Int. J. Nano Dimens., 7 (4): 308-320, Autumn 2016

ORIGINAL ARTICLE

Design of low power random number generators for quantum-dot cellular automata

***Abbas Rezaei*[1,*]*; Hamidreza Saharkhiz*[2]

[1]*Electrical Engineering Department, Kermanshah University of Technology, Kermanshah, Iran*
[2]*Electrical and Electronics Engineering Department, Razi University Tagh-E-Bostan, Kermanshah, Iran*

Received 04 April 2016; revised 22 August 2016; accepted 03 September 2016; available online 04 November 2016

Abstract

Quantum-dot cellular automata (QCA) are a promising nanotechnology to implement digital circuits at the nanoscale. Devices based on QCA have the advantages of faster speed, lower power consumption, and greatly reduced sizes. In this paper, we are presented the circuits, which generate random numbers in QCA. Random numbers have many uses in science, art, statistics, cryptography, gaming, gambling, and other fields. The base of these circuits is the linear feedback shift register (LFSR). In this paper, an optimized QCA LFSR is designed, and then different random number generators (RNGs) using XOR and adder are presented. These circuits generate different random numbers in each simulation. The results show that our QCA designs are really fast and optimized in comparison with the previous CMOS and QCA designs.

Keywords: *LFSR; Low power; Nanotechnology; Quantum-dot cellular automata; Random number generator.*

How to cite this article

Rezaei A, Saharkhiz H. Design of low power random number generators for quantum-dot cellular automata. Int. J. Nano Dimens., 2016; 7 (4): 308-320, DOI: 10.7508/ijnd.2016.04.006.

INTRODUCTION

QCA is one of the newest technologies to design the low power and fast switching circuits. According to Moore's law, it can be seen that the capacity of a computer chip grows exponentially with time [1]. Scientists are constantly trying into smaller and smaller MOSFET transistor dimensions, but they have limitations in physical dimension. For this reason, scientists began to study and research in this field, and a new method was devised that allow the continued growth of chip density. In 1993, Lent et al. prepared a physical implementation of an automaton with the quantum-dot cells. This technology was highly regarded in 1997 [2]. As a first result, this research found that the QCA can be used in electronic circuits and devices with lower scale, faster switching, lower power consumption and more importantly the alternative nanoscale technology for CMOS-technology. As shown in Fig.1a QCA cell consists of four Quantum dots, which are placed in a square pattern, and two electrons that can move between the dots inside the cell. Due to mutual electrostatic repulsion force between the electrons, they tend to keep the furthest distance between each other in the square pattern, thus only two stable states can be produced (Logic "0" and Logic "1"), in these states the electrons are in the minimum energy than each other [3-4]. A QCA wire is one of the most important elements we need in QCA circuit to transmit the signal. The wire is made up of a chain of cells. Logic values are passed from cell to cell due to the Coulomb's law. However, the system attempts to settle to a ground state. Any cells along the wire that are anti-polarized to the input would be at a higher energy level, and would soon settle to the correct ground state [5]. Fig.1b shows the standard cell placement in the QCA binary wire. If the QCA cells are placed in a diagonal direction, the electrostatic interaction is inversed, because the quantum-dots of different polarizations are misaligned between the cells [6]. QCA inverters have two different topologies as shown in Fig.2a [6].The most important element in the QCA circuits is the majority Gate. Using the majority Gate, AND and OR logic Gates can be created. This gate is also called a programmable Gate. The performance of this Gate is: M(A,B,C)=A.

* Corresponding Author Email: *unrezaei@yahoo.com*

B+A.C+B.C. With M(A,B,1), we have a logical OR Gate and with M(A,B,0) we have a logical AND Gate. Fig.2b shows the structure of the majority Gate. Fig.2c shows the topologies of logical AND and OR Gates created by the majority Gate.

In QCA, clocking is very important in the both sequential and combinational circuits. In QCA using the clocking, the information is not lost at the end of each wire and transmits correctly. This is similar with the power transmission systems, when the transferring electricity loses some of its power. This phenomenon also occurs in QCA. Thus, to avoid data loss, the data signal need to be recovered from time to time during the transmission process. The energy to restore the signal comes from the QCA clock. The cells gain a power in each stage and amplify a weak input signal to restore the logic level [7, 8]. In QCA clocking scheme as shown in Fig.2d, there are four clock phases in a clock zone, which are known as switch, hold, release, and relax. In QCA Designer simulation tool, there are four clock zones available, which are known as clock0, clock1, clock2 and clock3. Each clock zone is at a 90 degree phase shift from the others and the same phase repeats every clock cycle [9, 10].

In [11] a noninteracting quantum-dot arrays side coupled to a quantum wire has been studied. Also, transport through the quantum wire has been investigated by using a noninteracting Anderson tunneling Hamiltonian. In [12] a layout of four and eight bit universal shift register (USR) has been proposed. Also, initially QCA layouts of D flip-flop with clear and 4 to 1 multiplexer have been designed, which are extended to design 4 and 8-bit parallel in parallel out (PIPO) shift register. In [13], a QCA adder that shows significant fault-tolerance against all types of cell misplacement defects such as cell omission, cell displacement, cell misalignment and extra/additional cell deposition has been introduced. Some new designs of several types of registers including universal shift register up to N-bit and a dynamic register have been proposed and analyzed in [14]. In [15], novel serial decimal adder and adder/subtractor designs used the run-time reconfigurable wiring approach, which results in further significant QCA hardware simplification have been proposed. A new five input minority gate-based CAM cell has been introduced in [16]. Also, QCA designer has been used for simulation of the proposed structure and verifying its operation. In this paper, an optimized QCA LFSR is designed, and then different random number generators (RNGs) using XOR and adder are presented. These circuits generate different random numbers in each simulation.

EXPERIMENTAL

D Flip-Flop

The output function of D flip-flop is always its input, when the clock is high. In Fig.3a, a schematic of D flip-flop is shown. Using Fig.3a, the relationship between the inputs and the output of D flip-flop can be obtained by the following Equation:

$$D.Clock + \overline{Clock}.Q_{Old} = Q_{New} \quad (1)$$

Fig.3b shows the QCA layout of D flip-flop. This layout has 4 clock phases. Simulation results of this QCA layout using QCA Designer software is shown in Fig.3c.

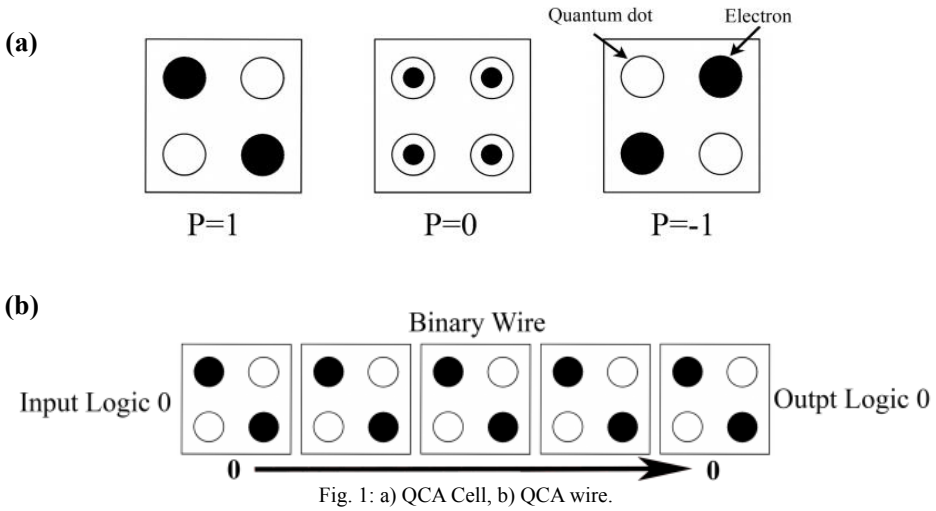


Fig. 1: a) QCA Cell, b) QCA wire.

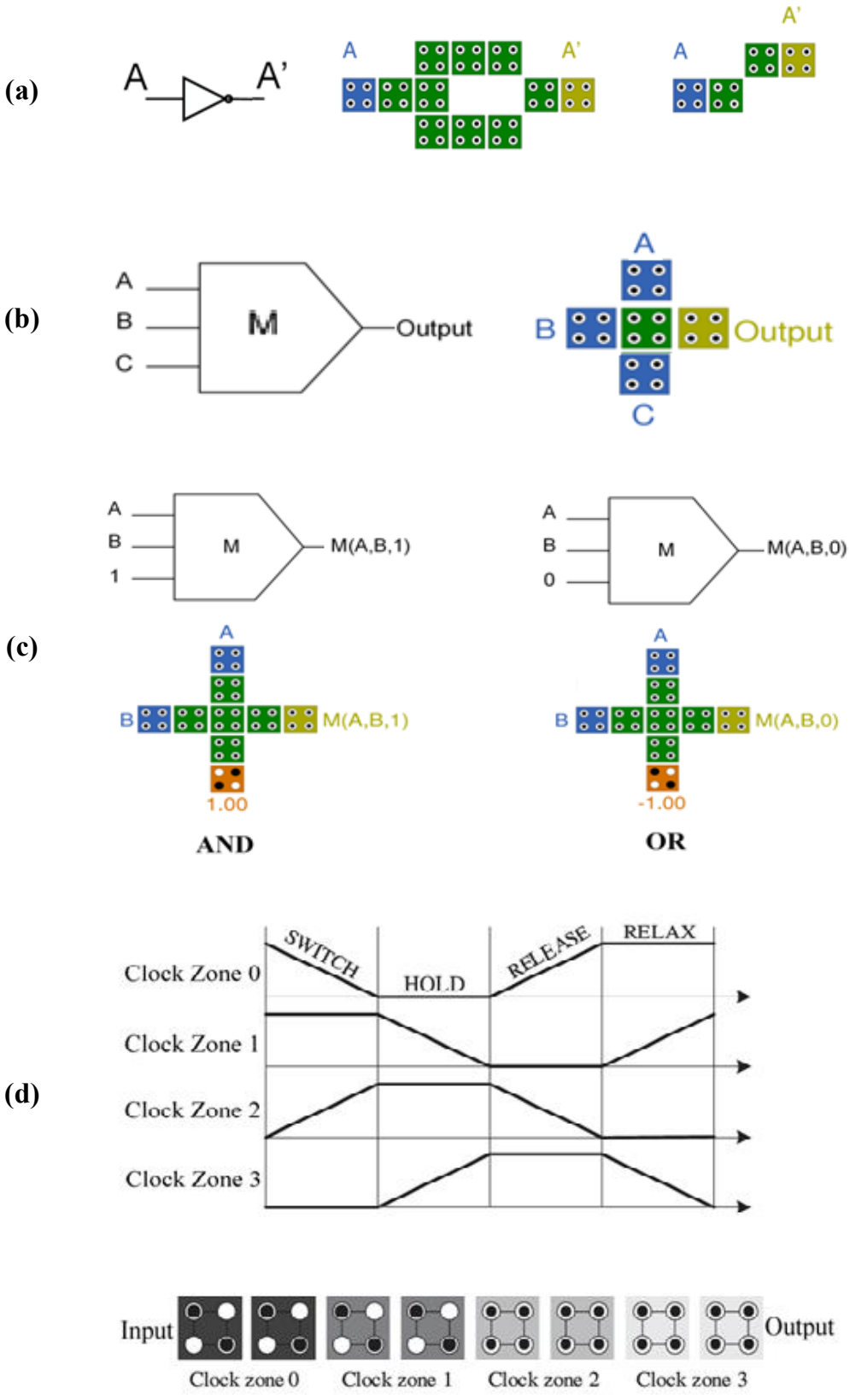


Fig. 2: a) QCA inverter. b) Majority Gate. c) AND and OR Gates. d) QCA Clocking.

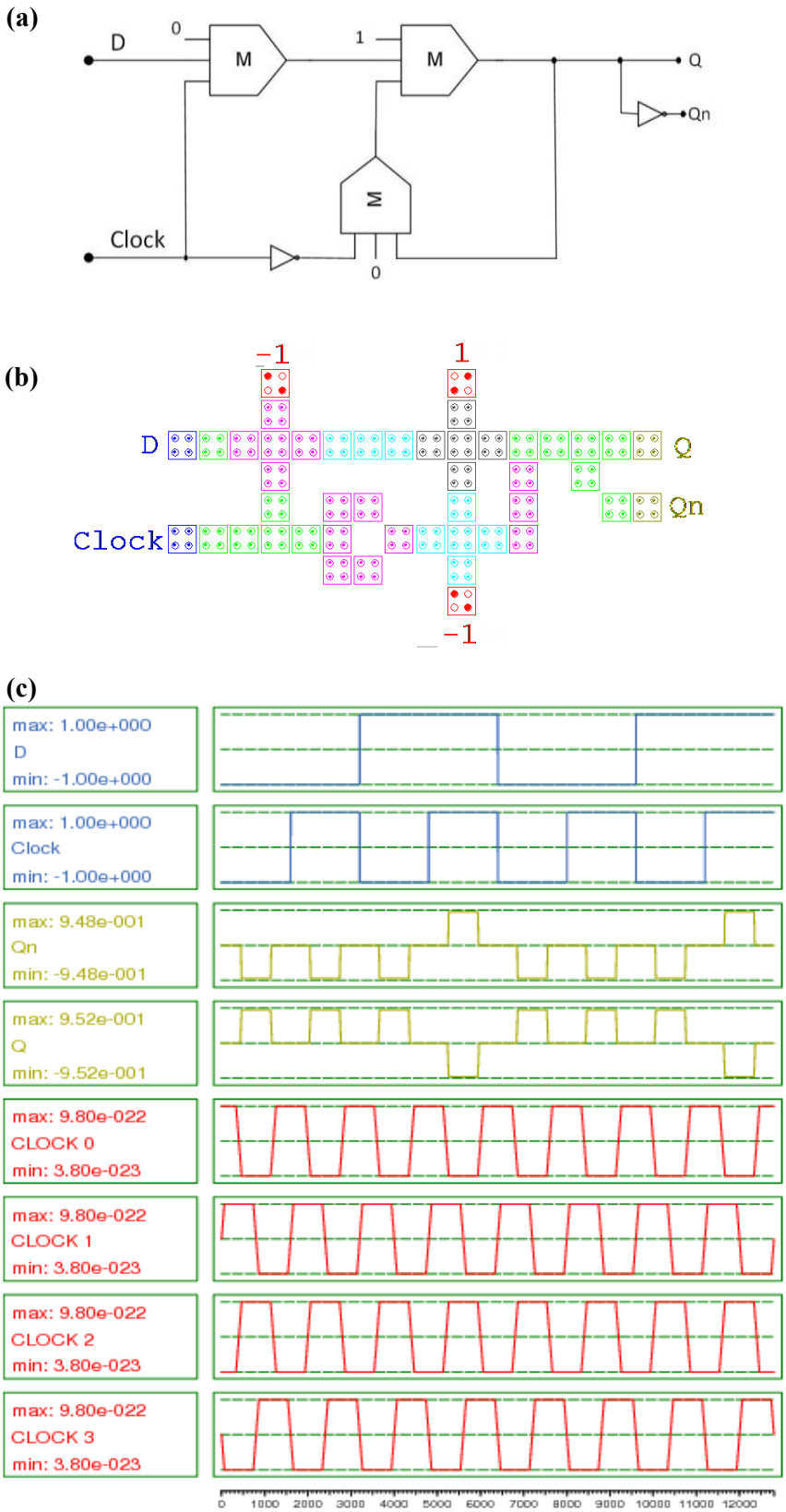


Fig. 3: a) Circuit diagram of D Flip-Flop. b) QCA layout of D Flip-Flop. c) Simulation result of QCA D Flip-Flop.

Shift Register

One of the most important applications of flip-flops is shift register. We can design the shift register with the D flip-flop by consecutive series of D flip-flops. In a 5-bit shift register we have five D flip-flops; each D flip-flop shifts the input to the output if the clock is high. In this paper, we are presented a shift register created by the D flip-flop in QCA technology. This circuit diagram of the 5-bit shift register contains five D flip-flops as shown in Fig.4a. Also, the QCA layout and the simulation result of the shift register in QCA Designer are shown in Fig.4b and Fig.4c, respectively. The 5-bit shift register has 5 clock cycles or 20 clock phases delay. Each bit needs one phase to shift and therefore the input of the first flip-flop goes to the output of the last flip-flop after 20 clock phases.

Linear feedback shift register (LFSR)

LFSR is a shift register whose input bit is a linear function of its previous state. LFSR is the heart of any digital system that relies on pseudo random bit sequences. We can create this circuit using shift register and XOR gate. Schematic of an LFSR is shown in Fig.5a. In this circuit, a shift register shifts an input bit to the output on the rising edge of clock signal.

QCA is a clock-based technology and if we set the Clock of this circuit in logic "1", the input is shifted to the output in the D flip-flop. The input of XOR is the output of the third D flip-flop and the output of the fifth D flip-flop. QCA layout of the LFSR is shown in Fig.5b. Fig.5c shows the simulation results of the LFSR in QCA designer, which shows the random output.

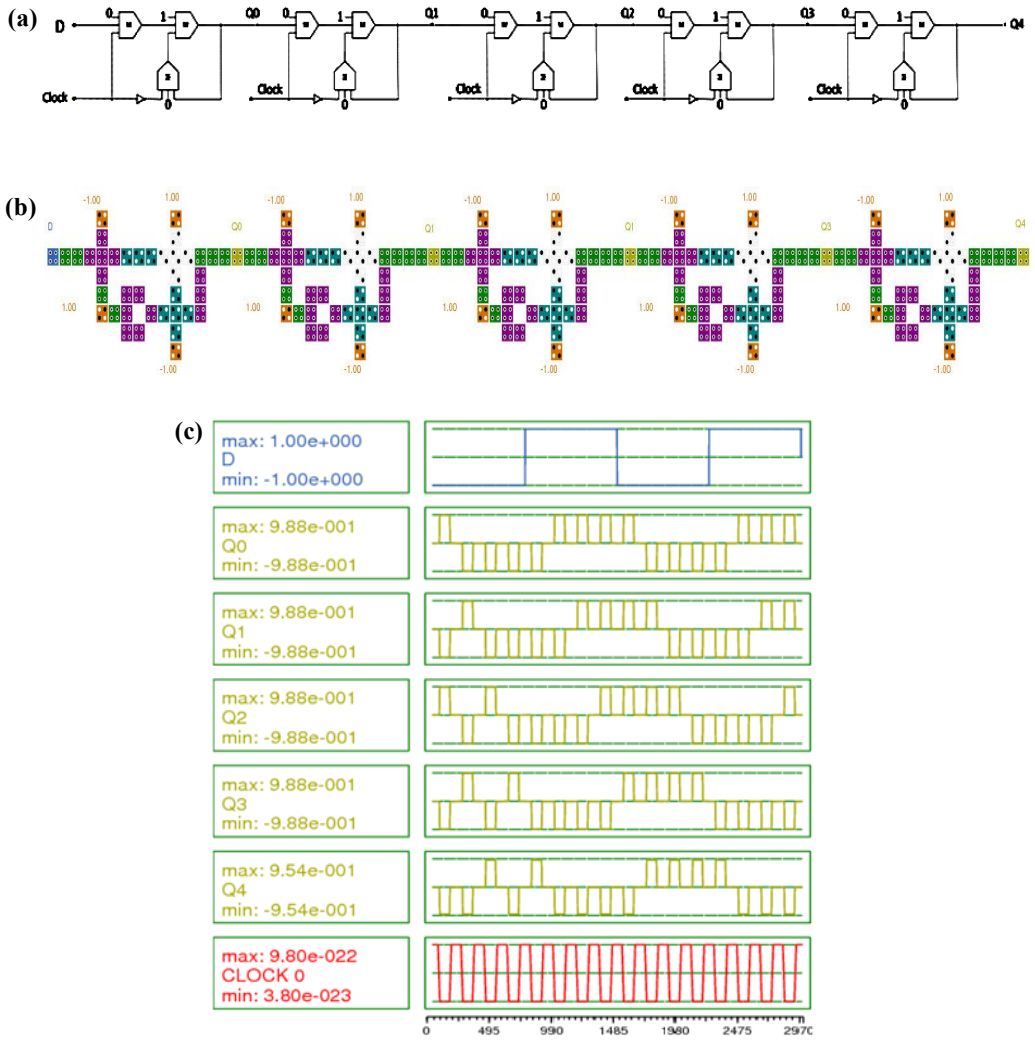


Fig. 4: a) Schematic of 5-bit shift register. b) QCA layout of 5-bit shift register. c) Simulation of 5-bit shift register in QCAdesigner.

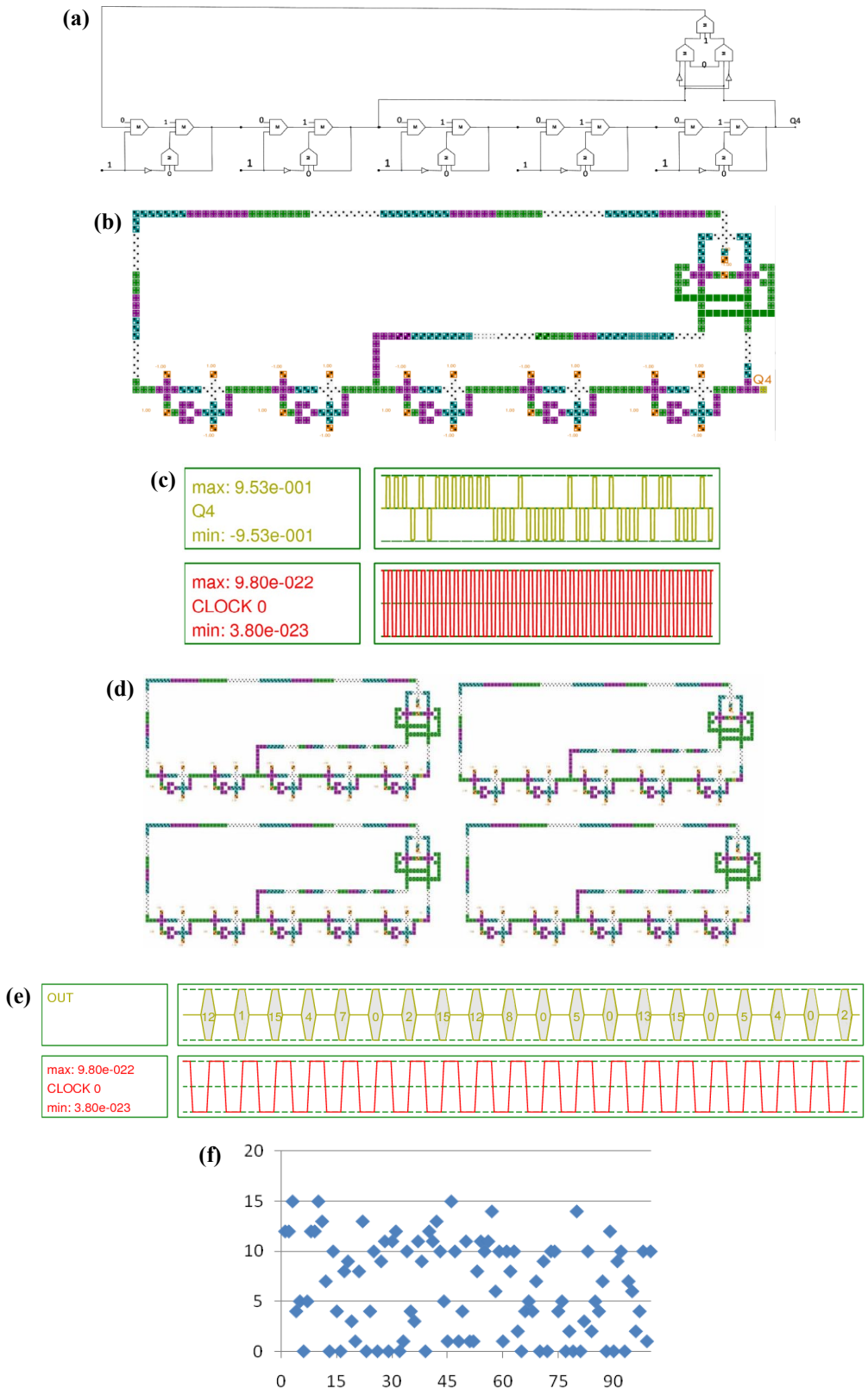


Fig. 5: a) Schematic of LFSR. b) QCA layout of LFSR. c) Simulation result of LFSR. d) QCA layout of 4-bit RNG. e) Simulation result of 4-bit QCA RNG. f) Distribution diagram of the random numbers generated by the proposed QCA 4-bit RNG.

To create each random bit we need one LFSR shown in Fig.5b. In Fig.5d and Fig.5e, the layout and the simulation results of a 4-bit RNG are shown. The distribution diagram of the random numbers generated by this circuit is shown in Fig.5f.

RNG using XOR

To optimize the previous circuit we can set the Clock of D flip-flops to logical "1". Therefore, we can replace these flip-flops with a wire. The resulting circuit is shown in Fig.6a. The QCA layout and its simulation results are shown in Fig.6b and Fig. 6d, respectively. This circuit contains one XOR Gate and a feedback. In Fig. 6c and Fig. 6e, the QCA layout and its simulation result of a 4-bit RNG using XOR Gate are shown. The distribution diagram of the random numbers generated by the proposed layout is shown in Fig. 6f. This diagram shows that the numbers generated by this circuit do not have any particular order and they are completely randomly.

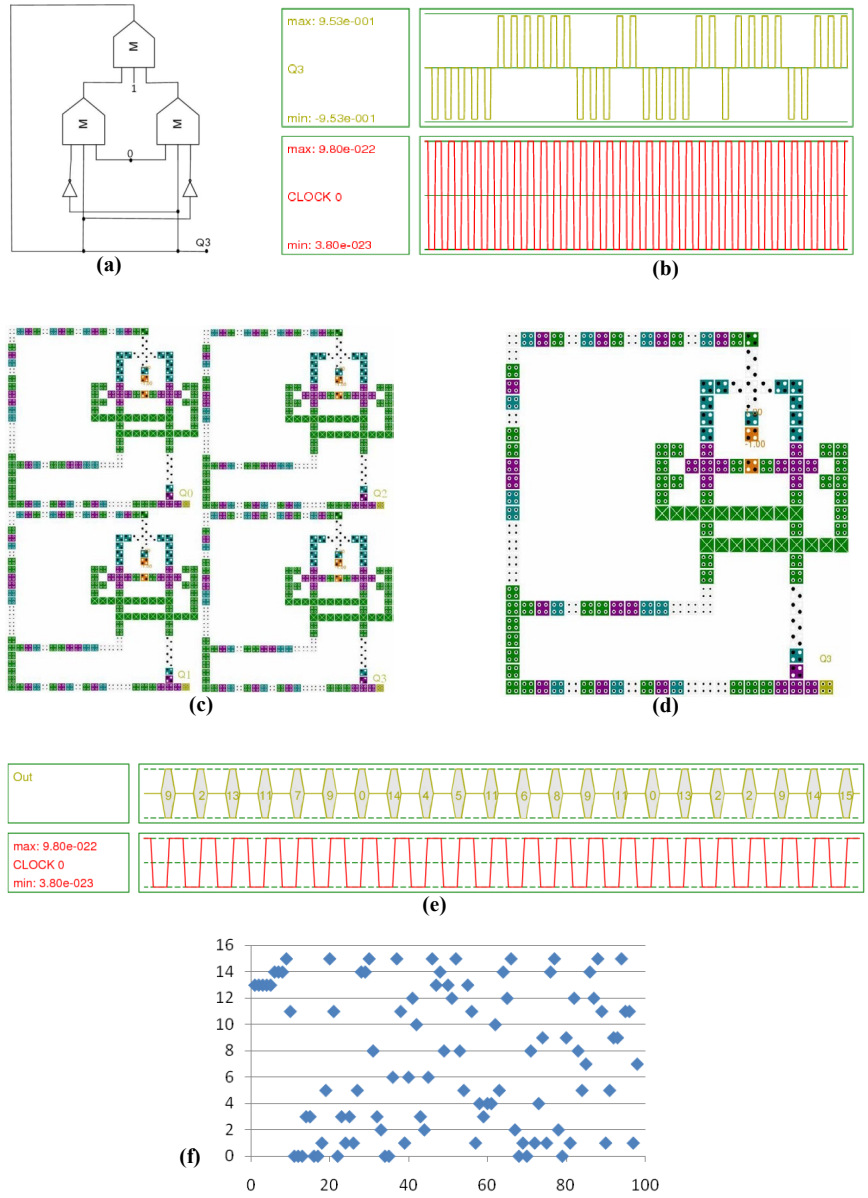


Fig. 6a: Schematic of one-bit RNG using XOR Gate. b) Simulation result of one-bit RNG using XOR Gate. c) QCA layout of 4-bit RNG using XOR Gate. d) QCA layout of one-bit RNG using XOR Gate. e) Simulation result of 4-bit RNG using XOR Gate. f) Distribution diagram of the random numbers generated by QCA 4-bit RNG using XOR Gate.

RNG using half adder

The addition of two random numbers can create a new random number. Therefore, we set the output of two RNGs as the inputs of a half adder. The half adder has two outputs (Sum and Carry). The simulation of this topology will show better results in comparison with the previous topology. Figs.7a-7e show the circuit diagram, equivalent QCA layout and simulation results of a one-bit RNG and a 2-bit RNG with half adder created using this method. The distribution diagram of the random numbers generated by the 2-bit RNG in Fig.7d is shown in Fig.7f. This diagram shows that the numbers generated by this circuit do not have any particular order and they are completely randomly.

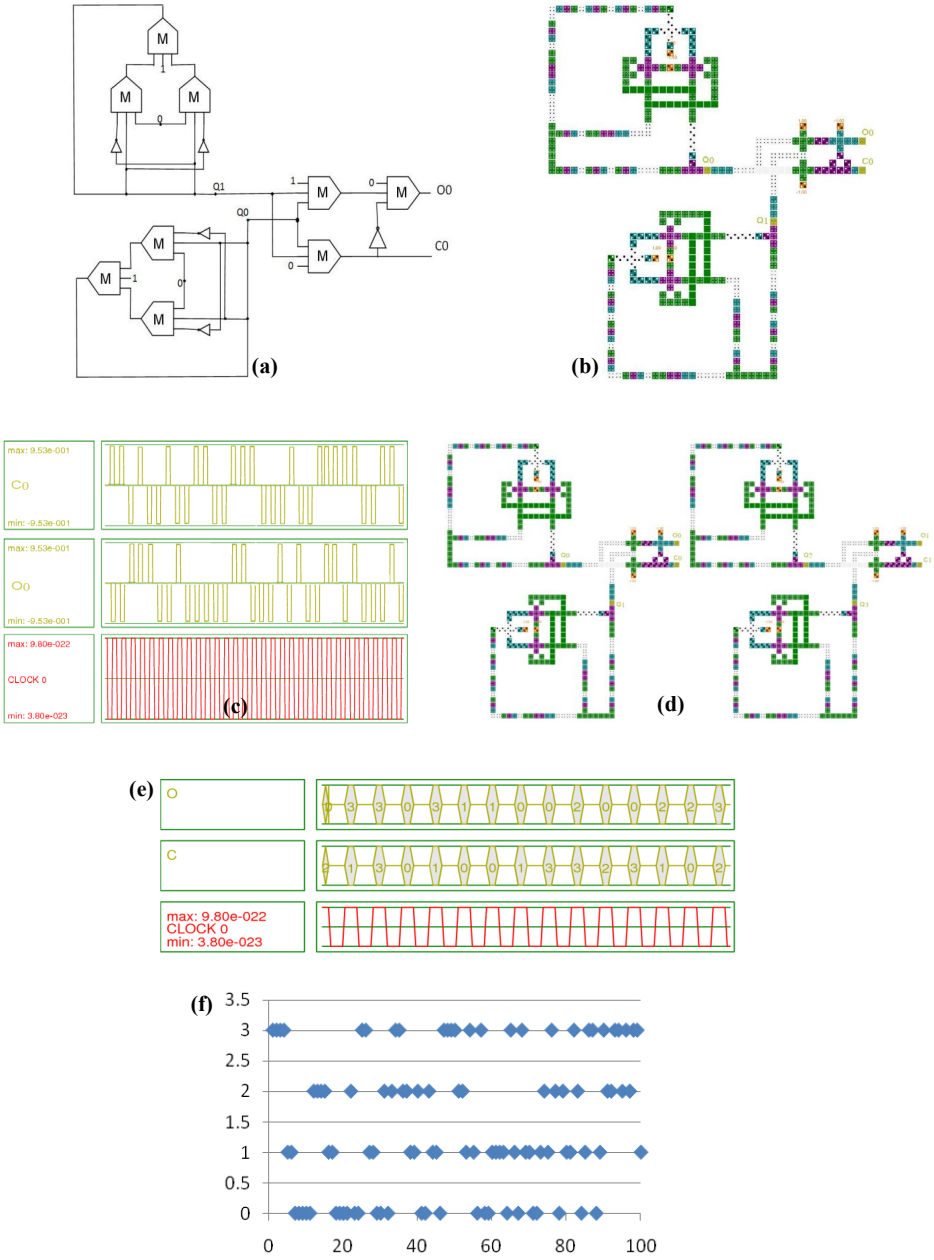


Fig. 7: a) Circuit diagram of one-bit RNG using half adder. b) QCA layout of one-bit RNG using half adder. c) Simulation result of one-bit RNG using half adder. d) QCA Layout of 2-bit RNG using half adder. e) Simulation result of 2-bit RNG using half adder. f) Distribution diagram of the random numbers generated by the QCA 2-bit RNG using half adder.

Fig.8a shows a QCA 4-bit RNG design using this method. In Figs.8b and 8c, the distribution diagrams of the random numbers generated by the proposed 4-bit RNG are shown. In this figure, the proposed 4-bit RNG is simulated twice (Fig.8b and Fig.8c) and two different sets of random numbers, which are completely randomly, are obtained.

According to the circuits presented in this paper, we can design the last circuit to generate the random numbers. This circuit contains four XORs and four half adders. Each XOR generates a one-bit random number and these numbers are added together. Using a feedback in this circuit from the last half adder to the first half adder, the random numbers are generated continuously. The proposed circuit diagram is shown in Fig.9a. The equivalent QCA layout shown in Fig.9b and the two simulations of this circuit are shown in Fig.9c, which have different results as we expect from RNGs.

In Figs.9d and 9e, the distribution diagrams of the random numbers generated by the proposed RNG are shown. In this figure, the proposed RNG is simulated twice and two different sets of random numbers, which are completely randomly, are obtained. The variance of the first data set (shown in Fig.9d) is 20.3737 and the standard deviation is 4.5137. The distribution of these numbers is shown in Fig.10a. The variance of the first data set (shown in Fig.9b) is 18.0076 and the standard deviation is 4.2435. Also, the distributions of these numbers are shown in Fig.10b. From these results, it can be concluded that this circuit can generate real random numbers.

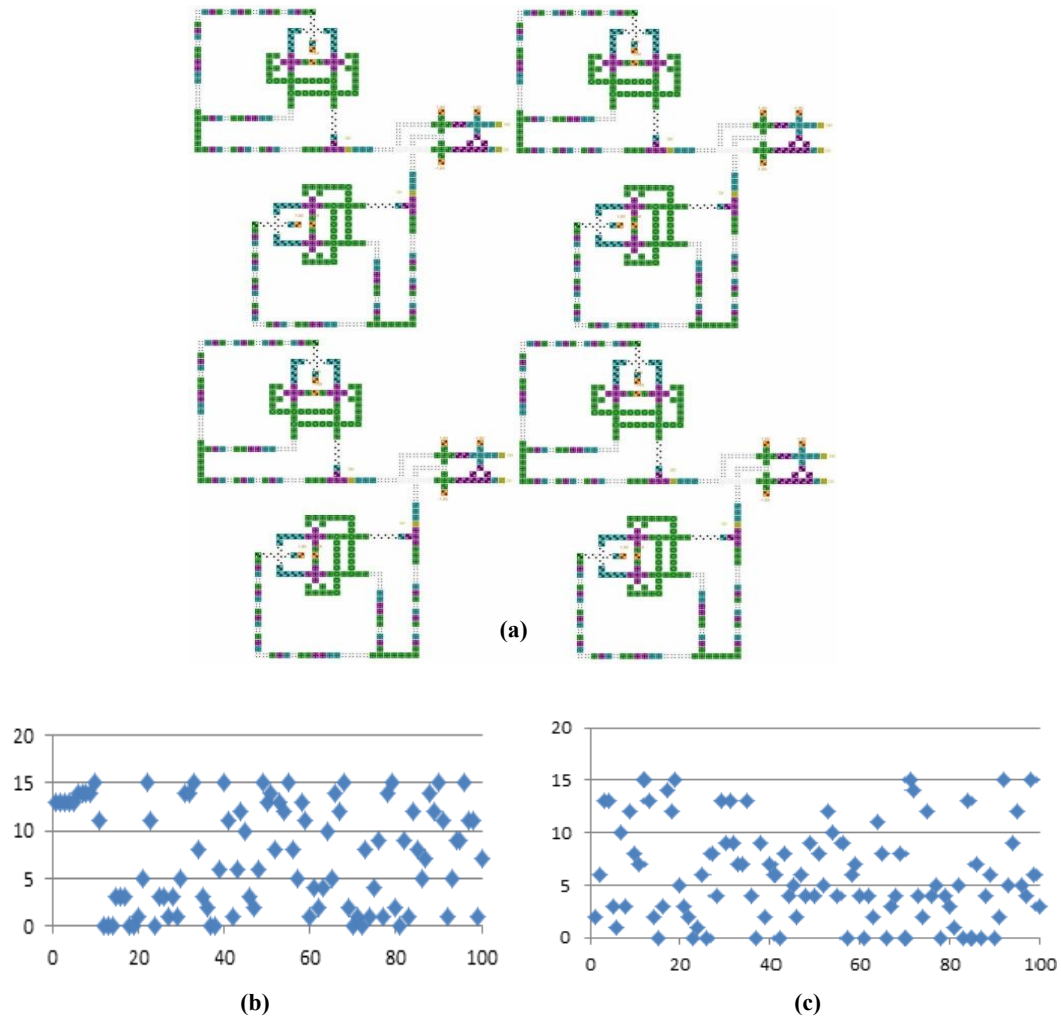
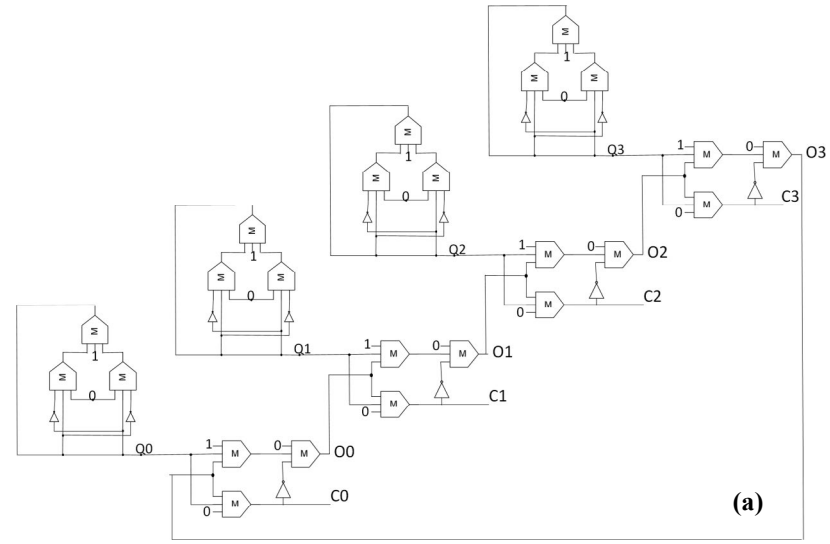
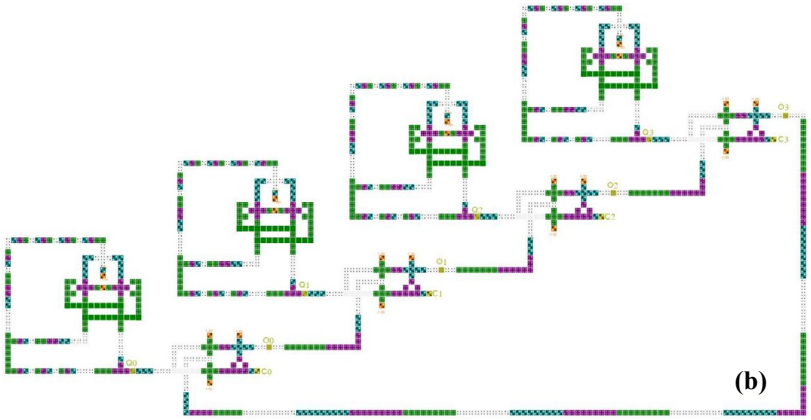


Fig. 8: a) QCA layout of the 4-bit RNG using half adder. (b, c): Distribution diagrams of the random numbers generated by the 4-bit RNG using half adder (b) the first simulation (c) the second simulation.





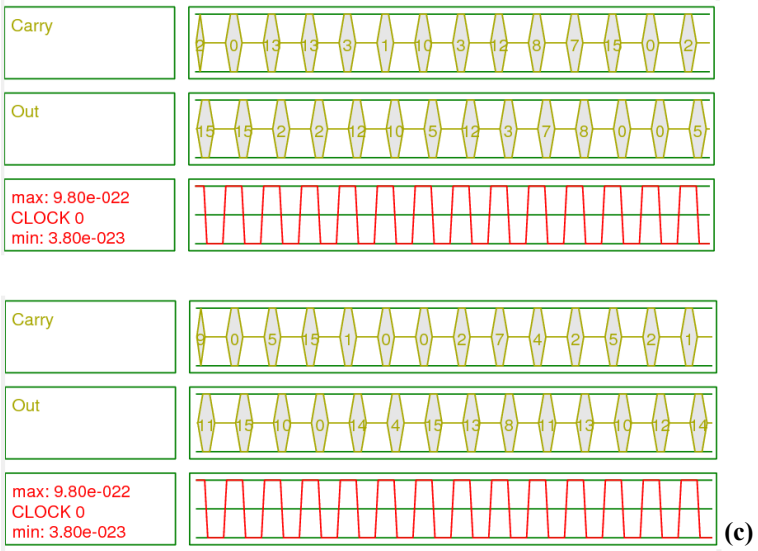


Fig. 9: a) Circuit diagram of the proposed RNG. b) QCA layout of the proposed RNG. c) Two simulation results of the proposed RNG. (To be continued)

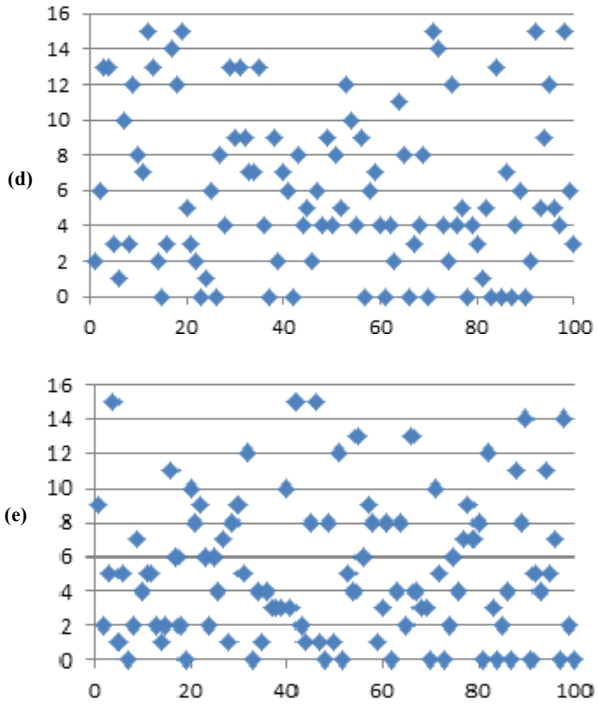


Fig. 9: Distribution diagrams of the random numbers generated by the proposed RNG (d) the first simulation (e) the second simulation.

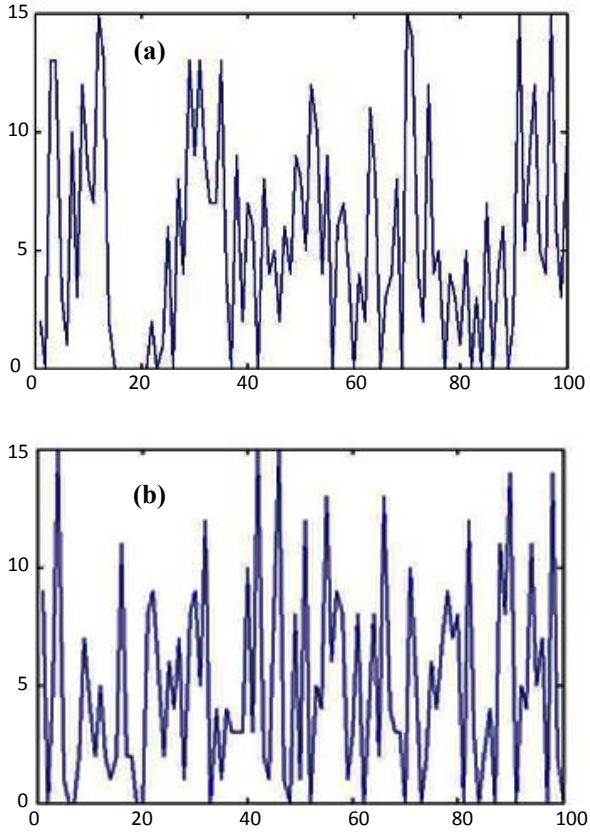


Fig. 10: a) Distribution of the random numbers in Fig. 9d. b) Distribution of the random numbers in Fig. 9e.

RESULTS AND DISCUSSION

Table 1 shows the specification of the proposed designs based on the number of cells, bit rate, clock frequency and power. The consumption power in the QCA arrays is $10^{-10}W$ per input bit [17, 18]. In this research, all designs do not have any input and thus the consumption power of all proposed circuits is 10^{-10} W . Table 2 shows the comparison between the proposed designs and the previous works in terms of the area, power and bit rate. As shown in Table 2, the proposed 4-Bit RNG using XOR has the smallest area. Also, all of our designs have the minimum power in comparison with the previous works because the proposed layouts don't have any inputs.

Table 1: Specification of the proposed designs.

Layout	Number of cells	POWER	Clock phase	BIT RATE
1-Bit RNG using LFSR	442 cells	$10^{-10}W$	11	1 Tb/S
4-Bit RNG using LFSR	1768 cells	$10^{-10}W$	11	1 Tb/S
1-Bit RNG using XOR	165 cells	$10^{-10}W$	9	1 Tb/S
4-Bit RNG using XOR	660 cells	$10^{-10}W$	9	1 Tb/S
1-Bit RNG using XOR and Half Adder	398 cells	$10^{-10}W$	12	1 Tb/S
2-Bit RNG using XOR and Half Adder	796 cells	$10^{-10}W$	12	1 Tb/S
4-Bit RNG using XOR and Half Adder	1592 cells	$10^{-10}W$	48	1 Tb/S
4-Bit RNG in Fig.34	1148 cells	$10^{-10}W$	23	1 Tb/S

Table 2. Comparison with the previous works.

Designs	Layout	POWER	BIT RATE	AREA
Proposed	1-Bit RNG using LFSR	$10^{-10}W$	1 Tb/S	$0.09\,\mu m^2$
	4-Bit RNG using LFSR	$10^{-10}W$	1 Tb/S	$0.36\,\mu m^2$
	1-Bit RNG using XOR	$10^{-10}W$	1 Tb/S	$0.02\,\mu m^2$
	4-Bit RNG using XOR	$10^{-10}W$	1 Tb/S	$0.08\,\mu m^2$
	1-Bit RNG using XOR and Half Adder	$10^{-10}W$	1 Tb/S	$0.08\,\mu m^2$
	2-Bit RNG using XOR and Half Adder	$10^{-10}W$	1 Tb/S	$0.16\,\mu m^2$
	4-Bit RNG using XOR and Half Adder	$10^{-10}W$	1 Tb/S	$0.32\,\mu m^2$
	4-Bit RNG in Fig.34	$10^{-10}W$	1 Tb/S	$0.2\,\mu m^2$
Previous	[20] (4-input RNG)	$4\times10^{-4}\,\mu W$	1 Tb/S	$0.01\,\mu m^2$
	[20] (5-input RNG)	$5\times10^{-4}\,\mu W$	1 Tb/S	$0.025\,\mu m^2$
	[19] (LFSR)	$2\times10^{-4}\,\mu W$	1 Tb/S	$0.08\,\mu m^2$
	[21]	3.9 mW	1.4Mb/s	$1.5\,mm^2$
	[22]	2.3 mW	10Mb/s	$0.0016\,mm^2$
	[23]	$50\,\mu W$	200Kb/s	$0.009\,mm^2$
	[24]	29 mW	40Mb/s	$0.518\,mm^2$
	[25]	1 mW	200Kb/s	$0.036\,mm^2$
	[26]	$9.39\,\mu W$	500b/s	$0.031\,mm^2$
	[26]	$180\,\mu W$	5Kb/s	$0.031\,mm^2$
	[26]	$180\,\mu W$	50Kb/s	$1.49\,mm^2$
	[27]	29 mW	40Mb/s	$0.234\,mm^2$
	[28]	1 mW	20Mb/s	_____
	[29]	$1.4\,\mu W$	40Kb/s	$0.05\,mm^2$

CONCLUSION

In this paper, we presented the methods to design random numbers in QCA. Linear feedback shift register (LFSR) is one of these circuits. We optimized a LFSR circuit using wires instead of D flip flops. This circuit was faster and the number of clock phases was less than the previous LFSRs. We also used XOR Gates and Half Adder to generate the random numbers.

Finally, we designed a circuit to generate 4-bit random numbers. The obtained results showed that these circuits were really random number generators and these random numbers can be used in any circuits or anywhere we need random numbers.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interests regarding the publication of this manuscript.

REFERENCES

[1] International Technology Roadmap for Semiconductors (ITRS), (Available from: http://www.itrs.net.) (2016).

[2] Lent C. S., Tougaw P. D., (1997), A device architecture for computing with quantum dots. *Proceed. IEEE*. 85: 541-557.

[3] Lent C. S., Isaksen B., Lieberman M., (2003), Molecular quantum dot cellular automata. *J. Am. Chem. Soc*. 125: 1056-1063.

[4] Bajec L., Zimic N., Mraz M., (2006), Towards the bottom-up concept: Extended quantum-dot cellular automata. *Microelect. Eng*. 83: 1826-1829.

[5] Snider G. L., Orlov A. O., Amlani I., Bernstein G. H., Lent C. S., Merz J. L., (1999), Quantum-dot cellular automata. *Microelectronic Eng*. 47: 261-263.

[6] Mardiris V. A., Ioannis G., (2010), Design and simulation of modular 2n to 1 quantum-dot cellular automata (QCA) multiplexers. *Int. J. Circuit Theory and Appl*. 38: 771-785.

[7] Yang X., Cai L., Huang H., Zhao X., (2012), A comparative analysis and design of quantum-dot cellular automata memory cell architecture. *Int. J. Circuit Theory and Appl.* 40: 93-103.

[8] Liu M., (2006). Robustness and power dissipation in quantum-dot cellular automata. PhD thesis. *Notre Dame University*.

[9] Askari M., Taghizadeh M., (2011), Logic circuit design in nano-scale using quantum-dot cellular automata. *Europ. J. Sci. Res.* 48: 516-526.

[10] Zhang R., Walnut K., Wang W., Jullien G., (2012), A method of majority logic reduction for quantum cellular automata. *IEEE*. 3: 443-450.

[11] Azari A., Zabihi S. A., Seyyedi S. K., (2012), Conductance in quantum wires by three quantum dots arrays. *Int. J. Nano Dimens.* 2: 213-216.

[12] Purkayastha T., De D., Chattopadhyay T., (2016), Universal shift register implementation using quantum dot cellular automata. *Ain Shams Engineering J*. In Press, Corrected Proof.

[13] Kumar D., Mitra D., (2016), Design of a practical fault-tolerant adder in QCA. *Microelectronic. J.* 53: 90-104.

[14] Majumder A., Singh P. L., Chowdhury B., Mondal A. J., Anand V., (2015), Efficient design and analysis of N-bit reversible shift registers. *Procedia Computer Sci.* 57: 199-208.

[15] Gladshtein M., (2016), Quantum-dot cellular automata serial decimal processing-in-wire: Run-time reconfigurable wiring approach. *Microelectron. J*. 55: 152-161.

[16] Heikalabad S., Navin R., Hosseinzadeh A. H. M., (2016), Content addressable memory cell in quantum-dot cellular automata. *Microelect. Eng*. 163: 140-150.

[17] Lent C. S., Tougaw P. D., Porod W., (1993), Quantum cellular automata. *Nanotechnol.* 4: 49-57.

[18] Kim K., (2006), Quantum-dot cellular automata design guideline. EICE transactions on funda-mentals of electronics. *Communicat. Computer Sci*. 89: 1607-1614.

[19] Mustafa M. , Beigh M. R., (2014), Novel linear feedback shift register design in Quantum Dot celullar automata. *Indian J. Pure and Appleid Physic.* 52: 203-209.

[20] Keikha A., Dadkhah C., Tehrani M., Navi K., (2011), A novel design of a random generator circuit in QCA. *Int. J. Comp. Applic.* 35: 30-36.

[21] Petrie C. S., Connelly J., (2000), A noise-based IC random number generator for applications in cryptography. *IEEE Transact. Circuits and Systems.* 47: 615-621.

[22] Bucci M., Germani L., Luzzi R., Trifiletti A., Varanonuovo M., (2003), A high-speed oscillator-based truly random number source for cryptographic applications on a smart card IC. *IEEE Transact. Computers*. 52: 403-409.

[23] Brederlow R., Prakash R., Paulus C., Thewes R., (2006), A low power true random number generator using random telegraph noise of single oxide-traps. *IEEE Int. Solid-State Circuits Conference (ISSCC)*. 1666-1675.

[24] Pareschi F., Setti G., and Rovatti R., (2006), A fast chaos-based true random number generator for cryptographic applications. *Proceedings of the 32nd European Solid-State Circuits Conference.* 130-133.

[25] Tokunaga C., Blaauw D., Mudge T., (2008), True random number generator with a metastability-based quality control. *IEEE J. Solid-State Circuits.* 43: 78-85.

[26] Holleman J., Member S., Bridges S., Otis B. P., Diorio C., (2008), A 3W CMOS true random number generator with adaptive floating-gate offset cancellation. *IEEE J. Solid-State Circuits.* 43: 1324-1336.

[27] Pareschi F., Setti G., Rovatti R., (2010), Implementation and testing of high-speed CMOS True random number generators based on chaotic systems. *IEEE Transact. Circuits and Systems.* 57: 3124-3137.

[28] Cao F., Li S., (2010), Random numbers from an integrated CMOS double-scroll IEICE Electronics Express. 7: 1382-1387.

[29] Chen W., Che W., Bi Z., Wang J., Yan N., Tan X., (2009), A 1.04 μW truly random number generator for gen2 RFID tag. *IEEE Asian Solid-State Circuits Conference*. 117-120.