International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:3, 2014

Design and Optimization of Parity Generator and Parity Checker Based On Quantum-dot Cellular Automata

Santanu Santra, Utpal Roy

***Abstract*—**Quantum-dot Cellular Automata (QCA) is one of the most substitute emerging nanotechnologies for electronic circuits, because of lower power consumption, higher speed and smaller size in comparison with CMOS technology. The basic devices, a Quantum-dot cell can be used to implement logic gates and wires. As it is the fundamental building block on nanotechnology circuits. By applying XOR gate the hardware requirements for a QCA circuit can be decrease and circuits can be simpler in terms of level, delay and cell count. This article present a modest approach for implementing novel optimized XOR gate, which can be applied to design many variants of complex QCA circuits. Proposed XOR gate is simple in structure and powerful in terms of implementing any digital circuits. In order to verify the functionality of the proposed design some complex implementation of parity generator and parity checker circuits are proposed and simulating by QCA Designer tool and compare with some most recent design. Simulation results and physical relations confirm its usefulness in implementing every digital circuit.

***Keywords*—**Clock, CMOS technology, Logic gates, QCA Designer, Quantum-dot Cellular Automata (QCA).

I. Introduction

IN last few decades, scaling the feature size and increasing the processing power have been successfully implemented by conventional lithography based on VLSI design [1]. Nanotechnology attracts much more attention from the researcher nowadays. As because the current CMOS technology faces different challenging problems like high power consumption and cannot ignored the difficulties in feature size reduction. A useful summarization report of future technologies has been published by ITRS [2]. Trusting on the unique properties of electronic devices at Nano level feature size, nanotechnology unlocks new prospects for computing systems and devices. QCA proposed by Lent [3], is an emerging technology that offers an innovative approach for computing at nano-scale by monitoring the position of a single electron. QCA technology has feature like high speed, extremely low power, fast switching speed [4] and much more dense circuit [5], which increases clock frequency of logic circuits and decreases the size and estimated power consumption of those circuits [6].

S.Santra is with Department of Computer Science & Engineering, Bengal Institute of Technology and Management, Santiniketan-731236, West Bengal, India (phone: 9474895712; e-mail: santrasantanu@gmail.com).

U. Roy is with Department of Computer & System Science, Visva-Bharati University, Santiniketan-731235, West Bengal, India (e-mail: roy.utpal@gmail.com).

XOR gate is probably a vital part of many modern designs of complex digital circuits. The research article propose an optimized XOR gate by which any complex digital circuit can be implemented, here we implement optimized parity generator and checker circuit using the proposed XOR gate.

Although the article assumes metal-based QCA designing, the underlying principles also apply to molecular QCA. There are various clocking schemes such as wave clocking which may be more suitable for molecular QCA.

The rest of the research article is organized as follows: In Section II, the fundamental of QCA technology and design approaches is described. Section III shows the design and implementation of XOR gates. Section IV compares the proposed design of XOR gate with some of most recent implemented issues. Section V describes the design and simulation of parity generator and parity checker using the proposed XOR gate for QCA circuits. In Section VI described comparative study of some most recent implementation. Section VII concludes the paper.

II. QCA Technology And Design Approaches

Quantum technology has gradually applied in various fields, Quantum dot cellular automata are projected as a promising nanotechnology for future nano ICs. QCA technology is based on the interaction of bi-stable [7], [8] QCA cells constructed from four quantum dots. The cell is charged with the help of two free electrons which are able to tunnel between adjacent dots. These electrons tend to occupy antipodal sites as a result of their mutual electrostatic repulsion. Thus there exist two equivalent energetically minimal arrangements of the two electrons in the QCA cell, but no current routed out of the cell [9], [10] as shown in Fig. 1.

A. QCA Logic Device

These arrangements are denoted as cell polarization P=+1 and P=-1, these two cell polarization P= +1 which represents "1" and P= -1 represents "0", binary information is encoded in the charge configuration of the QCA cell [11], [12]. Unlike conventional logic circuits in which information is transferred by electrical current, QCA operates by the Columbic interaction that connects the state of one cell to the state of its neighbors, which results as information transformation. One of the primary logic gates in QCA is the majority voter (MV). The basic logic function in majority voter is shown in (1). MV can be realized by only five QCA cells, as shown in Figs. 2 (a) and (b) shown the logical block diagram of a majority gate.

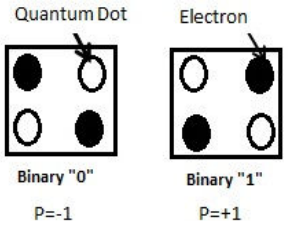


Fig. 1 Quantum cellular automata

$$MV(A,B,C) = AB + BC + CA \quad (1)$$

Logical AND (Fig. 3 (a)) and OR (Fig. 3 (b)) function can be implemented from the majority voter by fixing one input (control input) permanently to a 0 or 1. The inverter is another one basic gate in QCA and is shown in Fig. 3 (c). The interconnected fabric bus is shown in Fig. 3 (d).

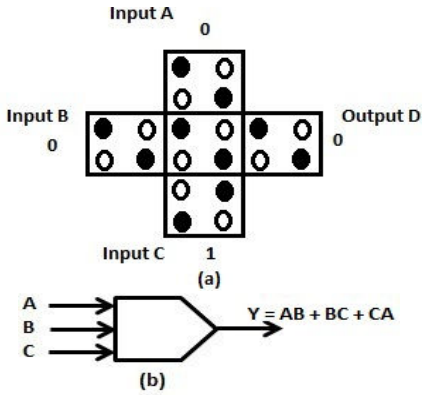
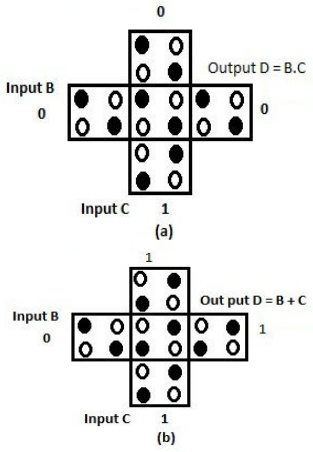


Fig. 2 (a) A QCA majority voter gate, (b) Basic block diagram of a QCA MV gate



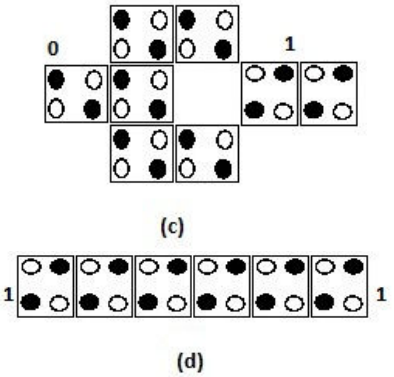


Fig. 3 (a) QCA AND gate, (b) QCA OR gate, (c) QCA inverter gate and (d) QCA bus

Several advantages can get using QCA technology like it is " edge driven" means an input is brought to an edge of a QCA block, that is evaluated and output at another edge. This also means that no power lines need be routed internally.

QCA systems should be very low power, because there is no current flowing. Only enough energy needs to add to lift the electrons from their ground states. The required space for a circuit is very small as because the size of QCA cells are very small.

B. QCA Clocking

QCA clocking is accomplished by four different and periodic timing state as shown in Fig. 4. A QCA circuit is divided into zones and every zone is maintained in a phase. The use of quasi-adiabatic switching technique for QCA circuits requires a 4-phased clocking signal, generated by CMOS wires buried under the QCA circuitry for modulating the electric field. Four phases clocking are treated as Relax, Switch, Hold and Release [13], [14]. During the first phase the actual computation takes place therefore the inter-dot barrier is slowly raised and a cell attains a definitive polarity under the influence of its neighbors. During the second phase, barriers are high and a cell remains its polarity. During the third phase, barriers are lowered and a cell losses its polarity and finally in the last phase, there is no inter-dot barrier and a cell remains un-polarized. Timing zones of a QCA circuit are arranged by the following periodic execution of these four clock phase and a latch must be present in between two clocking zones. A signal is latched when one clocking zone goes into Hold phase and acts as input to the subsequent zone. Inherent pipelining [15], [16] can be provided by clocking system and multi-bit information transfer for QCA can be produced by signal latching. As a zone in the Hold phase is followed by a zone in the Switch phase. Designs are partitioned along one dimension, thus effectively creating columns of clocking zones. The clocking signal is applied through an underlying CMOS circuitry which produces required electric field to modulate the tunneling barrier of all cells in the zones [17], [18].

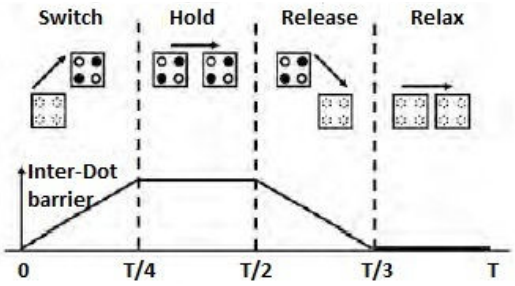


Fig. 4 Four phases QCA Clocking

C. QCA Faults

Manufacturing of QCA based structure are suffer from various types of faults which is categorized and illustrated as follows.

1. Displacement faults, the defective quantum cell are displaced from its original position. In Fig. 5 (b) shows the quantum cell with input A is displaced to the upper direction by Δnm from its original position.
2. Misalignment defect, direction of the defective quantum cell is not properly aligned. Fig. 5 (c) shows input cell A is misaligned to the right direction by Δnm from its original position.
3. Omission fault, the defective cell is missing as compared to defect-free case, as shown in Fig. 5 (d) where quantum cell with input A is not present.

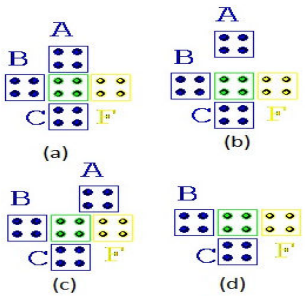


Fig. 5 (a) fault-free majority gate, (b) displacement fault, (c) misalignment fault and (d) omission fault

D. QCA Designer

Complex QCA circuit can be simulated using most standard platforms QCA Designer, developed at the ATIPS laboratory, University of Calgary. Now QCA Designer has attracted some important new developer including top researchers from university of Notre Dame. The current version of QCA Designer 2.0.2 has three different simulation engines. One is a digital logic simulator, where each is either fully polarized or null. Second one is nonlinear approximation engine, uses to design the nonlinear cell-to-cell response to iteratively determine the stable state of quantum cells. Third one is a two-state Hamiltonian to form an approximation of full quantum mechanical model of such a system. These three different engines have a different and important set of benefits and drawbacks. Each of these simulator types can perform an exhaustive verification of the system or a set of user selected input vectors [19], [20].

QCA designers which use to design basic gates and complex logical circuits require a rapid and accurate simulation and design layout to determine the functionality of QCA circuits. As well several simulation engines facilitate rapid and accurate simulation. The main problem to implement more accurate simulations is the lack of experimental data for QCA systems with a large number of cells.

III. XOR Gate Implementation

In [21], authors proposed three approaches to design a XOR gate, where first one require 44 no. of cells and four no. of clock are applied, second approach require 55 no. of cells and four no. of clock and third approach require 62 no. of cells and four no. of clock to design a XOR gate. In [17] authors proposed seven different implementation of XOR gate, where first one require 34 no. of cells, four no. of clock are required to execute the circuit and here 4 no. of majority gates are required to complete the design of the circuit. In [18] authors proposed a XOR gate, which required 58 no. of quantum cells and 3 no. of clock and 3 no. of majority gates are required to implement the circuit.

A. QCA Implementation

Basic logic gates AND and OR realized by fixing the polarization to any one of the input of the majority gate to either P=-1 treated as logic "0" or P=+1 treated as logic "1". The NAND can be implemented by inverting the output of AND gate Fig. 6 (a) shows the logic design of a NAND gate using QCA cells. Similarly the NOR gate is implemented by connecting OR gate followed by an inverting gate, Fig. 6 (b) shows the implementation of NOR gate by arranging the last two cells such that it acts like an inverting gate followed by the majority voter gate. These two cells inverting circuits can minimize the area and complexity of any circuits.

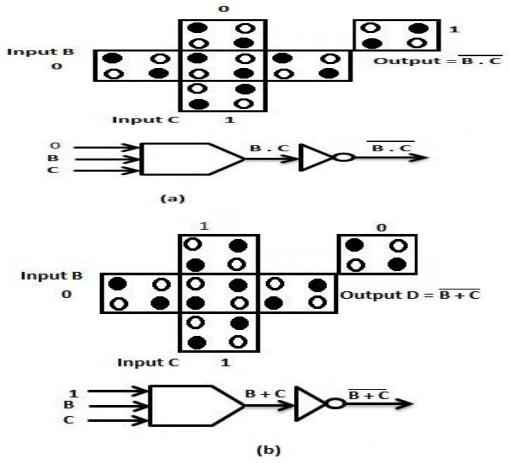


Fig. 6 QCA implementation (a) NAND gate (b) NOR gate

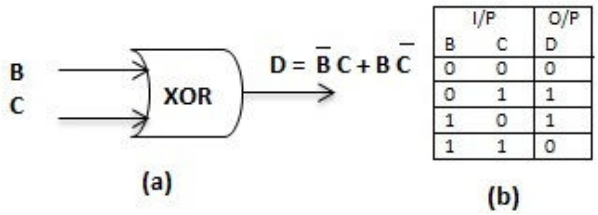
B. QCA Implementation of Proposed XOR Gate

The basic gates like AND, OR and NOT and the universal gates NAND and NOR are require to design a digital logic

circuits. In addition of these gates Exclusive-OR (XOR) and Exclusive-NOR (XNOR) gates are also used to design digital circuits. The XOR and XNOR gates are particularly useful in arithmetic operations as well as error-detection and correction circuits. These gates are usually found as two-input gates. There is no multiple-input XOR/XNOR gates are available since they are complex to fabricate with hardware.

The XOR gate can give the following logic operation shown in (2).

$$B \oplus C = \bar{B}C + B\bar{C} \quad (2)$$



I/P		O/P
B	C	D
0	0	0
0	1	1
1	0	1
1	1	0

(b)

Fig. 7 (a) XOR gate Graphical Symbol (b) XOR gate truth table

The representing symbol and the truth table of XOR gate is shown in Figs. 7 (a) and (b). In digital logic XOR is a logical value depending on the two inputs and the logical value of XOR is true only if odd number of input is true otherwise the logical value is false. This forms a fundamental logic gate in many operations to follow. In digital logic if the specific type of gate is not available then it may be constructing by other available gates. An XOR gate can be trivially constructed from the basic gate AND, OR and NOT gates. However, this approach requires five gates of three different kinds. The expression get from Fig. 8 is shown as (3).

$$D = \bar{B}C + B\bar{C} \quad (3)$$

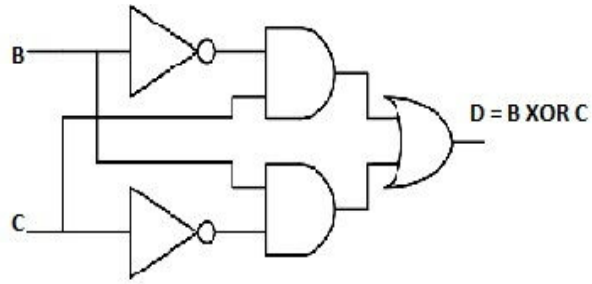


Fig. 8 Implementation of XOR gate

We propose the QCA design and layout of two different approaches of XOR gates based on basic logic gates arrangements, the proposed design layout is shown in Figs. 9(a) and (b), first one is used where both inputs are outside of the circuit, as because this design layout require 51 no. of quantum cells and 5 no. of clock pulses and an area of approximately 16524nm^2. In second approach of XOR gate implementation design layout require only four clock cycle and consist of only 30 no. of QCA cells(including inputs and output cells) and an area of 9720 nm^{2}approximately also there is no cross over in the circuit. Simulation results of these two proposed layouts are shown in Figs. 10 (a) and (b) respectively.

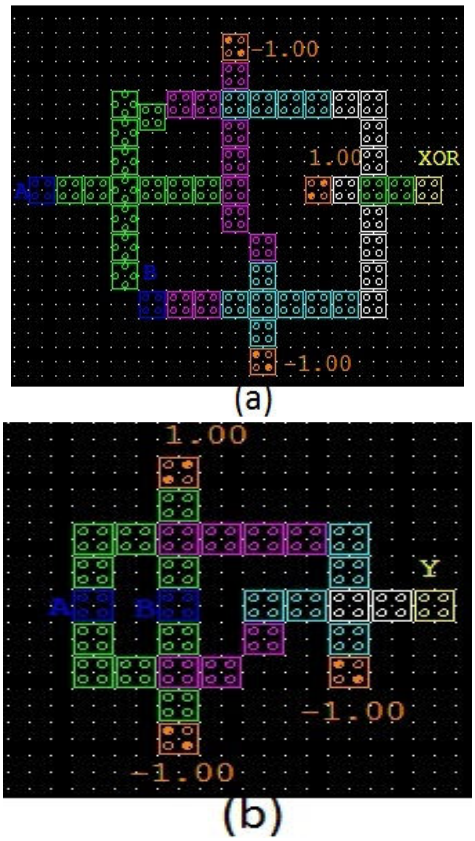
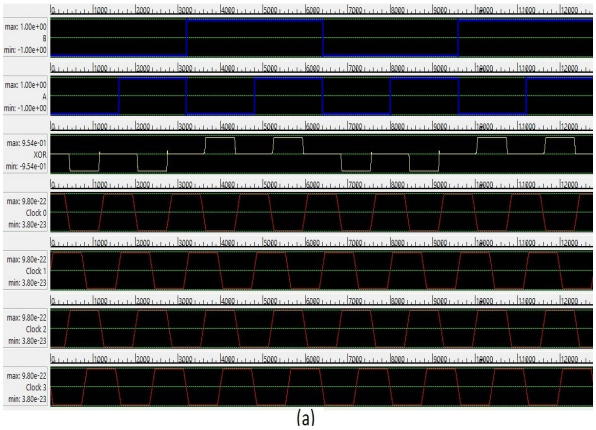
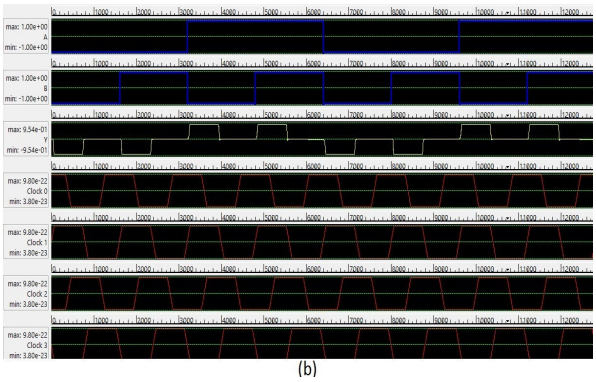


Fig. 9 Proposed QCA XOR design layouts.



(a)



(b)

Fig. 10 Simulation results of proposed XOR design layout

IV. Comparative Study of XOR Gate

For the proposed circuit layout and functionality checking, a simulation tool for QCA circuits QCA Designer [21] version 2.0.3 is used. The following parameters are used for a bitable approximation: cell size=18 nm, clock high = 9.800000e-022 J, clock low=3.800000e-023 J, Dot diameter= 5nm, Space between two cells = 2nm.Most of the mentioned parameters are default values in QCA Designer. Fig. 10 displays simulation result of the proposed design of two input XOR gates.

Table I gives the comparison of proposed design with several previous designs [17], [18], [22]-[23]. It is evident from Table I that the proposed designs are effective in terms of cell count, majority gate, delay, and area to simulate circuits.

TABLE I
Comparative Study ofProposed Design with Some Most Recent Design Layouts

XOR\Parameters		Cells	Majority gates	Delay	Total area(nm^2)	Area used(nm^2)	%of area used
As in [17]	a	41	3	4	35640	13284	37.27
	b	55	4	4	60588	17820	29.41
	c	62	4	4	57024	20088	35.22
As in [18]	a	34	4	4	25920	11016	42.50
	b	54	3	4	51840	17496	33.75
	c	52	4	4	50544	16848	33.33
	d	52	4	4	50544	16848	33.33
	e	48	3	3	35640	15552	43.63
	f	54	3	4	48600	17496	36.0
	g	42	3	3	29160	13608	46.66
As in [22]		58	3	3	50544	18792	37.18
As in [23]		121	3	3	157464	39204	24.89
proposed	a	51	4	5	58320	16524	28.33
	b	30	3	4	28044	9720	34.66

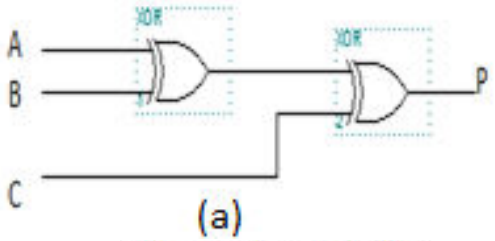
V. Parity Generator and Parity Checker Circuits

Exclusive-OR circuits are very convenient in design of parity bits for error-detection. A parity bit is used for detecting errors during transmission of binary information. A parity bit is an extra bit included with binary message to make the total number of 1's in this message (including parity bit) either odd or even. Including parity bit, message transmitted and checked at the receiving end for errors. An error is detected if the checked parity does not correspond with the one transmitted. The design layout that generates parity bit at the transmission node is called a parity generator and the design layout that checks the parity at the receiver side is called a parity checker.

In this article, a three bit message to be forward with an even parity bit and the truth table is shown in Fig. 11 (b). The three bits A, B and C constitute the message and the output of even parity generator circuit P is transmitted. Even parity generator layout produce parity bit P = 1 while there are odd numbers of inputs are high. Otherwise parity bit P = 0. The expression of even parity generator can be written in equation 4.

$$P = A \oplus B \oplus C \quad (4)$$

The basic building block for even parity generator circuit is shown in Fig. 11 (a).



(a)

Message			Even Parity Bit
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

(b)

Fig. 11 Parity Generator (a) Parity generator using XOR gate (b) Input and output vector

The design layout of even parity generator is shown in Fig. 12 (a), which is implemented by 2 numbers of two inputs XOR gates and consists of only 60 numbers cells and the circuit area is 19440 nm^2. Latency of the proposed design layout is 1 clock cycle and the parity bit P generates after one clock interval from the input vectors.

Parity generator reported in [23] has 99 no of cells count as compare to proposed implementation with same latency. Thus, the proposed design is simple in implementation, uses lesser number of cell counts and consumes less area and less power.

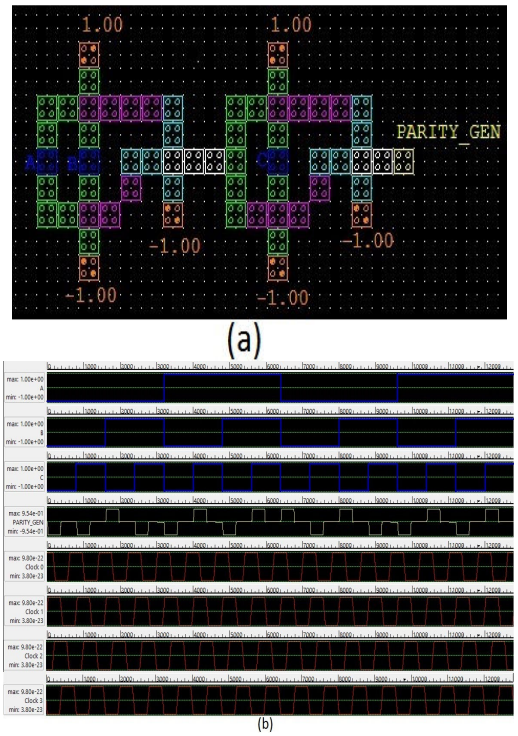
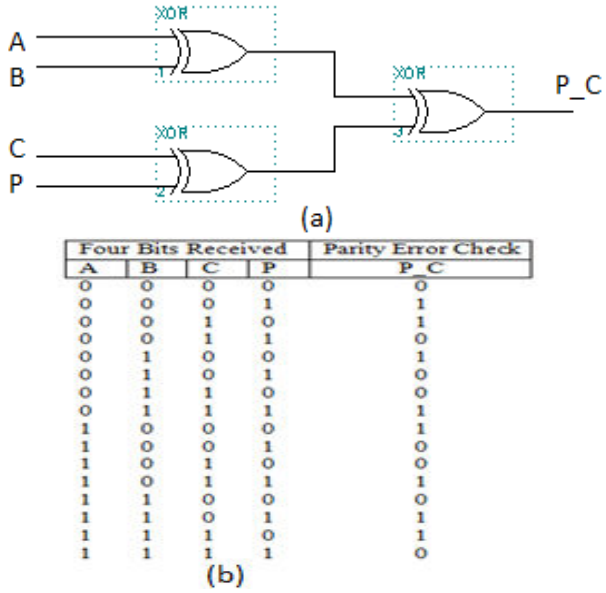


Fig. 12 Even Parity generator layout and Simulation results

3 bits message and one bit even parity are transmitted to their destination, where a parity checker circuit checks the possible errors in the transmission. Since the message was transmitted with even parity bit, the received four bits message must have an even number of 1's. Parity checker circuit generates an error. The logic diagram of 4 bit parity checker and the characteristic table is shown in Figs. 13 (a) and (b) respectively.

The design layout of Parity Checker is shown in Fig. 14 (a), which is implemented by 3 numbers of two inputs XOR gates and consist of only 117 cells and circuit area is 37908 nm^2. The simulation results of Parity checker circuit is shown in Fig. 14 (b). Latency of the proposed parity checker design layout is 2 clock cycles and hence the parity checker bits can be founded after 2 clock interval after receiving the 4bits message. The logic implementation of parity checker can be represented as in (5):

$$C = A \oplus B \oplus C \oplus P \quad (5)$$



Four Bits Received				Parity Error Check
A	B	C	P	P_C
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

Fig. 13 Parity Checker (a) Parity Checker using XOR gate (b) Input and output vector

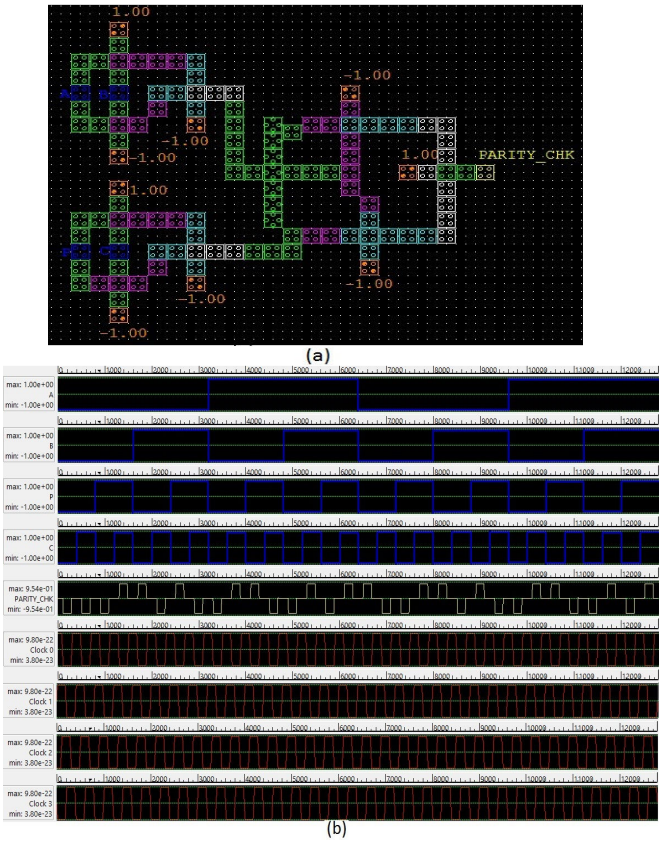


Fig. 14 (a) 4 bits QCA Parity checker circuit(b) Simulation result of 4 bits Parity Checker

Parity checker design reported in [23] has 299 no of cells and four times area and as compared and as reported in [24]

has 145 no. of cells count as compare to proposed implementation with same latency. Thus, the proposed design is simple in implementation, uses lesser number of cell counts and consumes less area and less power.

VI. Comparative Study of Parity Generator and Parity Checker

In this study we implement Parity Generator and Checker circuit to enable error checking in data transfer network using proposed XOR circuit. The novelty of this gate besides parameter like delay, majority gate and area is minimal in comparison to other design as proposed in the literature [17], [18] and [22], [23]. The complexity parameters such as cell count, time delay and area consumption of QCA circuit has been calculated with QCA Designer [21]. Attempt is taken to design parity generator and checker circuit with the proposed XOR gate only.

The comparison between various parameter of the circuit such as no. of cell count, time delay and total area consumption with the work available in literature has been compared.

Complexity in terms of cell count, delay and area of CA circuits can be easily obtained by QCA Designer. Table II demonstrates the comparison between the QCA based parity generator and checker with the proposed XOR gate. The result indicate that no. of cell count is considerably low in our design.

TABLE II
Comparative Study of Parity Generator and Parity Checker Circuit

		Complexity		Area (nm^2)		
		No. of cells	delay	Circuit	wasted	%of used
As in [24]	P G	99	3	113724	81648	28.20
	P C	145	3	197316	150336	23.80
As in [23]	PG	NA	NA	NA	NA	NA
	P C	299	3	428652	331776	22.60
Proposed	PG	60	2	52488	33048	37.04
	PC	117	2.25	135432	97524	27.99

*PG indicates Parity Generator and PC indicate Parity Checker.

VII. Conclusion

This article represents the design, layout and simulation of parity generator and checker circuits based on novel XOR circuit configuration. An optimal design for XOR base parity generator and checker circuits has been proposed. The proposed implementation is simulated using QCA based circuits simulation tools i.e. QCA Designer [25]. This design is efficient in terms of cell count, area. Moreover considering the less numbers of cell count, area and power consumption.

References

[1] R. Compano, L. Molenkamp, and D.J. Paul, European Commission IST Programme,FET Conf., 1999.
[2] S.Timarchi,K.Navi, *IEEE Trans. On Instrumentation and Measurement,* Vol. 58 Issue:9, pp. 2959-2968 2009. doi:10.1109/TIM.2009.2016793.
[3] A.O.Orlov,I.Amlani,G.H.Bernstein,C.S.Lent,G.L.Snider,Science Vol. 277 no. 5328 pp. 928-930, 1997.
[4] P.Tougaw,C.Lent, Journal of Applied Physics vol.75 no.3 pp. 1818 – 1825. 1994.
[5] C. Lent, P. Tougaw, W. Porod, and G. Bernstein,, Nanotechnology vol. 4 pp. 49-57, 1993.
[6] J D JacksonJhon Wiley and Sons, 1999.
[7] J. Huang, M. Momenzadeh, M. Ottavi, L. Schiano, F. Lombardi,Internal report, (2004).
[8] M. Momenzadeh, J. Huang, M. Ottavi, N. Park, F. Lombardi,Internal report, (2004).
[9] J. Timler and C.S. Lent, Journal of Applied Physics, vol. 91, pp. 823-831, (2002).
[10] C. S. Lent, P. D. Tougaw and W. Porod, Proc. of the Workshop on Physics and Computing, pp. 5–13, (1994).
[11] C. G. Smith, Science, vol. 284, 274, (1999).
[12] S. Santra, U. Roy,International Journal of Computer, Information Science and Engineering , WASET,Vol:8 No:1, PP 168-173 (2014)
[13] C. S. Lent and P. D. Tougaw, Proc. IEEE, vol. 85, no. 4, pp. 541–557, Apr. (1997).
[14] K. Hennessy and C. S. Lent, J. Vac. Sci. Technol. B, Microelectron. Process. Phenom., vol. 19, no. 5, pp. 1752–1755, Sep. (2001).
[15] D. A. Antonelli, D. Z. Chen, T. J. Dysart, X. S. Hu, A. B. Kahng, P. M. Kogge, R. C. Murphy and M. T. Niemier, Design Automation Conference (DAC), pp. 363–368, (2004).
[16] K. Walus, G. A. Jullien and V. S. Dimitrov, Proceedings of Asimolar Conference, (2003).
[17] M. Mustafa, M R Beigh, Indian Journal of Pure & Applied Physics, vol. 51 pp.60-66,(2013).
[18] M.R.Beigh,M.Mustafa,F. Ahmad, Circuits and Systems vol.4 pp. 147-156 (2013).
[19] K. Walus, T. Dysart, G. Jullien, and R. Budiman, IEEE Trans. Nanotechnol., vol. 3, no. 1, pp. 2629, Mar. 2004.
[20] Kyosun Kim, Kaijie Wu, and Ramesh Karri, IEEE transactions on computer-aided design of integrated circuits and systems, vol. 26, no. 1, January 2007.
[21] W Wang, K Walus, G AJullien, In proceedings of IEEE Conference on Nanotechnology, PP 461-464, 2003.
[22] H.S. Jagarlamudi, M Saha, PK Jagarlamudi, World Academy of Science, Engineering and Technology, Vol. 60 PP 486-490 2011.
[23] Vishnu T C, Polisetti&Santosh K, 3rd Annual IEEE-International conference on Nano/Micro Engineered Molecular Systems, China (2008).
[24] M Mustafa, M R Beigh, Indian Journal of Pure & Applied Physics, Vol. 51, PP 60-66 (2013).
[25] V.C. Teja, S. Poliseti, S. Kasavajjala, 3rd IEEE Int. Conf. on Nano/Micro Engineered and Molecular Systems, PP 758-763 (2008).