Novel Code Converters Based On Quantum-dot Cellular Automata (QCA)

Firdous Ahmad[1], **GM Bhat**[2]

[1,2]Department of Electronics & IT, University of Kashmir, (J&K) India – 190006

Abstract: ***Quantum-dot cellular automata (QCA) is an emerging nanotechnology and a possible alternative for complementary metal-oxide semiconductor (CMOS). In this paper, we have proposed novel XOR gates. The models are suitable for designing combinational circuits with less area and circuit complexity. Code converters like excess-3 code, gray code, parity code generator, and checker, with minimum area and circuit complexity have been attempted by using proposed XOR gates. The novel designs enjoy the features of small area, superior performance factors in respect of area, latency, circuit stability, and low power dissipation. The operation of QCA circuit is simulated using QCA Designer tool.***

Keyword: Nanotechnology, XOR Gate, Excess-3 Code, Gray Code, Parity Generator, Parity Checker

1. Introduction

Quantum-dot Cellular Automata (QCA) is a technology featuring computer operations at high speed and low power consumption. It was first proposed by Craig S. Lent et al [1], at the University of Notre Dame [2]. A comprehensive overview about this subject can be found in [3]. QCA is based upon the encoding of binary information in the charge configuration within quantum dot cells. Computational power is provided by the Coulombic interaction between QCA cells. No current flows between cells and no power or information is delivered to individual internal cells [4]. The local interconnections between cells are provided by the physics of cell-to-cell interaction due to the rearrangement of electron positions [5-6]. Recent papers show that QCA can achieve high density, fast switching speed, and room temperature operation [7].

The objective of this paper is to propose a detailed analysis and design strategy for code converters using QCA nanotechnology. XOR gate is the main component to design code converters. Researcher in [8] has shown different performances of XOR gates. We propose an optimal design of novel XOR gates, which consists of less area, circuit complexity in comparison with the already proposed design in [9]. The aim of our proposed design is to maximize the circuit density and focus on a layout that is minimal in its use of cell counts. XOR gate has wide applications in code converters like, pseudo-code generators, and error detection & correction circuits shown in [9], [16-17]. In this paper, we propose the seven novel implementations of QCA code converters based on XOR gate and presented the simulation results of these individual designs. A detail comparison with regard to various characteristics of these designs is also presented. The aim is to maximize the circuit density and focus on a layout that is minimal in its use of cells. The proposed QCA circuits have been designed and stimulated using the QCADesigner tool.

2. Materials and methods

2.1 Background

The fundamental unit of QCA is the QCA cell created with four quantum dots positioned at the vertices of a square [4]. The cell is loaded with two extra electrons, which tend to occupy the diagonals due to Coulomb electrostatic repulsion. The positive charge that compensates for the cell energy is fixed. It is assumed that tunneling to the outside of the cell is not allowed due to a high potential barrier [11]. Binary information is encoded in the two possible polarizations (i.e., +1 or -1) as shown in Fig. 1(a). If two cells are brought close together, Coulombic interactions between the electrons cause the cells to take on the same polarization. If the polarization of one of the cells is gradually changed from one state to the other, the second cell exhibits a highly quick, bistable switching of its polarization.

2.2 QCA logic

The fundamental QCA logical circuit is the majority gate (MV) [12]. The logic equation for a majority gate is F (A, B, C) = AB + AC + BC and can be implemented by 5 QCA cells arranged in a cross as shown in Fig. 1(b). By fixing the polarization of one input as logic 1 or 0, we can obtain an OR gate and an AND gate respectively. So more complex logic circuits can be constructed from OR and AND gates respectively as:

$$a+b = F(a, b, 1) \quad (1)$$

$$a.b = F(a, b, 0) \quad (2)$$

Fig. 1(c) shows placing several of QCA cells placed side-by-side forms a wire. Logic values pass from cell to cell due to the Coulombic interactions. The polarization of the input cell is propagated down the wire. As a result, the system attempt to settle to a ground state. Any cells along the wire that are anti-polarized to the input would be at a higher energy level, and would soon settle to the correct ground state. When cells are placed diagonally to each other, they tend to have reverse polarizations due to the repulsion between electrons. This characteristic is used to implement an inverter, such as the one shown in Fig. 1(d). Unlike conventional CMOS in which it is the simplest block, it consumes considerable area in QCA

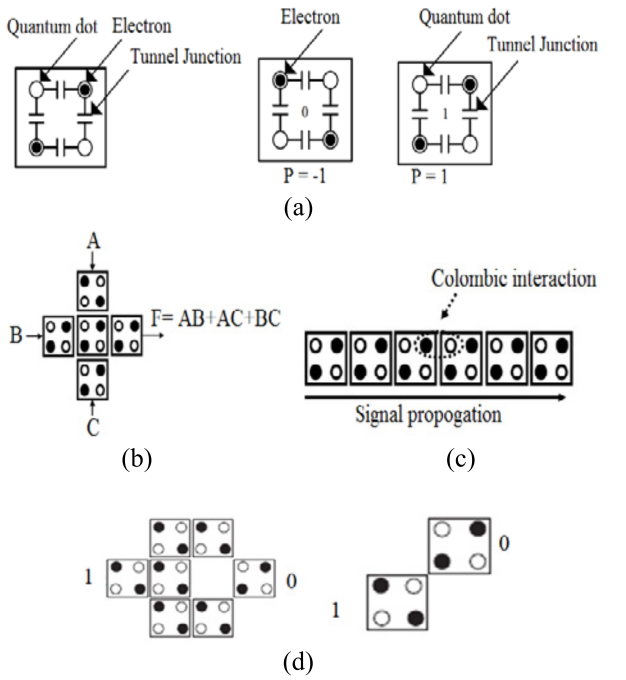


Figure 1: (a) QCA cells with four quantum dots in two possible cell polarizations (b) Majority gate (c) QCA Binary Wire (d) QCA Inverters

2.3 QCA Clock

An important thing about QCA information flow is the clocking scheme; that is to say for adjacent cells, in order to control the polarization reactions and effects, one should hold the polarization of the first cell fixed and lower the potential barrier of its adjacent cell in order to let the electrons of the adjacent cell relocate. This phenomenon should repeat repeatedly to pass the information through cells. It has been shown that for a QCA circuit to function correctly, only four clocking zones are necessary. Each clock signal lags 90 in phase with respect to the previous clocking [13].The four clock zones are shown in Fig. 2(a). These four clock zones are also called as one clock sequence. In QCA implementations, the power consumption of QCA circuits are mainly affected by clock signals. Therefore, in practice using a smoother clock, the power consumption will be very less but still it is data dependent. In order to remove the data dependency of power traces, Bennett clocking scheme can be used. As compared to its counterpart Landauer-clocked QCA circuit, Bennett clocking produces very low and very similar power traces for different inputs [14].Therefore, by using Bennett clocking, the power dependence of QCA circuits on the inputs can be effectively removed making it impossible to perform power analysis attack [15].

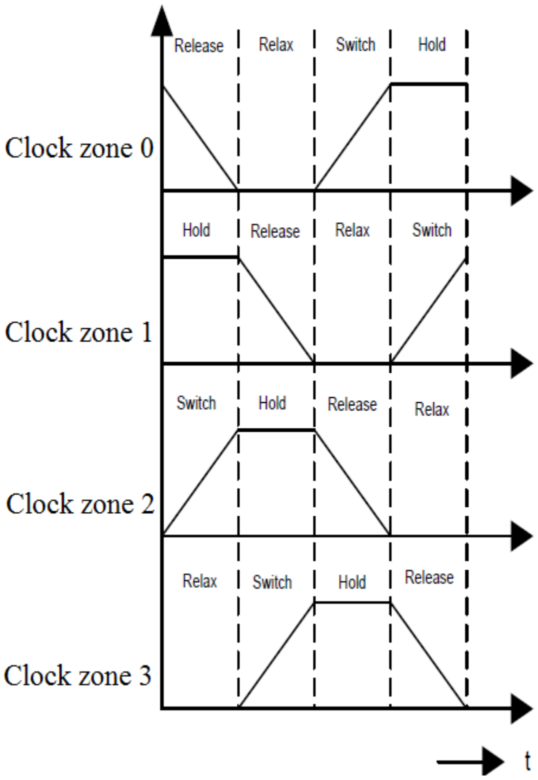
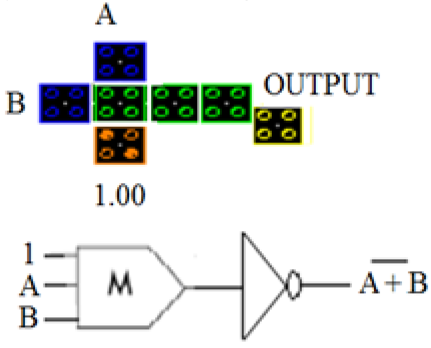


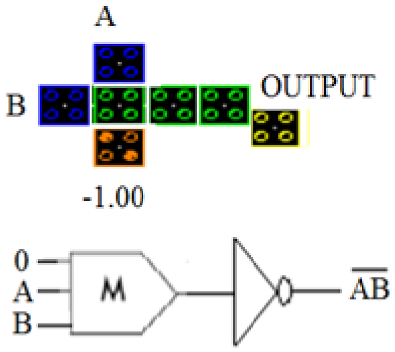
Figure 2: The four phases of the QCA Clock

3. QCA Implementations

To implement more complicated logical functions, a subset of simple logical gates is required. For example, it would be impossible to implement an XOR gate, code converters, pseudo-code generators, parity generators, and checkers in QCA without a logical AND gate, OR gate, or inverter. It has been demonstrated that a value's complement can be obtained simply by ripping it to a 45^0 wire at the proper location. Implementing the logical AND and OR functions is also quite simple by fixing the polarization to one of the inputs of majority vote as logic '1' or '0', we can obtain an OR gate and an AND gate respectively. NOR gate and NANAD gates can be implemented by complementing OR gate and an AND gate respectively. NOR gate is realized by connected OR gate followed by inverter shown in Fig. 3(a). Similarly, NAND gate is realized by connected AND gate followed by inverter shown in Fig. 3(b).



(a)

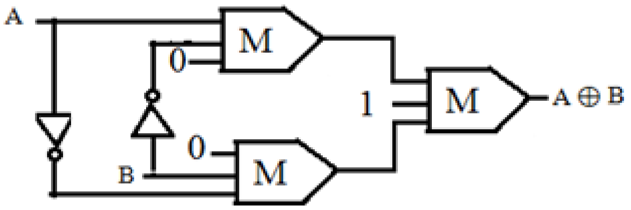


(b)

Figure 3: (a) Layout of NOR gate (b) Layout of NAND gate

3.1 The proposed XOR gate

XOR gate is a digital logic gate that implements an exclusive or; that is, a true output (1/HIGH) results if one, and only one, of the inputs to the gate is true. If both inputs are false (0/LOW) and both are true, a false output results. XOR represents the inequality function, i.e., the output is true if the inputs are not alike, otherwise the output is false. A way to remember XOR is "one or the other but not both". The schematic representation and truth table of an XOR gate is shown in Fig. 4



(a)

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

(b)

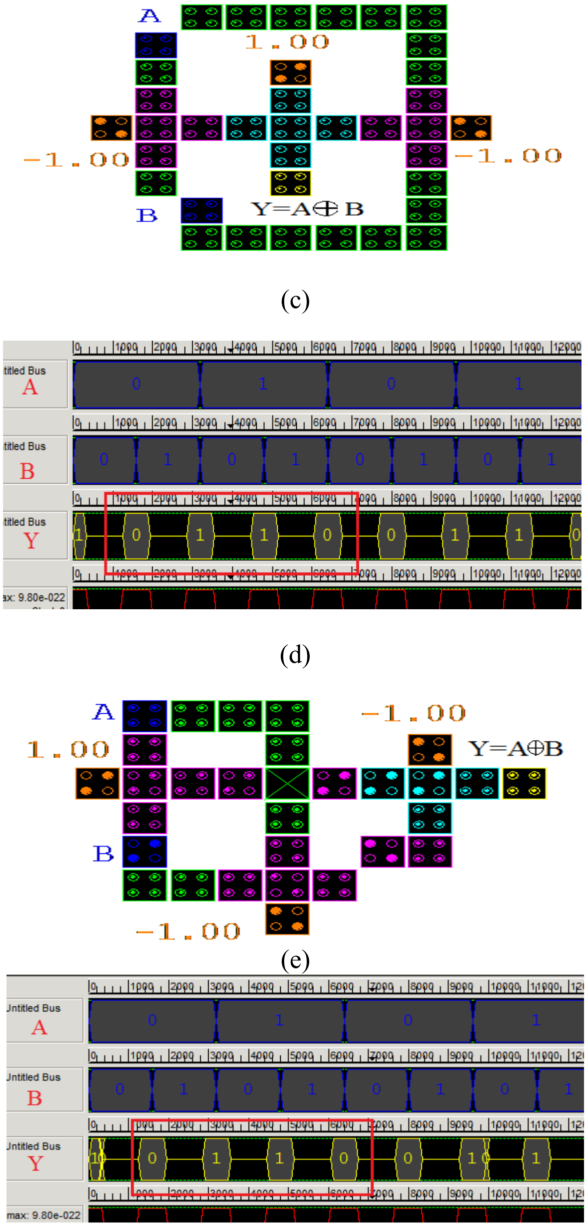
Figure 4: (a) Schematic XOR gate (b) Truth table

It is designed use of majority gates and inverters. Formula below presents the equation for this gate.

XOR = MV (MV (A', B, 0), MV (A, B', 0), 1) (3)

Different researchers [8-9], [18-20], have proposed the QCA implementation of XOR gate. XOR can also be viewed as addition modulo 2. As a result, XOR gates are used to implement binary addition in computers.

We have propose QCA layout of XOR gates with simple arrangement of basic cells and consist of very less area, circuit complexity with proper arrangement of clock delay as compared to previous designs [8-10]. The proposed structure shown in Fig. 5(c) consists of 37 cells and Fig. 5(e) consists of 30 cells. Each of design consists of less area 0.03um^2 and 0.5 clock delay. The simulation results of the proposed designs have shown in Fig. 5(d) & Fig. 5(f) respectively



(c)

(d)

(e)

(f)

Figure 5: (c) & (e) QCA layouts (d) & (f) Simulation results of proposed QCA layouts

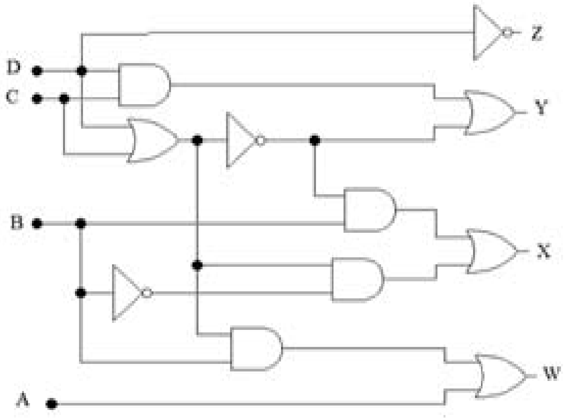
The proposed layouts can used to design more complex structure structures. We propose code converters like, excess-3 code, Binary to gray code converter, Parity generators, and checkers by using proposed layout XOR gates shown in Fig. 5

3.2 The proposed excess-3 code converter

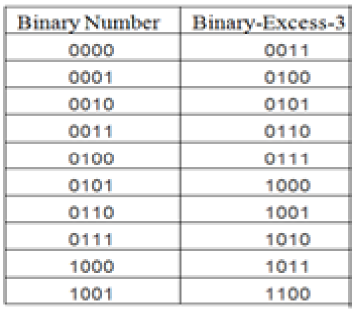
In digital electronics, codes are used to communicate the information between computers. These codes represent the information symbolically as a string of bits 0 and 1 and rules defined by the code decide the arrangement of these bits. The conversion circuit must be inserted between the two systems if each uses different codes for the same information. Thus, a code converter is a circuit that makes the two systems compatible even though each uses the different codes.

Excess-3 code is also known as **self-complimenting** code or **reflective code**. It is unweighted code used in decimal arithmetic units. The excess-3 code for the decimal number is performed in the same manner as BCD except that decimal number 3 is added to each decimal unit before encoding it to binary. The logic diagram of excess-3 code is shown in Fig. 6(a) and truth table for excess-3 code is shown in Fig. 6(b)

Implementation of QCA code converters introduced in [16] consists of very large area, circuit complexity, and latency. We have proposed an efficient excess-3 code converter based on proposed XOR gate. The proposed excess-3 code occupies a fraction of area 0.22um^2, less circuit complexity 143 cells compared to previous design [16]. The QCA layout of excess-3 code is shown in Fig. 6(c) and simulation results of excess-3 code is shown in Fig. 6(d)



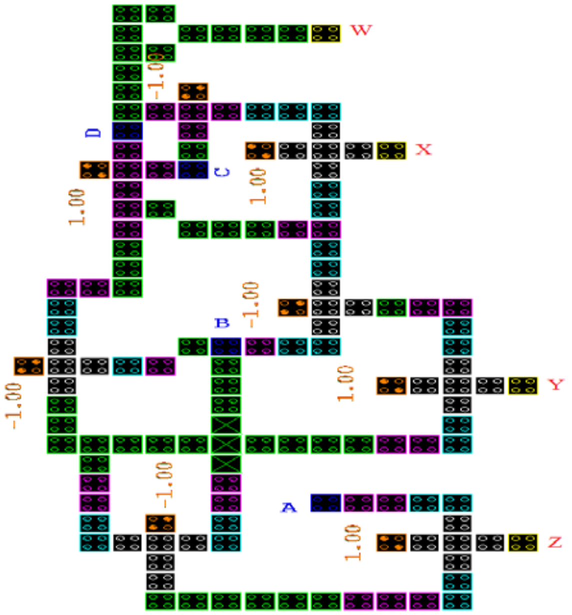
(a)



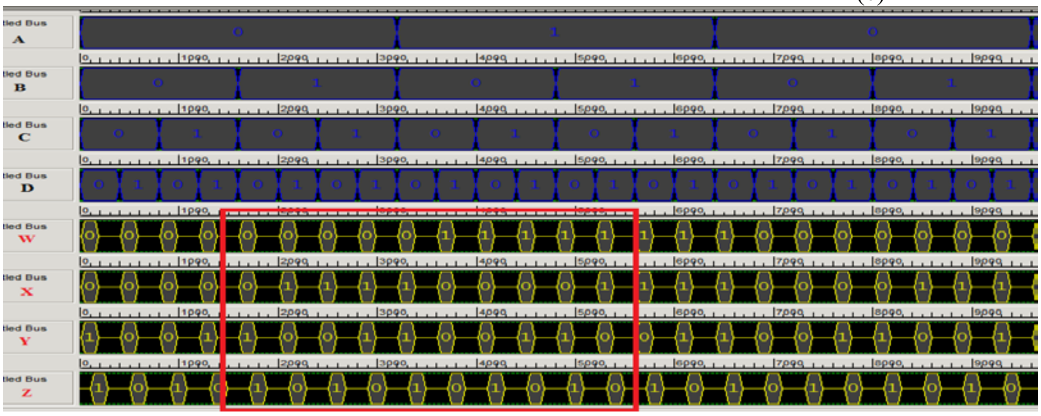
Binary Number	Binary-Excess-3
0000	0011
0001	0100
0010	0101
0011	0110
0100	0111
0101	1000
0110	1001
0111	1010
1000	1011
1001	1100

(b)

Figure 6: (a) Logic diagram (b) Truth table



(c)



(d)

Figure 6: (c) QCA layout of Excess-3 code (d) Simulation results

3.3 Gray code

Gray code, also known as reflected binary code, expresses all its values as a sequence of 1s and 0s. Unlike binary code, each value differs from the previous one by only a single bit. This has many practical applications, particularly where multiple simultaneous bit changes would result in errors. Gray codes can have any number of bits, and new gray codes can be calculated from binary codes that have one bit less than the proposed gray code. Gray codes are widely used to facilitate error correction in digital communications such as digital terrestrial television and some TV systems.

Grey code equivalent of the given binary number is computed as follows:

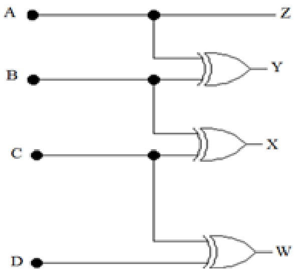
$$Z = A$$

$$Y = A \oplus B$$

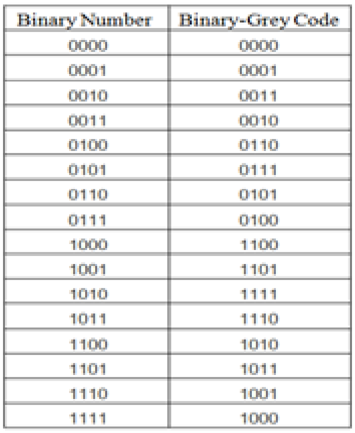
$$X = B \oplus C$$

$$W = C \oplus D$$

A binary to gray code converter can be implemented using XOR gates. For n input, n-1 gates are required. As shown in logic diagram Fig. 7(a) in the image below for 4 inputs, 3 XOR gates are used. The truth table of Gray code is shown in Fig. 7(b).



(a)

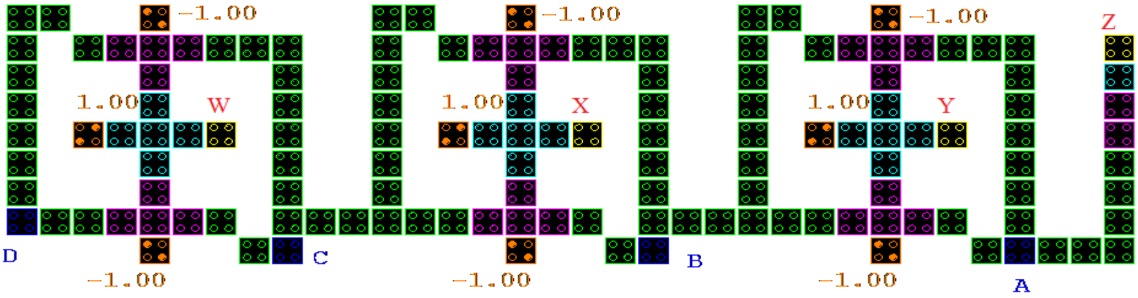


Binary Number	Binary-Grey Code
0000	0000
0001	0001
0010	0011
0011	0010
0100	0110
0101	0111
0110	0101
0111	0100
1000	1100
1001	1101
1010	1111
1011	1110
1100	1010
1101	1011
1110	1001
1111	1000

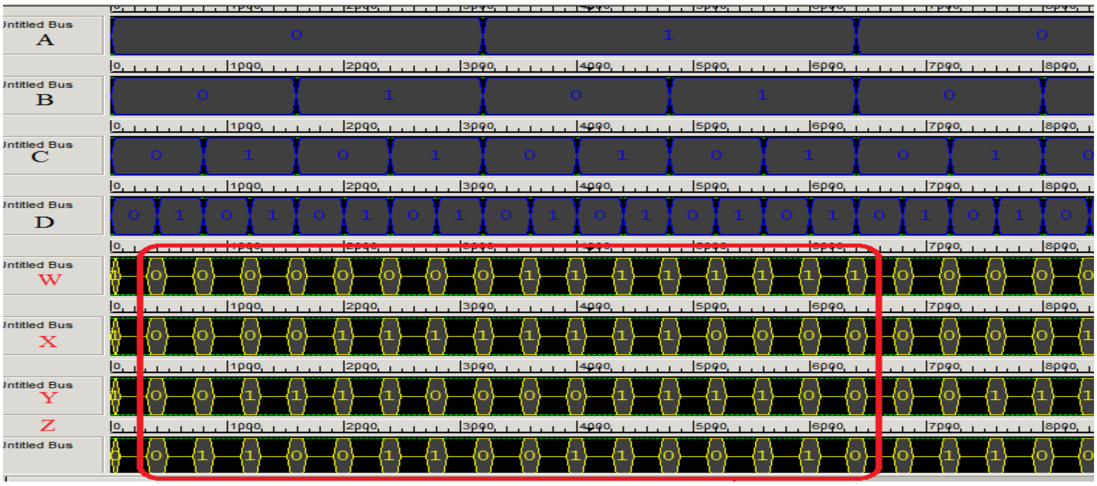
(b)

Figure 7: (a) Logic diagram of gray to binary Code Converter (b) Truth table

We have proposed a novel QCA layout of grey code converter shown in Fig. 7(c). A simulation result of the proposed design is shown in Fig. 7(d), have been checked using QCADesigner. Simplicity of the proposed grey code converter is that simple arrangement of QCA 90^0 cells, with proper latency. The circuit area of grey code converter is $0.16um^2$, latency 0.5, and circuit complexity is 137-cells. The conversion between gray to binary and binary to gray is very common in digital systems. The simulation results of grey code converter have been checked with QCADesigner tool and logical truth table. The advantage of computing with the truth table is that it ensures the computing process to generate same outputs.



(c)



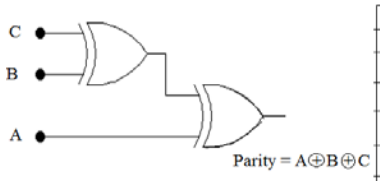
(d)

Figure 7: (c) QCA layout of gray code (d) Simulation result

3.4 Parity generator and Parity Checker:

In digital communications, a large amount of data is transmitted and received across various mediums. Mostly during the transfers, some noise gets added to the data and makes it difficult to recover signal. To Make the data recovery easier an extra bit is appended to the binary (0, 1) message to make the 'logic 1' count even or odd. The logic diagram of parity generator is shown in Fig. 8(a) consists of three message inputs A, B, and C, whose output is 'P'. This extra bit is known as parity_bit 'P' and used for error detection. In case of even parity, the parity bit is chosen, so that the total number of 1's in the coded message is even. Alternatively, odd parity can be used in which the total number of 1's in the coded message is made odd. During transfer of information, the message at the sending-end is applied, to a parity generator where the parity pit is generated. At the receiving-end, a parity checker is used to detect single bit error in the transmitted data word by regenerate the parity bit in the same fashion as the generator and then compare with the parity bit transmitted. Truth table of parity generator is shown in Fig. 8(b).

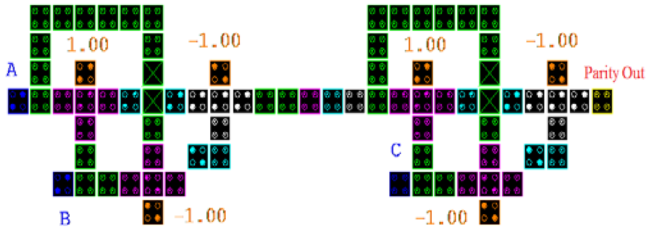
Parity generator proposed by some authors consists of large area and circuit complexity in [9]. The proposed XOR gate has used to implement an efficient parity generator, which consists of less circuit complexity of 79-cells and less circuit area 0.09um^2. The QCA layout of parity generator is shown in Fig. 8(c). Simulation results demonstrates that the latency of proposed layout is 3 clock cycles and hence parity bit 'P' can be found after an interval of 3 clocks from the inputs A, B, and C, shown in Fig. 8(d).



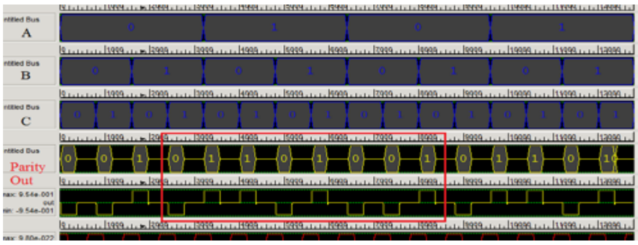
Message bits			Parity out
A	B	C	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(a) (b)

Figure 8: (a) Logic diagram of parity generator (b) Truth table



(c)

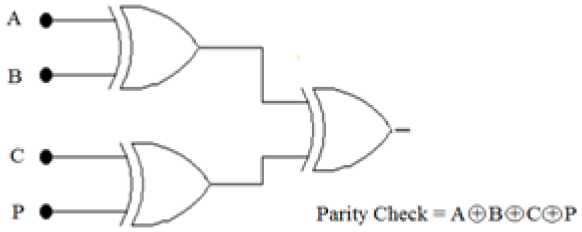


(d)

Figure 8: (c) QCA layout of parity generator (d) Simulation results

Parity systems are implemented on both transmitters and receivers. The transmitter is responsible for generating the parity bit 'P'. Thus, the 3 message inputs A, B and C along with parity bit 'P' is transmitted by the transmitter. The receiver is responsible for detecting the message including the parity bit 'P' shown in logic diagram Fig. 9(a). If message does not meet the parity check, an error flag is generated and transmitter is requested to re-transmit the packet. The truth table for parity checker is shown in Fig. 9(b).

We have propose an a novel design of even checker by introducing the proposed novel XOR gate, which consists of less circuit complexity 92-cells and area 0.12 um^2 . QCA layout of parity checker is shown in Fig. 9(c). Simulation result of proposed design is shown in Fig. 9(d).

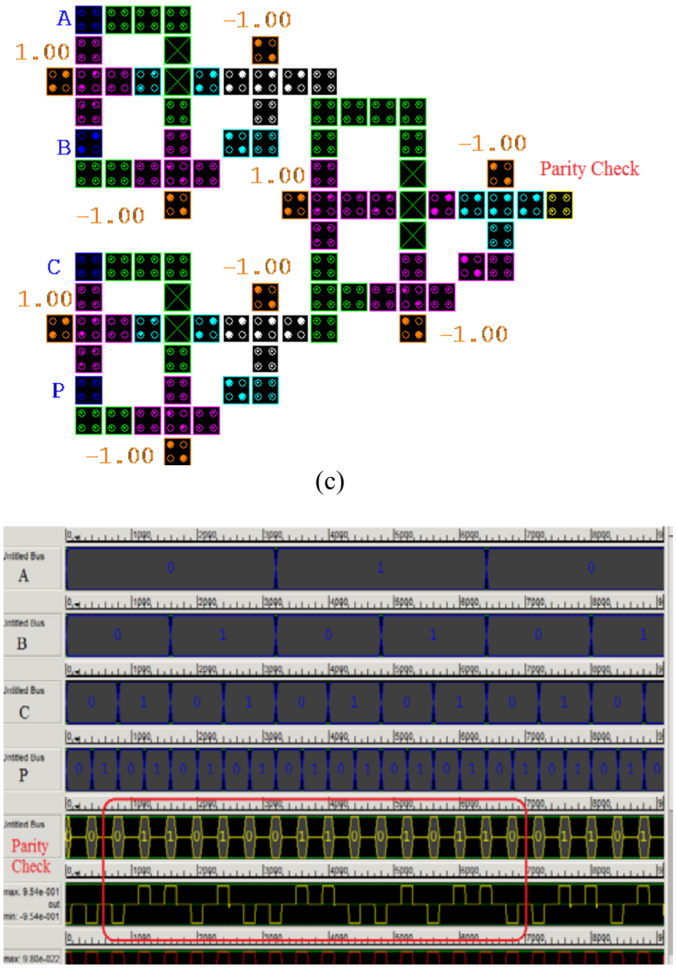


(a)

Receiveing message inputs				Erroe Check
A	B	C	P	Check
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

(b)

Figure 9: (d) Logic diagram of parity checker (b) Truth Table



(c)

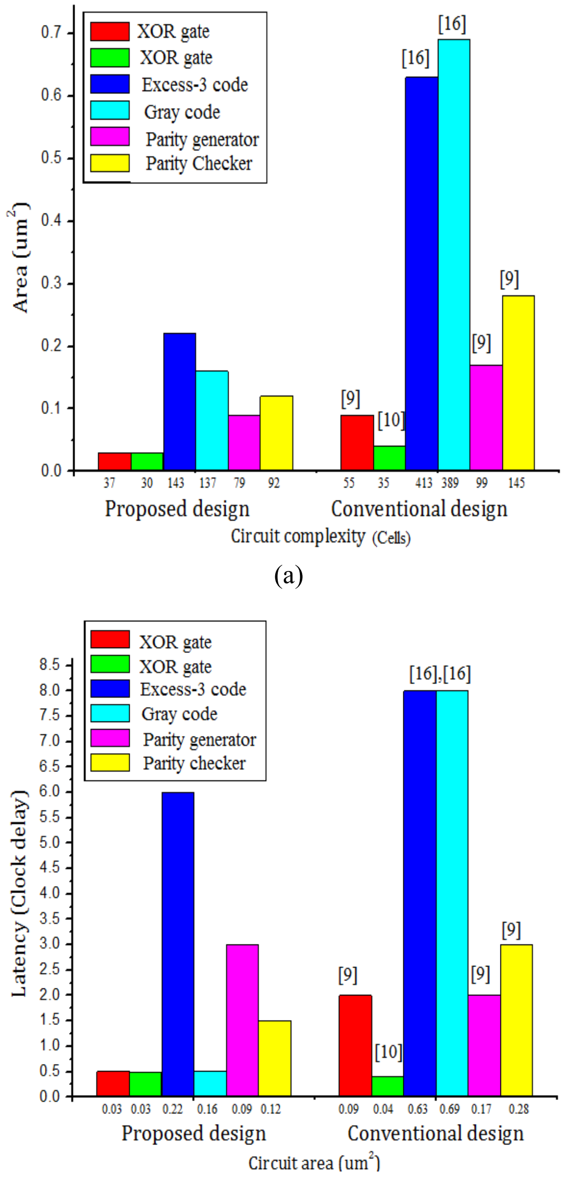
(d)

Figure 9: (c) QCA layout of parity checker (d) Simulation results

4. Comparison

In this section, the proposed QCA XOR gates and its implementations as code converters have compared with conventional designs with regards latency, occupied area, and number of used QCA cells. The efficiency and robustness of these code converters has verified according to circuit complexity, latency, and the area.

Table 3 gives the comparative study of the proposed and conventional XOR gates and various code converters with regard to various parameters. The proposed XOR gates consist of less area, circuit complexity as compared to previous designs [8-10]. Researcher [16] has proposed the QCA implementation of code converters like gray code and Excess-3 code. The design needs either needs either coplanar crossovers or multiple layers to implement. We have proposed efficient like gray code and excess-3 code, which consists of fraction of area and less circuit complexity. In addition, it can be visualized that the proposed QCA layouts of code converters are better than the conventional structures because of low cell count and less multilayer connections. The proposed designs show linear decrease of circuit area and improve the circuit efficiency. Parity generator and checker shown in [9] consist of very large area, complexity, and latency. We have proposed novel design of parity generator with less area, circuit complexity, and latency shown in comparative study Table 1. Fig. 9(a) shows that the proposed circuits are more efficient than conventional circuits [8-10], [16], in terms of area occupied. The proposed designs show linear decrease of circuit area and improve the circuit efficiency. Fig. 9(b) shows the comparative study of various structures in terms of latency (clock delays) and area occupied. The performance of proposed designs is more efficient than conventional designs



(a)

(b)

Figure 9 : (a) Efficiency of proposed designs versus conventional designs (b)comparison of area versus latency(clock delay)

Table 1: Comparative study of XOR structures and code converters with the results available in literature [8-10], [16]

QCA structures	Previous structures				Proposed structure		
	Complexity	Area	Latency		Complexity	**Area**	**Latency**
XOR gate [9]	55 cells	0.09 μm^2	2	Fig. 5(c)	37 cells	**0.03 μm^2**	0.5
[10]	35 cells	0.04 μm^2	0.4	Fig. 5(e)	30 cells	**0.03 μm^2**	0.5
Excess-3 code [16]	413 cells	0.63 μm^2	8	Fig. 6(c)	143 cells	**0.22 μm^2**	6
Gray code [16]	389 cells	0.69 μm^2	8	Fig. 7(c)	137 cells	**0.16 μm^2**	0.5
Parity generator [9]	99 cells	0.17 μm^2	2	Fig. 8(c)	79 cells	**0.09 μm^2**	3
Parity checker [9]	145 cells	0.28 μm^2	3	Fig. 9(c)	92 cells	**0.12 μm^2**	1.5

5. Conclusion

The proposed designs are a solution for implementations of QCA based circuits using minimum number of QCA cells, lesser clock delays and reduced area. This paper presents design of improved QCA XOR gates and its implementations. The proposed XOR gates have been used to implement complex structures. In this paper, novel code converters like excess-3 code, gray code, parity code generator, and checker have attempted by using proposed XOR gates. The novel code converters occupies less area circuit complexity, circuit stability and are more efficient than previous structures. All error detection, correction-controlling mechanisms has been studied. However, it has been found that parity generator and chacker is most efficient and simplest technique used for error detection & correction mechanism in long distance communication.

6. Acknowledgments

The author gratefully acknowledges very helpful conversations with Prof. (Dr.) G.M Bhat

References

[1] C.S. Lent, P.D. Tougaw, W. Porod, and G.H. Bernstein., “Quantum cellular automata”, Nanotechnology, vol. 4, pp. 49–57, 1993

[2] University of Notre Dame. Department of electrical engineering. web: http://www.nd.edu/ qcahome/.

[3] Konrad Walus and Graham A. Julien., “Design tools for an emerging SoC technology: Quantum-dot cellular automata”, In Proceedings of the IEEE, vol. 96, pp. 1225–1244, 2006

[4] Lent C. S. and Tougaw P. D, “Lines of interacting quantum-dot cells: a binary wire”, Journal of applied Physics, vol. 74, pp: 6227-33, 1993

[5] Askari, M., M. Taghizadeh, Kh. Fardad, “Digital Design using Quantum-Dot Cellular Automata (A Nanotechnology Method),” in Proc. International Conference on Computer and Communication Engineering, pp. 952-955, 2008

[6] Askari, M., M. Taghizadeh, Kh. Fardad, “Design and Analysis of a Sequential Ring Counter for QCA Implementation,” in Proc. International Conference on Computer and Communication Engineering, pp. 933-936, 2008

[7] Cho, H., E.E. Swartzlander, “Adder Designs and Analyses for Quantum-Dot Cellular Automata,” IEEE Transactions on Nanotechnology, vol. 6(3), pp. 374-383, 2007

[8] M R Beigh, M. Mustafa, F. Ahmad, “Performance Evaluation of Efficient XOR Structures in Quantum-Dot Cellular Automata (QCA),” Circuits and Systems, vol.4, pp.147- 156, 2013

[9] M R Beigh, M. Mustafa, “ Design and implementations of quantum-dot cellular automata base novel Parity generator and checker circuits with minimum cell complexity and cell count,” Indian Journal of Pure & Applied Physics, vol. 51, pp. 60-66, 2013

[10] Wani Shah Jahan, Peer Zahoor Ahmad, Peer M A3, Khan K A, “ Circuit Nanotechnology: QCA Adder Gate Layout Designs, “IOSR Journal of Computer Engineering (IOSR-JCE),vol 16, PP 70-78, 2014

[11] Omar Paranaiba, et al., “Neural Network Simulation and Evolutionary Synthesis of QCA Circuits,” IEEE Transactions on Computers, vol. 56(2), pp. 191-201, 2007

[12] Tougaw P. D, and Lent C. S, “Logical devices implemented using quantum cellular Automata,” Journal of Applied Physics, vol. 75(3), pp. 1818-1825, 1994

[13] G. Toth, C. S. Lent, “Quasiadiabatic switching for metal-island quantum-dot cellular automata,” Journal of Applied Physics, vol. 85, no. 5, pp. 2977-2984,1995

[14] Lent, Craig S., Mo Liu, and Yuhui Lu., “Bennett clocking of quantum-dot cellular automata and the limits to binary logic scaling,”Nanotechnology, vol. 17, no. 16, pp. 4240, 2006

[15] Liu, Weiqiang, et al., “Are QCA Cryptographic Circuits Resistant to Power Analysis Attack?,”IEEE transactions on nanotechnology, vol. 11, no. 6, pp. 1239-1251, 2012

[16] J. Iqbal, F. A. Khanday, N. A. Shah, “Efficient Quantum Dot Cellular Automata (QCA) Implementation of Code Converters,” Communications in Information Science and Management Engineering, vol. 3 Iss. 10, PP. 504-515, Oct. 2013

[17] Firdous, M. Mustafa, W.A. Nisar, and A.M. Feroz, “A novel idea of pseudo-code generator in quantum- dot cellularautomata (QCA),” Int. J. Simul. Multisci. Des. Optim, vol. 5, A04, 2014

[18] S.Karthigai lakshmi, G.Athisha, “Efficient Design of Logical Structures and Functions using Nanotechnology Based Quantum Dot Cellular Automata Design,” International Journal of Computer Applications (0975 – 8887) Volume 3 – No.5, June 2010

[19] P.D Tougaw and C.S Lent, “logical devices implementation using quantum dot cellular automata,”journal of applied physics, 75:1818, 1999

[20] Vishnu Teja Ch, Satish Polisetti and K. Santhosh, “ QCA based multiplexing of 16 arithmetic & logical subsystem-a paradigm for nano computing” ,3rd Annual IEEE-International conference on Nano/Micro Engineering Molecular System, Hainan Island, China, January 2008