A Review on QCA Multiplexer Designs

Mohsen Yoosefi Nejad[1*], Mohammad Mosleh[2]
1- Department of Computer Engineering, and Information Technology, Payame Noor University, Iran
Email: m_yoosefi@pnu.ac.ir (Corresponding author)
2- Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful, Iran
Email: mosleh@iaud.ac.ir

Received: August 2017 Revised: March 2017 Accepted: April 2017

ABSTRACT
Quantum-dot Cellular Automata (QCA), is a contemporary nanotechnology for manufacturing logical circuits which brings less power consumption, smaller circuit size, and faster operation. In this technology, logical gates are composed of nano-scale basic components called cells. Each cell consists of four quantum-dot arranged in a square pattern. Diagonal arrangement of two extra electrons resembles two logical states 0 and 1. Majority gate and inverter gate are considered as the two most fundamental building blocks of QCA. The effect of cells on their neighbor cells enables designing more diverse circuits. Multiplexer is a key component in most computer circuits. Researchers have presented various QCA designs for multiplexers since the introduction of QCA. In this research all presented designs are simulated in QCA Designer tool version 2.0.3 and investigated from different aspects such as complexity, occupied area, types of components used in circuit, number of layers, and delay.

KEYWORDS: Multiplexer, Quantum-dot Cellular Automata (QCA), Majority Gate, Digital Circuit.

1. INTRODUCTION

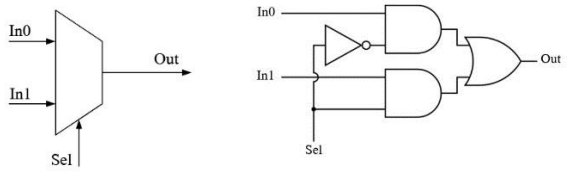
Quantum-dot Cellular Automata (QCA)[1], [2] as a known technology is an appropriate replacement for CMOS. Conventional CMOS technology has a lot of limitation while scaling into Nano-level [3]. This leads to fast development of molecular devices in Nano-scale. Several studies reported that QCA is able to develop systems with less power consumption and higher switching speed [4]. QCA logical circuits are built using a basic component called cell. Cells are nanometers in size. Since majority gates and inverters are the fundamental building blocks in QCA, various schemes and methods are proposed by researchers for other gates and circuits. Each one has its own characteristics, advantages and disadvantages.

Multiplexer is one of useful components in digital circuits. An electronic multiplexer makes it possible for several signals to share one device or resource, for example one A/D converter or one communication line, instead of having one device per input signal. In addition to classical use of multiplexers such as bus controllers and arithmetic logic unit (ALU), multiplexing has been utilized in multistage interconnection networks [5], cellular networks [6], information security [7], wideband digital communication [8], and on-chip networks [9]. The simplest form of a multiplexer is 2:1 multiplexer, which can transfer one of its two inputs to the output using the select line. Fig. 1 illustrates a 2:1 multiplexer and its logical circuit. When select line is zero (Sel=0) the input line In_0 is forwarded to the output and when select line is one (Sel=1) the input line In_1 is selected and forwarded to the output. This behavior can be formulated as Eq.1. Small multiplexer can be cascaded to develop a larger multiplexer. For example, one can make a 4:1 multiplexer using three 2:1 multiplexers [10]. Fig. 2 shows how to implement a 4:1 multiplexer using three 2:1 ones including select lines orders and corresponding output. Implementation of any logical function using multiplexers is an important characteristic. Fig. 3 illustrates the implementation of a function with three inputs using an 8:1 multiplexer. Generally a 2^N:1 multiplexer is needed to implement a logical function with N inputs. To date, various QCA designs have been proposed for multiplexers. In this research, in addition to simulating the proposed designs using QCADesigner, different aspects of the circuits such as number of cells, occupied area, types of components used in circuit, number of layers, and number of cycles for producing output have been investigated. QCADesigner is a rapid design and simulation tool for Quantum-dot Cellular Automata developed in University of Calgary[11].

$$Out = In0.\overline{Sel} + In1.Sel \quad (1)$$

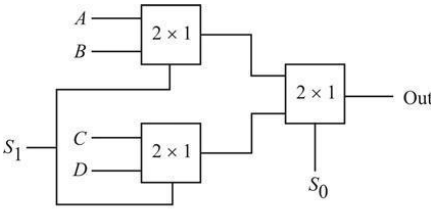
The remainder of this paper is organized as follows: In section 2, Quantum-dot Cellular Automata and its

components are introduced. Proposed multiplexer designs by different researchers and simulation results for each design are presented separately in section 3. Section 4 discusses about each design and compares them against each other. Conclusion comes at the end.



(a) (b)

Fig. 1. 2:1 multiplexer. (a) Block Diagram. (b) Logical Circuit.

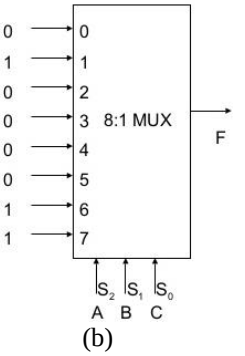


S_0	S_1	Out
0	0	A
0	1	B
1	0	C
1	1	D

Fig. 2. Implementation of a 4:1 multiplexer using three 2:1 multiplexers.

f(a, b, c) = a'b'c + ab

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



(a) (b)

Fig. 3. Implementation of logic function f(a,b,c) = a'b'c+ab using an 8:1 multiplexer. (a) The truth table of function. (b) Implementation

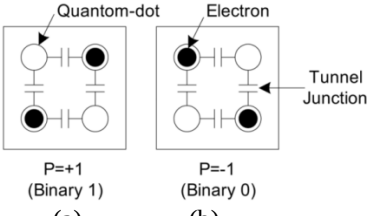
2. QCA OVERVIEW

In this section, different aspects of Quantum-dot Cellular Automat will be presented. This includes basic components of QCA, wire crossing, and clocking in QCA.

2.1. Basic Components of QCA

The building blocks in QCA circuits are cells. Each cell consists of four quantum dots. As can be seen in Fig. 4, four quantum dots are located in corners of a square. These quantum dots are sites electrons can occupy by tunneling to them. Totally six different states are possible for positioning these two electrons in four quantum dots, but due to coulomb repulsion between electrons, they tend to stay in a position having longest distance from each other, so all of these states are not stable, and just two states are stable in which electrons occupy quantum dots diagonally. Numbering the holes clockwise, the polarity is computed using Eq. 2. In this equation, if hole number *i* is occupied by an electron, P*i*=1, otherwise P*i*=0. The possible values for P are -1 and +1 which corresponds to logical 0 and 1 respectively[12].

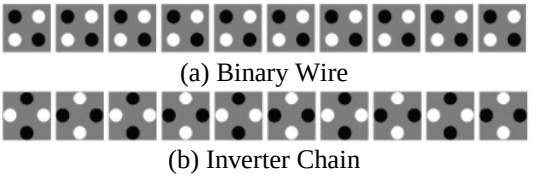
$$P = \frac{(p1 + p3) - (p2 + p4)}{p1 + p2 + p3 + p4} \quad (2)$$



(a) (b)

Fig. 4. (a) +1 Polarity (Logical 1) (b) -1 Polarity (Logical 0)

Electrons tunnels through holes in a nonlinear motion, when they move in a cell. The distances between holes are usually 20nm. In addition to the coulomb repulsion in a cell, each cell affects its neighbors. Two adjacent cells pose a state, which minimizes the coulomb repulsion. Grid arrangements of quantum-dot cells behave in ways that allow for computation. The simplest practical cell arrangement is given by placing quantum-dot cells in series, to the side of each other making a QCA wire. There are two types of wires possible in QCA: A simple binary wire as shown in Fig. 5a and an inverter chain, which is constituted by placing 45-degree inverted QCA cells side by side as shown in Fig. 5b [13].



(a) Binary Wire

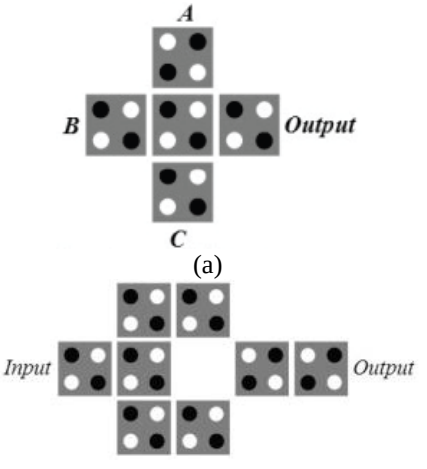
(b) Inverter Chain

Fig. 5. (a) Binary Wire (b) Inverter Chain

The basic QCA gates are inverter gate (NOT), and majority gate[14]. Fig. 6 illustrates inverter gate and majority gate. The logical function of majority gate is as Eq. 3.

$$F(A, B, C) = AB + AC + BC \quad (3)$$

In which, A, B, and C are inputs and F is the output. If one of majority gate inputs is fixed to logical 1, the gate performs as an OR gate. Also in case of fixing one input to logical 0, the majority gate performs as an AND gate. The proof is shown in table 1 which is the truth table of majority gate[15].



(a)

(b)

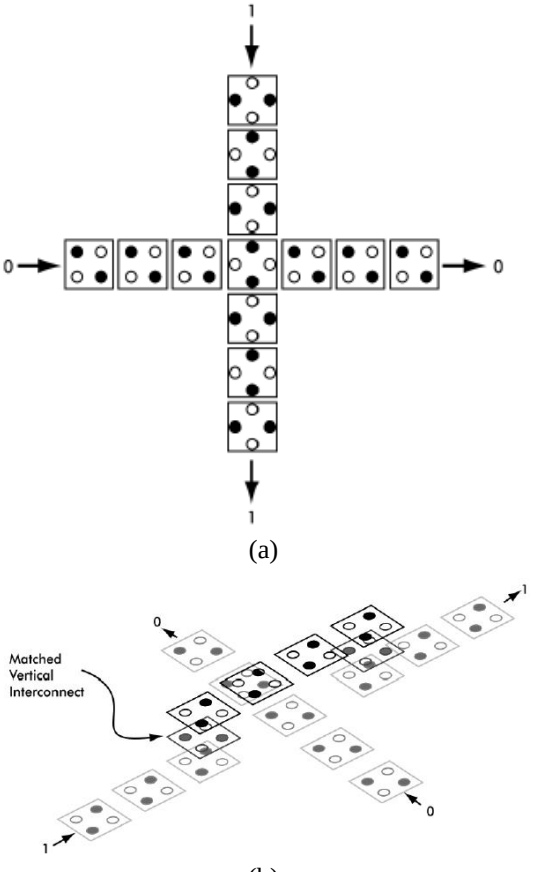
Fig. 6. (a) Majority Gate, (b) Inverter Gate

Table 1. Truth table of majority gate

A	B	C	F	Proof
0	0	0	0	A=0
0	0	1	0	F=BC
0	1	0	0	
0	1	1	1	
1	0	0	0	A=1
1	0	1	1	F=B+C
1	1	0	1	
1	1	1	1	

2.2. Wire Crossing

There are two approaches for crossing two wires. The in-plane approach, achieved by crossing a wire and an inverter chain. Although this crossing is in one plane, the passing signals do not affect each other. Another approach is to place each wire in a separate plane. In order to move a wire to another plane, intermediate cells are required. Fig. 7 shows two wire crossing approaches. Fig. 8 illustrates schematics for crossing two wires; via cells displayed as large circles in a square and crossover cells displayed as X signs.



(a)

(b)

Fig. 7. Crossing Wires. (a) In-plane Crossing. (b) Multilayer Crossing.

Fig. 8. Schematics for multilayer crossing

2.3. QCA Clocking

The clocks of a QCA system not only serve as controlling data flow direction, it also supplies the power of automaton. Signal energy lost to the environment is restored by the clock. This clock scheme, as shown in Fig. 9.a, consists of four phases: Switch, Hold, Release, and Relax. In the switch phase, the inter-dot barrier is slowly raised and the cell attains a definitive polarity under the influence of its neighbors. In the hold phase, barriers are high and a cell keeps its polarity and acts as input to the adjacent cells. During the release phase, barriers are lowered and the cell loses its polarity. In the relax phase, the cell has no polarity and electrons move freely in the cell. As can be observed in Fig. 9.b, the QCA circuit is divided into so-called clocking zones. All cells in a specific

zone are controlled by the same clock signal and perform a specific calculation. A subset of cells in same clock zone forms a sub array. When the sub array is in switch state, it is affected by the adjacent stable cells, which are in hold state. Fig. 10 displays QCA wire operation in different clock zones. As can be seen in Fig. 11, clock zones make QCA perform as a pipeline[16].

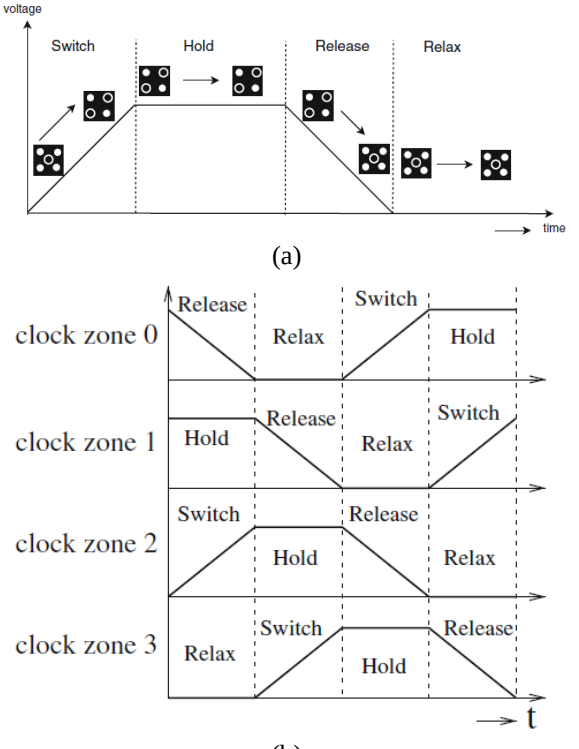


Fig. 9. (a) Four clock phases in QCA (b) Signal clock zones.

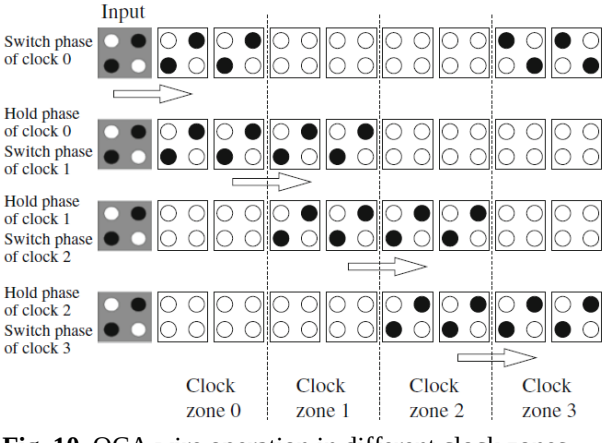


Fig. 10. QCA wire operation in different clock zones.

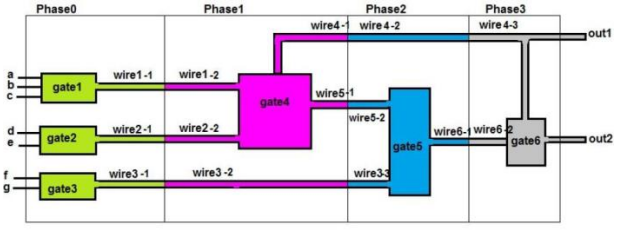


Fig. 11. Clock zones make QCA perform as a pipeline.

3. PREVIOUS WORKS

A variety of QCA multiplexer circuit designs has been proposed up to now. In this section, various proposed multiplexer designs will be presented and discussed.

3.1. Mardiris et al. Multiplexer

One of the earliest QCA multiplexer designs was a 2:1 multiplexer proposed by V.A. Mardiris et al. in 2008 [17]. A high-level block diagram of the design is illustrated in Fig. 12 where i0 and i1 are input lines; s is the selector line and out corresponds to the output of multiplexer.

In this design, Eq. 1 has been employed as the function of multiplexer. The design includes an AND block, an ($a.\bar{b}$) AND block and an OR block. It also includes a signal delay block at the i1 input of the multiplexer. All blocks are colored according to the clock zone they use. Using the pipelining structure the proper signals arrive simultaneously at the inputs of the AND and OR blocks. Fig. 13 shows the implementation of these blocks. The AND blocks are implemented using a majority gate with an input fixed to logic '0' which is equivalent to an AND gate with two inputs. The second input of majority gate is supplied from the left side of block, and the third input comes from top. The top input of block is an inverter chain walks vertically the block, crosses the horizontal line comes from the left input and fans out to the bottom output of the block and to the third input of majority gate. The output of majority gate arrives at the right side of the block. The OR gate is implemented using a majority gate with a fixed input to logic '1'. The block gate is also implemented using cells with different clock zones making a delay of one-quarter clock period. Fig. 14 illustrates the complete design. The simulation results are shown in Fig. 15.

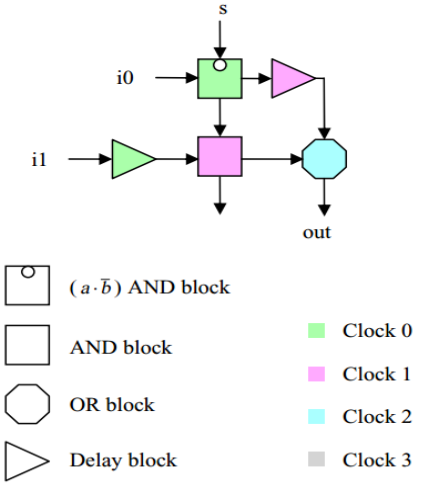


Fig. 12. High-level block diagram of 2:1 multiplexer presented by Mardiris et al.

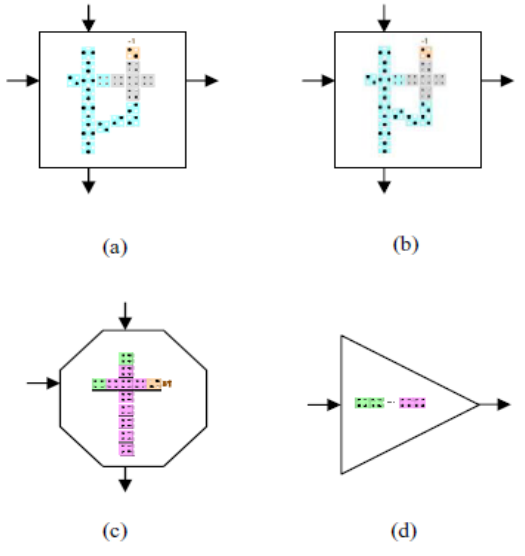


Fig. 13. Elementary blocks layout implementation of the 2x1 multiplexer designed by Mardiris et al., (a) AND block, (b) $(a.\bar{b})$ AND block, (c) OR block, and (d) Delay block

3.2. Teodosio and Sousa Multiplexer

In September, 2007, T. Teodosio and L. Sousa presented QCALG, a layout generation tool for QCA circuits [18]. For comparison purposes, they first presented a handmade version of a 2:1 multiplexer. The presented tool in the thesis also generated a layout for the same circuit automatically. In this section both manually elaborated and automatically generated layouts have been reviewed.

3.2.1. Manual multiplexer

As it is shown in Fig. 16, Eq. 1 is employed for implementing a 2:1 multiplexer. Two majority gates with fixed input to logic '0' (-1 polarity) are visible at top and bottom of layout perform as two AND gates. The left input of top majority gate is In0. Selector line, Sel, is negated using an inverter gate bringing $\overline{Sel}$ as the other input of top majority gate. The inputs of top majority gate supplied at clock0 and the output, $In0.\overline{Sel}$, produced at clock1. Similarly, the left input of bottom majority gate is In1 and the Selector line, Sel, arrives at top input of the majority gate, both at Clock0. The output, In0.Sel produced at Clock1. There is another majority gate at right side of the schematics with fixed input to logic '1' (+1 polarity) performing as an OR gate. The output of previous mentioned majority gates arrives at top and bottom inputs of this majority gate at Clock2. The final result, $In0.\overline{Sel} + In1.Sel$, which is the output of circuit is computed at Clock3.

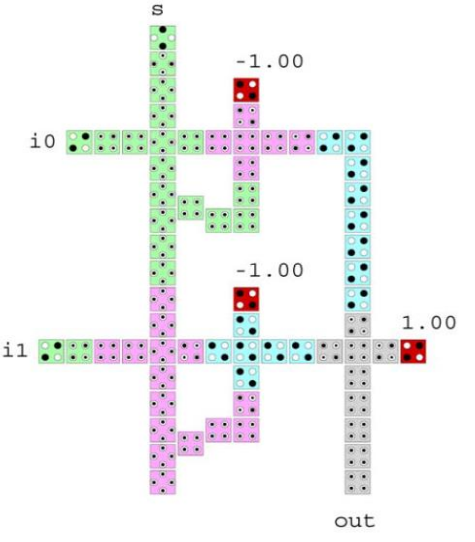


Fig. 14. 2:1 The QCA multiplexer presented by Mardiris et.al.

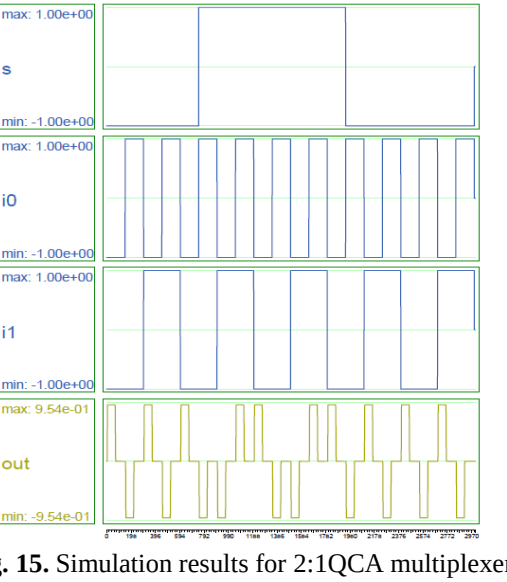


Fig. 15. Simulation results for 2:1QCA multiplexer presented by Mardiris et al.

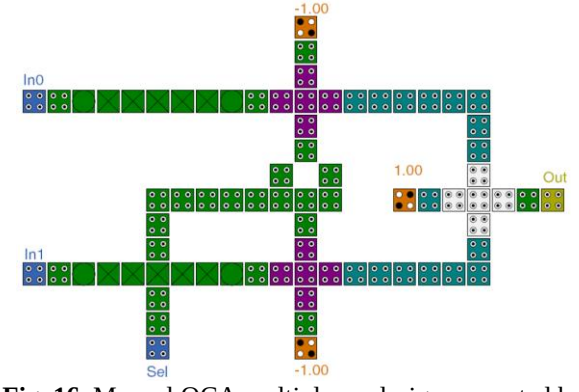


Fig. 16. Manual QCA multiplexer design presented by T. Teodosio and L.Sousa

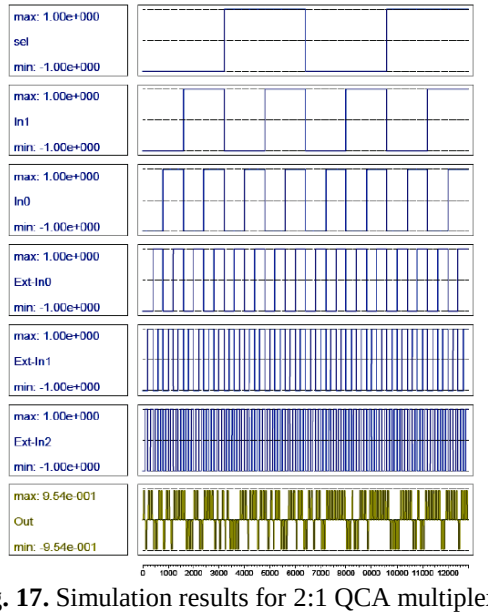


Fig. 17. Simulation results for 2:1 QCA multiplexer manually designed by T. Teodosio and L.Sousa

3.2.2. QCALG automatically generated layout

QCA Layout Generator (QCALG) is a tool developed by T. Teodosio and L. Sousa, which automatically create a layout for an input logic function. In-wire inverter is a new structure for inverting a signal proposed which is a half-cell along a binary wire. Fig. 18 compares usual and in-wire inverter for an input signal. Fig. 19 shows the automatically produced layout of 2:1 multiplexer by QCALG. The software does not use different symbols for via cells. As can be seen in the figure, Sel signal, which is inverted using in-wire inversion structure, enters right hand majority gate. In0 signal passes over the wire carrying Sel signal and enters into another input of the same majority gate. The third input of the mentioned majority gate is fixed to logic '0' to convert the majority gate to an AND gate, resulting In0.$\overline{\text{Sel}}$ as the output of the majority gate. At the left hand of the circuit, a similar majority gate exist which produces In1.Sel. At the bottom of the circuit, the third majority gate with logic '1' fixed input performs as a two input OR gate which its inputs are come from previously mentioned majority gates, results in Eq.1 as the final output of the multiplexer.

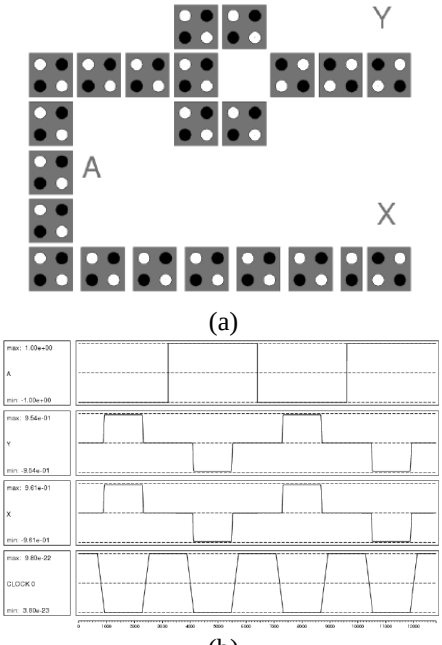


Fig. 18. (a) Comparison between usual and in-wire inverters. A is input cell. The usual inverter's output is labeled as Y, and the output of the in-wire inverter is X. (b) Simulation results.

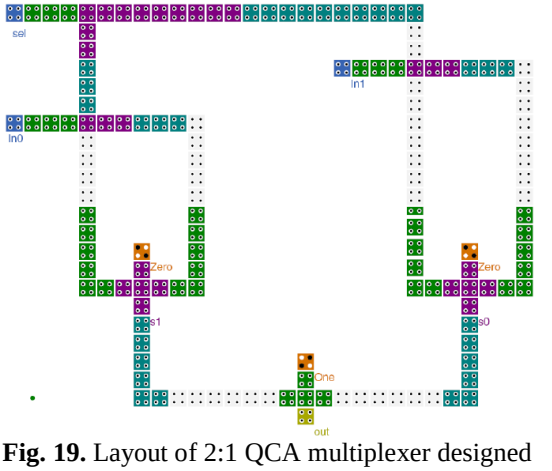


Fig. 19. Layout of 2:1 QCA multiplexer designed automatically by QCALG software.

3.3. Kim et al. Multiplexer

In another research conducted by K. Kim et al.[19] with the aim of designing a robust QCA adder, a 2:1 multiplexer is designed by the authors as a part of serial adder. In the design proposed by K. Kim and his coworkers, Eq.1 is employed as logic function of the

multiplexer. As can be observed in Fig. 20, two majority gates with fixed input '0', perform as two AND gates in the circuit. The left input of the top majority gate is In0, and the bottom input is $\overline{\text{Sel}}$. Both inputs arrive at Clock0. The output of top majority gate is In0.$\overline{\text{Sel}}$. Likewise, the left input of the bottom majority gate is In1 and Sel enters at its top input. The inputs of this gate arrive at Clock0 too. The output of the gate is In1.Sel. Both mentioned majority gates produce their outputs at Clock1 which enter the third majority gate at Clock2. Since one of the inputs of third majority gate is fixed to logic '1', it performs as an OR gate. The computed function of multiplexer is available at Clock3.

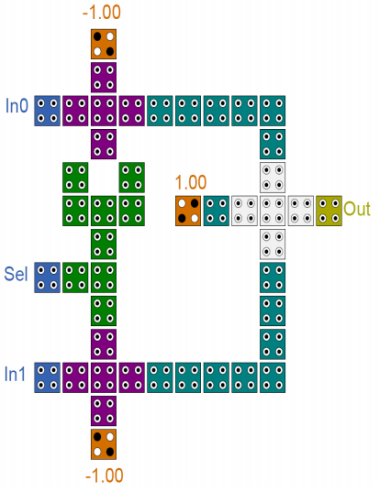


Fig. 20. K. Kim et al. 2:1 multiplexer

3.4. Hashemi et. al. Multiplexer

In 2008, another work done by Hashemi et al. [20] which leads to design a different 2:1 multiplexer, the authors use a different function for designing the multiplexer which is formulated as Eq. 4.

$$Out = M(M(In1, \overline{Sel}, 1), M(In1, Sel, 0), In0) \qquad (4)$$

n which, *M* is the majority function.

When Sel=0, $\mathrm{M}(\mathrm{In}_1, \overline{\mathrm{Sel}}, 1) = \mathrm{M}(\mathrm{In}_1, 1,1) = 1$. Furthermore, $\mathrm{M}(\mathrm{In}_1, \mathrm{Sel}, 0) = \mathrm{M}(\mathrm{In}_1, 0,0)=0$. So, $\mathrm{Out} = \mathrm{M}(1,0, \mathrm{In}_0) = \mathrm{In}_0$. In addition, when Sel=1, $\mathrm{M}(\mathrm{In}_1, \overline{\mathrm{Sel}}, 1) = \mathrm{M}(\mathrm{In}_1, 0,1) = \mathrm{In}_1$, and $\mathrm{M}(\mathrm{In}_1, \mathrm{Sel}, 0) = \mathrm{M}(\mathrm{In}_1, 1,0) = \mathrm{In}_1$. Replacing in Eq. 4 results $\mathrm{Out} = \mathrm{M}(\mathrm{In}_1, \mathrm{In}_1, \mathrm{In}_0) = \mathrm{In}_1$. This is exactly the operation of a 2:1 multiplexer. When Sel=0, Out=In0, and when Sel=1, Out=In1. Fig. 21 shows the layout of the implemented function.

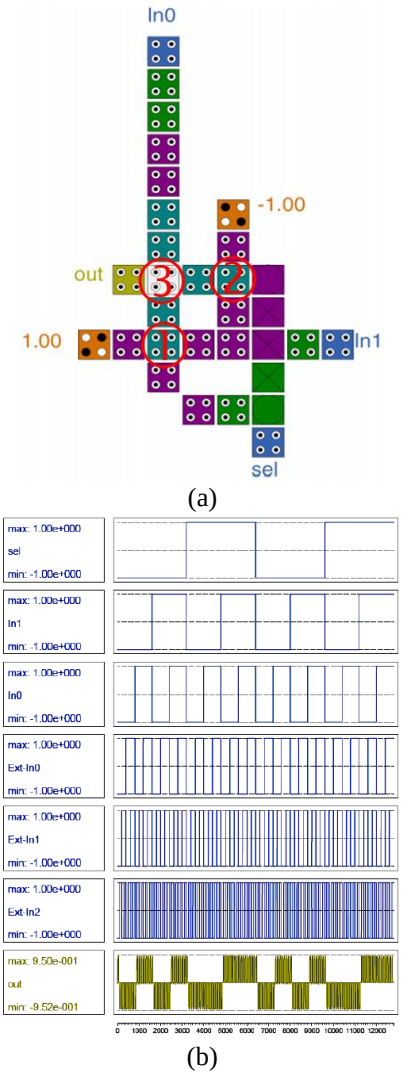


Fig. 21. (a) 2:1 The Multiplexer designed by Hashemi et al. (b) Simulation results.

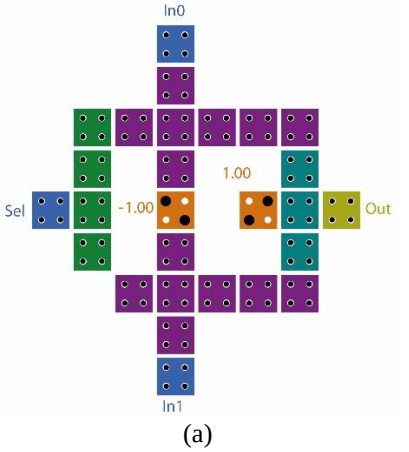
The first majority gate is equivalent to $\mathrm{M}(\mathrm{In}_1, \overline{\mathrm{Sel}}, 1)$ where In1 comes from right hand, $\overline{\mathrm{Sel}}$ from bottom and fixed value '1' comes from the left. The select signal is inverted with diagonally placement of two cells as can be seen in the figure. The second majority gate is implementation of $\mathrm{M}(\mathrm{In}_1, \mathrm{Sel}, 0)$. The bottom input is In1, the right hand input is Sel, and the top input is fixed zero value. The third majority gate implements the main function where its first input is supplied from the output of first majority gate, the second input comes from the second majority gate output and the third input is In0. This circuit has been constructed in three layers. The first layer is the main layer, which is the backbone of the circuit. The second layer from the bottom contained only via cells. The top layer only contains a wire, which transmits the select signal over the signal In1.

3.5. Roohi et al. multiplexer

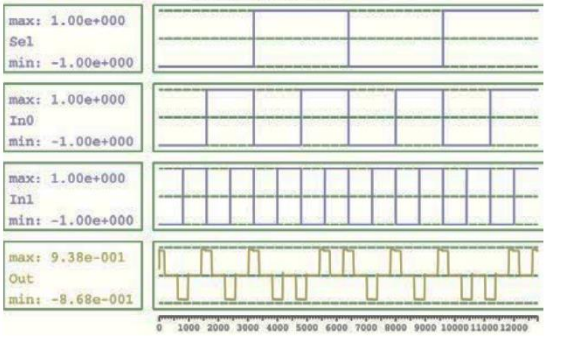
The 2:1 multiplexer design, proposed by A. Roohi et al. [21], which is illustrated in Fig. 22.a, implements Eq. 1. Two majority gates in top and bottom of the circuit share a fixed logical '0' input to act as AND functions. The top input of top majority gate is In0 and the left input comes from Sel. Likewise, the bottom input of bottom majority gate is In1 and the left input is $\overline{Sel}$, which comes diagonally from Sel input. Both majority gates perform the computation in Clock 1. Another majority gate can be seen on the right side of the circuit with a fixed logical '1' input. This majority gate plays the role of an OR gate and produces the final output in Clock 2. Figure 22.b shows the simulation results of the multiplexer.

3.6. Sen et al. multiplexer

One of the recent studies and presented designs of 2:1 multiplexer belongs to B. Sen and his coworkers in 2012 [22]. As can be observed in Fig. 23.a, the presented design is very compact. The multiplexer is implemented using Eq. 1 function. The first majority gate with a fixed logic '0' input performs as a two-input AND gate with first input I0 and second input $\overline{Sel}$. The select line is inverted with diagonally placement of two cells as shown in the figure. The second majority gate also performs as an AND gate with top input Sel and bottom input I1. The outputs of the majority gates enter the third majority gate, which has a fixed logic '1' input and performs as an OR gate and implements the main function. The simulation results are presented in Fig. 23.b.

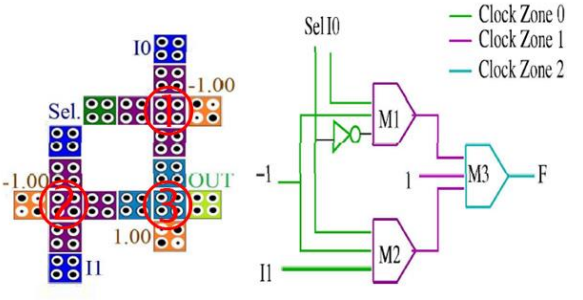


(a)

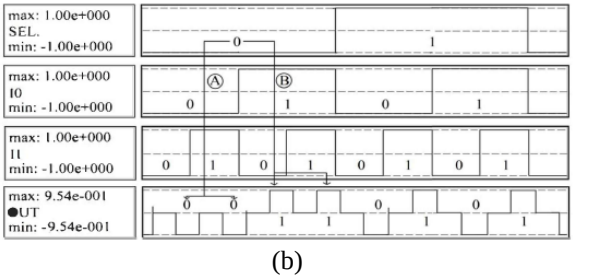


(b)

Fig. 22. (a) 2:1 The multiplexer designed by A. Roohi et. al. (b) Simulation results.



(a)



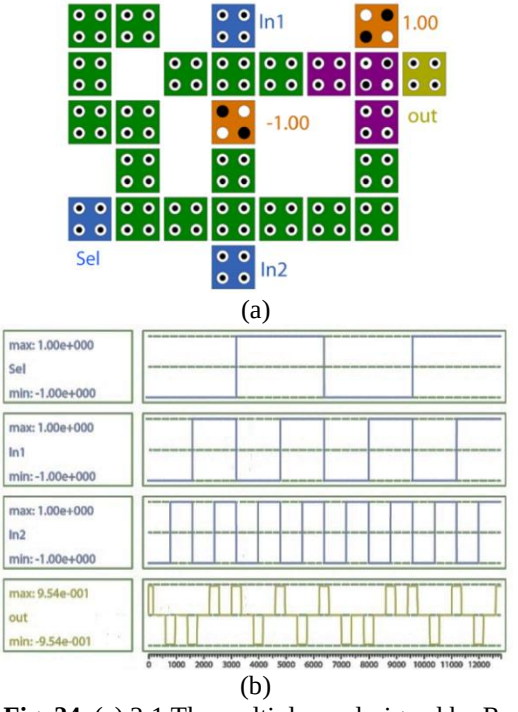
(b)

Fig. 23. (a) 2:1 The multiplexer designed by B. Sen et. al. (b) Simulation results.

3.7. Sabbaghi and Kianpour multiplexer

In a recent study, R. Sabbaghi and M. Kianpour [23] proposed a compact multiplexer design, which is depicted in figure 24.a. The circuit implements Eq. 1. Two majority gates are seen in top and bottom of the circuit with one input fixed to logical '0' acting as AND gates. Another input of these majority gates is an input, In1 for top gate and In2 for the bottom one. The bottom majority gate receives Sel signal directly from the left. The left input of the top majority gate is $\overline{Sel}$, which is inverted signal of Sel using a Tougaw inversion gate. All of the above-mentioned computations are performed in Clock 0. The majority gate on the right hand of the circuit with a fixed logical '1' as one of its inputs, acts as an OR gate, which combines the outputs of the AND gates and produces the final output in

Clock 1. The simulation results are presented in Fig 24.b.



(a)

(b)

Fig. 24. (a) 2:1 The multiplexer designed by R. Sabbaghi and M. Kianpour (b) Simulation results.

4. COMPARISON

All multiplexer designs have been implemented in QCA Designer version 2.0.3. The software configured for simulation in Bistable mode according to Table 2. The presented circuits have been compared by different aspects including number of cells, dimensions, area, latency, number of majority gates, type of inverter, and number of layers. The results of comparison are summarized in Table 3.

As it can be inferred from Table 3, the circuit, which was automatically designed by QCALG software, is the most complex and the largest circuit in terms of dimensions and the circuit presented by B. Sen et al. is the smallest. All presented designs use three majority gates. This is due to presence of three AND/OR operators in the Eq. 1, which has been employed in all designs except the presented design by Hashemi and her coworkers, which also used three majority functions in the proposed multiplexer function.

Table 2. The QCADesigner configuration for simulating in Bistable mode.

Attributes	Values
Number Of Samples	12800
Convergence Tolerance	0.001000
Radius of Effect [nm]	65.000000
Relative Permittivity	12.900000
Clock High	9.800000e-022
Clock Low	3.800000e-023
Clock Shift	0.000000e+000
Clock Amplitude Factor	2.000000
Layer Separation	11.500000
Maximum Iteration Per Sample	100

In terms of circuit delay, the multiplexer designed by Sabbaghi and Kianpour with 0.5 clock cycle is the fastest circuit. The multiplexers designed by B. Sen et al. and Roohi et al., both have latency of 0.75 cycle. The multiplexer which is automatically designed by QCALG has the latency of 2.25 cycles and is the slowest design. All other investigated circuits produce the output in one cycle. In terms of circuit area, the proposed designs by Sen et al., Sabbagi and Kianpour, Roohi el al., Kim et al., and Hashemi et al., are under 0.1 μm^2. Two designs proposed by Sen et al. and Sabbaghi and Kianpour with 0.02 μm^2 are minimum in circuit area. Four types of inverter structure can be observed in the investigated designs: 1) Inverter chain, 2) Tougaw inverter, 3) Inverter by placing two cells diagonally, and 4) In-wire inverter using a half-cell. Comparing the type of inverter against complexity, dimensions and area of the circuits shows that both smallest circuits (B. Sen et al. and S. Hashemi et al.) employ diagonally cell placement as the inversion structure.

Table 3. Presented multiplexers comparison

Multiplexer	Complexity (# cells)	Dimensions	Area (μm^2)	Latency (Clock cycles)	Majority gates	Inversion structure	Layers
V.A. Mardiris et al. [17]	67	14x18	0.15	1	3	Inverter Chain	1
T. Teodosio and L. Sousa [18], manual	84	22x14	0.19	1	3	Tougaw	3
T. Teodosio and L. Sousa [18], QCALG tool	151	29x22	0.39	2.25	3	In-wire	3
K. Kim et al. [19]	46	9x13	0.07	1	3	Tougaw	1
S. Hashemi et al. [20]	36	8x13	0.06	1	3	Diagonally cell placement	3
A. Roohi et al. [21]	27	9x8	0.03	0.75	3	Diagonally cell placement	1
B. Sen et al. [22]	19	7x7	0.02	0.75	3	Diagonally cell placement	1
R. Sabbaghi and M. Kianpour [23]	26	6x8	0.02	0.5	3	Tougaw	1

5. CONCLUSION

Multiplexers are important part of most logical circuits and control systems. In this study, various proposed designs for QCA multiplexer were reviewed. At first, six different designs were presented and the logic of the circuits was discussed. Afterward, the presented circuits were compared based on different aspects including number of cells, dimensions, area, latency, number of majority gates, type of inverter, and number of layers. The circuit, which was automatically designed by QCALG software, is the most complex and the largest circuit in terms of dimensions. In addition, the circuit also has the most latency. The proposed multiplexer by Sen and his coworkers, with the least number of cells and minimum size is the fastest circuit which is designed in a single layer. Four types of inverter structure has been observed in the investigated designs: 1) Inverter chain, 2) Tougaw inverter, 3) Inverter by placing two cells diagonally, and 4) In-wire inverter using a half-cell.

REFERENCES

[1] C. S. Lent, P. D. Tougaw, and W. Porod, **"Bistable saturation in coupled quantum dots for quantum cellular automata",** *Applied Physics Letters*, Vol. 62, pp. 714-716, 1993.

[2] C. G. Smith, "**Computation without current",** *Science*, Vol. 284, p. 274, 1999.

[3] J. Iqbal, F. Khanday, and N. Shah, "**Efficient Quantum Dot Cellular Automata (QCA) Implementation of Code Converters",** *Communications in Information Science and Management Engineering*, Vol. 3, p. 504, 2013.

[4] M. Kianpour, R. Sabbaghi-Nadooshan, "**Novel Design of n-bit Controllable Inverter by Quantum-dot Cellular Automata",** *International Journal of Nanoscience and Nanotechnology*, Vol. 10, pp. 117-126, 2014.

[5] S. Rajkumar, N. K. Goyal, "**Review of multistage interconnection networks reliability and fault-tolerance",** *IETE Technical Review*, pp. 1-8, 2015.

[6] M. I. Salman, M. Q. Abdulhasan, C. K. Ng, N. K. Noordin, A. Sali, and B. Mohd Ali, "**Radio resource management for green 3gpp long term evolution cellular networks: review and trade-offs",** *IETE Technical Review*, Vol. 30, pp. 257-269, 2013.

[7] H. W. Chang, W.-Y. Liang, and C. W. Chiou, **"Low Cost Dual-Basis Multiplier over GF (2 m) Using Multiplexer Approach",** in *Knowledge Discovery and Data Mining*, ed: Springer, 2012, pp. 185-192.

[8] S. Kamal, C. A. Azurdia-Meza, and K. Lee, "**Family of Nyquist-I Pulses to Enhance Orthogonal Frequency Division Multiplexing System Performance",** *IETE Technical Review*, Vol. 33, pp. 187-198, 2016.

[9] M. P. Kumar, S. Murali, and K. Veezhinathan, "**Network-on-chips on 3-D ICs: Past, present, and future",** *IETE Technical Review*, Vol. 29, pp. 318-335, 2012.

[10] S. Rawat, A. Sah, and S. Pundir, "**Implementation of Boolean functions through multiplexers with the help of Shannon expansion theorem",** *International Journal of Computer Applications (0975–8887)*, 2013.

[11] K. Walus, T. J. Dysart, G. A. Jullien, and R. A. Budiman, "**QCA Designer: A rapid design and simulation tool for quantum-dot cellular automata",** *IEEE Transactions on Nanotechnology*, Vol. 3, pp. 26-31, 2004.

[12] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernstein, "**Quantum cellular automata",** *Nanotechnology*, vol. 4, p. 49, 1993.

[13] C. S. Lent, P. D. Tougaw, and W. Porod, **"Quantum cellular automata: the physics of computing with arrays of quantum dot molecules",** in *Proceedings of Workshop on Physics and Computation, 1994. PhysComp'94.* , 1994, pp. 5-13.

[14] A. Safavi, M. Mosleh, "**An Overview of Full Adders in QCA Technology",** *International Journal of Computer Science & Network Solutions*, Vol. 1, pp. 12-35, 2013.

[15] I. Amlani, A. O. Orlov, G. Toth, G. H. Bernstein, C. S. Lent, and G. L. Snider, "**Digital logic gate using quantum-dot cellular automata",** *science*, Vol. 284, pp. 289-291, 1999.

[16] K. Hennessy, C. S. Lent, "**Clocking of molecular quantum-dot cellular automata",** *Journal of*

Vacuum Science & Technology, Vol. 19, pp. 1752-1755, 2001.

[17] V. Mardiris, C. Mizas, L. Fragidis, and V. Chatzis, **"Design and simulation of a QCA 2 to 1 multiplexer",** in *12th WSEAS International Conference on COMPUTERS, Heraklion, Greece*, 2008, pp. 572-576.

[18] T. Teodósio, L. Sousa, **"QCA-LG: A tool for the automatic layout generation of QCA combinational circuits",** in *Norchip*, 2007, pp. 1-5.

[19] K. Kim, K. Wu, and R. Karri, "**The robust QCA adder designs using composable QCA building blocks",** *IEEE transactions on computer-aided design of integrated circuits and systems*, Vol. 1, pp. 176-183, 2007.

[20] S. Hashemi, M. R. Azghadi, and A. Zakerolhosseini, **"A novel QCA multiplexer design",** in *International Symposium on Telecommunications, IST 2008.*, 2008, pp. 692-696.

[21] A. Roohi, H. Khademolhosseini, S. Sayedsalehi, and K. Navi, "**A novel architecture for quantum-dot cellular automata multiplexer",** *International Journal of Computer Science Issues*, Vol. 8, 2011.

[22] B. Sen, M. Dutta, D. K. Singh, D. Saran, and B. K. Sikdar, **"QCA multiplexer based design of reversible ALU",** in *IEEE International Conference on Circuits and Systems (ICCAS)*, 2012, pp. 168-173.

[23] R. Sabbaghi-Nadooshan, M. Kianpour, "**A novel QCA implementation of MUX-based universal shift register",** *Journal of Computational Electronics*, Vol. 13, pp. 198-210, 2014.