Design of Practical Parity Generator and Parity Checker Circuits in QCA

***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 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 in 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 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; Parity generator; Parity checker; Exclusive-OR (XOR) gate.**

I. 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 down-scaling 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.

The rest of the paper is organized as follows: Section II introduces the fundamentals of QCA technology. Section III reviews the related prior work. The proposed designs are presented in Section IV. Summary of comparative study between the proposed design with the existing ones is illustrated in Section V. Finally, Section VI draws the conclusion of this work.

II. Basics of QCA

The concept of QCA was first demonstrated by Metal-Island implementation [17]. Other possible implementation mechanism include semiconductor, molecular and magnetic [17]. In this work, we have considered semiconductor implementation of QCA. Basic operation of such QCA devices is based on the quantum mechanical effects and quantization of Coulombic charge [2]. The fundamental element, often referred to as QCA cell, is a square-shaped container-like structure to hold the charge. Each QCA cell has four potential wells (dots), one at each corner of the cell and two free electrons which are capable of tunnelling quantum mechanically, from one quantum dot to another. At equilibrium, the two electrons inside a cell always occupy the antipodal sites due to Coulombic repulsion. This gives way to two energetically equivalent arrangements. These two arrangements, as shown in the Fig. 1, are denoted as two different polarizations p = +1 and p = -1 which represent logic 1 and logic 0, respectively. Information flow in QCA is

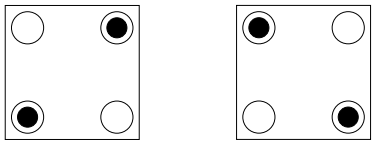
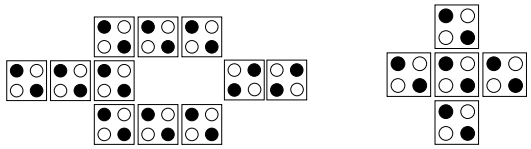


Fig. 1: Binary representation of QCA cells

achieved by Coulombic interactions between electrons present in neighboring cells without flow of electrons which leads to extremely low power dissipation. In a series of QCA cell, every cell just rearranges their polarized state according to the adjacent cell to make the flow of information possible.

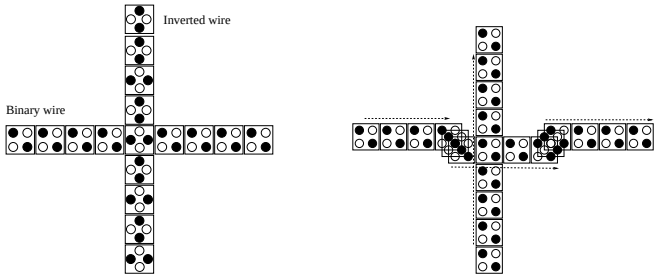
Majority gate or majority voter (M) and inverter gate (I) [6] are the two basic building blocks of any QCA circuits. Fig. 2 shows their design layout in QCA. Universal nature of the combination of these two gates facilitates implementation of any logic circuit using them.



(a) (b)

Fig. 2: Fundamental building blocks of QCA design layout (a) Inverter (b) Majority voter

QCA allows two wires to cross each other in the same layer without interfering each other. As shown in Fig. 3(a), such coplanar crossover [6] is realized using two different types of wires: a binary wire (consisting of a series of normal QCA cells) and an inverted chain (consisting of cells rotated 45° from their normal orientation). Wire crossings implemented using multiple layers (similar to metal wire crossovers in CMOS) are also possible in QCA (Fig. 3(b)). A quasi-adiabatic



(a) (b)

Fig. 3: (a) Coplanar and (b) Multilayer crossover in QCA

clocking mechanism is also used in QCA for synchronization of information and to meet the power requirement of the QCA device. A four-phase clock zone system, introduced by Lent et al. [2], with a 90° phase shift from one clock zone to the next is commonly used (Fig. 4). The four phases (zones) are named as switch, hold, release and relax, respectively. The orientations or state of electrons in a QCA cell is changed during switch or release phase only.

III. Related prior work

A significant part of the research on QCA so far has focused on the design and simulation of basic logic gates [6] and various digital modules including adders [18], [19], multipliers [7], multiplexers [8]. However, as mentioned in Section I, very few such efforts can be found in the literature in designing communication circuits and their components such as parity generators and checkers. A 4-bit even parity checker consisting of three 2-input XOR gates with both coplanar and multilayer crossovers was first proposed by Teja *et al.* [13]. Overall, the design consumes a large number of QCA cells (299 QCA cells) and incurs high latency (8 clock zones). With an intention of improving the design in terms of area, Mustafa and Beigh [15] proposed a 4-bit odd parity checker

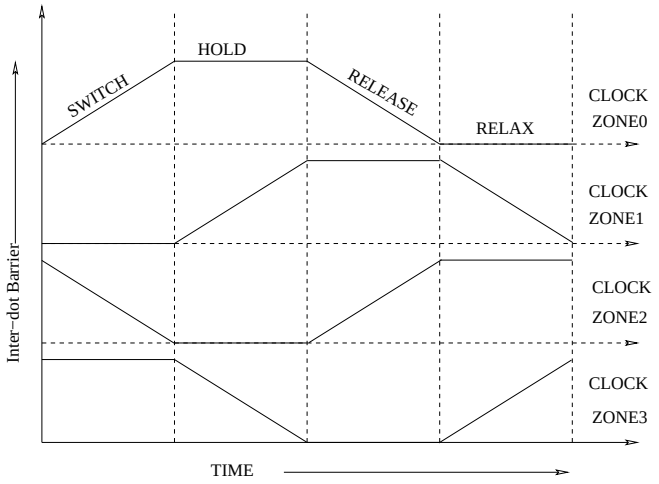


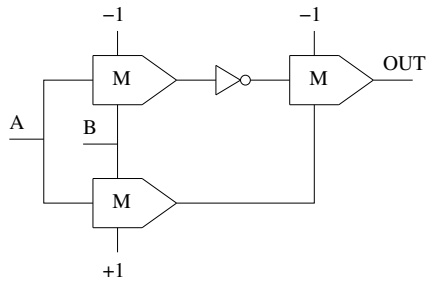
Fig. 4: QCA clocking

that consumes 145 QCA cells. However, this design is found to incur prohibitively large latency (12 clock zones). In [15], Mustafa and Beigh also presented a 3-bit even parity generator that consumes 99 QCA cells and incurs a latency of 8 clock zones. Later, Ahmad *et al.* [11] proposed designs of 4-input even and odd parity checkers both of which consume 94 QCA cells and incur a latency of 7 clock zones. Ahmad *et al.* [11] also proposed a 3-bit even parity generator and a 3-bit odd parity generator which consume 64 and 66 QCA cells, respectively, and incur a latency of 11 and 7 clock zones, respectively. An area efficient even parity generator and checker circuit was later proposed by Santra [14] where the 3-bit even parity generator consumes only 60 QCA cells and 4-bit even parity checker consumes 117 QCA cells. However, the latency incurred by them is 8 clock zones and 9 clock zones, respectively. A 3-bit odd parity generator and a 4-bit odd parity checker were proposed by Das and De in [10], using reversible logic for nano-communication. The design of parity generator consumes 72 QCA cells and incurs a latency of 7 clock zones whereas the design of the parity checker (that uses a 2×2 Feynman gate structure [20]) consumes 126 QCA cells and latency of 8 clock zones. It is apparent that all of these existing designs compromise a lot with one or more commonly accepted design metrics such as area, delay, complexity, and cost of fabrication. Moreover, most of these designs present smaller sized parity generators and checkers (3-bit/4-bit) and hardly provides any clue so that they can be extended to realize such circuits of bigger size (15-bit/16-bit or even bigger). The above mentioned drawbacks act as significant barrier against the practical realizability of these designs. Accordingly, efficient and practically realizable designs of parity generators and parity checkers have become very much essential.

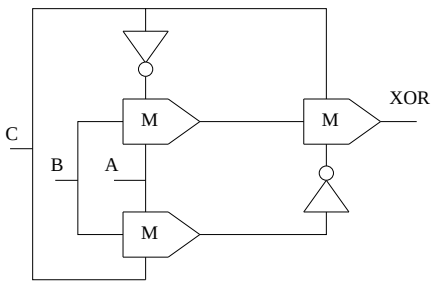
IV. Proposed QCA Parity Generator and Parity Checker Circuits

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 ($A\overline{B}$+$\overline{A}B$), representing 2-input XOR function can be re-written equivalently as $M[M(A, B, 1), \overline{M(A, B, 0)}, 0]$ 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 ($\bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$), representing a 3-input XOR function can be re-written as $M[M(\bar{A}, \bar{B}, \bar{C}), C, M(A, B, \bar{C})]$ using majority logic reduction, where $M(X, Y, Z)$ represents a 3-input majority gate with inputs X, Y, and Z. Fig. 5(b) shows the schematic diagram of the gate level implementation of the above expression. The logical expression for the



(a)



(b)

Fig. 5: Gate level implementation of (a) 2-input XOR and (b) 3-input XOR

output of 3-bit even parity generator is $A \oplus B \oplus C$ and hence, it can be implemented simply by using the 3-input XOR gate of Fig. 5(b). The logical expression for the

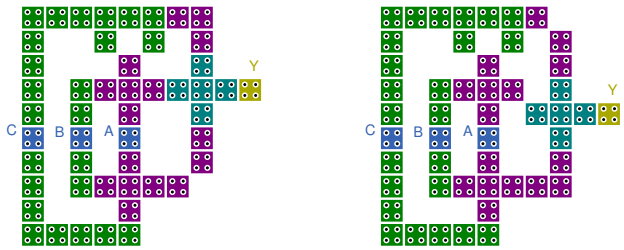


Fig. 6: Layout of the proposed 3-bit (a) even (b) odd parity generator

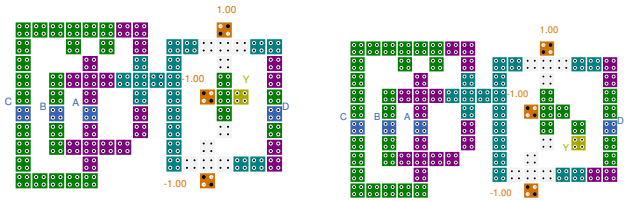


Fig. 7: Layout of the proposed 4-bit (a) even (b) odd parity checker

output of 3-bit odd parity generator (which is $A \oplus B \ominus C$) can be re-written as $(\bar{A}B\bar{C} + \bar{A}BC + A\bar{B}C + AB\bar{C})$ i.e., $M[M(A,B,C),\bar{C},M(\bar{A},\bar{B},C)]$ 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 $18nm \times 18nm$ with a gap of $2nm$ between two consecutive cells, each of the layouts consumes an area of $0.04\mu m^2$.

The logical expression for the output of 4-bit even parity checker (which is $A \oplus B \oplus C \oplus D$, 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 $A \ominus B \ominus C \ominus D$ which can also be realized using a 3-input XOR gate and a 2-input XOR gate. Fig. 7(a) and Fig. 7(b) show the layouts of the proposed 4-bit even and odd parity checkers, respectively. As apparent from the figures, the proposed designs consist of 84 QCA cells and 88 cells, respectively. However, both of them consume same area ($0.08\mu m^2$) assuming QCA cell size of $18nm \times 18nm$ 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: $18nm \times 18nm$ 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, Fig. 8 and Fig. 9 show the layouts of a 15-bit parity generator and a 16-bit parity checker circuit, respectively. 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. Table I shows the expressions for n-bit parity generator and checker. Figs. 10-11 may be referred for the graphical representations of the growth in area and latency, respectively. It is apparent that both the parameters grow somewhat linearly with the increase in the value of n.

TABLE I: Generalized expressions for various design metrics of proposed n-bit parity generator and parity checker

Circuits	Generalized expressions for		
	Area (μm^2)	Latency (clock cycles)	No. of Crossover
Parity Generator	$0.02(7n-11)\times$ $0.02(1.5n+11.5)$; $n \geq 5$	$n/4$	1.5(n-3)
Parity Checker	$0.02(7n-9)\times$ $0.02(1.5n+10)$; $n \geq 5$	$(n+1)/4$	1.5(n-4)

V. 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 ($Cost = (M^2 + I + C^2) \times T$) 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.

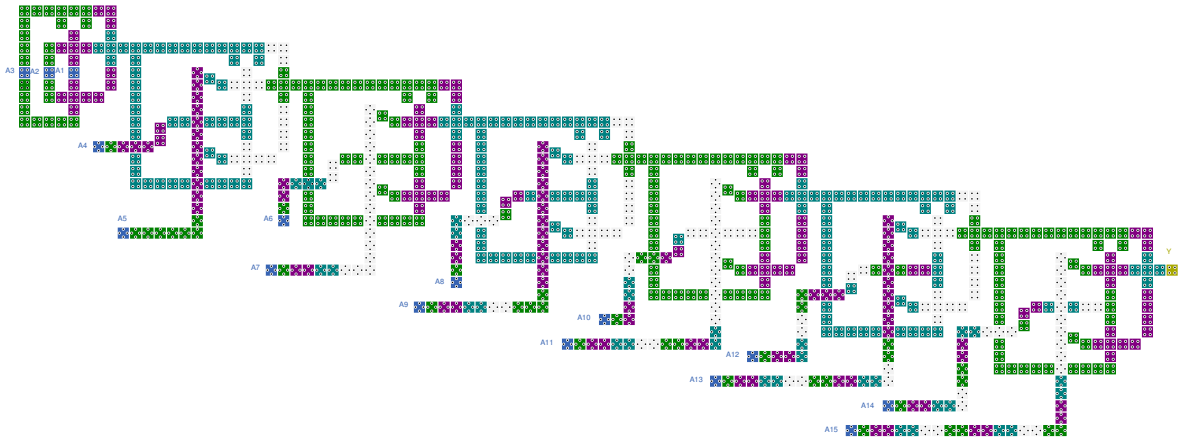


Fig. 8: Layout of the proposed 15-bit even parity generator

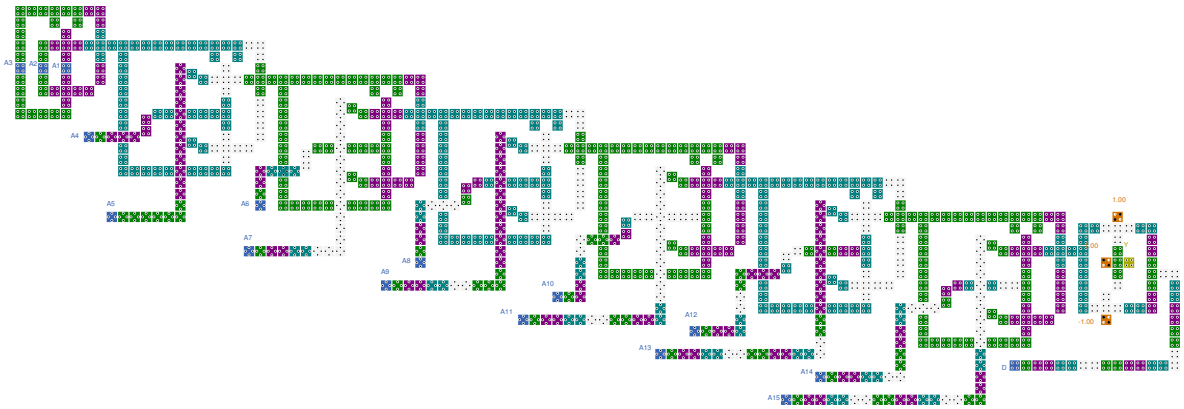


Fig. 9: Layout of the proposed 16-bit even parity checker

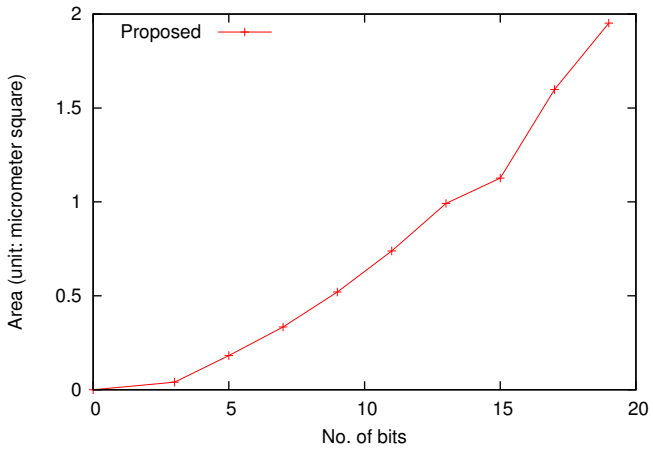


Fig. 10: Growth in area of the proposed n-bit parity generator with respect to increasing value of n

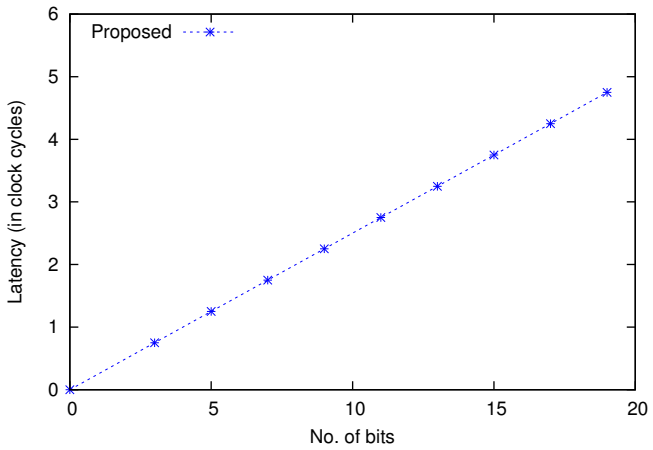


Fig. 11: Growth in latency of the proposed n-bit parity generator with respect to increasing value of n

VI. 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.

TABLE II: Comparisons of various 3-bit parity generators in terms of common design metrics

Parity Generator	Area (μm^2)	Latency (clock zones)	Type of crossover	Number of crossover	Complexity
[14], Even	0.06	8	None	–	8
[15], Even	0.17	8	None	–	8
[11], Even	0.09	11	Multilayer	2	14
[12], Even	0.09	12	Multilayer	2	14
Proposed (Even)	**0.04**	**3**	**None**	**–**	**5**
[10], Odd	0.08	7	None	–	11
[11], Odd	0.09	7	Multilayer	2	15
Proposed (Odd)	**0.04**	**3**	**None**	**–**	**6**

TABLE III: Comparisons of various 4-bit parity checkers in terms of common design metrics

Parity Checker	Area (μm^2)	Latency (clock zones)	Type of crossover	Number of crossover	Complexity
[14], Even	0.13	9	Coplanar	1	15
[15], Even	0.28	12	None	–	15
[11], Even	0.11	7	Multilayer	3	21
[12],Even	0.12	7	Multilayer	3	21
[13], Even	0.53	8	Coplanar & Multilayer	4 1	30
Proposed (Even)	**0.08**	**5**	**None**	**–**	**9**
[10], Odd	0.15	8	None	–	18
[11], Odd	0.13	7	Multilayer	3	22
Proposed (Odd)	**0.08**	**5**	**None**	**–**	**10**

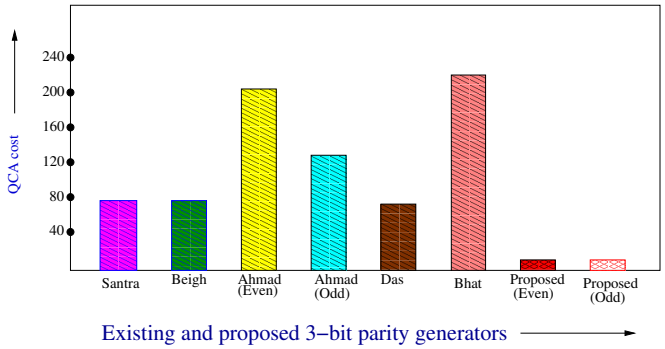


Fig. 12: Comparison of proposed 3-bit parity generator with the existing ones in terms of $Cost = (M^2 + I + C^2) \times T$

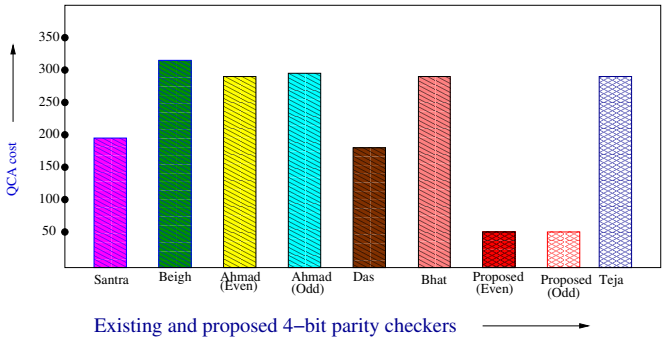


Fig. 13: Comparison of proposed 4-bit parity checker with the existing ones in terms of $Cost = (M^2 + I + C^2) \times T$

REFERENCES

[1] C. C. Mann. The end of Moore's law? MIT Technology Review, May 2000. http://www.technologyreview.com/featuredstory/400710/the-end-of-moores-law/.

[2] C. S. Lent et al. Quantum-dot cellular automata. *Nanotechnology*, 4:49–57, 1993.

[3] C. S. Lent and B.Isaksen. Clocked molecular quantum-dot cellular automata. *IEEE Trans. Electron Devices*, 50:1890–1896, 2003.

[4] R. P. Cowburn and M. E. Welland. Room temperature magnetic quantum cellular automata. *Science*, 287(5457):1466–1468, 2000.

[5] Y. Wang and M. Lieberman. Thermodynamic behavior of molecular-scale quantum-dot cellular automata QCA wires and logic devices. *IEEE Trans. on Nano*, 3(3):368–376, 2004.

[6] P. D. Tougaw and C. S. Lent. Logical devices implemented using quantum cellular automata. *Journal of Applied Physics*, 75(3):1818–1825, 1994.

[7] H. Cho and E. E. Swartzlander. Adder and multiplier design in quantum-dot cellular automata. *IEEE Trans. on Computers*, 58(6):721–727, 2009.

[8] V. A. Mardiris and I. G. Karafyllidis. Design and simulation of modular 2^n to 1 quantum-dot cellular automata (QCA) multiplexers. *International Journal of Circuit theory and Applications*, 38:771–785, 2010.

[9] M. M. Mano, editor. *Digital Logic and Computer Design*. Prentice Hall of India Pvt. Ltd., 2004.

[10] J. C. Das and D. De. Quantum-dot cellular automata based reversible low power parity generator and parity checker design for nanocommunication. *Frontiers of Information Technology and Electronic Engineering*, 17:224–236, 2016.

[11] F. Ahmad et al. Design and analysis of odd and even parity generators and checkers using QCA. In *INDIACom*, pages 187–193, 2015.

[12] F. Ahmad and G. M. Bhat. Novel code converts based on QCA. *Intl. journal of science and research*, 3(5), 2014.

[13] V. C. Teja et al. QCA based multiplexing of 16 arithmetic & logical subsystems-a paradigm for nano computing. In *Intl. Conf. on Nano/Micro Engineered and Molecular Systems*, pages 758–763, 2008.

[14] S. Santra and U. Roy. Design and optimization of parity generator and parity checker based on quantum-dot cellular automata. *IJCEACIE*, 8(3), 2014.

[15] M. Mustafa and M. R. Beigh. Design and implementation of QCA based novel parity generator and checker circuit with minimum complexity and cell count. *Indian Journal of Pure and Applied Physics*, 51:60–66, 2013.

[16] D. Y. Feinstein and M. A. Thornton. ESOP transformation to majority gates for quantum-dot cellular automata logic synthesis. In *Reed-Muller Workshop*, pages 43–50, 2007.

[17] M. O'Neill, A. Lau, and E. E. Swartzlander. *Design of Semiconductor QCA Systems*. Artech House Publishers, 2014.

[18] R. Zhang et al. Performance comparison of quantum-dot cellular automata adders. In *ISCAS*, pages 2522–2526, 2005.

[19] D. Kumar and D. Mitra. Design of a practical fault-tolerant adder in QCA. *Microelectronics*, 53:90–104, 2016.

[20] R. P. Feynman. Quantum mechanical computer. *Optics News*, 11:11–20, 1985.

[21] M. R. Beigh et al. Performance evaluation of efficient XOR structures in quantum-dot cellular automata (QCA). *Circuits and Systems*, 4:147–156, 2013.

[22] W. S. Jahan et al. Circuit nanotechnology: QCA adder gate layout designs. *IOSR Journal of Computer Engineering*, 16:70–78, 2014.

[23] R. Zhang et al. A method of majority logic reduction for quantum cellular automata. *IEEE Trans. Nanotechnol*, 3(4):443–450, 2005.

[24] H. Mahmoud. Systematic minimization technique for majority-majority digital combinational circuits. *Science alert*, 11(5):832–839, 2011.

[25] K. Kong et al. An optimized majority logic synthesis methodology for quantum-dot cellular automata. *IEEE Trans. Nanotechnol*, 9(2):170–183, 2010.

[26] K. Walus et al. QCADesigner: A rapid design and simulation tool for quantum-dot cellular automata. *IEEE Trans. Nanotechnol*, 3(1):26–31, 2004.

[27] M. LaRue et al. Stray charge in quantum-dot cellular automata: A validation of the intercellular hartree approximation. *IEEE Transactions on Nanotechnology*, 12(2):225–233, 2013.

[28] W. Liu et al. A first step toward cost functions for quantum-dot cellular automata designs. *IEEE Trans. Nanotechnol*, 13:476–487, 2014.