Design of Practical Parity Generator and Parity Checker Circuits In QCA

K. Raghavendra[1], Nalla Sarika[2], Nagandla Deepthi[2], Muppidi Vahini[2]

[1,2]Department of Electronics and Communication Engineering

[1,2]Malla Reddy Engineering College for Women (A), Maisammaguda, Medchal, Telangana.

Abstract

Quantum-dot Cellular Automata (QCA) has emerged as a possible alternative to CMOS in recent era of nanotechnology. Some attractive features of QCA include extremely low power consumption and dissipation, high device packing density, high speed (in order of THz). QCA based design of common digital modules have been studied extensively in recent past. Parity generator and parity checker circuits play an important role in error detection and hence act as essential components in communication circuits. However, very few efforts have been made for efficient design of QCA based parity generator and checker circuits so far. Moreover, these existing designs lack practical realizability as they compromise a lot with commonly accepted design metrics such as area, delay, complexity, and cost of fabrication. This paper presents new designs of parity generator and parity checker circuits in QCA which outperform all the existing designs in terms of the above-mentioned metrics. The proposed designs can also be easily extended to handle large number of inputs with a linear increase in area and latency.

Keywords: Quantum-dot Cellular Automata (QCA), CMOS, Parity generator.

1. INTRODUCTION

1.1 Overview

Optimizations in VLSI have been done on three factors: Area, Power and Timing (Speed).Area optimization means reducing the space of logic which occupy on the die. This is done in both front-end and back-end of design. In front-end design, proper description of simplified Boolean expression and removing unused states will lead to minimize the gate/transistor utilization. Partition, Floor planning, Placement, and routing are perform in back-end of the design which is done by CAD tool .The CAD tool have a specific algorithm for each process to produce an area efficient design similar to Power optimization. Power optimization is to reduce the power dissipation of the design which suffers by operating voltage, operating frequency, and switching activity. The first two factors are merely specified in design constraints but switching activity is a parameter which varies dynamically, based on the way which designs the logic and input vectors. Timing optimization refers to meeting the user constraints in efficient manner without any violation otherwise, improving performance of the design.

Quantum-dot cellular automata (QCA) are an attractive emerging technology suitable for the development of ultra dense low-power high-performance digital circuits. Quantum-dot cellular automata (QCA) which employs array of coupled quantum dots to implement Boolean logic function. The advantage of QCA lies in the extremely high packing densities possible due to the small size of the dots, the simplified interconnection, and the extremely low power delay product. A basic QCA cell consists of four quantum dots in a square array coupled by tunnel barriers. Electrons are able to tunnel between the dots, but cannot leave the cell. If two excess electrons are placed in the cell, Coulomb repulsion will force the electrons to dots on opposite corners. There are thus two energetically equivalent ground state polarizations can be labeled logic “0”and “1”.The basic building

blocks of the QCA architecture are AND,OR and NOT. By using the Majority gate we can reduce the amount of delay.i.e by calculating the propagation and generational carries.

1.2 Motivation

Quantum Dot Cellular Automata (sometimes referred to simply as quantum cellular automata, or QCA) are proposed models of quantum computation, which have been devised in analogy to conventional models of cellular automata introduced by von Neumann. Standard solid state QCA cell design considers the distance between quantum dots to be about 20 nm, and a distance between cells of about 60 nm. Just like any CA, Quantum (-dot) Cellular Automata are based on the simple interaction rules between cells placed on a grid. A QCA cell is constructed from four quantum dots arranged in a square pattern. These quantum dots are sites electrons can occupy by tunneling to them.

2. LITERATURE SURVEY

This section starts with the literature review of existing designs, which were focused on the design of efficient XOR structures and its applications. Further, a brief discussion on the overview of recent existing designs follows in this section. In the recent years, many combinational circuits are designed using QCA. The key element in 20 combinational gates is XOR, which has wide applications in arithmetic circuits and digital communications also

Concept regarding deep-sub-micron in CMOS has faced a challenge due to further scaling. The scaling is a limitation of channel length to face these obstacles. It is noticeable that further downscaling become more complex. In this way, nanoelectronics needs to be strong research area to develop best outcomes with nanometric scale (Zhang, Walus, Wang, & Jullien, 2004). The QCA technology focuses on Columbia force based information flow to the interaction of cells to produce outcomes in high-speed computation (Walus, Dysart, Jullien, & Budiman, 2004). In recent years, an exploration regarding research of QCA for the reversible circuit is seen (Misra, Wairya, & Sen, 2017). In the low power era, it is mandatory to have circuits which help of reversible gates (Parhami, 2006). As reversible logic synthesis technology has been progressed in the digital logic circuit scenario, recovered outputs form inputs have been employed to facilitate of no loss of information (Misra, Sen, & Wairya, 2017). They from the new emerging era in which the low power circuit for the reversible structure. Landauer's (1961) concept that, logic computations that are irreversible, also produce heat (KTln2 Joules) for a single bit of information lost, where denoted symbol are standard meaning. Bennett (1973) concept not the loss of information is known as reversible. It normally takes attention of energy dissipated due to loss of bits in logic computations. Reversible circuits can be designed using reversible gates, which have a one-to-one mapping between the inputs and outputs. Because of this mapping, reversible circuits can recover bit loss; however, they cannot detect bit errors in the circuit. QCA-based devices have a high probability of these type of errors owing to undesired manufacturing defects. Parity checking is the most widely used approach in digital systems that compare the input parity with the output parity to detect faults. Hence, this fault detection approach can be accomplished in parity-preserving gates, where the input parity is equal to the output parity (Parhami, 2006). The above scenario motivates us to investigate a new gate structure based on the QCA that is reversible, also realizing the universal logic function with the parity-preserving property. It has been observed that with the new parity-preserving, a reversible gate for circuit design in QCA foregrounds its primitive's results in changes. The relevant aspects of fault coverage and energy dissipation of this gate are discussed. The novelty of the proposed circuits in QCA foreground lies in its ease of computation, and low QCA primitives which show the cost effectiveness. Further simulation results are validated of proposed circuit in QCA. The main advantage of the new parity-

preserving, reversible gate is that, it has effectively implemented 13 standard functions and go through logic circuit implementation.

Currently, the design of logic gates in QCA is a preferred research area for building efficient QCA devices. These logic gates can be categorized into mainly two types: reversible and parity-preserving. Reversible gates have the advantage of low power dissipation; whereas, parity-preserving gates have the advantage of fault-tolerance (Parhami, 2006). Several QCA reversible gates such as the Feynman (Toffoli, 1980), QCA1 (Ma, Huang, Metra, & Lombardi, 2008), RUG (Sen, Saran, Saha, & Sikdar, 2011), TR (Das & De, 2016a), RM (Sen, Dutta, Goswami, & Sikdar, 2014), RQCA (Sen et al., 2014), and a reversible gate (Chabi et al., 2016) are mentioned in the literature. Similarly, several parity-preserving gates such as the CQCA (Thapliyal & Ranganathan, 2009), MX-QCA (Thapliyal, Ranganathan, & Kotiyal, 2013), t-QCA (Sen, Dutta, & Sikdar, 2014), TPC-QCA (Karkaj & Heikalabad, 2016), and the t-Adder (Goswami, Sen, Mukherjee, & Sikdar, 2017) are reported in QCA. Previously, only three logic gates, i.e., the Fredkin (Fredkin & Toffoli, 1982) and the parity-preserving reversible QCA gate (PPRG) (Roohi, Zand, Angizi, & Demara, 2016) and R-CQCA (Misra et al., 2017) had reported both a reversible logic and parity-preserving capability. All these gates are 3 × 3 (three inputs and three outputs) in size, except the Feynman gate, which is 2 × 2 and the R-CQCA, which is 4 × 4. The Fredkin gate is a universal gate; however, it is difficult to construct large circuits with this gate (Sen et al., 2014; Thapliyal et al., 2013). The PPRG gate is capable of self-checking active fault recovery using a cascaded arrangement (Roohi et al., 2016), but the disadvantage is that it requires a large number of QCA cells.

Lots of works are available on synthesizing reversible parity generator, parity checker and D latch. Some literature reviews on these testable circuits are given as Das and De (2016b) proposed a reversible parity generator and parity checker followed by the QCA implementation. A reduced set of QCA primitives was performed on designs. The cell count of 72, area of 0.078 μm2 and six majority gate was reported for reversible odd parity generator. Reversible odd parity checker correctly classified cell count of 130, area of 0.143 μm2 , and nine majority voter gates. The limitation of these designs is the more cell count involved in the QCA architecture. In this work, we compared the primitives results based on QCA paradigm were reported (Das & De, 2016b). Recently, a lot of D latch are implemented in reversible logic based (Hari, Shroff, Mahammad, & Kamakoti, 2006; Misra et al., 2017; Thapliyal et al., 2013). Misra et al. (2017) proposed D latch approach for logic synthesis and R-CQCA and F2G gates be utilized to reduce the garbage output and make representation more compact. Here the quantum cost of D latch is tremendously reduced. The design has quantum schematics and synthesizes output. Since most of the testable reversible circuits utilize gate architecture, this work is valuable in understanding the logic computation behavior of this QCA paradigm.

Initially, an Arithmetic and Logical Function Generator (ALFG) was designed by Teja VC et al., (2008) with sub-modules such as a one-bit Full Adder, 2-input XOR gate, the parity checker, 4:1 multiplexer and 4-bit serial multiplier for implementing 16 different functions [4]. Lakshmi, S. K and Athisha G (2010) implemented the design of various logical structures such as basic logic gates, universal gates, XOR and XNOR using the conventional methodology [5]. One year later, they proposed cell minimization techniques and these were employed to design Half Adder, Full Adder, and serial adder and those structures were analyzed [34]. Shahidinejad A and Selamat A (2012) investigated the practical XOR gate design in QCA, and the first Adder/Subtractor was implemented by using this XOR gate [6].

Jagarlamudi HS et al., (2011) focused on the implementation of combinational (NAND, NOR, XOR, XNOR and Half Adder) and sequential structures (SR latch and Flip-flop and D flip-flop) [51]. Beigh MR et al., (2013) implemented the XOR gate structures in various ways by using the basic gates. They evaluated the performance of XOR gate designs with respect to the cell count, area and latency [7]. Mustafa M and Beigh MR (2013) designed the parity generator and checker using QCA [8]. They utilized the optimized XOR gate in the design of Parity generator and checker. Waje MG and Dakhole PK (2013) focused on developing 4-bit ALU in a simple structure and they implemented the XOR gate structure. Namit Gupta et al., (2013) implemented the digital logic designs using QCA. Goswami M et al., (2014) investigated to implement multiplexer using the XOR, XNOR logic. They also designed the multiplexer based ALU and sequential elements.

Jagarlamudi HS et al., (2011) focused on the implementation of combinational (NAND, NOR, XOR, XNOR and Half Adder) and sequential structures (SR latch and Flip-flop and D flip-flop) [9]. Beigh MR et al., (2013) implemented the XOR gate structures in various ways by using the basic gates. They evaluated the performance of XOR gate designs with respect to the cell count, area and latency [10]. Mustafa M and Beigh MR (2013) designed the parity generator and checker using QCA [11]. They utilized the optimized XOR gate in the design of Parity generator and checker. Waje MG and Dakhole PK (2013) focused on developing 4-bit ALU in a simple structure and they implemented the XOR gate structure [12]. Namit Gupta et al., (2013) implemented the digital logic designs using QCA. Goswami M et al., (2014) investigated to implement multiplexer using the XOR, XNOR logic. They also designed the multiplexer based ALU and sequential elements.

XOR gate function is very advantageous in systems with parity bits for error detection. A parity bit is utilized with the primary objective of detecting the errors through the transmission of binary information. A parity bit is an additional bit incorporated with a binary message to create the total number of 1's in the message together with the parity bit either odd or even. The message, together with the parity bit, is transferred and then verified at the receiver end to detect the errors. An error is noticed if the patterned parity bit does not match with the one transmitted. The circuit used at the transmitter side to produce the parity bit is called a Parity generator. The circuit that verifies the parity at the receiver side is called Parity checker.

Design of a 4-bit ripple adder using a combinational concept from the conventional RCA and the CLA was presented [13]. Efficient full adder design using five-input majority gate and single-bit, multi-bit subtractors using 3-input majority gate. Performance evaluation of efficient XOR structures [14] and implementation of QCA based novel parity generator and checker circuits with minimum complexity and cell count. Implementation of 4-bit arithmetic logic unit and realization of digital designs using QCA. Two novel Full Adders implemented in one layer, and the second one is implemented in three layers using five-input majority gate [15]. To minimize the design complexities, an 8-bit Arithmetic Logic Unit (ALU) was implemented using a novel Bit-slice ALU. The efficient design of Baugh-Wooley multiplier and an overview of existing full adders presented. Defect and temperature effects on complex QCA devices [160] and a new wire-crossing technique on QCA.

Efficient realization of the digital logic circuit using QCA multiplexer [16], adder circuits, new XOR gate and code converters using QCA with the reduced number of wire crossings [58]. Parity Generator and Parity Checker based on QCA [17], a new F-shaped XOR gate and its implementations as novel adder circuits and an n-bit controllable inverter by QCA. A novel design of 8-bit adder/subtractor , a wire-crossing technique based on the difference of cell state in QCA and 3- input XOR logic gate. LFSR design, pseudo-code generator and exclusive-OR and multiplexer circuits based on a nano-electronic compatible designing approach. Efficient parallel binary comparators and

high-speed binary comparator presented. The first step toward cost functions for quantum-dot cellular automata designs discussed. Area-delay efficient binary adders in QCA and simple D flip-flop based sequential logic circuits for QCA implementation were presented in 2014.

A review on Reversible logic gates and it's QCA implementation was presented. To improve the thermal behavior of QCA systems a new sequential signal distribution network demonstrated by the complete design and simulation of a two-bit counter, a three-bit counter, and a pattern detection circuit [18]. Adder design presented using multilayer gate paradigm and shown that the used multilayer a gate does not need to be in a plane, one can shift the gate itself in three-dimensional space, rather than crossing over the wire [19]. Hence, reducing the length of connecting wire, which makes the circuit more compact, reducing the requirement for QCA cells and improving input–output delays. Abedi D et al., presented the design of QCA Coplanar full adders architecture that leads to the reduction of QCA cell count and area consumption without any latency penalty. They used non-adjacent clock zones for the two crossing wires. Ahmad F et al., designed the parity generators and checkers based on XOR/XNOR gates [20]. These circuits present a simple design effective technique using a homogenous layer of cells. Liu W and Swartzlander Jr EE designed a 3-D QCA adders that occupies less area and offers an additional dimension for computation. B. Ramesh and M. Asha Rani designed a low area Binary to BCD code converter based on area-optimized adder. An 8-bit adder/subtractor based on coplanar clock-zone based crossover was designed

3. PROPOSED ARCHITECTURE

3.1 Introduction

Last six decades have seen tremendous growth in CMOS based integrated circuits. However, threatened by many physical constraints, further down-scaling of chip size seems to be reaching its limit. Consequently, the signs of deviation of chip production from the predicted course of Moore's Law have started to show [1]. Hence, the focus is shifting towards new emerging nanotechnologies which can make further downscaling of integrated circuits possible. Quantum-dot cellular automata (QCA) is one of the promising nanotechnologies which has the potential to replace CMOS in upcoming nanotechnology era [2]. One of the most interesting feature of QCA is extremely low power dissipation and consumption. This is achieved by the fact that information flows in QCA devices without any flow of current [2]. Low power consumption and dissipation, high device packing density, high speed (in order of THz) enable realization of more dense circuits with fast switching speed, achieving room temperature operations [3]– [5] using QCA. Design and simulation of common computing modules like adders, multipliers, multiplexers [6]–[8] have been studied enormously. However, lesser effort has been observed in the direction of designing communication circuits. Parity based method is one of the most widely used error detection techniques for the data transmission [9]. In digital systems, binary data being transmitted and processed, may be subjected to noise that may alter data bits from 0s to 1s and vice versa. A parity bit, that indicates whether the number of 1s present in the data word is even or odd, is added to the original data word during transmission from the transmitter. At the receiving end, parity bit of the received word is counted by counting the number of 1s in it and is compared with the transmitted one to detect the presence of an error in the data. A parity generator is a combinational logic circuit that generates the parity bit in the transmitter [9]. On the other hand, a circuit that checks the parity in the receiver is called parity checker [9]. A combined circuit or device consisting of parity generator and parity checker is commonly used in digital systems to detect the single bit errors in the transmitted data word. A parity generator accepts an $(n - 1)$-bit stream data and generates the additional parity bit that is to be transmitted with the bit stream. In even parity bit scheme, the parity bit is 0 (1) if there are even (odd) number of 1s in the

data stream. In odd parity bit scheme, the parity bit is 1 (0) if there are even (odd) number of 1s in the data stream. A parity checker accepts an n-bit stream including (n − 1)-bit data and the parity bit transmitted along with it and generates the parity bit for the data thus received. Parity checker at the receiver can be even or odd depending on the type of parity generator used at the transmitter end. For an even parity checker, an error is indicated by the output 1 (i.e., the number of 1s in its input is found to be odd instead of even). Similarly, for an odd parity checker, an error is indicated by the output 1 (i.e., the number of 1s in its input is found to be even instead of odd). A few designs of parity generator and parity checker circuits in QCA have been presented in the literature [10]– [15]. However, existing designs lack in practical realizability as they compromise a lot with commonly accepted design metrics such as area, delay, complexity, and cost of fabrication. It may be noted that the basic principle involved in the implementation of parity circuits is that sum of odd number of 1s is always 1 and sum of even number of 1s is always zero. Hence, XOR function, that produces 0 (1) output when there are even (odd) number of 1s in the inputs, plays a pivotal role in implementing such circuits. For example, an (n−1)-bit parity generator can be realized by implementing an (n−1)-bit XOR function. Similarly, an n-bit parity checker, for checking the parity thus generated, can be realized by implementing an n-bit XOR function. Accordingly, overall efficiency of such circuits depends a lot on the implementation of XOR functions. A careful scrutiny of the existing designs of parity generator and checker circuits reveal that all these designs use cascaded 2-input XOR gates (without putting much effort in optimizing the individual XOR gates) for implementing the desired XOR function. In this paper, we have used a combination of 2-input and 3-input XOR gates to implement the desired XOR function for realizing parity generator and checker circuits in QCA. We have effectively utilized the fact that implementation of an n-bit XOR function in QCA can be optimized by using a combination of 2-input and 3-input XOR gates using ESOP based transformations [16] rather than using 2-input XOR gates only. It also helps in realizing larger parity generator and checker circuits using the smaller versions in a systematic manner. Simulation experiments performed to compare the proposed designs of QCA parity generator and checker circuits with the existing ones also demonstrate the expected benefit. The proposed ones are found to outperform all the existing designs in terms of commonly accepted design metrics.

3.2 Proposed QCA Parity Generator and Parity Checker Circuits

As mentioned in Section I, XOR function, that produces 0 (1) output when there are even (odd) number of 1s in the inputs, plays a pivotal role in implementing parity generator and checker circuits. An n-input XOR function is usually implemented by combining several 2-input XOR gates. Moreover, use of existing designs of QCA 2-input XOR gates [15], [21], [22], which use a large number of majority gates, lead to significant compromise with the area as well as latency. For instance, implementation of a 4-bit XOR function using three 2-input XOR gates [15] takes at least 9 majority gates. However, we have observed that more efficient implementation of n-input XOR function is possible by using combinations of 2-input and 3-input XOR gates following ESOP based transformation [16]. Accordingly, we have used combination of 2-input and 3-input XOR gates to implement n-bit XOR function instead of relying solely on 2-input XOR gates which was the case in existing implementations. We have also used majority logic reduction [23]–[25] to further optimize the designs of individual XOR gates (both 2-input and 3-input). The logical expression (AB+AB), representing 2- input XOR function can be re-written equivalently as M[M(A, B, 1), M(A, B, 0), 0] using majority logic reduction, where M(X, Y, Z) represents a 3-input majority gate [6] with inputs X, Y, and Z. The above Boolean expression can be implemented using three 3-input majority gates and one inverter as shown in Fig. 5(a). Similarly, the logical expression ($A\bar{B}\bar{C}$ +$A\bar{B}\bar{C}$ +$A\bar{B}\bar{C}$ +ABC), representing a 3-input XOR function can be re-written as M[M(A, $\bar{B}$, $\bar{C}$), C,M(A, B,

C¯)] using majority logic reduction, where M(X, Y, Z) represents a 3-input majority gate with inputs X, Y, and Z. Fig. 1 (b) shows the schematic diagram of the gate level implementation of the above expression. The logical expression for the output of 3-bit even parity generator is A ⊕ B ⊕ C and hence, it can be implemented simply by using the 3-input XOR gate of Fig. 5(b).

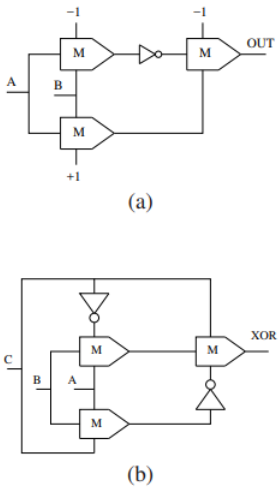
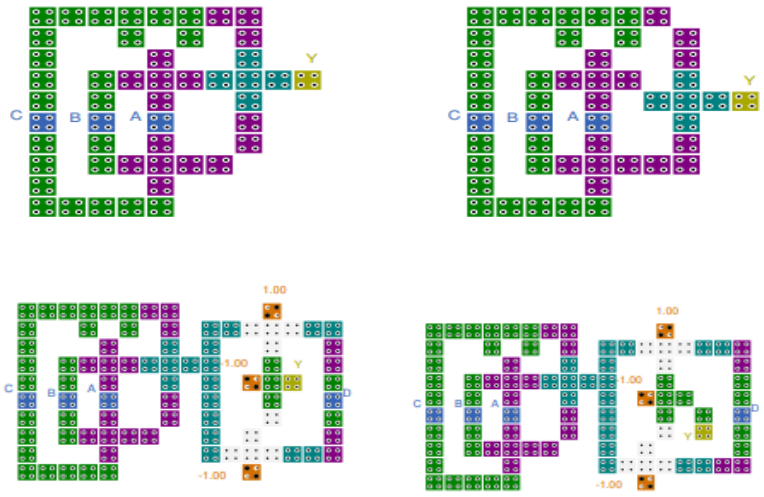


Fig. 1: Gate level implementation of (a) 2-input XOR and (b) 3-input XOR.

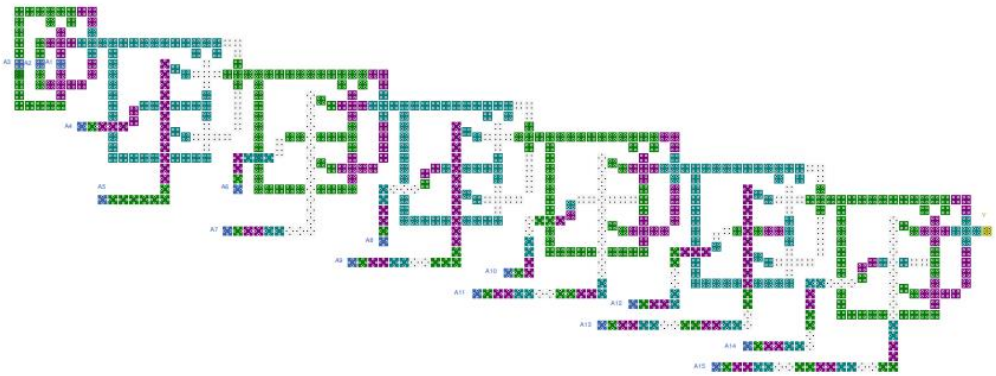
The logical expression for the output of 3-bit odd parity generator (which is A ⊕ B C) can be re-written as (A¯B¯C¯ + ABC ¯ + ABC¯ + ABC¯) i.e., M[M(A, B, C), C,M¯ (A, ¯ B,C ¯)] using majority logic reduction. The above expression indicates that 3-bit odd parity generator can also be implemented by using three majority gates. Fig. 6(a) and Fig. 6(b) show the layouts of the proposed 3-bit even and odd parity generators, respectively. As apparent from the figures, both the designs consist of 49 QCA cells without any crossover and incur 3 clock zones (0.75 clock cycle) latency. Assuming QCA cell size of 18nm × 18nm with a gap of 2nm between two consecutive cells, each of the layouts consumes an area of 0.04μm2. The logical expression for the output of 4-bit even parity checker (which is A ⊕ B ⊕ C ⊕ D, where 'D' represents the transmitted parity bit) is same as that of a 4-bit XOR gate and hence, it can be implemented by combining a 3-input XOR gate and a 2-input XOR gate. Similarly, the logical expression for the output of 4-bit odd parity checker is A B C D which can also be realized using a 3-input XOR gate and a 2-input XOR gate.

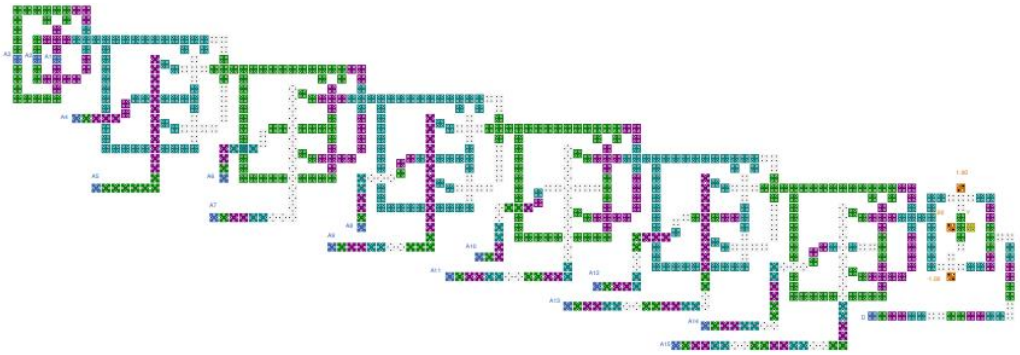


As apparent from the figures, the proposed designs consist of 84 QCA cells and 88 cells, respectively. However, both of them consume same area (0.08μm2) assuming QCA cell size of 18nm×18nm with a gap of 2nm between two consecutive cells. Moreover, both the designs incur latency of 5 clock zones (1.25 clock cycles) and have no crossover. In order to verify the functional behavior of the proposed parity generator and checker circuits, we carried out simulations using the bistable simulation engine of QCADesigner [26] (version 2.0.3) with the following parameters: (i) QCA cell dimension: 18nm×18nm with a gap of 2nm between two consecutive cells (ii) Radius of effect: 65nm, (iii) Relative permittivity: 12.9, (iv) Convergence tolerance: 0.001000. It may be noted that the bistable simulation engine of QCADesigner uses intercellular Hartree approximation (ICHA) assuming a simple two-state system to represent each QCA cell. A little compromise in accuracy as compared to full-basis computation is often compensated by the significantly better scalability [27]. Simulation results are found to show significantly strong polarization (more than 0.954) at the output of all the circuits. It may also be noted that the proposed designs can be systematically extended to handle any number of inputs (n). For example, In order to estimate the growth of various design metrics as a function of the number of inputs (n), we have computed the general expressions for area, latency, and number of crossovers.

Comparative Study

In order to evaluate the effectiveness of the proposed designs of parity generators and checkers, we have compared each of them with existing ones in terms of commonly accepted design metrics such as area, latency, complexity, and the type and number of crossovers used. Note that the complexity of a QCA circuit can be expressed as M+I+C [28], where M, I, and C refer to the number of majority gates, the number of inverters and the cost of crossovers used in the circuit, respectively. Table II and Table III show the summary of the comparative study made on 3-input parity generators and 4-input parity checkers, respectively. It is apparent from the tables that the proposed designs outperform all the existing designs in terms of all the design metrics. As suggested by Liu et al. [28], instead of considering the individual metrics for comparison, cost functions combining multiple metrics may be more effective. For the sake of completeness, we have also included a case of comparison based on a cost function (Cost = (M2 +I +C2)×T) specifically designed for QCA circuits [28]. Figs. 12-13 illustrate the comparison for 3-bit parity generators and 4-bit parity checkers, respectively. The proposed designs are found to be superior with respect to this cost function too.





4. SIMULATION RESULTS

4.1 Waveforms

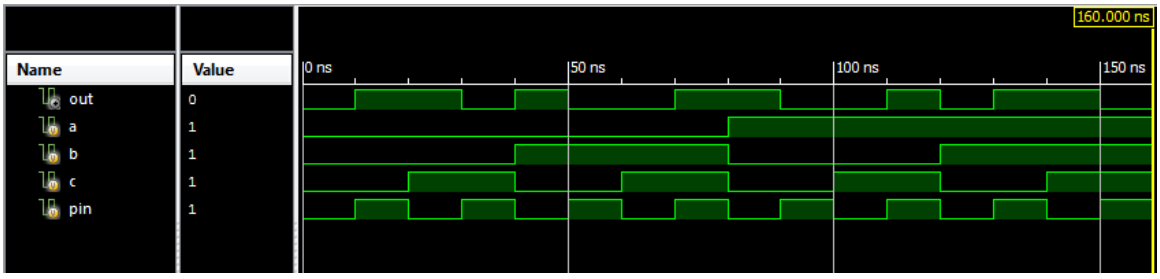


Fig. 2: Even parity checker.

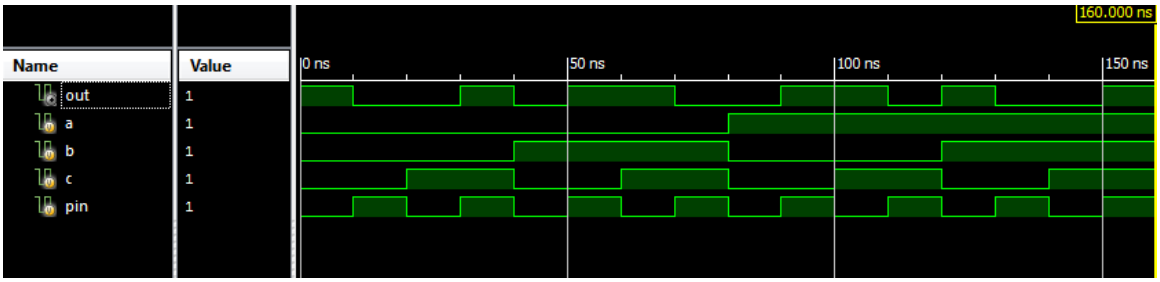


Fig. 3: Odd parity checker.

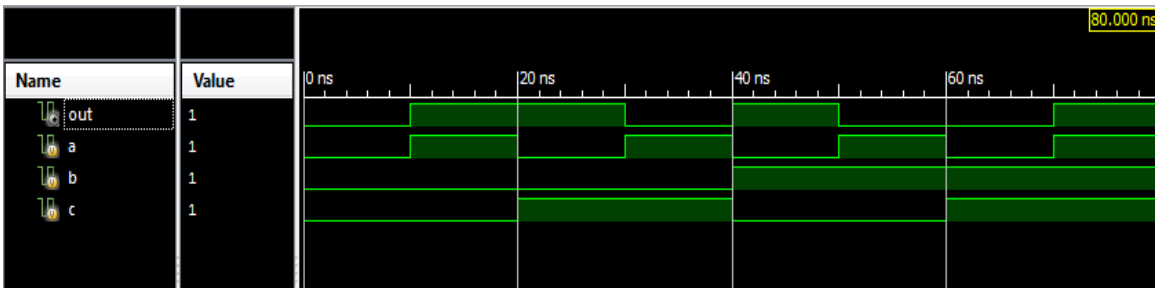


Fig. 4: Parity generator even.

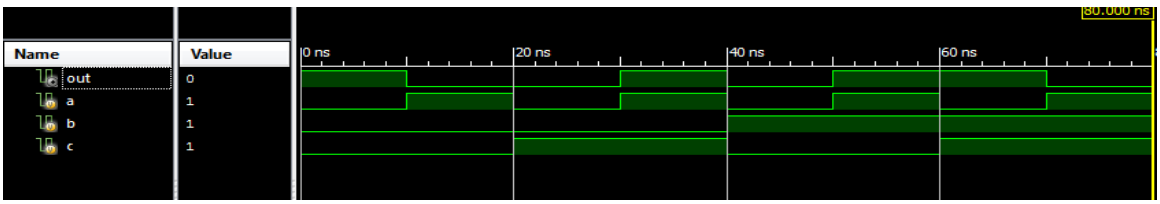


Fig. 5: Parity generator odd.

5. CONCLUSION

Efficient designs of 3-bit parity generator and 4-bit parity checker circuits in QCA have been presented. Both the designs are found to outperform all the existing designs in terms of common design metrics such as area, latency, and more importantly in-terms of cost function specially designed for QCA circuits. Moreover, both the designs can be extended easily for large number of inputs with linear increase in area and latency, thereby, making them suitable for practical realization.

REFERENCES

[1] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernestein, "Quantum cellular automata," *Nanotechnology*, vol. 4, no. 1, pp. 49–57, 1993.

[2] M. T. Niemer and P. M. Kogge, "Problems in designing with QCAs Layout = Timing," *Int. J. Circuit Theory Appl.*, vol. 29, no. 1,pp. 49–62, 2001.

[3] J. Huang and F. Lombardi, *Design and Test of Digital Circuits by Quantum-Dot Cellular Automata*. Norwood, MA, USA: Artech House,2007.

[4] W. Liu, L. Lu, M. O'Neill, and E. E. Swartzlander, Jr., "Design rules for quantum-dot cellular automata," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2011, pp. 2361–2364.

[5] K. Kim, K. Wu, and R. Karri, "Toward designing robust QCA architectures in the presence of sneak noise paths," in *Proc. IEEE Design, Autom. Test Eur. Conf. Exhibit.*, Mar. 2005, pp. 1214–1219.

[6] K. Kong, Y. Shang, and R. Lu, "An optimized majority logic synthesis methology for quantum-dot cellular automata," *IEEE Trans. Nanotechnology*.vol. 9, no. 2, pp. 170–183, Mar. 2010.

[7] K. Walus, G. A. Jullien, and V. S. Dimitrov, "Computer arithmetic structures for quantum cellular automata," in *Proc. Asilomar Conf.Sygnals, Syst. Comput.*, Nov. 2003, pp. 1435–1439.

[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).