCA Designer-2.0.3.

Keywords: Quantum Dot-Cellular Automata (QCA), Logic functions, NNI, AOI, and Electrostatic repulsion.

1. INTRODUCTION

In today's electronic world, CMOS technology is a needed one in present electronic environment. We are at the edge of a current CMOS technology. Reducing the scale of the IC and increasing the quantity of transistors could cause the unconditional affects inside the region of scaling down the nanometer gadget. CMOS equipment experiences by physical side effects of short-channel effects, overpriced heavy process and fabrication, and complex interconnection. Our world requires some innovative technological advancement to surpass the issues of CMOS technology. From this aspect, Quantum electronics will give better results in reality, and it's miles the viable candidate for effectively act to the downsides of the CMOS era. This advanced quantum electronics would create a new generation of 'Quantum Physics' changing modern-day Physics. Moreover, this high-tech innovation would make a new world, and it will interchange all the existing techniques.

The impression of QCA was initially coined by scientist Craig S. Lent in 1993 [1][2]. Ever since developed the new technology, many inventions and research have been performed for all field of applications. However, the QCA researchers were trying to reduce the quantum cell, which is the building block of the quantum electronic device and to enhance the proficiency of the QCA circuits. The scientists were struggled to construct enormous and composite digital circuits which is deployed in IC fabrication. The number of a quantum cell can be curtailed by utilizing three logic structures, namely majority gate, nand-nor-inverter and and-or-inverter. Majority gate is the underlying logic gate of the QCA technology, but it could not perform the inverter function [3]. The NNI logic is capable of performing all the logic functions, including inverter function also [4]. The AOI logic is harnessing both the logic structures majority gate and NNI, to form a new logic [5]. In [6], there are some sequential circuits has been designed and analyzed using the majority gate logic method in QCA implementation. The circuit complexity is worrisome for the designer perspective. However, in [7], the author designed logical circuits with no crosswire and thereby reduced the number of QCA cells. Some researchers were undergoing to accomplish fault-tolerant, area-efficient and fabricate QCA designs [8-11]. An excellent study is under progression on the manufacturing and fabrication process of the QCA. The evolution of the majority gate as a universal gate has been described in [12-13]. Moreover, the idea of multi-input to the majority gate is explained in [14-15]. Further, the Quantum technology provides some practical applications such as quantum computing, quantum information processing, quantum cryptography, quantum communication, quantum internet, and quantum optics and so on. In 2015, Samsung Company launched fourthgeneration QLED TV in the market, which is the rival to the OLED technology. The technology behind QLED is, the TV uses Quantum dot technology on an LED panel which provides better colour contrast and clarity. Similarly, quantum cryptography is the developing field for the quantum computer which is examined to be the best secure communication. Besides, Google has built 53-qubits of quantum computer called 'Sycamore' which do the problem-solving in 200 seconds, but conventional supercomputer takes 10,000 years to crack a similar problem [16]. At present, the processing speed of 'Sycamore' quantum computer is 1.5 trillion times of current supercomputer. So this invention is considered to be the breakthrough in the digital world and future, the digital world would become a quantum world in no doubt.

This paper is explained a brief stint of Quantum-dot Cellular Automata and the design of logic gates using different logic structures. This research analysis helps a lot for the beginning researchers in the field of Quantum Computing. This paper is structured as follows: Section II acquaints with the structure and operation principle of QCA and section III expounding the principles and theories related to QCA. Section IV described the proposed work, and section V discussed the simulation results and the comparative study. Finally, section VI concludes the paper.

2. PRELIMINARIES OF QCA

2.1 Quantum Dot Cellular Automata

Quantum dot Cellular Automata considered as a novel electronic prototype for information process and secure communication. The main characteristic of QCA compared to other Nano electronic architectures is the use of the same cell for the design of the logic circuits and for the wire which carries the information. In addition, the QCA offers incredible high speed operation in THz range, also furthermore, integration or volume of the device can be approximately 900 times superior than its current CMOS scaling limit that is at the - beyond the current CMOS scaling range. This functionality is not at all possible with today's technology. For these reasons, the ITRS semiconductor industries have been predicted that CBDs are the future nanotechnologies. In QCA, logical operations and data transfer can be done with the columbic interaction between each QCA cell rather than the current flow. Let's talk in detail about the basics and elements of the QCA.

2.2 QCA Cell

QCA cell is act as a fundamental block of QCA circuit which is created by 4 quantum dots that is structured inside a square form which is displayed in Figure 1. The researcher Zhang [17] is discovered that the cell has two excess electrons, when the cell gets charged, the excess electron in one cell tends to make a tunnel to the other cell through a clocking system. This electron tends to inhabit their respective antipodal places employing electrostatic repulsion. At the time of transferring information from one cell to the adjacent cell the columbic repulsion is possible. There are two kinds of cell structures based on the polarization of the electrons, and it is denoted by 'P' as presented in Figure 1. The digital circuits operated by switching activity of the transistor that means 'on' and 'off'. The conventional computer works by using logic '1' and logic '0'. Similarly, the logic '0' and logic '1' can be represented as P=-1 and P=+1 in QCA technology. Figure 1 demonstrates the arrangement of cell is based on polarization. By doing so, the binary information has been stored as a charged configuration in a QCA cell.

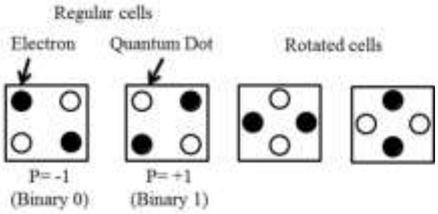


Figure 1. Basic QCA Cell & Two Possible Polarizations

The electrostatic energy of two quantum cells ‘a’ and ‘b’ for different polarities P_a and P_b, is expressed by the relation (1) and (2). From both the equations, we compute the energy of the two quantum cells by the sum of both four quantum cell of ‘a’ and ‘b’. Where, q_i^a and q_j^b represent the charges of the cells ‘a’ and ‘b’ and also, $r_i^a$and r_j^b denotes the locations of both the cells. Furthermore, the limits ‘i’ and ‘j’ signifies the range of the quantum dots in a cell, here it denotes four, as each cell consists of four quantum dots.

$$E^{a,b} - \frac{1}{4\pi\varepsilon}\sum_{i=1}^{4}\sum\nolimits_{j=1}^{4}\frac{q_i^a q_j^b}{\left|r_i^a - r_j^b\right|}$$

$$E_{kink} = E^{a,b}_{p_a \neq p_b} - E^{a,b}_{p_a = p_b}$$

2.3 QCA Wire

The rows arranged as horizontal in QCA cells were acting like a wire, and through this, binary signal can be spread from i/p to o/p owing to the electrostatic interfaces among an adjoining cells. If this cell is charged with two additional electrons, the electrons will be situated in crossways. This kind of arrangement in the cell is because of columbic repulsion forces which cannot permit in another way of position. In addition to that, the polarization (logic 1) of one cell prompted to make the same polarization in the next cell also, owing to the columbic repulsion theory.

The construction of the QCA cell has two types, such as 90° wire and 45° wire. The 90° wiring method facilitates to build normal cells and 45° wiring technique used to make rotated cells. The 45° wire make the QCA cell rotates between two polarizations. The model of 90° wire & 45° wire QCA has been presented in Figure. 2(a) and 2(b).

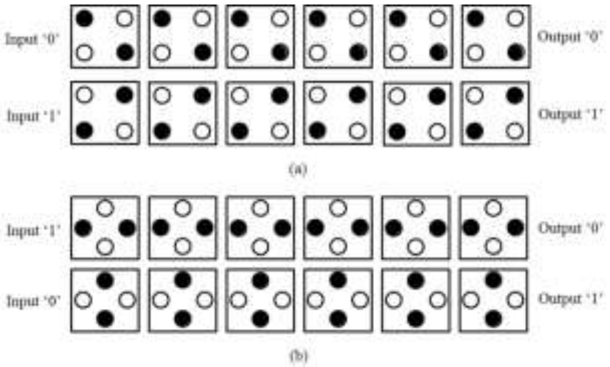


Figure 2. (a) QCA Wire 90° (b) QCA Wire 45°

2.4 QCA Clocking

The clock signal is an essential part of any digital circuit. Likewise, the QCA clock helps control the flow of information and determines the actual power gain in QCA. Mainly, it is used to restore the signal lost in the environment. Typically, the QCA valve has four distinct phases and is required for combinatorial and sequential circuits. The source of the QCA clock signal is the electric field, which is applied to rise or drop the tunnel obstruction among points in the QCA cell. If the barrier is low, cells are in unpolarized form and if the barrier is high, the cells states cannot be changed. Heumpil Cho et al [18] explained about the clock of QCA circuits. The QCA clock concept is completely different from the CMOS approach since the QCA cell was not powered by any external source other than the clock. The QCA circuit has four arming zones, using these clocks the information has been verified. The four types of timing zones have been shown in Figure. 3. In this figure, each tap has been placed out of phase by 90 ° with respect to the previous one.

All cells can be grouped together to form a zone, and zone loops have four phases. In [19], Kyosun Kim et al. they described the four levels of the clock zones. The four phases of rooster zone have been illustrated in figure 4. In the switching phase, by raising the tunnel barrier, the electrons of a cell were therefore inclined by an electron of the nearest zone with the theory of colombic repulsion. The tunnel barrier will be high in the waiting phase, so the state will not change, but will affect other areas nearby. The release and relaxation phases reduce the tunnel barrier so that the area does not affect the different stimulation environment.

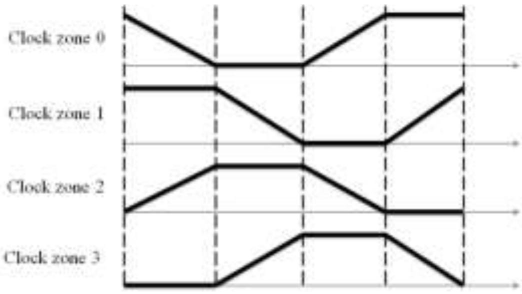


Figure 3. QCA Clocking Zones

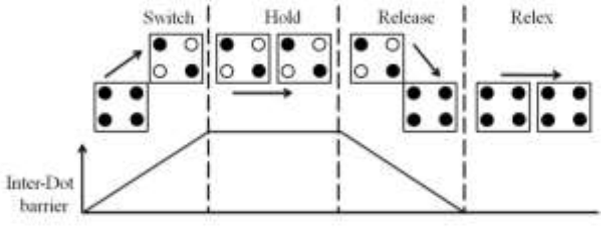


Figure 4. Four phases of clock

3. PROPOSED WORK

The existing literature is evident that the QCA design mostly depends on different gate technique. Walus [20] and Zhang [17] had predicted the QCA gate-based design mainly similar to the CMOS design process. The first step is to design the circuit for logic function and followed by synthesis process of logic function, which may responsible for obtaining the netlist. The logic function has been performed using the logic gate. The logic circuits are created based upon logic gates. OR, AND, NOT, NAND and NOR are some of the basic gates used. In this, some of the universal gates are NAND and NOR. The XOR gate is alternative gate that is made of the basic gates.

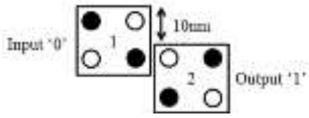


Figure 5. QCA Two Cell Inverter

Surface area and complexity are the essential issues in the design of QCA circuits. For cell count alleviation, QCA follows certain cell reduction techniques. By using the two-cell inverter, we can further reduce the number of QCA cells. In general, the inversion function was performed by 13 cells. But by using the two-cell inverter technique, the number of cells has been reduced. The two-cell inverter model was shown in Figure 5 and this type of inverter is used in this work for the efficiency of the area. This proposed work, all logic gates are designed and simulated using three QCA logic majority gate, NandNorInverter & AndOrInverter. This designed doors were simulated using software called QCADesigner. The simulated outcomes of a proposed QCA logic were equated and tabulated for reference.

3.1 Logic functions using Majority Gate

Different types of QCA circuits as well as devices are fabricated using various orientations of physical cells. The Majority Voting (MV) port is considered the base port in QCA technology, but the other base port is the inverter (INV). In QCA technology, logical operations were performed by interactions between neighboring cells. In this type of implementation, particular care must be taken to improve the stability of the circuit. The majority gate structure reduces the complexity of the QCA circuit area by arranging a non-overlapping cell of neighboring cells and also by using two cell inverters. Therefore, this type of implementation makes it possible to further reduce the number of cells.

3.1.1 AND and OR Gates

To construct a logic structure in QCA, the cells can be arranged in rows or columns. The propagation of the information or binary signal has been done by electrostatic repulsion. Because in each cell, the electrons repel each other. Consider the function A+B has to be constructed, the input A, B and C affects the structure of centre and out cells displayed in Figure 6.

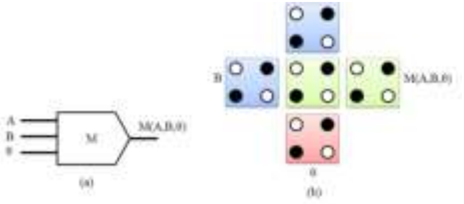


Figure 6. AND Gate (a) Schmetic (b) Layout

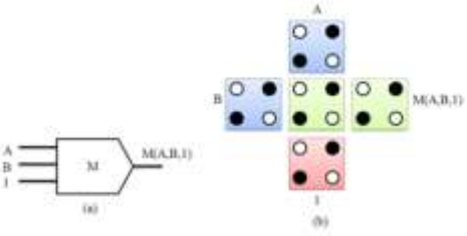


Figure 7. OR Gate (a) Schmetic (b) Layout

Subsequently, the mainstream of the inputs is fixed to be '1', so the middle and output cells are enforced to set into '1' position, astonishingly, which is the appropriate output for A+B (OR) function. From this, the theory has come out; if input C is set to '1', the logic structure acts as a OR gate, otherwise, if the input C is set to '0', the logic structure acts as a AND gate. Therefore, the logic operation AND and OR can be recognized by changing divergence of any one of the inputs of a majority gate by P= -1 (logic 0) or P=1 (logic 1). QCA logic structure of AND and OR functions are shown in Figure 6 and 7.

3.1.2 NAND and NOR gates

The position of the cells accomplishes the computation of QCA with respect to the polarization of the adjacent cells. Nevertheless, the inverter operation can be implemented in two ways, such as arranging and replacement of the cells. In this work, the arranging of the QCA cells has been utilized to do an inverter operation. Process of two cell inverter arrangement has been done as follows, the first cell is located usually, also second cell is located nearby location of first cell however 10 nm perpendicularly under the cell, as displayed in Figure 8.

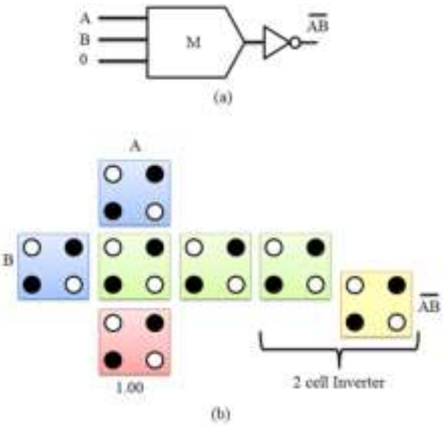


Figure 8. NAND Gate (a) Schmetic (b) Layout

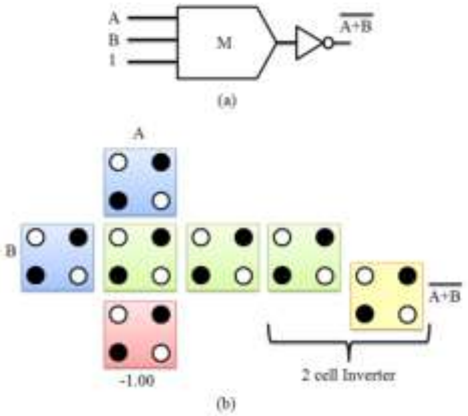


Figure 9. NOR Gate (a) Schmetic (b) Layout

Electrostatic interaction is reversed here because the polarizations of quantum dots are uneven and different among cells. As NAND gate is a complement of AND function, the NAND function can be realized by connecting the inverter to the AND (MG) gate as shown in Figure 8. Similarly, the NOR function is realized by connecting the inverter to the OR gate, as shown in Figure 9. In this implementation, the area complexity has been alleviated by harnessing of two cell inverter.

3.1.3 XOR & Ex-NOR gates

The XOR logical operation produces output logic '1' if and only if any one of the operands is logic '1' but not both the operands. Through this gate, many logical functions have been constructed. A complement of XOR function is an Ex-NOR function. Which was recognized via fixing inverter to the XOR gate. Logic function of a XOR gate can be,

$$A \oplus B = A'B + B'A \qquad (3)$$

This function can be realized by using the majority gate as,

$$A \oplus B = M(M(A', B, 0)M(A\ , B'\ ,0), 1) \qquad (4)$$

A function $A'B$ has been performed by using the QCA gate function $M(A', B, 0)$ and the next term $B'A$ has been realized by the majority gate function $M(A\ , B'\ ,0)$, then to add both the results uses OR logic gate in equation (4).

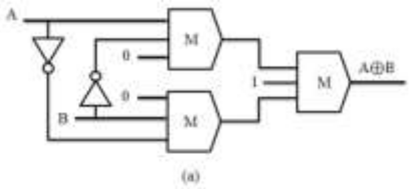
The Ex-NOR function is constructed by,

$$(A \oplus B)' = AB + A'B' \qquad (5)$$

Equation (6) shows a majority gate function of above equation,

$$(A \oplus B)' = M(M(A, B, 0), M(A', B', 0), 1) \qquad (6)$$

Equations (4) and (6) are the expression for XOR and Ex-NOR function in majority gate logic.



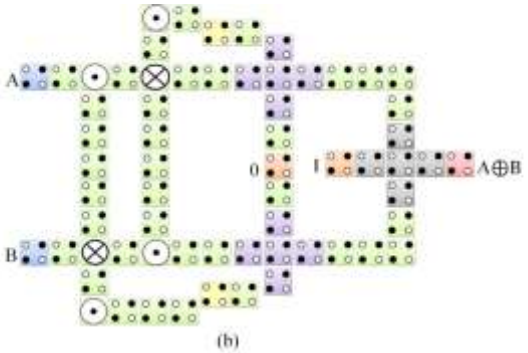


Figure 10. (a) XOR Gate Schmetic (b) Layout of XOR Gate

There are 3 majority gate and 2 inverters are used for constructing a XOR gate as presented in Figure 10. The particular QCA implementation is shown in Figure 10. There are 64 cells required to design XOR using majority gate structure, and it inhabits a region of 64000 nm^2, which is smaller than the earlier implementation. Likewise, the Ex-NOR gate was structured and executed by QCA cells, as Figured in 11.

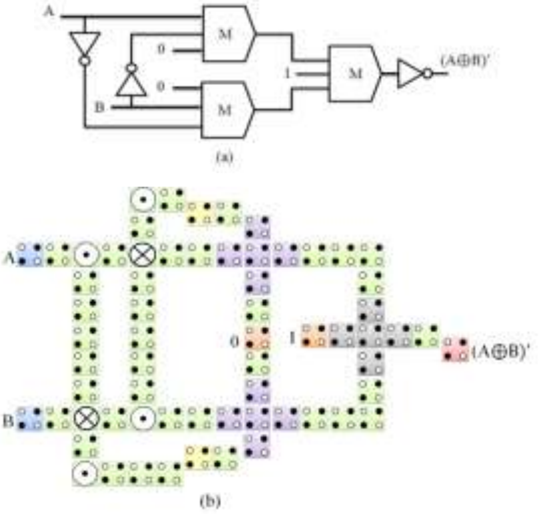


Figure 11. XNOR Gate (a) Schmetic (b) Layout

3.2 Logic functions using Nand-Nor-Inverter (NNI)

The logic gate in a circuit is responsible for performing all the logical functions. The NNI logic used to implement the features with the minimized QCA cells. This logic style consists of two inverted inputs. Therefore, this logic has its own design rules and using these rules, all design functions can be implemented. The design rules are,

- Rule 1: if the first 2 inputs of the NNI gates are zero then NNI gate act like inverter.
- Rule 2: if the last 2 inputs of the NNI gates are zero then NNI gate act like inverter.
- Rule 3: The NNI gate output is always one if A=C=0 and B=1 or 0.
- Rule 4: The NNI gate acts as a NAND gate if B=1, and the other 2 inputs are not changed.
- Rule 5: The NNI gate acts as a NOR gate if B=0, and the other 2 inputs are not changed.

3.2.1 Inverter Design

The majority gate didn't have the facility to design the inverter function. This motivated the NNI logic to construct the inverter operation. The inverter design using NNI logic can be described down,

The gate of NNI is denoted as,

$$\mathrm{NNI}(A,B,C) = M(A',B,C') = A'B + BC' + C'A' \tag{7}$$

If B=C=0 is substituted to the equation (7) then,

$$\mathrm{NNI}(A,0,0) = A' \tag{8}$$

Equation (8) can be realized from rule 1 and 2 of NNI design rules. The corresponding NNI gate implementation is exposed by Figure 12.

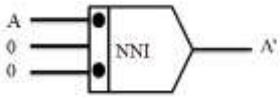


Figure 12. Inverter Using NNI

3.2.2 AND and OR Design

The AND logic function is realized by setting the input B=1 in equation (7) is given as,

$$\mathrm{NNI1}(A, 1, C) = A' + C' + C'A' \quad (9)$$

$$= A' + C' = (AC)' \quad (10)$$

This is the result of the NAND gate, but the required function is AND gate. This is accomplished by giving the output of NNI1 to any one of the inputs of NNI2 and makes the remaining 2 inputs as zero, that can be act as inverter. Therefore, this provides logical operation of AND gate. Figure 13 shows that the NNI-AND gate implementation.

Hence,

$$\mathrm{NNI2}(\mathrm{NNI1},0,0) = \mathrm{NNI2}((AC)', 0,0) = AC \quad (11)$$

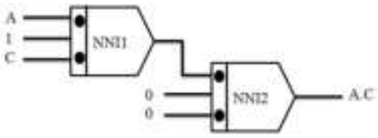


Figure 13. AND Gate using NNI

The OR logic function has been achieved by putting B=0 in equation (7) is given as,

$$\mathrm{NNI1}(A, 0, C) = C'A' = (A + C)' \quad (12)$$

This is the result of NOR gate, but the required function is OR gate. This is accomplished by giving the output of NNI1 to any one of the inputs of NNI2 and makes the remaining 2 inputs as zero, that perform as inverter. Therefore, this provides logical operation of the OR gate. Figure 14 shows that the NNI-OR gate implementation.

$$\mathrm{NNI2}(\mathrm{NNI1},0,0) = \mathrm{NNI2}((A + C)', 0,0) = A + C \quad (13)$$

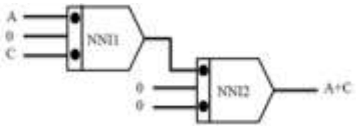


Figure 14. OR Gate Using NNI

3.2.3 NAND and NOR Design

In general, the NAND logic was a complement form of the AND function. That was recognized by attaching an inverter with the AND gate. Similarly, the NOR function is accomplished by connecting an inverter to the OR gate. However, at NNI logic, no requirement of inverter operation since it's a universal gate. So, single gate is enough for constructing both NAND & NOR gates. Thereby, area requirement and difficulty may be reduced further. From the above discussion, the equations (10) and (12) are known as the functions of NAND & NOR. NAND gate is recognized while setting a NNI gate input to B=1 and B=0 for NOR gate. Implementation of NAND & NOR gates are given by Figure 15.

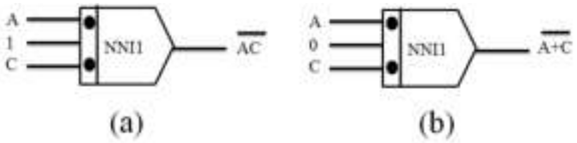


Figure15. (a) (b) NAND and NOR Gate Using NNI

3.2.4 XOR and XNOR Design

The XOR logical operation produces output logic '1' provided any one of the operands is logic '1' but not both the operands. Through this gate, many logical functions have been constructed. The complement of XOR function is an Ex-NOR function. This is recognized as attaching the inverter to the XOR gate. Logic function of a XOR gate is,

$$A \oplus B = A'B + B'A \quad (14)$$

There are some steps to realize the above equation.

- Step 1: To achieve the term $A'B$, put C=1 in (7) then, $\mathrm{NNI1}(A, B, 1) = A'B + B.0 + 0.A' = A'B$
- Step 2: To realize the next term $B'A$, interchange the variables and put C=1 in equation (7),

$$\mathrm{NNI2(B, A, 1) = B'A}$$

- Step 3: to add both results, NNI1 and NNI2 outputs are specified as the input to NNI3, the NNI3 input is denoted as 0.

$$\mathrm{NNI3 = NNI(NNI1, NNI2,0) = (A'B + B'A)'} \qquad (15)$$

- Step 4: to invert the above function, $\mathrm{NNI4 = NNI(NNI3,0,0) = A'B + B'A}$

Thus, we achieved an XOR operation. Therefore, the formula for XOR operation is given as,

$$\mathrm{A \oplus B = NNI4((NNI3(NNI1(A, B, 1), NNI2(B, A, 1), 0), 0,0} \qquad (16)$$

Similarly, as the XNOR operation is precisely the opposite of the XOR operation, there is no necessity of step 4. The XNOR realization is accomplished at the third step only. Let us discuss in detail, We know that

$$\mathrm{A \odot B = AB + A'B' = (AB + A'B')'} \qquad (17)$$

Since, $\mathrm{(AB + A'B')' = (A'B)'(B'A)'}$

$$\mathrm{= (A + B')(B + A') = AB + A'B'}$$

Thus, the complete expression for XNOR-NNI gate realization is,

$\mathrm{A \odot B = (A \oplus B)' = (NNI3(NNI1(A, B, 1), NNI2(B, A, 1), 0)}$

Thus, equation (15) proves the XNOR operation of equation (17). Thereby, the additional inverter step is not required. Figure 16 depicted the XOR implementation in NNI logic.

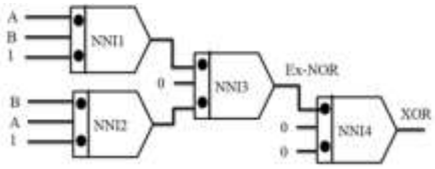


Figure 16. XOR Gate Using NNI

3.3 Logic Functions using AND-OR-INVERTER (AOI)

The majority gate alone not capable of constructing all the Boolean functions but it requires inverter also. Therefore, the QCA designers proposed new cell preparation for the l of the logical functions. Thus, to implement majority gate with NOT gate, a new QCA gate has been made as And-Or-Inverter (AOI) [21]. By using this AOI gate, many logical functions can be designed. The AOI is a five input gate and considered as a complex one. Hence, functional rules have been discussed here. These rules help to implement the design in an easier way. The AOI gate has five inputs. So the design complexity has reduced by considering the inputs D and E as constant values. This assumption leads to define the characteristic equation for the AOI gate is,

$$\mathrm{F = DE + (D + E)(A'C' + A'B + BC')} \qquad (18)$$

AOI gate design instructions are suggested for a fast and simplify the execution of compound circuits. Some rules are mentioned bellow. The design rules are applicable when the inputs DE=01 or 10 in (18) equation,

- Rule 1: AOI gate acts as an inverter when the first 2 inputs A and B is zero.
- Rule 2: AOI gate acts as an inverter when the last 2 inputs B and C is zero.
- Rule 3: The output of AOI gate is always one if A=C=0 and B=1 or 0.
- Rule 4: The AOI gate acts as the NAND gate when B=1, then there will be no change in further inputs.
- Rule 5: The AOI gate acts as NOR gate when B=0, and there will be no change in further inputs.

3.3.1 Inverter Design

The majority gate has an issue of not able to construct the inverter operator. This limitation is tackle by the AOI gate. Since the AOI gate embedded with AND, OR and INV functions, all the Boolean functions has been designed. The design of inverter function using AOI gate as follows, From (18),

$$\mathrm{F\ = DE + (D + E)(NNI(A, B, C))} \qquad (19)$$

We know that, $\mathrm{NNI(A, B, C) = M(A', B, C') = A'B + BC' + C'A'}$

Substitute DE=01 or 10 in (18) so the resultant equation becomes,

$$F = AOI(0,1,A,B,C) = A'B + BC' + C'A' \quad (20)$$

If we set B=C=0 to the above equation then, $F = AOI(0,1,A,0,0) = A'$

Thus, an inverter function has implemented using AOI gate, and Figure 17 shows that the inverter implementation using AOI gate.

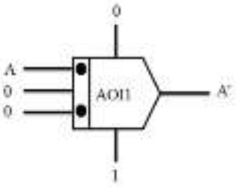


Figure 17. Inverter Using AOI

3.3.2 AND and OR Design

To design AND logic function, put B=1 in equation (20) then,

$$F = AOI1(A,1,C) = A' + C' + C'A'$$

$$= (AC)' \quad (21)$$

To make the invert function of the above result, the output of the AO1 is given as the input of AOI2. Then, by using the design rules, make other two inputs of the AOI2 is zero, the inverter function is implemented. Also, the implementation of AND-AOI gate is displayed in Figure 18. There by,

$$AOI2(AOI1,0,0) = AOI2((AC)',0,0) = AC \quad (22)$$

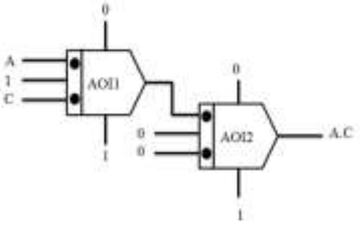


Figure 18. AND Gate Using AOI

To design OR gate, put B=0 in eqn. (20) then,

$$AOI1(A,0,C) = C'A' = (A+C)' \quad (23)$$

To make the invert function of the above result, the output of the AO1 is given as the input of AOI2. Then, by using the design rules, make other two inputs of the AOI2 is zero, the inverter function is implemented. Also, the OR-AOI gate implementation is shown in Figure 19.Therefore,

$$AOI2(AOI1,0,0) = AOI2((A+C)',0,0) = A + C \quad (24)$$

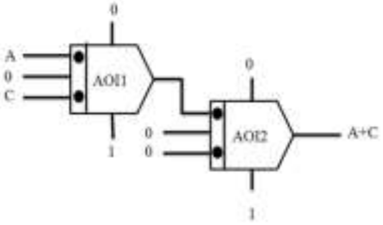


Figure 19. OR Gate Using AOI

3.3.3 NAND and NOR Design

In general, the NAND logic is the complement form of the AND function. It is recognized by joining an inverter with the AND gate. Similarly, the NOR function is accomplished by connecting an inverter to the OR gate. However, here no requirement of AOI logic inverter operation since it’s a universal gate. So, a single gate is enough for constructing both NAND and NOR gates.

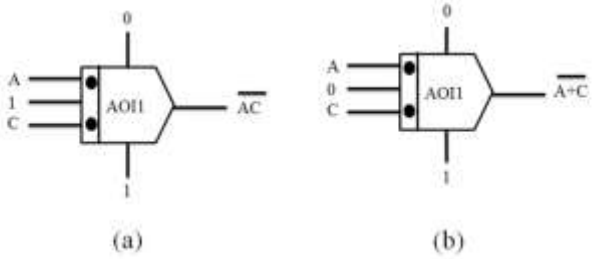


Figure 20. (a) NAND Gate Using AOI (b) NOR Gate Using AOI

Thereby, the area requirement and complexity can be reduced further. From the above discussion, the equations (21) and (23) are known as the functions of NAND and NOR. NAND gate was recognized as setting an AOI gate input to B=1 and B=0 for NOR gate. Implementation of NAND & NOR gates given in Figure 20.

3.3.4 XOR and XNOR Design

The XOR logical operation produces output logic '1' if and only if any one of the operands is logic '1' but not both the operands. Through this gate, many logical functions have been constructed. A complement function of XOR is an Ex-NOR function. This can be recognized by linking the inverter to the XOR gate. The logic function of the XOR gate is,

$$A \oplus B = A'B + B'A \qquad (25)$$

There are some steps to realize the above equation.

- Step 1: the first term is obtained by setting C=1 in AOI1(A,B,1)=A′B
- Step 2: the second term is obtained by interchanging the variables and setting C=1 in AOI2(B,A,1)=B′A
- Step 3: to add both the outputs, the results for AOI1 and AOI2 are represented as the inputs of AOI3 gate also setting the 3rd input to zero. Therefore,

$$\mathrm{AOI3} = \mathrm{AOI}(\mathrm{AOI1}, \mathrm{AOI2}, 0) = (A'B + B'A)'$$

- Step 4: to obtain the final result, we have to invert the above result. so, the output of AOI3 is given as one of the inputs of AOI4 and fixing the other inputs to zero, therefore,

$$\mathrm{AOI4} = \mathrm{AOI}(\mathrm{AOI3}, 0, 0) = (A'B + B'A) \qquad (26)$$

Finally, whole expression for XOR-AOI gate is,

$$A \oplus B = \mathrm{AOI4}((\mathrm{AOI3}(\mathrm{AOI1}(A, B, 1), \mathrm{AOI2}(B, A, 1), 0), 0, 0) \qquad (27)$$

Similarly, as XNOR operation is precisely the complement of the XOR operation, there is no need for step 4. The XNOR realization is accomplished at the third step only. Let us discuss in brief,

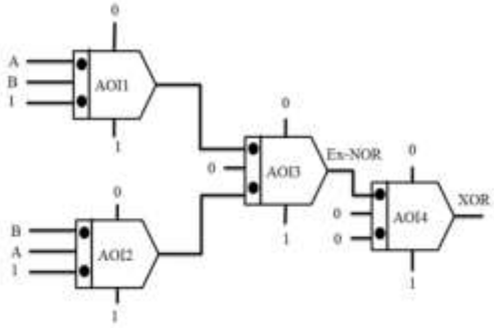


Figure 21. XOR and XNOR Gate Using AOI

We know that,

$$A \odot B = AB + A'B' = (AB + A'B')' \qquad (28)$$

Since, $(AB + A'B')' = (A'B)'(B'A)'$

$$= (A + B')(B + A') = AB + A'B'$$

Thus, the complete equation for XNOR-AOI realization is,

$$A \odot B = (A \oplus B)' = (\mathrm{AOI3}(\mathrm{AOI1}(A, B, 1), \mathrm{AOI2}(B, A, 1), 0)$$

Thus, equation (26) proves the XNOR operation of equation (28). Thereby, the additional inverter step is not required. Figure 21 depicted the XOR and XNOR implementation in AOI logic.

4. SIMULATION RESULTS AND DISCUSSION

4.1 Simulation Waveforms

The functionality of the logic gates is tested by QCA Designer version 2.0.3. The waveform simulation for XOR and XNOR gates are shown in Figures 22 and 23. The QCA design implementation of AND, OR, NAND, and NOR gate required clock 0 zone. But implementing a QCA for XOR and XNOR gate involves four clocking zones by referring the Figure 21. The clock zone 0 represent receive an inputs from A and B; a path for common logic gate is created by clock 1, clock 2 was utilized by discovering common logic also clock 3 utilized for calculating output. Again output is displayed at clock 0. Here, the clock signal is used for governs the movement of data in QCA circuits; they are hold, relax and release phase.

Table 1. Analysis of Engine Type Simulation Results (MG Logical Structures)

Simulation Engine	Simulation Type	Simulation time of XOR (sec)	Simulation time of XNOR (sec)
Bistable Approximation	Exhaustive	3	2
	Vector table	3	3
Coherence vector	Exhaustive	84	68
	Vector table	92 for 6 input	67

Table 1 describes the various kinds of simulation environment under the QCADesigner suit. There are two simulation engines available in the software, and they execute the program in two different simulation types such as exhaustive and vector table.

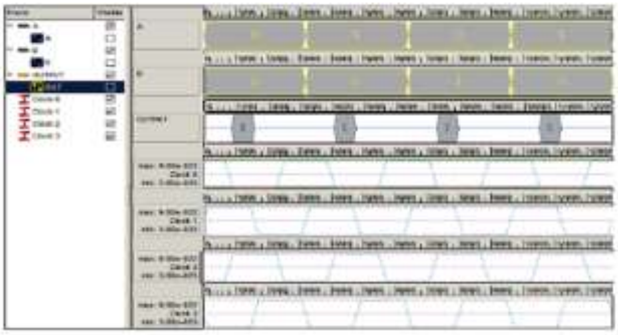


Figure 22. Simulation result of XOR Gate

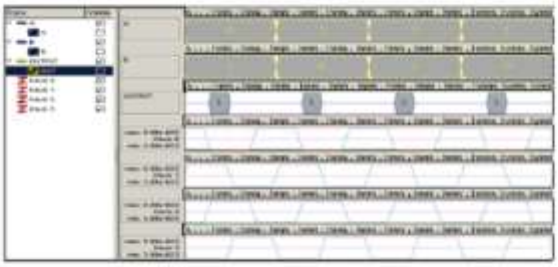


Figure 23. Simulation result of XNOR Gate

From an analysis report, we have been understood that coherence vector simulation engine provides results in somewhat slower however compared to than the bistable simulation engine it gives specific results. The waveforms outcomes are plotted for simulation, and selectively the buses are assembled as shown by Figures 22 and 23.

4.2 Comparative Analysis of Different Logic

Table 2. Comparison of Logical Structures

Type of Design	Design of Logical Structures using number of gates					
	AND	**OR**	**NAND**	**NOR**	**XOR**	**XNOR**
CMOS Design	4	4	4	4	8	8
Majority gate (MG) Design	1	1	1 1 Inverter	1 1 Inverter	3 2 Inverter	3 3 Inverter
NNI Gate Design	2NNI gate	2NNI gate	1NNI gate	1NNI gate	4NNI gate	3NNI gate
AOI Gate Design	2AOI gate	2AOI gate	1AOI gate	1AOI gate	4AOI gate	3AOI gate

Table 2 shows that the performance analysis of different QCA logic methods. The QCA logic method is mainly used for reducing the cell count and size. Thereby, the QCA circuit is area efficient and enhancing the processing speed. Here, the proposed logic can be compared with CMOS logic and identified the counts of transistors and QCA gates. It is clearly shown, proposed logic is far enhanced than conventional CMOS structure in terms of area, power and speed. Thus, in future, it is evident that quantum computing is a promising replacement for CMOS technology.

5. CONCLUSION AND FUTURE DIRECTIONS

Quantum computing is a brand new paradigm of transistor-much less generation and proved to be the robust alternative technique for the traditional CMOS logic. In this work, three types of QCA logic have been performed to design all the logic gates to ease the Boolean functions and expressions. The logical functions are designed using QCA and verified using QCADesigner software. From the simulation results and analysis, these QCA implementations of different logics are efficient than our conventional CMOS logic. The QCA structure provides breakneck processing speed which is almost not viable with CMOS logic. Moreover, the QCA logic is more attracted since it is an area-efficient. Because, in CMOS logic, by using N transistors, it can perform 2N operations. But in the case of QCA, one qubit has the capability of doing 2n operations in a parallel manner. Thus, QCA implementation becomes faster and creates a more intelligent network. The implementation of this design may be useful for designing complex circuits for various applications.

5.1 Challenges and Future Research Directions

Quantum Computing has a vast number of research fields in the world. Significantly, the quantum computing research and innovation is required for the field of information processing, computing and cryptography. There is a prerequisite of proposing efficient algorithms for providing strong entanglement authentication and for tackle the problem of de-coherence. Most of the quantum cryptographic protocols are in theoretical only. But in practice, they are prone to lose data, error-prone qubit detection and susceptibility to noise. Thereby, a more effective fault-tolerant algorithm has to be suggested. Another prominent research area is to design a revocable quantum identity authentication protocol. Similarly, in all kinds of fields researches are undergoing.

Data Availability Statement:

No datasets were used and analysed in this work. Therefore, data sharing also not applicable.

REFERENCES

1. C. S. Lent, P. D. Tougaw, W. Porod and G. H. Bernstein, *Quantum Cellular Automata*, Nanotechnology, Vol. 4, No. 1, 1993, pp. 49-57. doi:10.1088/0957-4484/4/1/004.
2. B. Isaksen and C. S. Lent, *Molecular Quantum-Dot Cellular Automata*, Proceedings of the Third IEEE Conference on Nanotechnology, Vol. 2, 2003, pp. 5-8.
3. Rumi Zhang, K. Walus, Wei Wang and G. A. Jullien, *A method of majority logic reduction for quantum cellular automata*, in IEEE Transactions on Nanotechnology, vol. 3, no. 4, pp. 443-450, Dec. 2004, doi: 10.1109/TNANO.2004.834177.
4. A. Bharti and A. Sharma, *Implementation of Ex-OR gate using QCA with NNI logic*, 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, 2017, pp. 2027-2031, doi: 10.1109/WiSPNET.2017.8300117.
5. Kunal Das and Debashis De, *A novel approach of And-Or-Inverter (AOI) gate design for QCA*, 2009 4th International Conference on Computers and Devices for Communication (CODEC), Kolkata, 2009, pp. 1-4.
6. Lee Ai Lim, A. Ghazali, S. C. T. Yan and Chau Chien Fat, *Sequential circuit design using Quantum-dot Cellular Automata (QCA)*, 2012 IEEE International Conference on Circuits and Systems (ICCAS), Kuala Lumpur, 2012, pp. 162-167, doi: 10.1109/ICCircuitsAndSystems.2012.6408320.
7. L. Balakrishnan, T. Godhavari and S. Kesavan, *Effective design of logic gates and circuit using quantum cellular automata (QCA)*, 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, 2015, pp. 457-462, doi: 10.1109/ICACCI.2015.7275651.
8. G. Singh, B. Raj and R. K. Sarin, *Fault-tolerant design and analysis of QCA-based circuits*, in IET Circuits, Devices & Systems, vol. 12, no. 5, pp. 638-644, 9 2018, doi: 10.1049/iet-cds.2017.0505.
9. A. N. Bahar and K. A. Wahid, *Design of an Efficient N × N Butterfly Switching Network in Quantum-Dot Cellular Automata (QCA)*, in IEEE Transactions on Nanotechnology, vol. 19, pp. 147-155, 2020, doi: 10.1109/TNANO.2020.2969166.
10. S. Babaie, A. Sadoghifar and A. N. Bahar, *Design of an Efficient Multilayer Arithmetic Logic Unit in Quantum-Dot Cellular Automata (QCA)*, in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 66, no. 6, pp. 963-967, June 2019, doi: 10.1109/TCSII.2018.2873797.
11. R. K. Nath, B. Sen and B. K. Sikdar, *Optimal synthesis of QCA logic circuit eliminating wire-crossings*, in IET Circuits, Devices & Systems, vol. 11, no. 3, pp. 201-208, 5 2017, doi: 10.1049/iet-cds.2016.0252.
12. K. Das and D. De, *A study on diverse nanostructure for implementing logic gate design for QCA*, International Journal of Nanoscience, vol. 10, no. 1&2, pp. 263–269, 2011.
13. K. Das and D. De, *Characterization, Test and logic synthesis of novel conservative logic gates for QCA*, Internal Journal of Nanoscience, vol. 9, no. 3, pp. 201–214, 2010.
14. K. Navi, R. Farazkish, S. Sayedsalehi, and M. R. Azghadi, *Five-input majority gate, a new device for quantum-dot cellular automata*, Journal of Computational and Theoretical Nanoscience, 2010.
15. P. Agrawal Kumar and B. Ghosh, *A novel nine input majority gate design in quantum-dot cellular automata*, Journal of Applied Global Research, vol. 6, pp. 33–41, 2013.
16. Arute, F., Arya, K., Babbush, R. et al. *Quantum supremacy using a programmable superconducting processor*, Nature 574, 505–510 (2019). https://doi.org/10.1038/s41586-019-1666-5
17. Zhang, R.., Walus, K., Wang, W. and Jullien, G. A. *A method of majority logic reduction for quantum cellular automata*, IEEE Trans. Nano., Vol.3, No. 4, pp. 443-450, 2004.
18. Heumpil Cho and Earl Swartzlander, E. *Adder Designs and Analyses for Quantum-Dot Cellular Automata*, IEEE Trans. on Nanotechnology, Vol. 6, No. 3, pp. 374-383, 2007.
19. Kim, K. Wu, K. and Karri, R. *The Robust QCA Adder Designs Using Composable QCA Building Blocks*" IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems, Vol. 26, NO. 1, Jan. 2007.
20. Walus, K. Jullien, G.A. and Dimitrov, V. *Computer arithmetic structures for quantum cellular automata*, Asilomar Conference on Signals, Systems, and Computers, 2003.
21. Mariam Momenzadeh, B. *Characterization, test and logic synthesis of and or Inverter Gate for QCA Implementation*, IEEE transactions on computer aided design of integrated circuits and systems, Vol. 2, pp.1881-1893, 2005.
22. C. Gopala Krishnan, Y. Harold Robinson, E. Golden Julie, A. M. Anusha Bamini, Raghvendra Kumar, Pham Huy Thong, Le Hoang Son, Hybrid Cache Management in Ad Hoc Networks, Wireless Personal Communications, Issue 4/2021.
23. Gopala Krishnan C, Gomathi S, Anusha Bamini A.M, High energy efficient lifetime management system and trust management framework for manet using self-configurable cluster mechanism, Peer-to-Peer Networking and Applications, Issue 3/2021.