Feed Forward Neural Network Approach for Reversible Logic Circuit Simulation in QCA

Arijit Dey, Kunal Das, Sanjoy Das and Mallika De

Abstract Quantum dot Cellular Automata (QCA) is becoming a new paradigm in nanoscale computing. Artificial Neural Network model is a promising model to design and simulate QCA circuits. This study proposes a new approach to design, model and simulate small circuit as well as large circuit. Feed Forward Neural Network (FFNN) model is used to design and simulate the reversible circuit as well as conservative circuit. The simulation result of this proposed FFNN model gives better result than exhaustive simulation of QCADesigner.

Keywords QCA · Artificial neural network · Feed forward neural network (FFNN) · Reversible circuit · Conservative circuit

A. Dey (✉)
Department of Computer Application, B.P. Poddar Institute of Management
and Technology, Kolkata 700052, West Bengal, India
e-mail: ad.computerapplication@gmail.com

K. Das
Department of Information Technology, B.P. Poddar Institute of Management
and Technology, Kolkata 700052, West Bengal, India
e-mail: kunaldasqca@gmail.com

S. Das
Department of Engineering and Technological Studies, Kalyani University,
Kalyani 741235, West Bengal, India
e-mail: dassanjoy_kalyani@yahoo.co.in

M. De
Dr. Sudhir Chandra Sur Degree Engineering College, Kolkata 700052,
West Bengal, India
e-mail: demallika@yahoo.com

J.K. Mandal et al. (eds.), *Information Systems Design and Intelligent Applications*,
Advances in Intelligent Systems and Computing 339,
DOI 10.1007/978-81-322-2250-7_7

1 Introduction

The Complementary Metal Oxide Semiconductor (CMOS) technology provides high density and low power Large Scale Integrated Circuit (VLSI) in micro scale computing. Now a days, this technology is facing new challenges like high leakage of current, power dissipation in terms of heat and reaches its limit. Researchers are still finding the alternatives of CMOS technology in nanoscale computing for VLSI design. Semiconductor Industries Association's International Roadmap for Semiconductors has reported that the circuit size is becoming double in every 18 months [1]. Quantum dot Cellular Automata (QCA) is an emerging technology and an alternative of CMOS technology. QCA was first introduced by Lent et al. [2], Tougaw and Lent [3] in the year 1993. In QCA electrons are confined within the cell so that there is no current and output capacitance in the circuit [4]. The electrons are tunneled through tunnel junction. Each QCA cell consists of four quantum dots and two extra electrons are confined within the cell. These extra electrons are positioned diagonally to hold maximum distance of electrons and these two positions define two states of polarization +1.00 as 'logic 1' and −1.00 as 'logic 0' respectively as shown in Fig. 1a. The three input majority voter is shown in Fig. 1b. Figure 1c, d shows the basic logic functions AND operation and OR operation by putting one input to fixed polarized −1.00 and +1.00 respectively.

The irreversible computation perform the same way as the conventional computers i.e., once the output is generated from the logic block the input bits are lost, so that the power is retained in the system [5]. Reversible computing computes with almost zero power dissipation. Landauer [6] has proved that each bit information loss produce $K_B T \ln 2$ J of heat energy for irreversible logic computation where K_B is Boltzman's constant and T is the absolute temperature. Bennett [7] has proved zero power dissipation in case of reversible logic computing. Basically, Feynman gate, Fredkin gate and Toffoli gate [8–10] perform as reversible logic gate. In reversible logic computing, the mapping of input vector I_V and output vector O_V is bijective i.e., each input yields to a distinct output [8]. In Conservative logic is one type of reversible logic. Conservative logic gate inputs from input vector I_V and outputs from output vector O_V are mapped in a way that parity of inputs I_V and

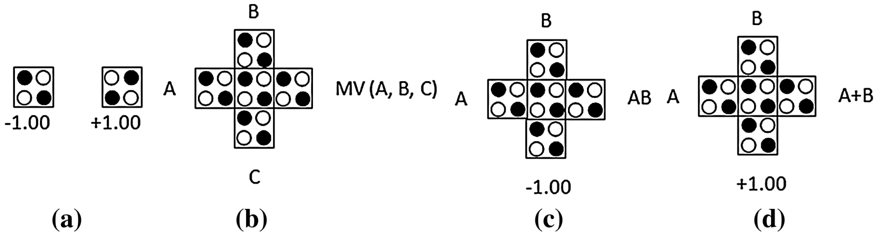


Fig. 1 **a** QCA cell polarization. **b** Three input majority voter. **c** AND gate. **d** OR gate

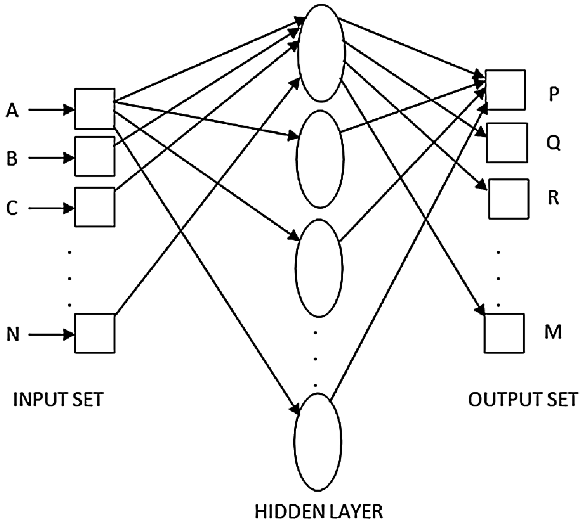
outputs O_V are preserved i.e., number of 1's present in each input and number of 1's present in output must be same [11]. In earlier works, several tools such as QCADesigner [12] have been used for designing and simulation of small QCA circuits. Recently, VHDL based simulation tools [13, 14], Hopfield neural networks [15, 16] have been introduced for presenting the simulation of QCA circuit. In [17] tansig method has been reported for simulation of QCA circuits.

In this paper, feed forward neural network [18] (FFNN) model is proposed for modeling and simulation of QCA circuit. This method illustrates the modeling and simulation of QCA reversible circuit by means of Knik energy. One cell impresses its neighboring cell due to coulomb interaction, known as Knik energy. The Knik energy is inversely proportional to distance between the charges of two cells q_i and q_j and is defined as

$$E_{i,j}^{knik} = \frac{1}{4\prod \varepsilon_0 \varepsilon_r} \sum_{n=1}^{4} \sum_{m=1}^{4} \frac{q_n^i q_m^j}{\left| r_n^i - r_m^j \right|} \tag{1}$$

where ε_0 is the permittivity of free space and ε_r is the relative permittivity. In this approach the polarization of input cells are imposed on the device cell by the effect of Knik energy which calculates the polarization of device cell. This polarization of device cell is transferred to the output cell as resultant polarization. The basic feed forward neural network is shown in Fig. 2. Each input from the input set is connected to each of the process in the hidden layer and the connection of each process of hidden layer is connected to one of the outputs in output set.

Fig. 2 Feed forward neural network model



2 Proposed Feed Forward Neural Network Model

The proposed feed forward neural network (FFNN) model is applied over the reversible logic computing as well as conservative logic computing for modeling and simulation. This FFNN model is very simple to design the reversible logic gate and conservative logic gate. In this study, the reversible logic gate is designed and simulated with few steps. This FFNN model consists of input set, one hidden layer and output set. An artificial feed forward neural network model FFNN is proposed here to demonstrate an experimental study of modeling and simulation of reversible circuits. The FFNN model is illustrated with QCA cell of size 18 nm and distance between a pair of QCA cells of 2 nm. This FFNN model has been tested and simulated by MATLAB 7.7 using a set of training data. The steps involved to design and simulate a reversible logic gate are discussed below:

Steps of FFNN model

1. Take the number of inputs (NI) of the reversible circuit in the input layer of FFNN model
2. Find the number of process of the hidden layer from the truth table of number of inputs
   Number of Processes (NPr) in the hidden layer = 2^{NI}
3. Set number of outputs (NO) in the output layer as the same number as inputs in the input layer.
   Set NO = NI and also Set I = 1.
4. Repeat Step 5 to Step 6 while (I <= NPr)
5. Find the polarization of process P(I)
6. Set I = I + 1
7. The polarization of each process is imposed on the output layer
8. To find a particular output of output layer set the polarization of the processes either the ‘imposed polarization’ or ‘0’ according to the output function.

2.1 Study on Feynman Gate

Feynman Gate is a 2 × 2 reversible logic gate i.e., it has 2 inputs and 2 outputs. The input vector I_V (A, B) is mapped to output vector O_V ($P = A \oplus B$, $Q = A$). Figure 3 shows the block diagram of a Feynman gate and the equivalent FFNN model design is shown in Fig. 4. The inputs A and B are put on the processors of input layer. These processors of input layer are connected to each of the process of hidden layer.

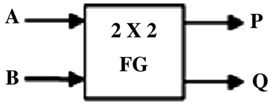


Fig. 3 Block diagram of Feynman gate

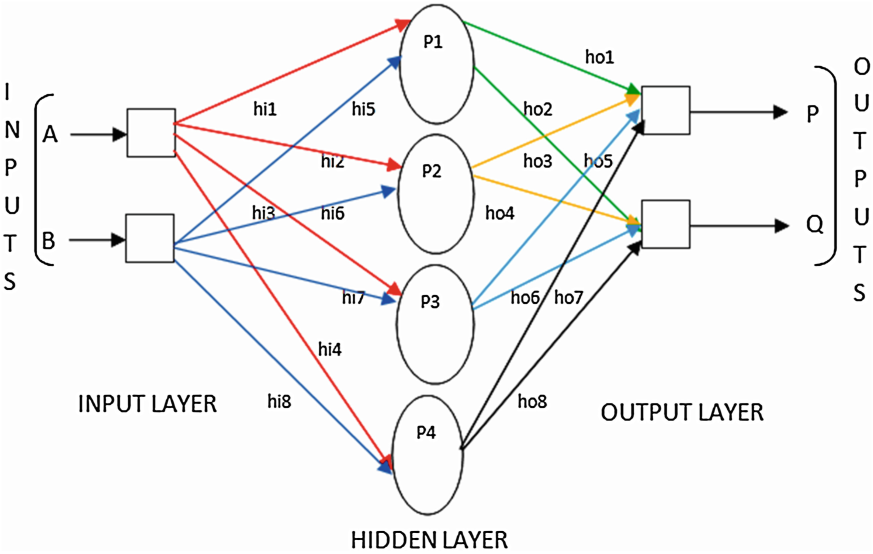


Fig. 4 Feed forward neural network (FFNN) model of Feynman gate

The lines hi1, hi2, hi3, hi4 show the connection between the processes P1, P2, P3, P4 of hidden layer from input A. Similarly, the lines hi5, hi6, hi7, hi8 show the connection to the processes P1, P2, P3, P4 of hidden layer from input B. Now, the polarizations of each connection hi1 to hi8 are imposed on the processes P1–P4 of hidden layer and these polarizations propagate each process to calculate the input combinations. The processes of the hidden layer give all the input combinations that are found from the input set. All these input combinations are connected to each of the input of output layer. The lines ho1, ho2 (input binary combination 00) which are found as outputs from the process P1 of hidden layer, act as inputs to the processors of the output layer. Similarly, ho3, ho4 are the input binary combination 01 produced by the process P2 of hidden layer, ho5, ho6 are the input binary combination 10 produced by the process P3 and the connection ho7, ho8 are the input binary combination 11 produced by the process P4 of the hidden layer. All these processes of hidden layer produce all input combinations from the truth table. All these are connected from hidden layer to each of the output processors of the output layer. The desired results are found by controlling the connections ho1 to ho8 in the processors of output layer. Finally the output processors of output layer drive the outputs P and Q by controlling the polarizations which are found from the each process of hidden layer. The polarization of the connections ho1, ho7 are set to 0, and ho3, ho5 are set to 1 to produce the output $P = \sum (ho3, ho5)$ (SOP form) of Feynman gate. The polarization of the connections ho2, ho4 are set to 0 and ho6, ho8 are set to the polarization that has found from the processes of hidden layer to get the desired result at output $Q = \sum (ho6, ho8)$ (SOP form) of Feynman gate.

3 Feed Forward Neural Network Simulation of Full Adder Circuit

This proposed FFNN model is also applied on QCA circuit design, modeling and simulation. Earlier, one bit conservative, lossless, zero garbage full adder circuit was designed and simulated by exhaustive simulation of QCADesigner [5] as shown in Fig. 5. In this study, the equivalent FFNN model of the conservative, lossless, zero garbage full adder circuit is designed and simulated by a set of training data. The FFNN model simulation is done by MATLAB 7.7. The architecture of the FFNN model of the said full adder circuit design is shown in Fig. 6. In this proposed FFNN model the processes of hidden layer produce all the combinations of inputs that have taken from the input layer i.e., the number of process in the hidden layer is eight as a full adder circuit has 3 inputs. The polarizations of all

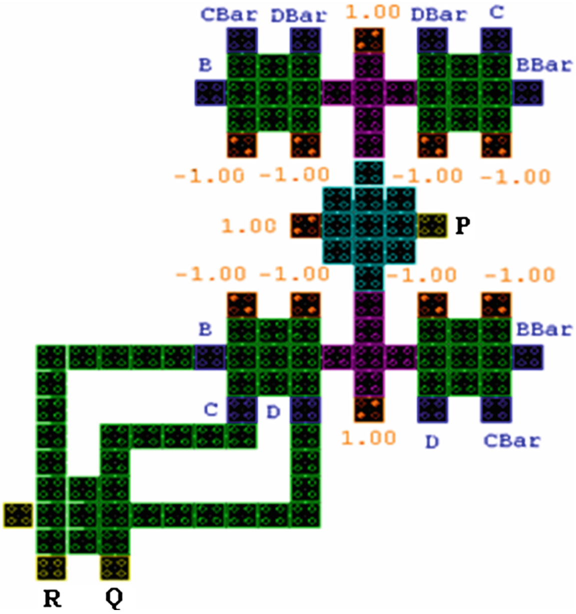


Fig. 5 Conservative full adder circuit design by QCADesigner

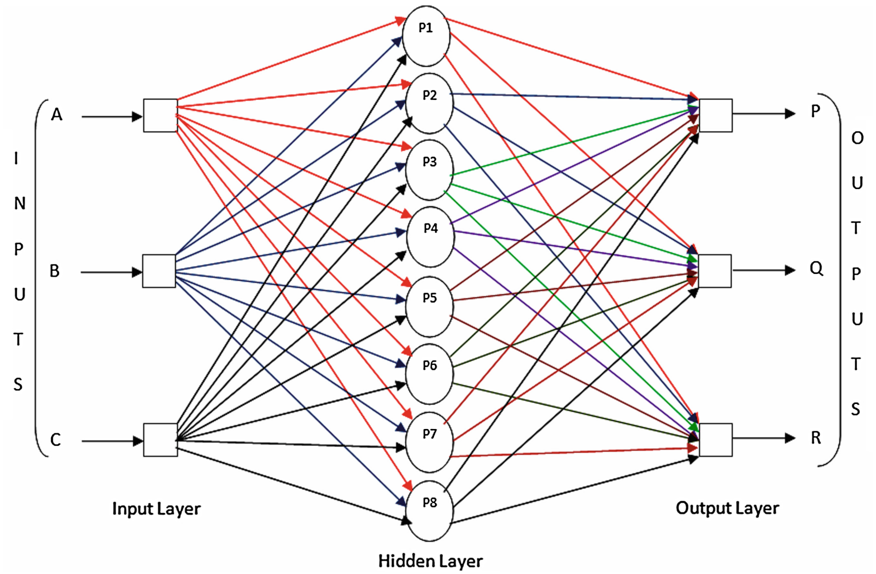


Fig. 6 FFNN model of conservative full adder

processes of hidden layer are calculated using Eq. 2, where E_K is the Knik energy of a QCA cell,$\Delta\mathrm{E} = \hbar/\tau$, *polarization* gives the polarization of previous cell and te is the tunneling energy. All these calculated polarizations of the processes of the hidden layer have imposed on the output layer. The weight/polarization of each connection between the processes of hidden layer and output layer control the weight/polarization of each output of the conservative full adder circuit. In this FFNN approach the polarization of the processes are found from the polarization of inputs that are imposed on the process of hidden layer. Once the polarizations of all processes are generated, they are imposed on all the outputs of output layer. Now, the polarization '0' is set to those connections that are not used to find out a particular output. Finally, the proposed FFNN model produces the output of conservative full adder $P = \sum(1, 2, 4, 7), Q = \sum(3, 5, 6, 7), R = \sum(3, 5, 6, 7)$ (SOP form).

$$Final\,Polarization = \frac{(E_K/2\Delta\mathrm{E}) \times polarization}{\sqrt{((1 + Ek)/te) \times polarization}} \tag{2}$$

4 Simulation Result Analysis

The design of zero garbage, lossless, conservative full adder circuit is done by QCADesigner as shown in Fig. 5. The exhaustive simulation result by QCADesigner is shown in Fig. 7. The polarization value of P is 0.877, Q is 0.931 and R is 0.930 whereas the proposed FFNN model has given the polarization values of P, Q and R as 0.922, 0.922 and 0.922 respectively. The FFNN model simulation result is given in Table 1. The polarization of the processes of hidden layer is computed and then the polarization of the final output is calculated. In Table 1 the simulation result of FFNN model is given. This simulation result of FFNN model gives a better polarization of output than exhaustive simulation by QCADesigner. The comparison of QCADesigner simulation result (represented by 1) and proposed FFNN model simulation result (represented by 2) for different outputs is shown in Fig. 8.

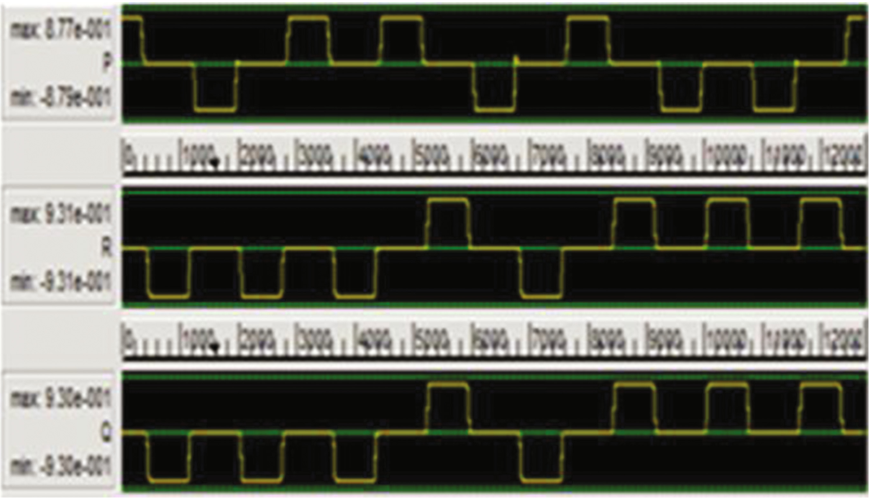


Fig. 7 Simulation result of conservative full adder circuit by QCADesigner

Table 1 Simulation result of conservative full adder circuit by FFNN model compared with QCADesigner

Initial input polarization of three inputs set for respective min terms	Process No. of hidden layer with calculated polarization of respective processes	Polarization set for output P at output layer	Polarization set for output Q at output layer	Polarization set for output R at output layer	Final polarization at output layer (MAILAB simulation)	Final polarization at output layer (QCADesigner simulation)
A = 1.00 (logic l) or −1.00 (logic 0)	Pl = 0.943	P1 = 0	P1 = 0	P1 = 0	Output P = 0.922	Output P = 0.877
	P2 = 0.943	P2 = 0.943	P2 = 0	P2 = 0		
	P3 = 0.943	P3 = 0.943	P3 = 0	P3 = 0		
B = 1.00 (logic l) or −1.00 (logic 0)	P4 = 0.943	P4 = 0	P4 = 0.943	P4 = 0.943	Output Q = 0.922	Output Q = 0.931
	P5 = 0.943	P5 = 0.943	P5 = 0	P5 = 0		
	P6 = 0.943	P6 = 0	P6 = 0.943	P6 = 0.943		
C = 1.00 (logic l) or −1.00 (logic 0)	P7 = 0.943	P7 = 0	P7 = 0.943	P7 = 0.943	Output R = 0.922	Output R = 0.930
	P8 = 0.943	P8 = 0.943	P8 = 0.943	P8 = 0.943		

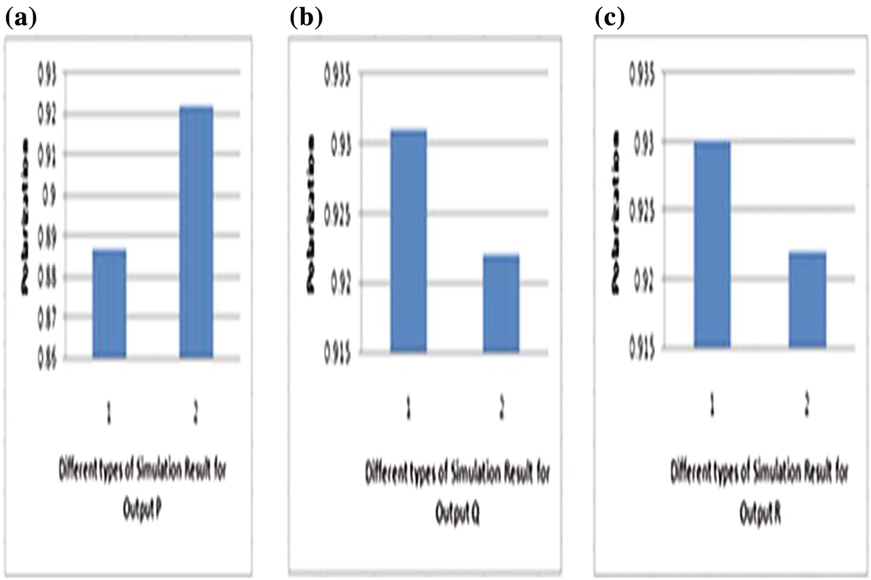


Fig. 8 QCADesigner and proposed FFNN model simulation results **a** for output P, **b** for output Q, **c** for output R

5 Conclusion

In this study, the artificial intelligence technology is used to design and simulate QCA reversible as well as conservative circuit. The modeling of QCA circuit using FFNN is very simple and the simulation with an acceptable precision of polarization is done by MATLAB. The accuracy of the polarization at each output is also compared with the exhaustive simulation result of QCADesigner. The result found from MATLAB simulation shows that this FFNN model is efficient and gives an acceptable precision at each output of QCA reversible as well as conservative circuit.

References

1. International technology roadmap for semiconductors. Semiconductor Industries Association, San Jose, CA. http://public.itrs.net (2001)
2. Lent, C.S., Tougaw, P.D., Porod, W., Bernstein, G.H.: Quantum cellular automata. Nanotechnology **4**(1), 49 (1993)
3. Tougaw, P.D., Lent, C.S.: Dynamic behavior of quantum cellular automata. J. Appl. Phys. **80** (8), 4722–4736 (1996)
4. Smith, C.G.: Computation without current. Science **284**(5412), 274 (1999)

5. Dey, A., Das, K., De, D., De, M.: Online testable conservative adder design in quantum dot cellular automata. In: Emerging Trends in Computing and Communication, pp. 385–393. Springer India, Berlin (2014)
6. Landauer, R.: Irreversibility and heat generation in the computing process. IBM J. Res. Dev. **5** (3), 183–191 (1961)
7. Bennett, C.H.: Logical reversibility of computation. IBM J. Res. Dev. **17**(6), 525–532 (1973)
8. Toffoli, T.: Reversible Computing, pp. 632–644. Springer, Berlin (1980)
9. Fredkin, E., Toffoli, T.: Conservative Logic, pp. 47–81. Springer, London (2002)
10. Feynman, R.P.: Quantum mechanical computers1. Found. Phys. **16**(6), 986 (1985)
11. Das, K., De, D.: Characterization, test and logic synthesis of novel conservative and reversible logic gates for Qca. Int. J. Nanosci. **9**(03), 201–214 (2010)
12. Walus, K.: ATIPS laboratory QCADesigner homepage. ATIPS Laboratory, University Calgary. Calgary, Canada (2002)
13. Henderson, S.C., Johnson, E.W., Janulis, J.R., Tougaw, P.D.: Incorporating standard CMOS design process methodologies into the QCA logic design process. IEEE Trans. Nanotechnol. **3** (1), 2–9 (2004)
14. Ottavi, M., Schiano, L., Lombardi, F., Tougaw, D.: HDLQ: a HDL environment for QCA design. ACM J. Emerg. Technol. Comput. Syst. (JETC) **2**(4), 243–261 (2006)
15. Behrman, E.C., Niemel, J., Steck, J.E., Skinner, S.R.: A quantum dot neural network. In: Proceedings of the 4th Workshop on Physics of Computation, pp. 22–24 (1996)
16. Neto, O.P.V., Pacheco, M.A.C., Hall Barbosa, C.R.: Neural network simulation and evolutionary synthesis of QCA circuits. IEEE Trans. Comput. **56**(2), 191–201 (2007)
17. Hayati, M., Rezaei, A.: New approaches for modeling and simulation of quantum-dot cellular automata. J. Comput. Electron. **13**(2), 537–546 (2014)
18. Bebis, G., Georgiopoulos, M.: Feed-forward neural networks. Potentials IEEE **13**(4), 27–31 (1994)