

# A novel QCA implementation of MUX-based universal shift register

Reza Sabbaghi-Nadooshan · Moein Kianpour

Published online: 5 September 2013  
© Springer Science+Business Media New York 2013

**Abstract** QCA (Quantum-dot Cellular Automata) is an alternative technology for CMOS that has a low power consumption and high density. QCA extensively supports the new plans in the field of nanotechnology. Applications of QCA technology as an alternative method for CMOS technology in nano-scale have a hopeful future. This paper presents the successful design, implementation and simulation of 2 to 1, 4 to 1 and 8 to 1 multiplexer with the minimum area as compared to the previous models in QCA technology. In this paper, by means of 4 to 1 multiplexers including D-Flip Flop (D-FF) structure in QCA, we present an 8-bit universal shift register. The structure of the 8-bit universal register is extendable to 16-bit, 32-bit and etc. In this paper, the successful simulation of 2 to 1, 4 to 1 and 8 to 1 multiplexers, including D-FF and finally 8-bit universal register structure in QCADesigner is provided. The multiplexers and D-FF presented in this paper have the minimum complexity, area and delay compared to the previous models. In this paper, the implementation of 8-bit universal shift register, by means of 4 to 1 multiplexers and D-FF are presented in QCA technique which have the minimum complexity and delay. In the proposed design of the 8-bit universal shift register, the faults are likely to occur at 2 to 1 multiplexers and D-FF. In this article, 2 to 1 multiplexers and D-FF are investigated from the cell missing and possible defects. Considering the pipeline being the virtue of QCA, the 8-bit universal shift register has a high speed function. This 8-bit universal shift register may be used in the high speed processors as well as cryptography circuits.

**Keywords** Defect · Majority gate · Module · Multiplexer · Quantum-dot cellular automata (QCA) · Universal shift register

## 1 Introduction

QCA (Quantum-dot Cellular Automata) is one of the most attractive alternatives for CMOS technology. QCA uses the paired order of quantum dots to implement the Boolean logic functions. QCA is physical implementation of a classic cellular automata from mechanic quantum effect. The common digital technologies use the voltage or current ranges for showing the logic values, whilst in QCA technology, the situation of electrons in quantum dots shows the binary values. The advantages of this technology include:

1. High operational speed (Tera Hertz range),
2. Low power consumption (approximately 100),
3. High device density (approximately 10) [1].

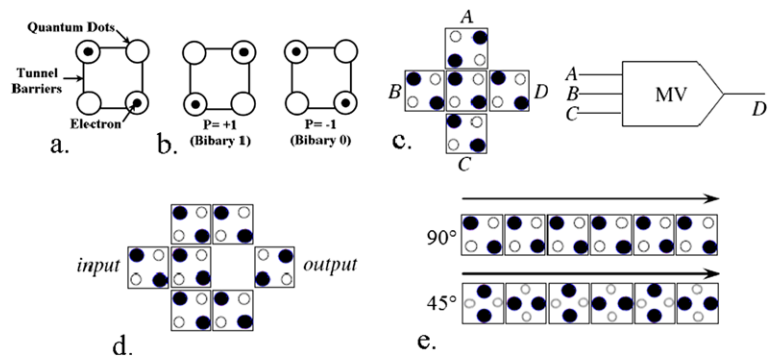
Minimum feature in CMOS has reduced after several decades, however, facing some limitation. This subject caused the rapid development of molecular plans in nano-scale. QCA is a hopeful sample in nanotechnology, suggested by Lent et al. [2] and created in 1997 [3]. According to the considerable features of QCA such as high density, low power consumption, high speed function potential and pipeline being advantage, QCA is changed as an interesting alternative technology for CMOS technology.

In the previous works, several multiplexer plans based on QCA have been presented [4–6]. Mardiris et al. presents a new design model of 2 to 1 multiplexer in QCA [4, 5]. In this paper, the implementation of multiplexer has been presented based on modular design. A modular design and principle simulation has been developed by means of which

---

R. Sabbaghi-Nadooshan (✉) · M. Kianpour  
Dept. of Electrical Engineering, Islamic Azad University, Central  
Tehran Branch, Tehran, Iran  
e-mail: r\_sabbaghi@iauctb.ac.ir

**Fig. 1** Basic QCA logic devices (a) QCA cell; (b) QCA cell polarization; (c) Majority voter (MV); (d) Inverter; (e) Binary wire



$2^n$  to 1 multiplexers may be designed by made blocks. In the design, AND, OR, and delay modular blocks have been used to implement the multiplexer. But due to using a lot of delay blocks and extra interface circuits, the circuit delay and complexity is high.

Roohi et al. have designed a new model of 2 to 1 multiplexer [6]. Then, they implemented a 4 to 1 multiplexer by this 2 to 1 multiplexer. But in this paper larger multiplexers or further applications of multiplexer have not been examined.

In this article, design, implementation and simulation of a basic circuit by QCA technology will be presented. Also, by using this circuit in QCA, implementation and simulation of an 8-bit universal shift register will be provided. Designing method in QCA technology is different from CMOS technology so that in QCA, the quantum dots are used instead of transistor in CMOS technology. In this paper, a new design of 2 to 1 multiplexer with the area, delay and complexity has been provided. Using this 2 to 1 multiplexer as a module, 4 to 1 and 8 to 1 multiplexers are implemented. This 2 to 1 multiplexer module may be used for extension of a  $2^n$  to 1 multiplexer as well. In this article, 2 to 1 multiplexers and D-FF are examined from the cell missing and possible defects. The 8-bit universal shift register presented in this paper is implemented by using eight 4 to 1 multiplexers and eight D-flip flops. This 8-bit universal shift register performs four actions in a circuit simultaneously:

1. Remaining in an unaltered state,
2. Shift to right,
3. Shift to left,
4. Parallel transmission of data to the output.

We are seeking two goals in our designs: to implement the circuits designed in QCA with the minimum complexity, and to optimize and simplify the designs and fittings for reducing delay and increasing the processing speed.

This paper is organized as follows: Sect. 2 reviews the QCA. In Sect. 3, the design and implementation of the multiplexers and D-FF are shown. Section 4 presents 2 to 1 multiplexer and D-FF, analysis defects. In Sect. 5, 8-bit uni-

versal shift register implementation in QCA is presented. In Sect. 6, simulation results are shown. Conclusions are given in the last section.

## 2 QCA review

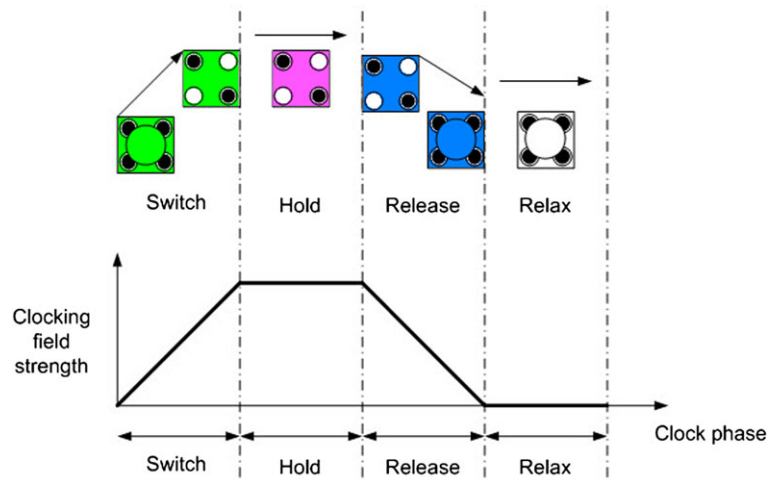
### 2.1 Background

A standard QCA cell is constituted of four quantum dots at four corners of a square cell. In this cell, four quantum dots have been paired together by the tunnel barriers. Two electrons existing in each cell may tunnel between the quantum dots inside the cell. The high intercellular potential barriers ensure that no electron tunnels between QCA cells. Figure 1(a) shows a standard QCA cell with four quantum dots located at its corners. The efficacy of coulombic interaction have run two electrons each to the cell diameters. The polarization of both stable states in cell diameters provides binary logic 0 and binary logic 1. Figure 1(b) exhibits the state of electrons placed in cell diameters and 0 and 1 binary formation. If two cells are located next to each other, the coulombic interaction between the electrons causes the cells to have equal polarization and the same value of its left side cell. The most fundamental logic gate in QCA is majority gate. The logic equation for a majority gate is as follows [7, 8]:

$$F(A, B, C) = AB + AC + BC \tag{1}$$

Majority gate is implemented by five QCA cells as Fig. 1(c). By holding the polarization of one of majority gate inputs in the constant values 1 or 0, OR and AND gates are obtained respectively. The other high consuming gate in QCA is inverter gate; Fig. 1(d) shows the common sample of inverter gate. Then, upon having AND, OR and NOT gates at our disposal, the most complex logic circuits may be implemented. In Fig. 1(e) some of QCA cells have been located beside each other and formed a wire in QCA. In QCA technique, the wires are 45 and 90 degrees. Both wire types are used in the cross over and arrays intensively [9, 10].

**Fig. 2** Four phases of the QCA clocking



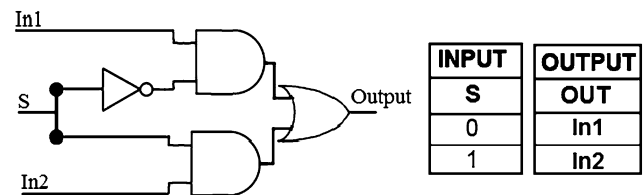
## 2.2 QCA clocking

In VLSI technology, the timing is controlled by a reference signal (means one clock) and is required essentially for sequential circuits. QCA clocking has been performed through timing in four distinct phases and required for both combinational and sequential circuits. Clocking not only controls the data current but provides the actual power in QCA circuits. The signal energy lost by the medium is recovered by a new clocking. In QCA, clocking signals have been generated by an electrical field so that to control tunnel barriers in the quantum dots inside a QCA cell. This electrical field may be provided by CMOS wires or Carbon Nano Tubes (CNT) which are embedded under QCA circuit. When the barriers are down, the cells are depolarized and when the barriers are up, the cells are not allowed to change the state [3, 11].

Clocking in QCA has been applied by controlling the potential barriers between neighbor quantum dots. The clocking used in QCA includes four phases: hold, release, relax and switch, as shown in Fig. 2. Each one of these phases is 90 degrees backward from the others. During switch phase, the potential barriers ascend through the quantum dots gradually and QCA cell is placed in one of two existing polarizations states, considering its neighbor cell. During hold phase, the barriers between quantum dots remain on their highest extent that prevents the electrons tunneling and fixes the polarization state of QCA cell. During release and relax phases, the barriers between quantum dots are reduced to their lowest extent in relax state. Then electrons can move in their cell [12, 13].

## 3 Proposed multiplexer in QCA

In general, a multiplexer allows selecting one input signal among several inputs, and sending it to the output. This capability has changed the multiplexer to a highly important



**Fig. 3** Schematics of 2 to 1 multiplexer with truth table

device in each system such as a communication line or shift register instead of having several devices for each signal. The other advantage of multiplexer is its programmable and controllable lines by means of selector BUS, that allows the system to select one input from among several inputs. In this section, we will present the design and implementation of a 2 to 1 multiplexer in QCA, that its complexity, area and delay have been optimized saliently more than previous works. To continue this section, 2 to 1 multiplexer is used as a module and so 4 to 1 and 8 to 1 multiplexers are implemented. This method may be extended to  $2^n$  to 1 multiplexer. By using 4 to 1 multiplexer presented in this section and D-FF presented in the rest of this section, an 8-bit universal shift register in QCA will be implemented in the next section.

### 3.1 Implementation of proposed 2 to 1 multiplexer in QCA

In the logic implementation of a 2 to 1 multiplexer, two AND gates in the input and one OR gate in the output are used. The selector line is joint in two rails of AND gate and two other lines of AND gates are connected to In1 and In2. OR gate in output sends the result of selecting the input line by the selector to the output. Figure 3 depicts the logic implementation of a 2 to 1 multiplexer and its truth table.

In QCA technology, if the logic implementation has more than three inputs, it may be implemented by using multiple cascading of three-input majority gates. The implementation of 2 to 1 multiplexer using the majority gate in QCA

is observed in Fig. 4. Whenever selector “S” is in “0” or “1” mode, In1 and In2 inputs are selected respectively.

The implementation of 2 to 1 multiplexer presented in this article by QCA technology consisted of 26 cells that have occupied the area equal to  $0.02 \mu\text{m}^2$ . Figure 5 shows the implementation of 2 to 1 multiplexer in QCA technology presented in this paper. Considering the implementation of 2 to 1 multiplexer in QCA, it is observed that its delay is 0.5 clock cycle (two phases), which has been optimized from the previous works.

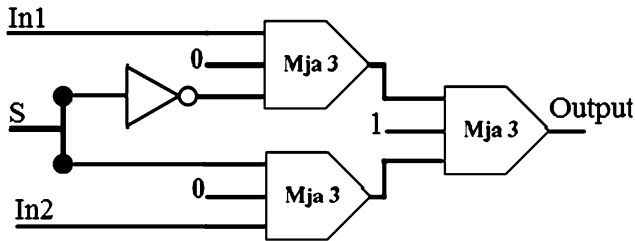


Fig. 4 Logical 2 to 1 multiplexer implemented in QCA

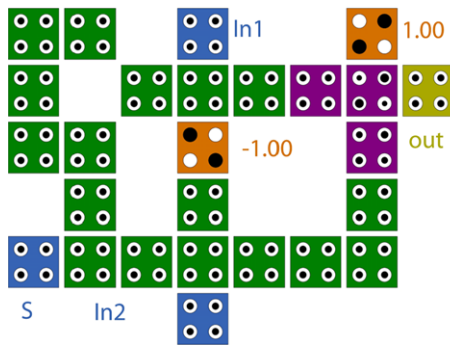
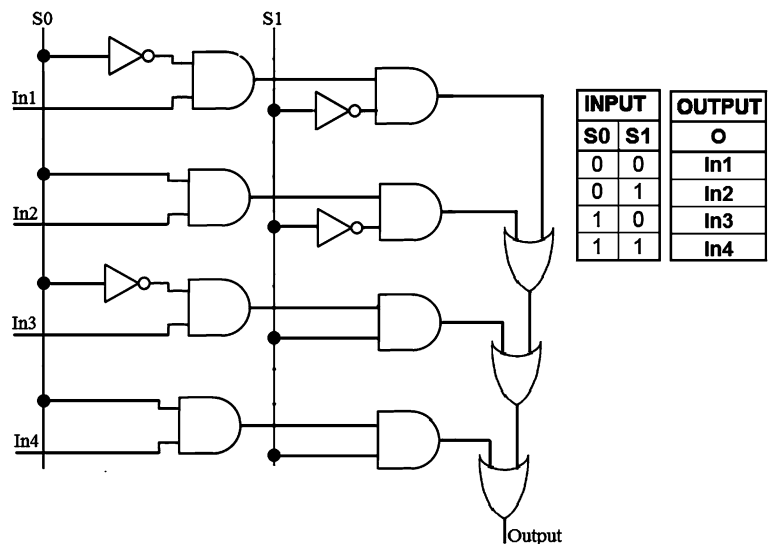


Fig. 5 Proposed 2 to 1 multiplexer implemented in QCA

Fig. 6 Schematics of 4 to 1 multiplexer with truth table



### 3.2 Implementation of proposed 4 to 1 multiplexer in QCA

A customary 4 to 1 multiplexer uses four AND gates at the first stage. The input of selector “S0”, and the inputs of In1 to In4 are connected to this stage by the required NOT gates. In the second stage, seven gates are used. Four gates are used as AND gate including selector “S1” and the outputs of the first stage. Three other gates are used as OR gate for establishing the output. Figure 6 shows the logic implementation of a customary 4 to 1 multiplexer. The logic function of circuit is so that if lines S0 and S1 respectively equal “00”, “10”, “01” and “11”; each one of inputs In1, In2, In3 and In4 runs toward the output.

The 4 to 1 multiplexer exhibited in Fig. 6 may be implemented by applying modules of 2 to 1 multiplexer. Figure 7 shows the implementation of 4 to 1 multiplexer by modules of 2 to 1 multiplexer. As observed, by using three modules of 2 to 1 multiplexer, 4 to 1 multiplexer may be implemented that has a simpler and more efficient structure.

For implementation the block diagram of Fig. 7 in QCA, we used Signal Distribution Network (SDN) method to avoid the coplanar problem of crossing wires [17, 18]. In the SDN, the implementation of 4 to 1 multiplexer divided into two parts: signal distribution network and combinational logic gates. The separation of signal distribution network from combinational logic gates improves the system performance. Figure 8 shows the implementation of 4 to 1 multiplexer in QCA. The implementation of 4 to 1 multiplexer in QCA with SDN method uses four standard clock signals (shown in Fig. 2). By compliance stringent delay in design of the signal network will not be any problem in passing the signal correctly. The 4 to 1 multiplexer presented in QCA consists of 271 cells and occupies an area equal to  $0.39 \mu\text{m}^2$ . The signal distribution network consists of 161 cells and occupies an area equal to  $0.24 \mu\text{m}^2$ , and the com-

binational logic consists of 110 cells and occupies an area equal to  $0.13 \mu\text{m}^2$ .

As specified in the figure, In1 to In4 imply the multiplexer inputs and S0 and S1 show the selector lines of 4 to 1 multiplexer. According to selection lines S0 and S1, one of four inputs are selected and transmitted to the output. In this modular implementation technique of multiplexer, the number of 2 to 1 multiplexers required for design of each multiplexer of higher grades is  $n - 1$ ; n implies the number of inputs of multiplexer. Here, to implement a 4 to 1 multiplexer, because of having four inputs, three modules of 2 to 1 multiplexer are required. As it is observed in Fig. 8, the delay of 4 to 1 multiplexer in signal distribution network equals 3.5 clock cycles (14 phases) and in combinational logic equals 1.25 clock cycles (5 phases) which has been improved from the former works.

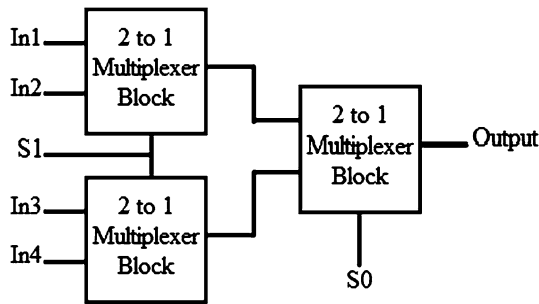
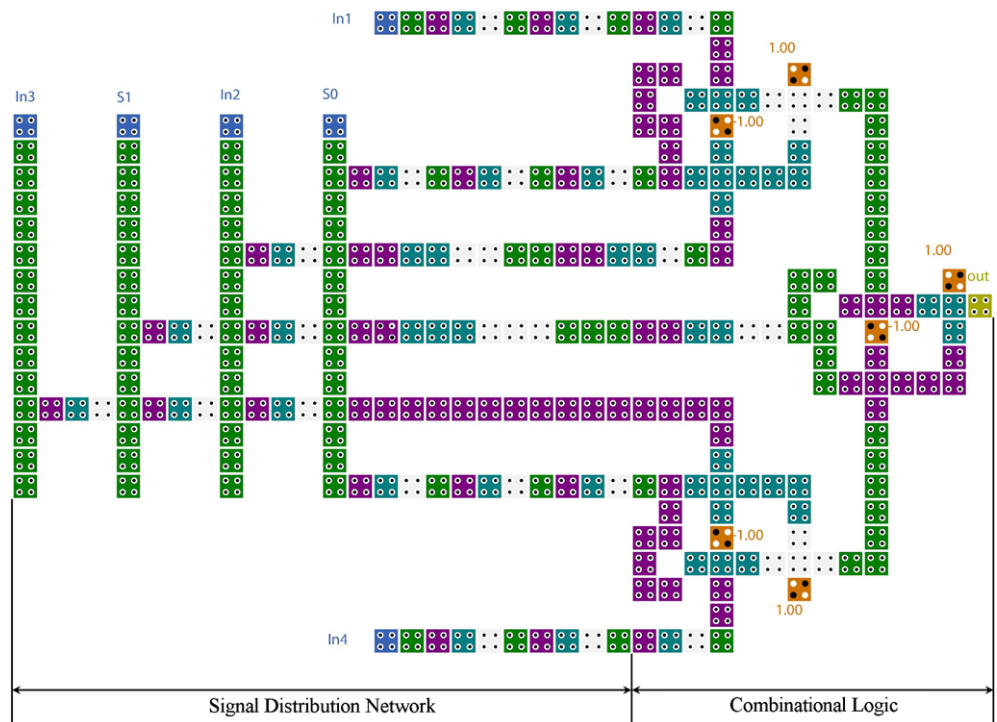


Fig. 7 Schematics of 4 to 1 multiplexer by 2 to 1 multiplexer module

Fig. 8 Proposed 4 to 1 multiplexer implemented in QCA with Signal Distribution Network (SDN) method



### 3.3 Implementation of proposed 8 to 1 multiplexer in QCA

In this section, the implementation of an 8 to 1 multiplexer by means of modules of 2 to 1 multiplexer has been proposed. As mentioned above, to implement an 8 to 1 multiplexer, because of including eight inputs, seven modules of 2 to 1 multiplexer are required. Accordingly, an 8 to 1 multiplexer is presented by seven modules of 2 to 1 multiplexer presented in this paper and suitable conjunctions between them. Figure 9 exhibits the block diagram of an 8 to 1 multiplexer implementation by modules of 2 to 1 multiplexer in QCA. For Fig. 9, we used Signal Distribution Network (SDN) method to avoid the coplanar problem of crossing wires [17, 18]. Figure 10 shows the final implementation of 8 to 1 multiplexer in QCA with SDN. Here, In1 to In8 imply the multiplexer inputs and S0, S1 and S2 the selector lines of 8 to 1 multiplexer. In this paper, 8 to 1 multiplexer consists of 1311 QCA cells and occupies an area equal to  $1.85 \mu\text{m}^2$ . The signal distribution network consists of 1016 cells and occupies an area equal to  $1.45 \mu\text{m}^2$ , and the combinational logic consists of 295 cells and occupies an area equal to  $0.38 \mu\text{m}^2$ . According to Fig. 10, the delay of 8 to 1 multiplexer in signal distribution network is 8.5 clock cycles (34 phases), and in combinational logic is 2 clock cycles (8 phases). So, it may be extended by use of 2 to 1 multiplexer module and considering the number of inputs of multiplexer.

### 3.4 Implementation of proposed D-flip flop

In this section, an overview of QCA memory architectures is presented. These architectures are based on the memory-in-

motion paradigm [14] by which the value of the stored data is moved through different cells in a closed loop spanning at least four clocking zones. The basic D-flip flop of this archi-

tecture is shown in Fig. 11(a). The data bit is stored in a loop, until the CLK control signal is low. When CLK is raised high, then the input bit is stored in the loop. The loop must be implemented using all zones of the four phase adiabatic switching technique for the clock, thus allowing the motion of the stored bit. In a QCA pipelined system, this allows the stored value to be preserved in the loop until the CLK is low. The right AND gate is called an enable gate and operates independently from the rest of the circuit. Figure 11(b) shows the layout of the D-flip flop in QCADesigner and Table 1 shows the D-flip flop operation. The proposed D-flip flop in QCA consists of 52 cells and layout of this design has an area of  $0.06 \mu\text{m}^2$ .

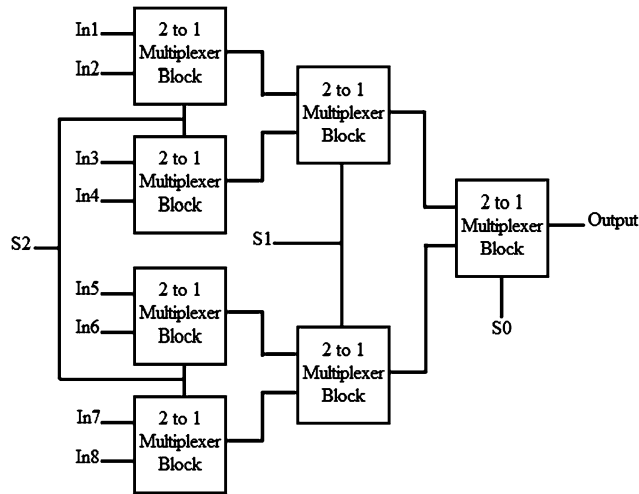


Fig. 9 Schematics of 8 to 1 multiplexer by 2 to 1 multiplexer module

#### 4 Defect analysis of 2 to 1 multiplexers and D-FF in QCA

Recently considerable progress has been made in the molecular manufacturing of QCA, which in this method each QCA cell is a molecule. In manufacturing, defects can occur in

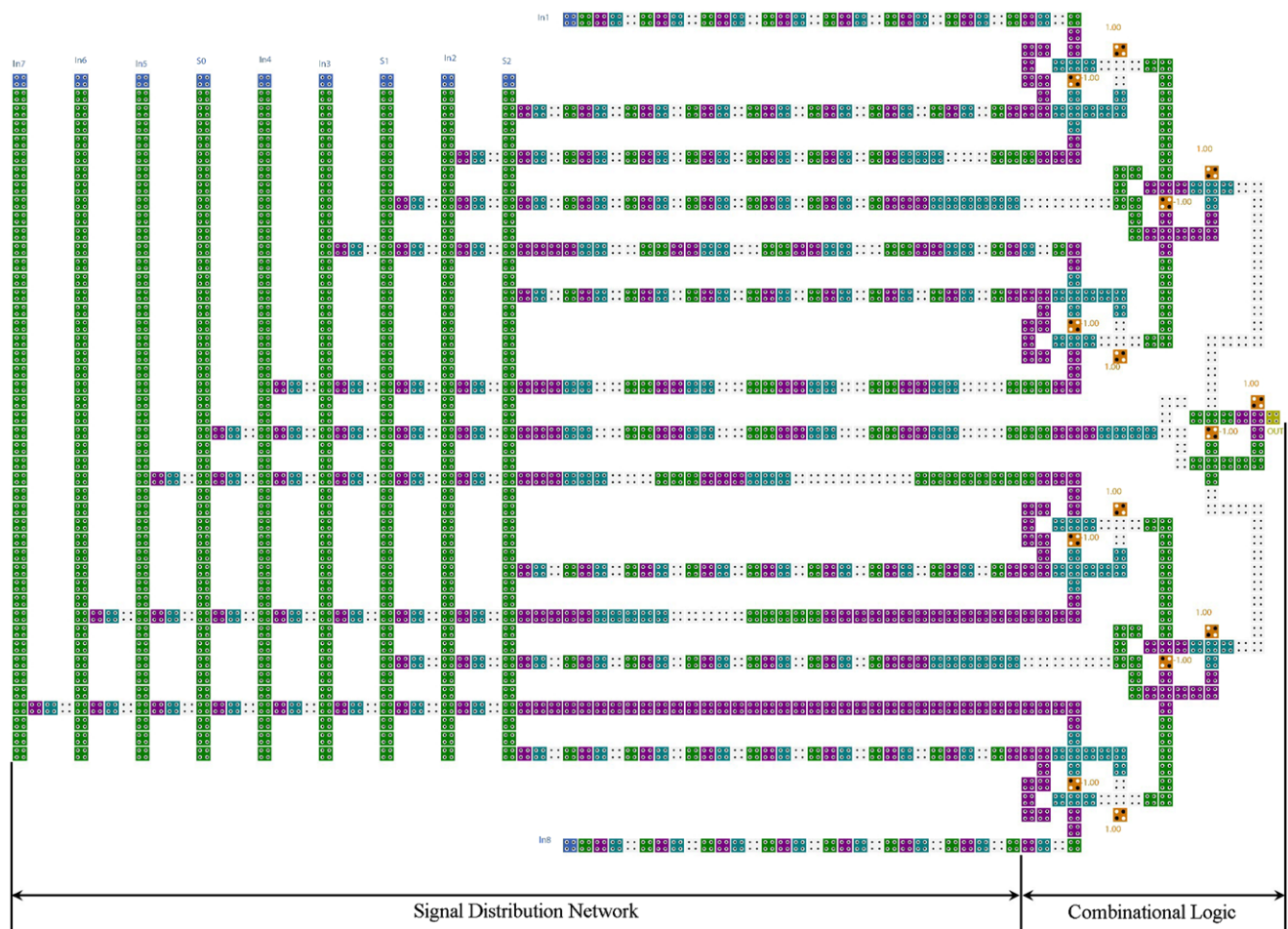
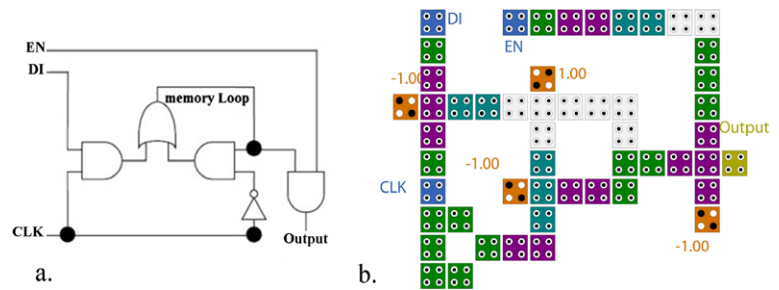
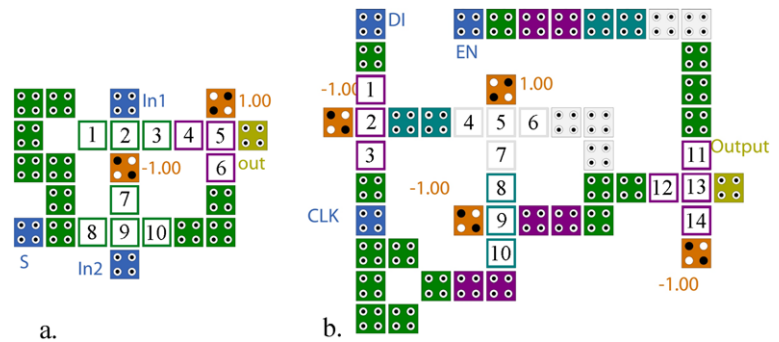


Fig. 10 Proposed 8 to 1 multiplexer implemented in QCA with Signal Distribution Network (SDN) method

**Fig. 11** (a) Basic D-flip flop [14]; (b) proposed D-flip flop in QCA



**Fig. 12** QCA layout of possibility missing cell of (a) 2 to 1 multiplexer [M1...M9]; (b) D-type flip flop [FF1...FF12]



**Table 1** D-flip flop operation

CLK	EN	DI	Memory Loop	Output
0	0	X	Unchanged	0
0	1	X	Unchanged	Stored Value
1	X	0	0 (next cycle)	NA
1	X	1	1 (next cycle)	NA

NA—not applicable

both phases of chemical synthesis and deposition phase. Reports in the literature has shown that the defects are very likely to occur in the deposition phase than chemical synthesis phase, which can result in created, producing cells completely deficient in the substrate. In this section, 8-bit universal shift register elements presented in this paper is investigated from the cell missing and possible defects. A comprehensive test at QCADesigner has been investigated using the eight possible input vectors and possible missing of cells in the majority gates. In other words, in each step one cell is omitted, then this obtained circuit is simulated by QCADesigner. In the proposed design of the 8-bit universal shift register, the faults are likely to occur at two places:

- 2 to 1 Multiplexer
- D-flip flop

Figure 12 shows 8-bit universal shift register components cells that have the possibility of missing [15].

The results of 2 to 1 multiplexer for the defects are shown in Table 2. All possible missing cell defects in MV have been simulated. From the simulation it can be observed that defect

result in the same fault pattern. The exhaustive testing of the 2 to 1 multiplexer with 8 input (by In1, In2 and S) patterns and all possible single missing cell defects is done using the QCADesigner simulator. The exhaustive testing for 2 to 1 multiplexer generated 8 unique fault patterns.

For D-FF, the results are shown in Table 3. For all the possible missing cell defects, fault patterns are generated at the outputs. The exhaustive testing of the D-FF with 8 input (by DI, CLK and EN) patterns and all possible single missing cell defects is done using the QCADesigner simulator. The exhaustive testing for D-FF generated 12 unique fault patterns.

Finally, according to the results can be a determined testable implement for the 8-bit universal shift register components to be considered and used it to design.

### 5 Proposed 8-bit universal shift register in QCA

In this section, universal shift register, one of the most applicable and essential electronic circuits is presented by QCA technology. If the outputs of flip flop’s shift register are available, then the serial input data may be outputted in parallel from flip flop output by shifting. If the shift register is added parallel to load ability, the parallel input data may be outputted in series form by shifting. Some shift registers have the required terminals for parallel transmission. These circuits may be displaceable to the left and right. The register that is only displaceable to one direction is referred to as unidirectional shift register and in the event of displacing to two directions is referred to as bi-directional shift register. If

**Table 2** Simulation results of 2 to 1 multiplexer for the defects

Input Vector	Faulty Output of missing cell ((F): Faulty Output, (T): True Output)										
	Fault Free Output	M1 Fault	M2 Fault	M3 Fault	M4 Fault	M5 Fault	M6 Fault	M7 True	M8 True	M9 Fault	M10 Fault
$A_0 = 000$	$O_0 = 0$	0	1	1	1	0	1	0	0	1	1
$A_1 = 001$	$O_1 = 0$	0	1	1	1	0	1	0	0	0	1
$A_2 = 010$	$O_2 = 1$	0	1	1	1	0	1	1	1	1	1
$A_3 = 011$	$O_3 = 1$	1	1	1	1	0	1	1	1	1	1
$A_4 = 100$	$O_4 = 0$	0	1	1	1	0	1	0	0	1	1
$A_5 = 101$	$O_5 = 1$	1	1	1	1	0	1	1	1	0	1
$A_6 = 110$	$O_6 = 0$	0	1	1	1	0	1	0	0	1	1
$A_7 = 111$	$O_7 = 1$	1	1	1	1	0	1	1	1	0	1

**Table 3** Simulation results of D-FF cell for the defects

Input Vector	Fault Free Output	FF1	FF2	FF3	FF4	FF5	FF6	FF7	FF8	FF9	FF10	FF11	FF12	FF13	FF14
		Fault	Fault	Fault	Fault	Fault	True	Fault	Fault	Fault	Fault	Fault	True	Fault	Fault
$A_0 = 000$	$O_0 = 0$	0	0	0	0	0	0	0	0	0	0	1	0	1	1
$A_1 = 001$	$O_1 = 0$	0	0	0	0	0	0	0	0	0	0	0	0	1	0
$A_2 = 010$	$O_2 = 1$	0	0	1	1	0	1	1	1	1	0	1	1	1	1
$A_3 = 011$	$O_3 = 1$	0	0	1	1	0	1	1	1	1	0	1	1	1	1
$A_4 = 100$	$O_4 = 0$	0	0	0	0	0	0	0	0	0	0	1	0	1	1
$A_5 = 101$	$O_5 = 0$	0	0	0	0	0	0	0	0	0	0	1	0	1	1
$A_6 = 110$	$O_6 = 0$	0	0	0	1	0	0	1	1	1	0	0	0	1	0
$A_7 = 111$	$O_7 = 1$	0	0	0	1	0	1	1	1	1	1	1	1	1	1

**Table 4** 8-bit universal shift register operation

S1	S0	Register Operation
0	0	Unchanged
0	1	Shift-Right
1	0	Shift-Left
1	1	Parallel Data Loading

the register can be shifted into two directions and loaded parallel, it is referred to as universal shift register. The block diagram of an 8-bit universal shift register consisted of eight 4 to 1 multiplexers and eight D-flip flop is observed in Fig. 13.

Eight 4 to 1 multiplexers have two common selector inputs S0 and S1. Input In1 is selected in any multiplexer when  $S1S0 = "00"$ . Input In2 is selected by  $S1S0 = "01"$  and two other inputs are selected similarly. The selector inputs control the register operation mode according to Table 4.

If  $S1S0 = "00"$ , the current value of register is applied to D-flip flop inputs. This status creates a path from each flip flop output to its input. If  $S1S0 = "01"$ , the input In2 of multiplexers has a path to D-FF inputs. It causes a shift to right, therein serial input is inputted to In8. When  $S1S0 = "10"$ ,

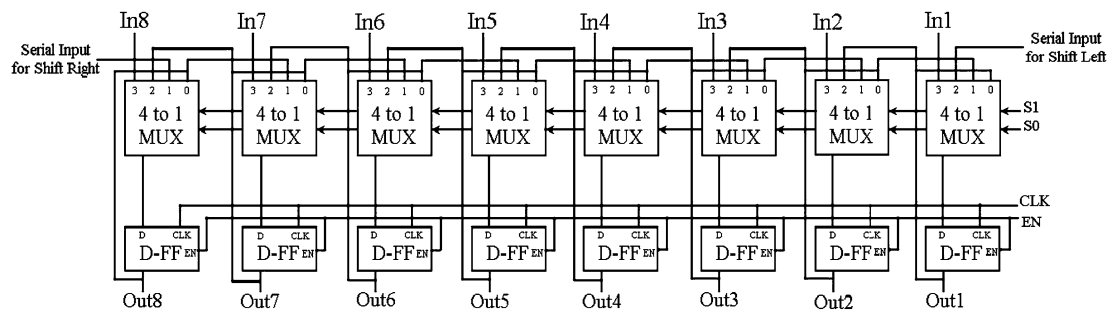
a shift to left is performed and the other serial input is transmitted to In1. Ultimately, when  $S1S0 = "11"$ , the binary data on the parallel input lines inputted to the register simultaneously. Implementation of this 8-bit universal shift register presented in this paper by QCA is observed in Fig. 14. For implementation the 8-bit universal shift register, we used SDN method to avoid the coplanar problem of crossing wires [17, 18]. This implementation of 8-bit universal shift register includes 4053 cells and occupies an area equal to  $6.33 \mu m^2$  and its maximum delay is 8.75 clock cycles.

### 6 Simulation result

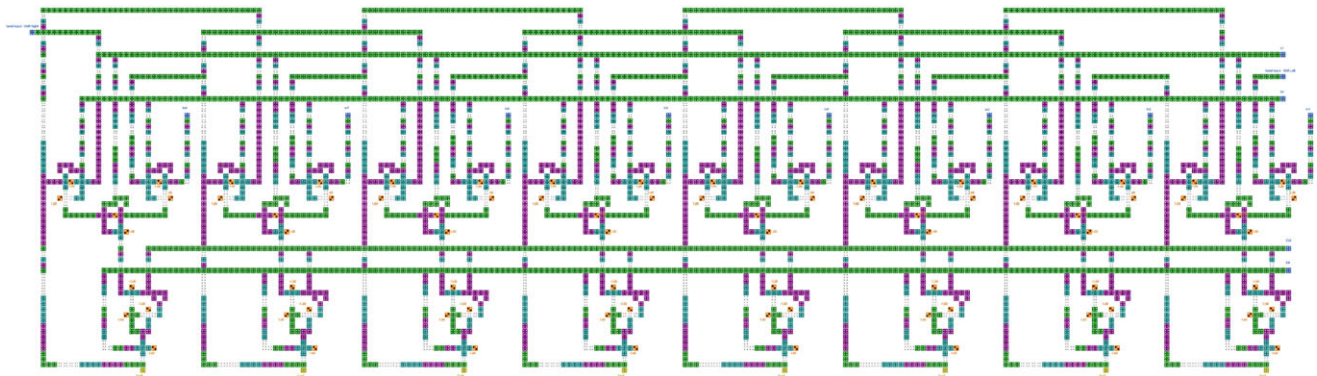
In our design we used QCADesigner Ver. 2.0.3 in the bistable approximation. Table 5 presents a brief description for each parameter used for a simulation engine [16].

Figure 15 depicts the simulation of D-flip flop in QCADesigner. Results reach to output after 1.5 clock cycles. When EN is "1", output is enabled and when EN is "0", output is "0". When CLK is "1", write state is enable and data value save in memory loop and when CLK is "0", read state is enable and saved bit is placed on output.





**Fig. 13** Block diagram of proposed 8-bit universal shift register



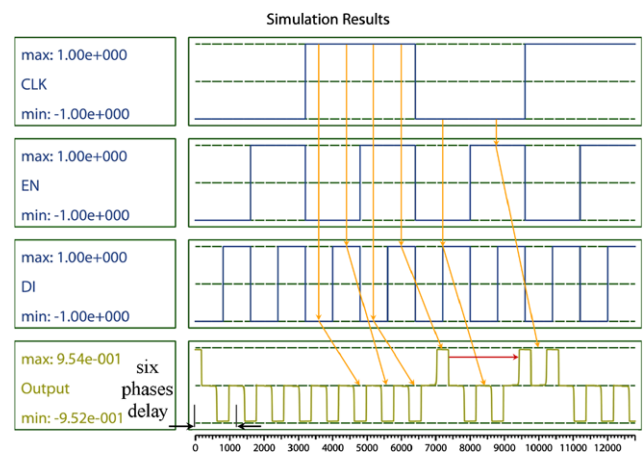
**Fig. 14** Proposed 8-bit universal shift register implemented in QCA with Signal Distribution Network (SDN) method

**Table 5** Parameters Model In QCADesigner simulator

Parameter	Value
Cell width	18 nm
Cell height	18 nm
Dot diameter	5 nm
Number of samples	50000–1000000
Convergence tolerance	0.001
Radius of effect	65 nm
Relative permittivity	12.9
Clock high	9.8e–22 J
Clock low	3.8e–23 J
Clock amplitude factor	2
Layer Separation	11.5 nm
Maximum iteration per sample	100

The simulation result of the 2 to 1 multiplexer is presented in Fig. 16. Two waveforms with different frequencies are applied at the inputs In1 and In2, and the multiplexer outputs the signal at In1 when the select signal S is low and it outputs the signal at In2 when select signal S is high.

Figure 17 shows simulation result of proposed combinational logic of 4 to 1 multiplexer. In this simulation, the results have reached output correctly after 1.25 clock cycle delays. Four waveforms with different frequencies are ap-



**Fig. 15** Simulation of D-flip flop

plied at the inputs In1, In2, In3 and In4, and the multiplexer outputs the signal at In1 when the select bus “S0S1” is “00” and it outputs the signal at In2 when select bus “S0S1” is “01”. Also, the multiplexer outputs the signal at In3 when the select bus “S0S1” is “10” and it outputs the signal at In4 when select bus “S0S1” is “11”.

The simulation result of combinational logic of the 8 to 1 multiplexer is presented in Fig. 18. As the simulation result is observed, delay in this implementation is 2 clock cycle. Eight waveforms with different frequencies are applied

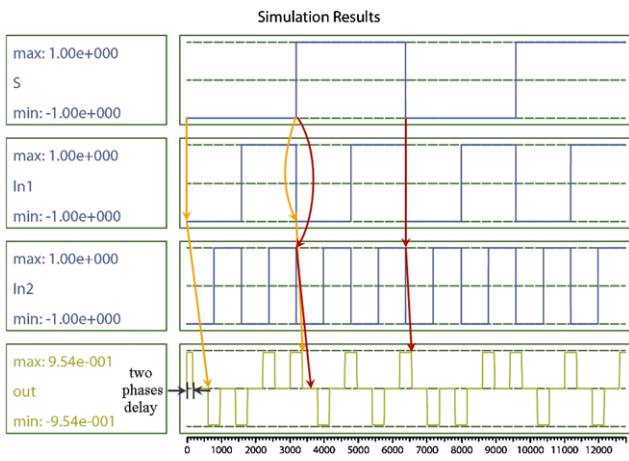


Fig. 16 The simulation results of 2 to 1 multiplexer

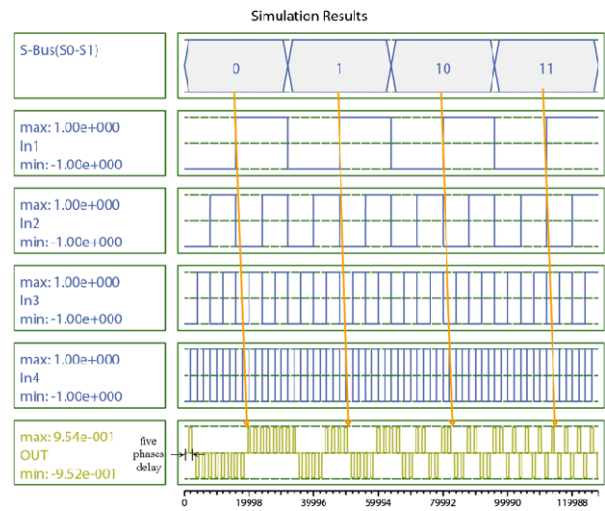


Fig. 17 The simulation results of combinational logic of 4 to 1 multiplexer

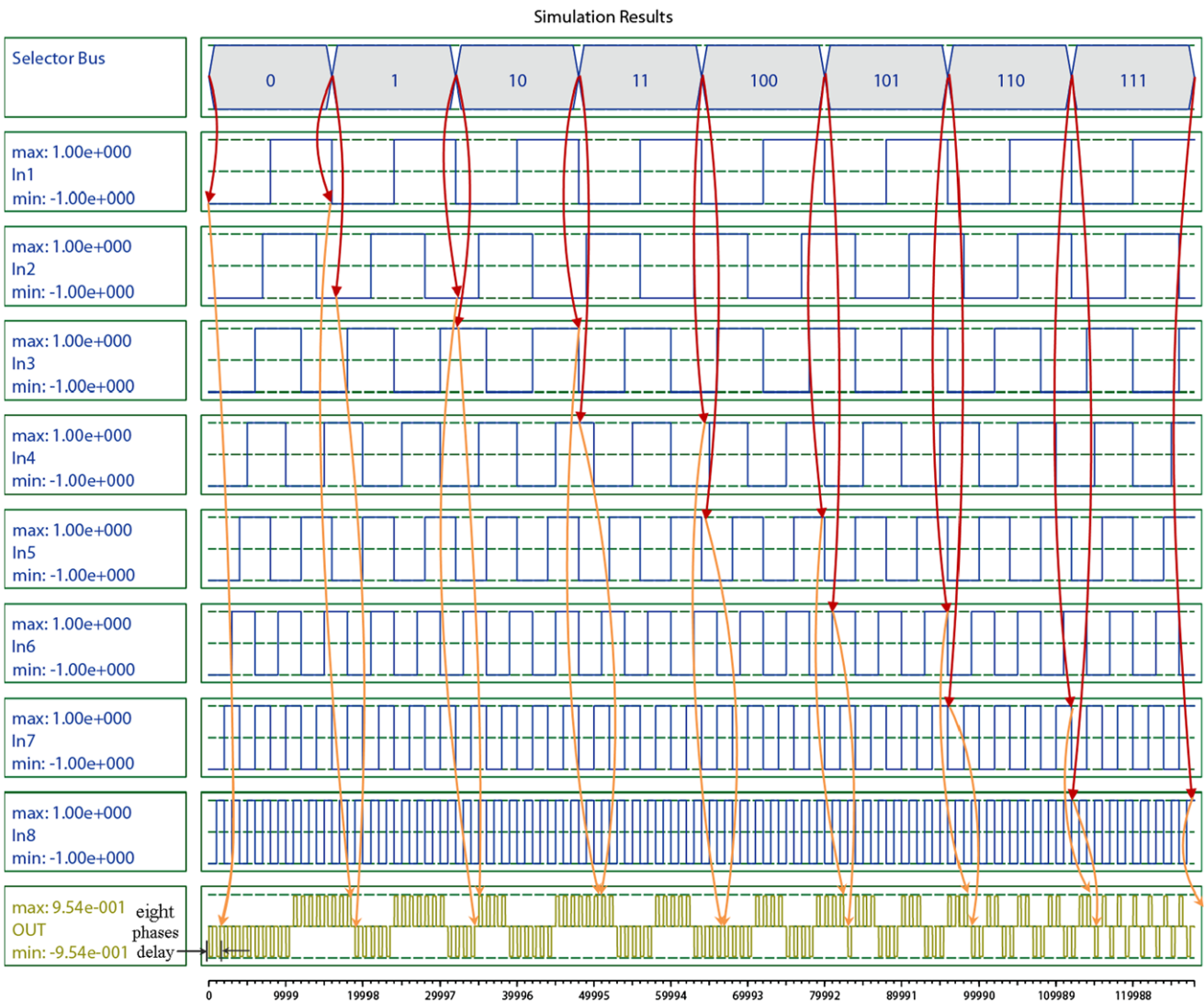
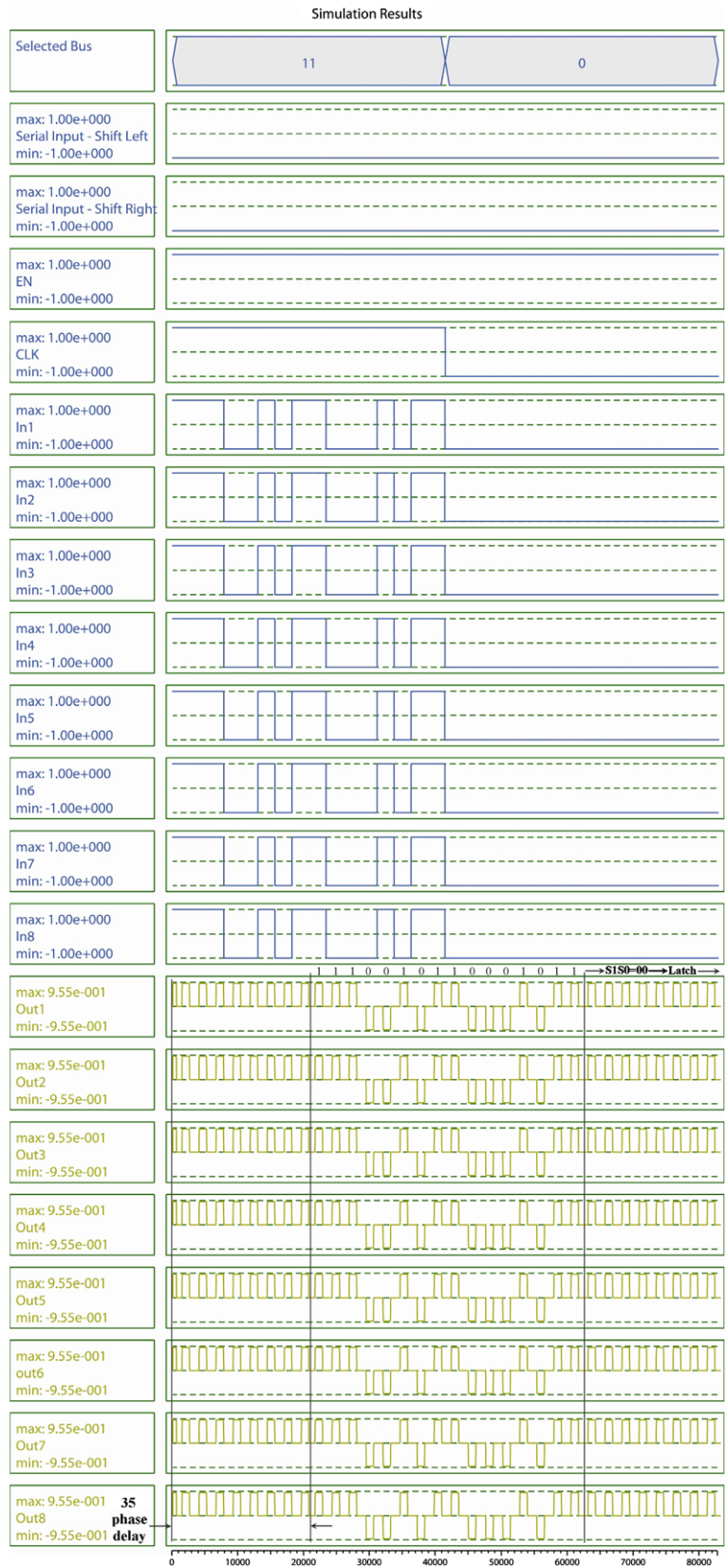
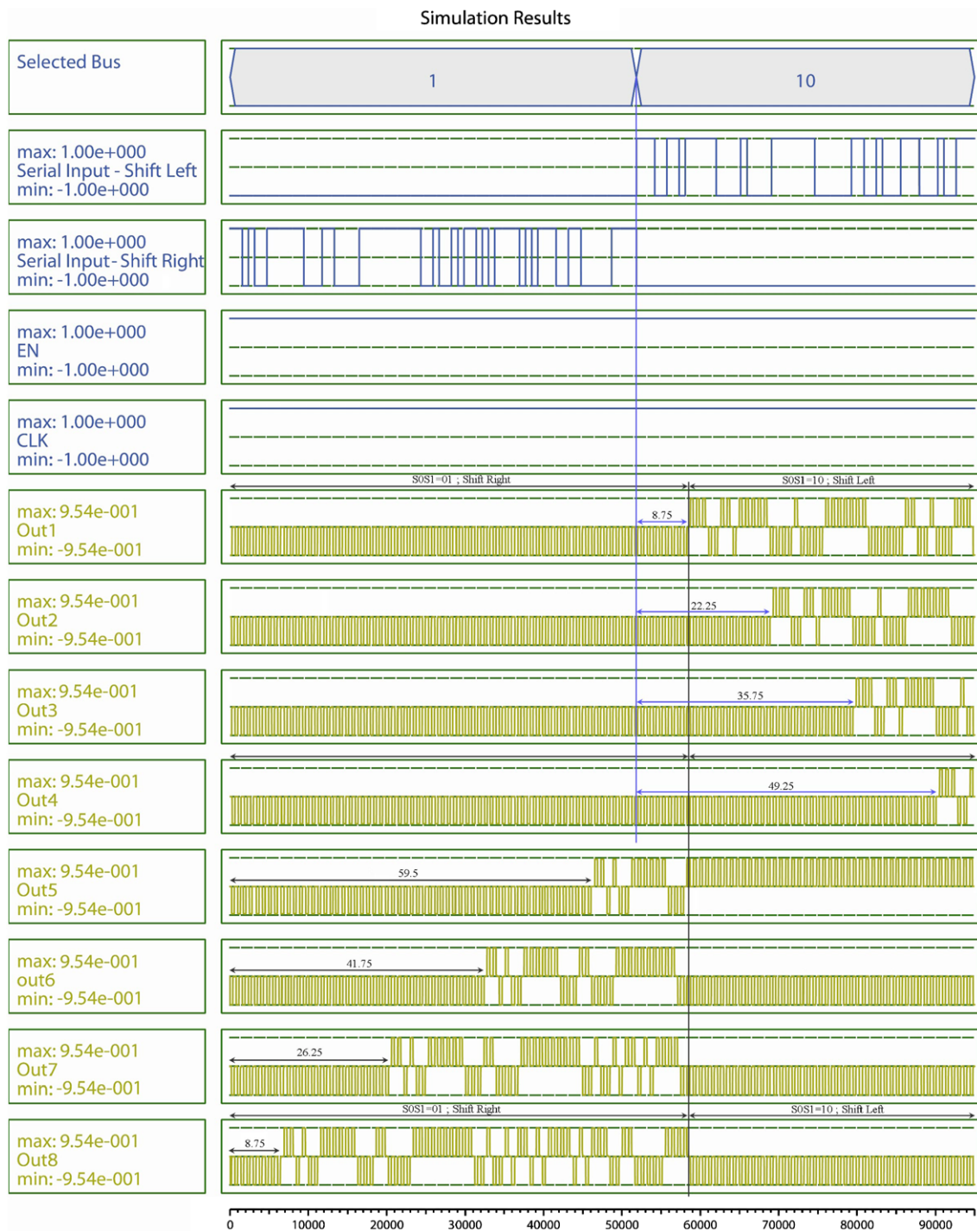


Fig. 18 The simulation results of combinational logic of 8 to 1 multiplexer

**Fig. 19** The simulation of the 8-bit universal shift register, when SR = “11 & 00”





**Fig. 20** The simulation of the 8-bit universal shift register, when SR = “01 & 10”

at the inputs In1, In2, In3, In4, In5, In6, In7 and In8, and the multiplexer outputs the signal at In1 when the select bus “S0S1S2” is “000” and it outputs the signal at In2 when select bus “S0S1S2” is “001”. Also, the multiplexer outputs the signal at In3, In4, In5, In6, In7 and In8 when the select bus “S0S1S2” are “010”, “011”, “100”, “101”, “110” and “111” respectively.

To verify the operation of the 8-bit universal shift register, the bit string is implemented and simulated for different combinations of the EN and CLK inputs of the D-flip flops and selected bus (S0S1) of 4 to 1 multiplexers. For different combinations of the S0 and S1, the operation of the 8-bit universal shift register is verified for the expected output of the 8-bit universal shift register. When S0S1 = “11”, the outputs

**Table 6** Comparison between multiplexers presented in this paper and previous work in [5, 6]

	2 to 1 multiplexer			4 to 1 multiplexer				8 to 1 multiplexer			
	Proposed in this paper	Design proposed by [5]	Design proposed by [6]	Proposed in this paper		Design proposed by [5]	Design proposed by [6]	Proposed in this paper		Design proposed by [5]	Design proposed by [6]
				SDN	CL			SDN	CL		
Area ( $\mu\text{m}^2$ )	0.02	0.07	0.03	0.24	0.13	0.24	unknown	1.45	0.38	0.67	unknown
Complexity (# Cell)	26	56	27	161	110	215	94	1016	296	633	unknown
Delay ( $10^{-12}$ s)	0.5	1	0.75	3.5	1.25	1.75	1.75	8.5	2	3	unknown

**Table 7** Proposed 8-bit universal shift register in QCA

	Area ( $\mu\text{m}^2$ )	Complexity (#cells)	Maximum Delay ( $10^{-12}$ s)
Proposed 8-Bit Universal Shift	6.33	4035	8.75

of the 8-bit universal shift register are the binary data on the parallel input lines to the register and when  $S1S0 = "00"$ , the current value of register are latched to D-flip flop inputs (Fig. 19). Maximum delay in this state of implementation is 8.75 clock cycle.

When  $S0S1 = 01$ , the outputs of the 8-bit universal shift register are shifted from out8 to out1 verifying the shift right operation (Fig. 20) and maximum delay in this state of implementation is 17.5 clock cycle. When  $S0S1 = 10$ , the outputs of the 8-bit universal shift register are shifted from out1 to out8 verifying the shift left operation (Fig. 20) and maximum delay in this state of implementation is 13.5 clock cycle.

Table 6 compares area, complexity and delay of multiplexer presented in this paper with the multiplexer in previous works [5, 6].

Finally Table 7 shows the area, complexity and delay in proposed 8-bit universal shift register.

## 7 Conclusion

We have presented a new extendable design of 2 to 1 multiplexer and 8-bit universal shift register in QCA technology. Upon applying this 2 to 1 multiplexer in design, we have achieved high efficiency. This 2 to 1 multiplexer has been used as a module in design and implementation of 4 to 1 and 8 to 1 multiplexers. In this paper, we have presented a new and optimal design of an 8-bit universal shift register in QCA which have used 4 to 1 multiplexer and D-FF implementation. In this paper, 2 to 1 multiplexers and D-FF are studied from the cell missing and possible defects. This 8-bit universal shift register may be used in processors with high operational speed or for connecting digital systems communication located far from each other. In addition, it may be extended to n-bit universal shift register.

## References

- Huang, J., Lombardi, F.: Design and test of digital circuits by quantum-dot cellular automata. Boston, MA (2008)
- Lent, C.S., Tougaw, P.D., Porod, W., Bernstein, G.H.: Quantum cellular automata. *Nanotechnology* **4**, 49–57 (1993)
- Lent, C.S., Tougaw, P.D.: A device architecture for computing with quantum dots. *Proc. IEEE* **85**, 541–557 (1997)
- Mardiris, V., Mizas, C.H., Fragidis, L., Chatzis, V.: Design and simulation of a QCA 2 to 1 multiplexer. In: 12th WSEAS International Conference on COMPUTERS, Heraklion, Greece, pp. 572–576 (2008)
- Mardiris, V.A., Karafyllidis, I.G.: Design and simulation of modular  $2^n$  to 1 quantum-dot cellular automata (QCA) multiplexers. *Int. J. Circuit Theory Appl.* **38**, 771–785 (2010)
- Roohi, A., Khademolhosseini, H., Sayedsalehi, S., Navi, K.: A novel architecture for quantum-dot cellular automata multiplexer. *Int. J. Comput. Sci. Issues* **8**, 55–60 (2011)
- Amlani, I., Orlov, A.O., Bernstein, G.H., Lent, C.S., Snider, G.L.: Realization of a functional cell for quantum-dot cellular automata. *Science* **227**, 928–930 (1997)
- Amlani, I., Orlov, A.O., Toth, G., Bernstein, G.H., Lent, C.S., Snider, G.L.: Digital logic gate using quantum-dot cellular automata. *Science* **284**, 289–291 (1999)
- Toth, G., Lent, C.S.: Quasiadiabatic switching for metal-island quantum-dot cellular automata. *J. Appl. Phys.* **85**, 2977–2984 (1999)
- Lu, Y., Lent, C.S.: Theoretical study of molecular quantum-dot cellular automata. *J. Comput. Electron.* **5**, 115–118 (2005)
- Cho, H., Swartzlander, E.E.: Adder and multiplier design in quantum-dot cellular automata. *IEEE Trans. Comput.* **58**, 721–727 (2009)
- Bennett, C.H.: Logic reversibility of computation. *IBM J. Res. Dev.* **17**, 525–532 (1973)
- Landauer, R.: Irreversibility and heat generation in the computing process. *IBM J. Res. Dev.* **5**, 183–191 (1961)
- Frost, S.E., Rodrigues, A.F., Janiszewski, A.W., Rausch, R.T., Kogge, P.M.: Memory in motion: a study of storage structures in QCA. In: First Workshop on Non-silicon Computing (2002)
- Fijany, A., Toomarian, B.N.: New design for quantum dots cellular automata to obtain fault tolerant logic gates. *J. Nanopart. Res.* **3**, 27–37 (2001)
- Blair, E.P., Yost, E., Lent, C.S.: Power dissipation in clocking wires for clocked molecular quantum-dot cellular automata. *J. Comput. Electron.* **9**, 49–55 (2009)
- Graunke, C.R., Wheeler, D.I., Tougaw, D., Will, J.D.: Implementation of a crossbar network using quantum-dot cellular automata. *IEEE Trans. Nanotechnol.* **4**, 435–440 (2005)
- Tougaw, D., Khatun, M.: A scalable signal distribution network for quantum-dot cellular automata. *IEEE Trans. Nanotechnol.* **12**, 215–224 (2013)