Design of Fault-Tolerant XOR/XNOR Gate Using Fault-Tolerant NNI Gate

Fatemeh Kiayi [1], Behnaz Gharekhanlou [2]*, Alireza Kashani Nia [3]

[1]Department of Electrical Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran, fatemehkiayi@gmail.com
[2]Department of Electrical Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran, beh.gharekhanlou@iauctb.ac.ir
[3]Department of Electrical Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran, ali.kashaniniya@iauctb.ac.ir

Abstract

Today more than ever, we need high-speed circuits with low-occupancy and low-power as an alternative to CMOS circuits. Therefore, we proposed a new path to build nanoscale circuits such as Quantum-dot Cellular Automata (QCA). This technology is always prone to failure due to its very small size. Therefore, designers always try to design fault-tolerant gates and provide methods to increase the reliability of QCA. By adding redundant cells, the possibility of some defects such as cell omission and cell addition is somewhat reduced. However, in the face of defects such as stuck-at 0/1 faults, Clock fault and bridging fault. We can greatly increase the fault tolerance by appropriate placement and using fault-tolerant gates with a suitable structure. In this paper, we design the XOR/XNOR gate with the approach of preventing stuck-at 0/1 fault, clock fault, and bridging fault using the first NNI gate tolerating cell addition fault.

Keywords: Quantum-dot Cellular Automata, fault-tolerant XOR gate, fault-tolerant XNOR gate, fault-tolerant NNI, clock defect, bridging defect, stuck at 0/1 defect
Article history: Received 2021-07-20; Revised 2021-08-09; Accepted 2021-08-18.

1. Introduction

Today, information transmission is done using conventional switches of CMOS technology [1], [2]. According to Moore's Law, that the number of transistors doubles every 18 months [3], [4], this technology led to the introduction of a new solution to create high-speed, high-density, low-power circuits by reducing the size of transistors to nanoscale. In 1993, Lent first introduced the QCA to the world [5], a technology that is very new and practical for designing nanoscale circuits. In this technology, binary information is stored in a cell based on an electric charge and transmitted from one cell to another through the Coulomb force between cells. As in CMOS, for the construction of complex circuits, the use of basic gates as modules is common in the implementation of QCA circuits; As a result, the design of basic gates with high reliability improves the tolerance. This led us to design the XOR/XNOR gate which is more resistant to some faults.

In this study, we will first refer to the basic concepts, then review the concept of reliability and types of faults. In the following, we will discuss the concept of reliability in QCA circuits and finally we will design the XOR/XNOR gate using the fault-tolerant NNI with the approach of preventing errors such as stuck-at 0/1 faults, clock fault, and bridging fault and at the end we will compare the proposed gate with its counterparts.

2. The Basic Concepts

In its simplest form, the QCA cell is a square cell with four cavities (dots) and two electrons. The polarity of the cell is determined by the arrangement of electrons in the cell. Since electrons have similar charges, the repulsive force between them causes the cell to reach its most stable state when the electrons are at the farthest distance from each other. Therefore, in general, a cell has pole -1 and pole 1. There are two models to display a cell: standard and 45º [6]. Figure 1 shows these two cell models with their polarities. The contact angle of the water droplet (a) $0° < \theta < 90°$ (b) $90° < \theta < 150°$.

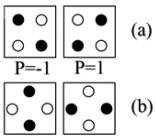

Fig. 1. A QCA cell (a) Position of electrons in a 90º cell (b) Position of electrons in a 45º cell

A) Wire

According to the definition of cell-to-cell response [5], a cell can be affected by the polarization of its neighboring cells. The interaction between the cells causes the information to be transferred from one cell to the next. It means that one can arrange cells into a row to create a wire [7]. In general, there are two types of wiring. In the first method, the standard cells are used to create the wire, and the input information itself is transferred to the output. In the second method, 45º rotated cells are used to create the wire, and one can transfer the input or its inverted value to the output by using an even or odd number of cells [7]. Also, in this technology, there is no limit to the number of fan-out wires and the cells are easily stacked together (Figure 2, a and b).out any text that may try to fill in next to the graphic.

B) Basic Logical Element

We can implement all QCA circuits using majority and inverter gates [6], [8]. The other two gates, AOI [9], [10] and NNI [11], [12] also play important roles in designing QCA circuits. Majority gate: One of the most widely used QCA gates is the majority gate, which has odd number of inputs, one output, and one or more decision-making cells that transfer a stronger polarization to the output. This gate is used to build AND OR gates [5]. (Figure 2, c). Inverter: Another popular gate in QCA circuits are inverters that transmit the input signal to the output in reverse, thus they have one input and one output [5]. (Figure 2, d).

AOI gate: Although it was very easy to implement AND and OR through the majority gate, the biggest problem with this gate was that it did not have an internal inverter. Since the inversion operation is very expensive in this technology, the seven-cell AOI gate with five input cells, one decision-making cell, and one output cell was designed and implemented. As its name implies, in addition to AND and OR operations, this gate is also an inverter [9], [10]. (Figure 2, e).

NNI gate: In [11], [12], the NNI gate is universally introduced as a suitable solution to reduce the occupied area and increase the strength in the QCA circuits. This gate can provide significant services as a majority gate, an inverter gate, and NAND and NOR gate in the implementation of all QCA circuits. (Figure 2, f)

C) Clock

There is no current flow in the QCA, so we use the clock to control the flow path from the input cell to the output cell. The clock controls the flow by increasing or decreasing the potential barrier between the dots in a cell. When the potential barrier is low, the cells are in a state without special polarization, and when the potential barrier is at its highest value, the cell's state doesn't change [13]. The clock signal for QCA circuits typically consists of four phases: switch, hold, release, and relax [14], to indicate that the change in cell state is not abrupt but adiabatic [2]. There are two proposed clocking design for QCA circuits: Landauer [1] and Bennett [1]. As shown in Figure 3, c, the QCA Designer has four Landauer clocks and each clock has a 90-degree phase difference with the next clock [14].

3. Reliability in QCA Circuits

The existence of defects and their impact on systems and their results are undeniable; QCA circuits have a high potential for defects due to their small scale. So far, many studies have been conducted on the types of defects that may occur for QCA circuits [10], [16], [17]. Figure 5 indicates the types of defects and also the solutions to tolerate some of these defects [18], [19], [20]. One of these solutions is to design basic fault-tolerant gates [21], [22], [23]. Before designing fault-tolerant gates, it is necessary to identify the critical area or cell in each gate [24].

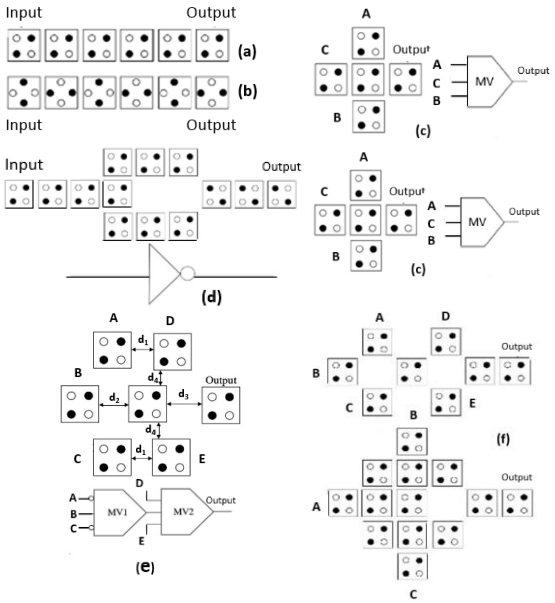

Fig. 2. Basic gates in QCA technology

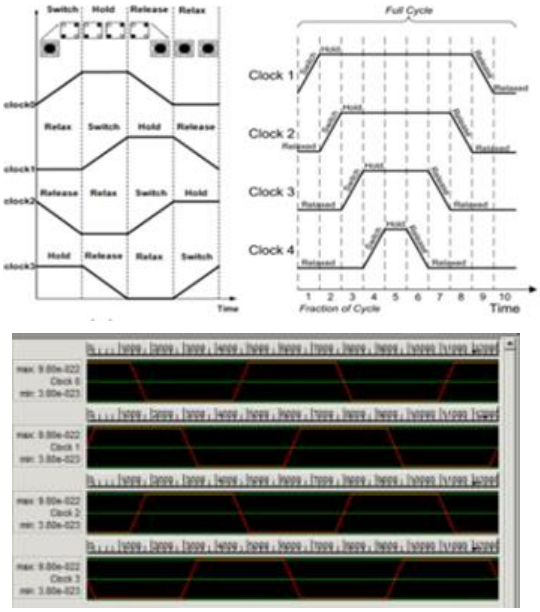

Fig. 3. QCA Zone clocking (a) Landauer clocking waveform [1] (b) Bennett clocking waveform [1] (c) clocking wave form in QCADesigner

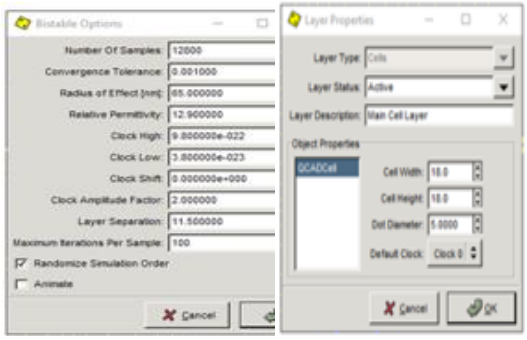

Fig. 4. Default parameters in QCA Designer

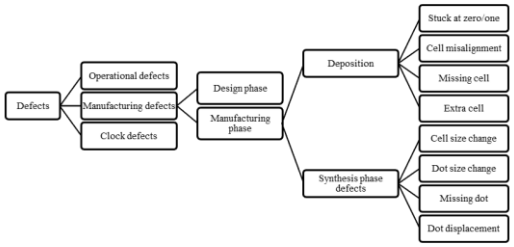

Fig. 5. Categorizing different defect types in QCA circuits

Fault injection method in QCA Designer was used in order to simulate and evaluate the tolerance of QCA circuits against cell omission, cell addition, clock, stuck at 0/1 and bridging defects in a gate or a part of QCA circuit which will be studied in the following.

A) Missing Cell Defect

If for some reason a cell was missing, the neighbouring cells cannot have their ideal performance, and the lack of effect on these cells, which is due to the missing cell, is not negligible. If the normal distance of the cells is relatively high, the circuit cannot work properly with a missing cell [22]. (Figure 6)

B) Extra Cell Defect

This defect occurs when an extra cell is placed next to the original cells of the gate on the bed by mistake [25].(Figure 6).

C) Clock Defect

Clock phase defect can be considered as a result of phase mismatch in the clock in QCA circuits, which will cause unwanted delay or inversion at the output [26]. (Figure 6)

D) Stuck at 0/1 Defect

Occurrence of this defect means that QCA cells are fixed at 1 or -1 polarization. In other words, electrons are not able to perform tunnelling operation properly within QCA cells [18]. (Figure 6)

E) Bridging Defect

If the distance between two QCA wires is equal to the width of a QCA cell, then bridging fault will occur in the event of the occurrence of cell addition fault [19]. Fig. 9 shows a schematic diagram of the missing cell, extra cell defects in a majority gate and stuck at 0/1, clock and bridging defects in a part of circuit. (Figure 6)

4. Design and Implementation of Fault-Tolerant NNI Gate

In the NNI gate, the critical area that is prone to cell addition fault is the central part of the gate, and in the event of such fault, the NNI gate becomes the majority gate. (Figure 7). The first fault-tolerant NNI gate is designed in Figure 8. This gate, with 40 extra cells, is fully tolerable against cell addition fault in the critical area. Figure 9 shows the fault tolerant NNI gate and its simulation results in a normal state and in the event of extra cell defect.

A) Design XOR/XNOR Gate

In the design of digital circuits, in addition to OR/NOR, AND/NAND, and inverter, two XOR/XNOR gates with a very special function play an important role in designing circuits such as generating even or odd parity, fault detection and correction, etc. These two gates are usually in the form of two inputs and one output. Using Boolean functions, these two gates can be implemented in different ways. Figure10 shows an example of this

implementation with logic gates (a) fault-tolerant NNI gate simulation in QCA Designer, (b) fault-tolerant NNI gate in the event of defects.

B) Design Fault-tolerant XOR/XNOR Gate

In order to design an XOR/XNOR gate which is tolerant of some defects in QCA technology, some actions have been taken as follows:

- Using the majority gate [19] and fault-tolerant NNI (Figure 11)
- Designing a new prototype of the majority gate in a 4×4 full-block with the aim of preventing the stuck-at 0/1 faults [19] (Figure 12).
- Strengthening the wires using extra cells to reduce rotation fault or cell omission in the information transmission path [19].
- Inserting at least three cells in a clock to prevent clock defect in the absence of one cell [19].
- Spacing at least two cells between wires to prevent clock defect in the event of a cell addition fault [19].

Finally, the schematic and implementation of XOR and XNOR gates is as shown in Figure 13. Also, the XOR/XNOR gate and its simulation results are as shown in Fig. 14, which is 100% tolerant of stuck-at 0/1, clock, and bridging fault. It is also 52% and 89% tolerant of cell omission and cell addition faults. In Table I and II the result of cell addition and cell omission are shown.

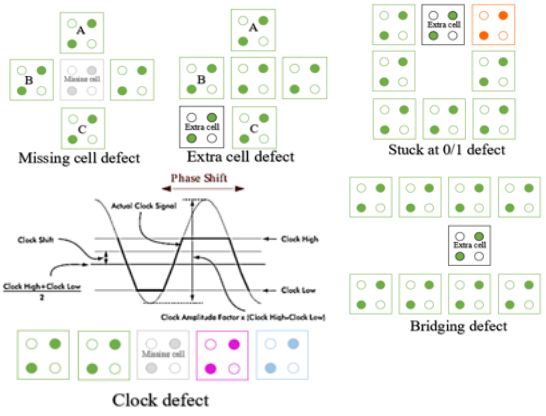

Fig. 6. A schematic diagram of the missing cell, extra cell defects in a majority gate and stuck at 0/1, clock [26] and bridging defects in a part of circuit

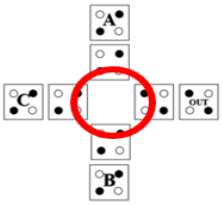

Fig. 7. Critical area at the NNI gate

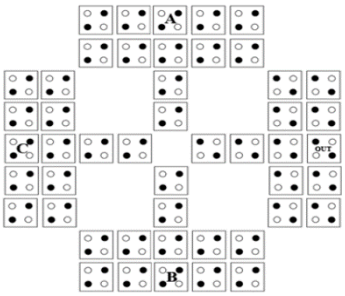

Fig. 8. Fault-tolerant NNI gate

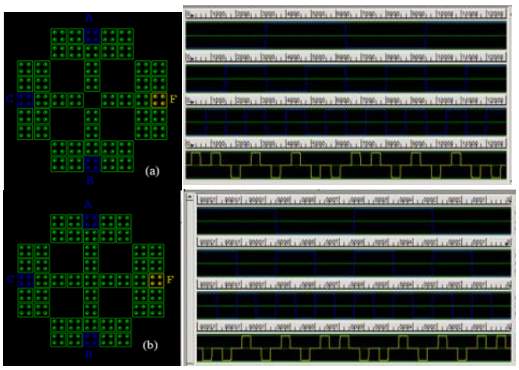

Fig. 9. Schematic, implementation, and simulation of XOR and XNOR

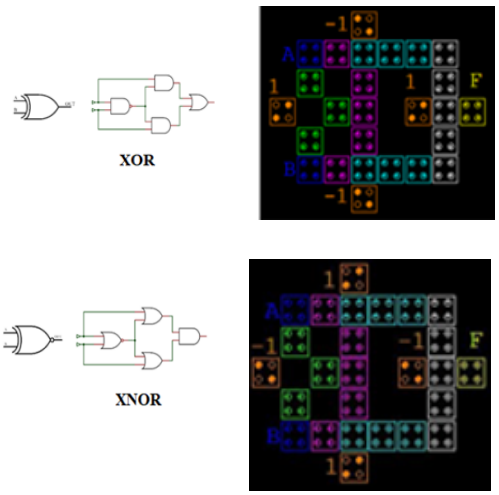

Fig. 10. Schematic . (a) fault-tolerant NNI gate –(b) fault-tolerant Majority gate [19]

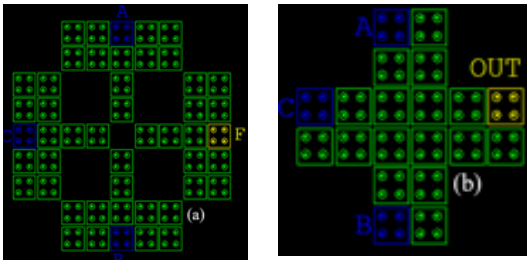

Fig. 11. 4×4 block majority gate

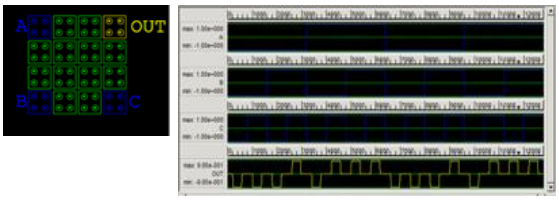

Fig. 12. Schematic and implementation of XOR and XNOR gates

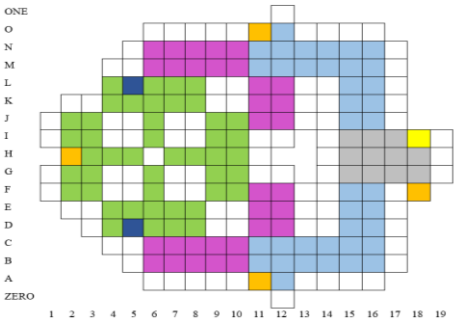

Fig. 13. XOR and XNOR gates simulation results

Fig. 14. Comparing the Results

So far, various models of XOR/XNOR gates were proposed, each pursues a specific purpose; therefore, it is essential that the proposed XOR/XNOR gate be compared with those XOR/XNOR gates which pursue exactly the same goal. In [27]v, [28], two models of XOR gate are presented with the approach of combining the three-input and five-input majority gates. (Figure 15)

In addition to their advantages, both gates presented in [27], [28], may have defects in the areas shown in Figure 16 However, in the proposed gate, these defects are avoided as much as possible. Table 3 compares the physical condition of these two gates with the proposed gate. The columns compare the number of cells used, latency, area, wiring, freedom of inputs and output, respectively. Figure 17 shows that in comparison with other gates, the proposed gate is tolerant of cell omission and cell addition faults.

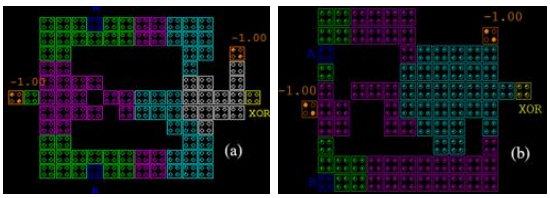

Fig. 15. XOR gates presented in part (a) of the gate presented in [27] (b) of the gate presented in [28]

Table.1.
Simulation results of cell addition fault in the proposed gate

Cell	*Out*	*Cell*	*Out*	*Cell*	*Out*	*Cell*	*Out*	*Cell*	*Out*	*Cell*	*Out*
A6	T	C4	T	E9	F	G8	F	I7	F	J14	T
A7	T	C5	T	E10	F	G11	T	H13	T	J17	T
A8	T	C17	T	E13	T	G12	T	I8	F	K2	T
A9	T	D3	T	E14	T	G13	T	I11	T	K3	T
A10	T	D9	T	E17	T	G14	T	I13	T	K9	F
A11	T	D10	F	F1	T	H1	T	I12	T	K10	T
A12	T	D13	F	F4	T	H6	T	I14	T	K13	T
A13	T	D14	T	F5	F	H11	T	J1	T	K14	T
A14	T	D17	T	F7	T	E18	T	J4	T	J18	T
A15	T	E2	T	F8	F	O8	T	O9	T	O10	T
A16	T	E3	T	F13	T	O15	T	O16	T	H1	T
B5	T	H12	T	F14	T	L3	T	L14	T	M5	T
B17	T	F19	T	F18	T	L9	T	K17	T	M17	T
Z11	T	Z12	T	O7	T	L10	T	L17	T	N5	T
O12	T	O13	T	O14	T	L13	F	M4	T	N17	T
G1	T	H14	T	J5	T	L14	T	O11	T	O6	T
G4	T	I1	T	J7	T	K17	T	ONE12	T	M4	T
G5	T	I4	T	J8	F	L17	T	O11	T	J13	T
G7	T	I5	T								

Table.2.
Checking cell omission fault

Cell	*Out*	*Cell*	*Out*	*Cell*	*Out*	*Cell*	*Out*	*Cell*	*Out*	*Cell*	*Out*	*Cell*	*Out*
A12	T	C9	T	D16	T	F11	F	M8	F	K8	F	I14	T
B6	T	C10	F	E4	T	F12	F	M9	T	K11	F	I15	T
B7	T	C11	T	E5	F	F15	T	M10	F	K12	F	I16	F
B8	T	C12	F	E6	F	F16	F	M11	T	K15	T	I17	F
B9	T	C13	T	E7	F	G2	T	M12	F	K16	F	J2	T
B10	T	C14	T	E8	F	G3	T	M13	T	L4	T	J3	T
B11	F	C15	F	E11	T	G6	F	M14	T	L6	F	J6	F
B12	F	C16	T	E12	T	G9	F	M15	F	L7	F	J9	F
B13	F	D4	T	E15	T	G10	F	M16	T	L8	F	J10	F
B14	F	D6	F	E16	T	G15	T	N6	T	L11	T	J15	T
B15	T	D7	F	F2	T	G16	F	N7	T	L12	F	J16	F
B16	T	D8	F	F3	T	G17	T	N8	T	L15	T	K4	T
C6	T	D11	T	F6	F	G18	T	N9	T	L16	T	K5	T
C7	T	D12	F	F9	F	H2	T	N10	T	M6	T	K6	F
C8	T	D15	T	F10	F	H3	T	M7	T	O12	T	H7	T
H4	F	I10	F	K7	F	N16	T	H5	F	H8	T	H9	F
H10	F	H16	F	I2	T	I9	F	H14	T	H17	F	I3	T
N15	T	H15	F	H18	T	I6	F						

Table 3.
Comparison between XOR gates and the proposed gate

XOR	*Cell count (complexity)*	*Area um^2*	*Latency (clock cycle)*	*Cross*	*Accessible I/O*
a	96	0.09	1	No	yes/yes
b	85	0.07	0.75	No	yes/yes
Proposed	133	0.1	1	No	yes/yes

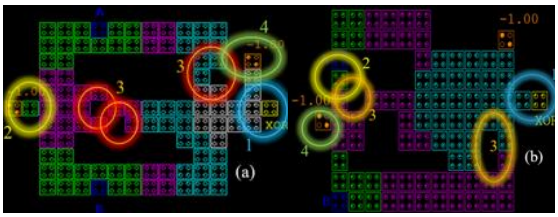

Fig. 16. XOR Defects that may occur in the two gates presented in [27] and [28]: 1- Poor output transmission, 2- Possibility of clock fault, 3- Possibility of bridging fault, 4- Possibility of stuck-at-0/1 fault.

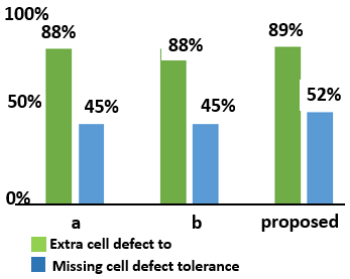

Fig. 17. XOR Comparing the tolerance of the proposed gates against defects with those of the gates presented in [27], [28] (shown in Figure 15)

5. Conclusion

QCA is one of the candidates and has its own advocates. Due to its low power consumption, high density and speed, small size, and, QCA is one of the most prominent alternatives for CMOS. However, despite these advantages, just like any other technology, it has some problems too. The most notable of those is its relatively high fault-rate. Therefore, the design and implementation of highly reliable integrated circuits while keeping the complexity at a minimum level is imperative for this technology. In this paper, we presented the first NNI gate which is tolerant of cell addition fault. Also, in order to prevent the stuck-at 0/1fault, we implemented a new sample of majority gate. Using these gates and following some rules based on the experience, the fault-tolerant XOR/XNOR gate was designed and implemented. This gate is completely tolerant of stuck-at 0/1, clock and bridging fault and is more tolerant of cell addition and cell omission fault in comparison with its counterparts.

References

[1] I. Hänninen, “Computer arithmetic on quantum-dot cellular automata nanotechnology,” Tampere University of Technology , 2009

[2] M. Liu, “Robustness and power dissipation in quantum-dot cellular automata”, University of Notre Dame, 2006, DOI: 10.7274/3t945q49m56

[3] G. E. Moore, “Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp. 114 ff,” IEEE solid-state circuits society newsletter, vol. 11, no. 3, pp. 33-35, 2006.

[4] M. I. Mendelson, “Learning Bio-micro-nanotechnology, ” CRC Press, 2013, ISBN-13: 978-1138076310

[5] P. D. Tougaw, and C. S. Lent, “Logical devices implemented using quantum cellular automata,” Journal of Applied physics, vol. 75, no. 3, pp. 1818-1825, 1994, DOI: 10.1016/S0016-0032(97)00041-0

[6] C. S. Lent et al., “Quantum cellular automata,” Nanotechnology, vol. 4, no. 1, pp. 49, 1993, DOI: 10.1088/0957-4484/4/1/004

[7] C. S. Lent, and P. D. Tougaw, “Lines of interacting quantum-dot cells: A binary wire,” Journal of applied Physics, vol. 74, no. 10, pp. 6227-6233, 1993, DOI: 10.1063/1.355196

[8] C. S. Lent, and P. D. Tougaw, “A device architecture for computing with quantum dots,” Proceedings of the IEEE, vol. 85, no. 4, pp. 541-557, 1997, DOI: 10.1109/5.573740

[9] J. Huang et al., “Design and characterization of an and-or-inverter (AOI) gate for QCA implementation,” Proceedings of the 14th ACM Great Lakes symposium on VLSIApril 2004, pp. 426–429, DOI: 10.1145/988952.989054

[10] S. Ghosal, and D. Biswas, “Study and Defect Characterization of a Universal QCA Gate,” International Journal of Computer Applications, vol. 975, pp. 8887, 2013

[11] K. Das, and D. De, “Characterization, test and logic synthesis of novel conservative and reversible logic gates for QCA,” International Journal of Nanoscience, vol. 9, no. 03, pp. 201-214, 2010, DOI: 10.1142/S0219581X10006594

[12] K. Das, and D. De, “A study on diverse nanostructure for implementing logic gate design for QCA,” International Journal of Nanoscience, vol. 10, no. 01n02, pp. 263-269, 2011, DOI: 10.1142/S0219581X11007892

[13] S. E. Frost et al., “Carbon nanotubes for quantum-dot cellular automata clocking,” 4th IEEE Conference on Nanotechnology, 2004, DOI: 10.1109/NANO.2004.1392286

[14] T. D. Lantz, “A QCA implementation of a look-up table for an FPGA,” 2006 IEEE International Conference on Reconfigurable Computing and FPGA's (ReConFig 2006), DOI: 10.1109/RECONF.2006.307763

[15] R. Farazkish, “A new quantum-dot cellular automata fault-tolerant full-adder,” Journal of Computational Electronics, vol. 14, no. 2, pp. 506-514, 2015, DOI: 10.1007/s10825-015-0668-2

[16] T. J. Dysart et al., “An analysis of missing cell defects in quantum-dot cellular automata,” Physics, 2005, Corpus ID: 14799148

[17] M. B. Tahoori et al., “Defects and faults in quantum cellular automata at nano scale,” 22nd IEEE VLSI Test Symposium, 2004. Proceedings, DOI: 10.1109/VTEST.2004.1299255

[18] M. Poorhosseini, “Novel defect Terminolgy beside evaluation and design fault tolerant logic gates in quantum-dot cellular automata,” Journal of Advances in Computer Engineering and Technology, vol. 2, no. 1, pp. 17-26, 2016.

[19] D. Kumar, and D. Mitra, “A systematic approach towards fault-tolerant design of QCA circuits,” Analog Integrated Circuits and Signal Processing, vol. 98, no. 3, pp. 501-515, 2019, DOI: 10.1007/s10470-018-1270-x

[20] M. Poorhosseini, and A. R. Hejazi, “A fault-tolerant and efficient XOR structure for modular design of complex QCA circuits,” Journal of Circuits, Systems and Computers, vol. 27, no. 07, pp. 1850115, 2018, DOI: 10.1142/S0218126618501153

[21] A. Fijany, and B. N. Toomarian, “New design for quantum dots cellular automata to obtain fault tolerant logic gates,” Journal of nanoparticle Research, vol. 3, no. 1, pp. 27-37, 2001, DOI: 10.1023/A:1011415529354

[22] R. Farazkish, S. Sayedsalehi, and K. Navi, “Novel design for quantum dots cellular automata to obtain fault-tolerant majority gate,” Journal of Nanotechnology, vol. 2012, 2012, DOI: 10.1155/2012/943406

[23] D. Kumar, D. Mitra, and B. B. Bhattacharya, “On fault-tolerant design of exclusive-OR gates in QCA,” Journal of Computational Electronics, vol. 16, no. 3, pp. 896-906, 2017, DOI: 10.1007/s10825-017-1022-7

[24] M. J. Beard, “Design and simulation of fault-tolerant quantum-dot cellular automata (QCA) NOT gates,” Wichita State University, College of Engineering, Dept. of Electrical and Computer Engineering, 2006.

[25] S.-S. Ahmadpour, and M. Mosleh, “A novel fault-tolerant multiplexer in quantum-dot cellular automata technology,” The Journal of Supercomputing, vol. 74, no. 9, pp. 4696-4716, 2018, DOI: 10.1007/s11227-018-2464-9

[26] M. Ottavi et al., “On the error effects of random clock shifts in quantum-dot cellular automata circuits,” 22nd IEEE

International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT 2007), DOI: 10.1109/DFT.2007.58

[27] F. Shabani, and S. Sayedsalehi, "Design of a Novel Fault Tolerant XOR Structure in Quantum Dot Cellular Automata," Majlesi Journal of Electrical Engineering, vol. 12, no. 3, pp. 35-40, 2018

[28] S. Aghababaei, and S. Sayedsalehi, "New Approach to Design and Implementation XOR Gate in QCA Technology," Signal Processing and Renewable Energy, vol. 2, no. 2, pp. 15-24, 2018, DOI: 20.1001.1.25887327.2018.2.2.2.2