



# Optimized area efficient quantum dot cellular automata based reversible code converter circuits: design and energy performance estimation

Aishwarya Kaity<sup>1</sup> · Sangeeta Singh<sup>1</sup>

Accepted: 15 February 2021 / Published online: 19 March 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

Quantum-dot cellular automata (QCA) based circuit designs are creating a surge in transistorless computational technologies. Due to its quasi-adiabatic switching resulting in extremely low leakage power dissipation as there is no continuous path. These circuits also enjoy extremely high packaging density of the order of  $10^{12}$  devices/cm<sup>2</sup> because of its extremely scaled area of  $18\text{ nm} \times 18\text{ nm}$  along with very high 100 GHz frequency of operation. Further the loss of bit information could be abolished by reversible logic computing. This thereby realizes an energy efficient logic operations owing to bijective relation between inputs and outputs in reversible logic. This work investigates the code converter circuits which converts 4-bit binary code to excess-3 code and vice versa based on reversible QCA logic gates for the first time. Moreover an area efficient design for 4-bit binary to gray and vice-versa code converters also designed here. All these four code converter circuits are designed using reversible logic gate Feynman and Peres gates by deploying the QCA Designer and Designer-E tool v2.0.3. Finally the in depth performance estimation of the proposed circuits in terms of circuit complexity, quantum cost and energy dissipation are also presented here. Moreover, these QCA based circuits provide a strong evidence that reversible logic based QCA circuits can be efficiently deployed for these code converter circuits.

**Keywords** Quantum dot cellular automata · Binary code · Gray code · Excess-3 code · Code converter circuit

---

✉ Aishwarya Kaity  
kaity.aishwarya@gmail.com

<sup>1</sup> Microelectronics and VLSI design, National Institute of Technology Patna, Patna, Bihar, India

## 1 Introduction

Modern day society shows huge surge in the demands for fast, reliable, predictable and affordable scaling of computing performance. This can only be realized by the development of consumer electronics. Further these electronics domains find huge applications in various fields such as, ‘big data’ and data centres (Google, Facebook), national security, scientific discovery, control for avionics systems for aircraft, the automotive industry for self-driven cars and smart grid technologies. Till now these applications has been realized by the continuous scaling of transistors with enhanced functionality. The complementary metal oxide semiconductor (CMOS) transistors are used to design circuits, have been gradually scaled down to atomic level which increases the device density as well as the operating frequency. However these seem to be severely hindered due to the end of lithographic scaling, there by significantly obstructing 4 trillion dollar electronics industry growth severely [1]. Further for the higher integration density the architecture, logic and circuit designs are very crucial. But when this scaling reaches to its ultimate limit then the transistors face several short channel effects (SCEs), high leakage power dissipation, low transition speed etc. [2]. Under such circumstances many alternative technologies are being explored according to ITRS report. This includes tunnel field effect transistor (TFET) [3], single electron transistor (SET) [4], quantum dot cellular automata (QCA), tunnelling phase logic (TPL) [5] and carbon nanotube (CNT) [6]. Among all these, the transistorless QCA technology has been reported as promising technology [7–9] because it does not have the limitations of transistor at ultra-scaled dimensions. This is based on the quasi-adiabatic switching to realize the ultra-low power of operation, extremely low leakage due to no continuous path and transistorless computational model. Its core is the QCA cell with the area of  $18\text{ nm} \times 18\text{ nm}$ , hence it is very strong candidate for implementing high device density of the order of  $10^{12}$  devices/cm<sup>2</sup> along with very high 100 GHz frequency of operation. QCA cells are not physically connected but it passes information through “Coulomb’s interaction” process, which results in ultra-low power and extremely low leakage energy [10, 11]. The basic gates designed by QCA cells are three input majority gate and five input majority gate. By using these majority gates, several logic gates [12] namely, AND, OR, NOT, NOR, NAND, XOR etc., combinational circuits like, adder [13], subtractor [14], multiplexer [15], decoder [16], encoder [17], code converter [18] etc. and several sequential circuits [19] are already designed. Moreover, Landauer et al. [20] proved that  $kT \ln 2$  Joules of heat energy dissipates when each bit of information loss occurs in case of irreversible logic design. Here  $k$  is Boltzmann constant,  $T$  is absolute computing temperature in Kelvin. Following this Bennett et al. [21] coined that there is no loss of energy in case of reversible circuits owing to its bijective or one-to-one relation between inputs and outputs. This initiated huge research interest in deploying reversible logic gates for various combinational and sequential circuits realization. Several reversible logic gates are proposed by several researchers such as, Feynman gate [22, 23], Toffoli gate [24], Peres gate [25], Fredkin gate [26] etc. These all gates have one-to-one

mapping between input and output. By using these reversible gates [27], reversible adder [28], subtractor [29], multiplexers [30], decoder [31], priority encoder [32], reversible latches [33], sequential circuits [34] have been already designed. QCA wires cross each other by two crossover techniques, multilayer crossover [35] and coplanar crossover [36]. The multilayer crossover requires multiple layer designing which makes the actual circuit realization complicated. However, in coplanar crossover only one layer is required to design the complete circuit hence it is easier to implement. Further for digital communication system, code converter circuits are extremely crucial. Sometimes for the security of the transmitted data, it is converted to another codes. These codes include binary codes, such as binary coded decimal (BCD) code, gray code, excess-3 code etc. Binary code is converted to these respective codes for computation ease and reduction in power consumption. Post this to retrieve the original data, these codes are again needed to be converted in binary code. Gray code requires less bit transition at the time of counting, which leads to less power requirement. Excess-3 code is very useful for the subtraction operation as each excess-3 code has its complement, which reduces the circuit complexity.

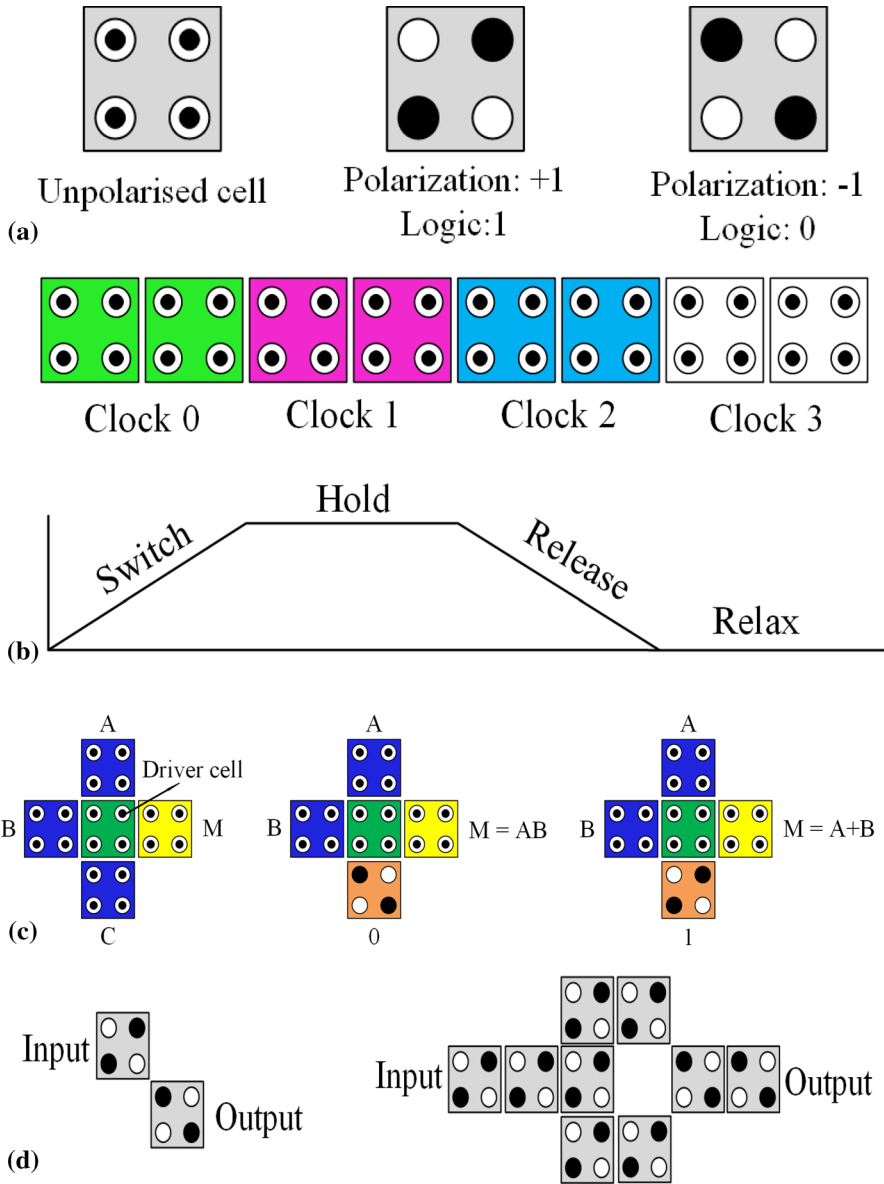
Inspired by these concurrent findings in the research domain, our work reports the code converter circuits which converts binary code to excess-3 code and vice versa based on reversible QCA logic gates for the first time. Apart from this, here an area efficient design for binary to gray code and vice-versa code converters are also reported. All these four code converter circuits are designed using reversible logic gate Feynman and Peres gates. The reversible Peres gate is realized using Feynman reversible gate, which is used to design binary-to-excess-3 code and excess-3-to-binary code. Moreover, these circuits include inverters and majority gates also. Code conversion of 4-bit data has been considered here. The performance of the circuits is also analysed in terms of number of cells, area, delay, quantum cost, number of garbage outputs and energy consumption. However, the actual circuit realization is still a limitation that can be addressed by can be metal island-based QCA devices for cryogenic temperature QCA, semiconductor quantum dots are nanostructures formed using electron beam lithographically [41, 42].

The rest of this manuscript is organized as follows: Sect. 2 demonstrates the background of QCA and different types of reversible logic gates. Then the basics of code converter circuits and their requirements are described in Sect. 3 and Sect. 4 covers the simulation framework. Section 5 shows the results and discussion along with the performance analysis of all the reported circuits. Finally, Sect. 6 concludes the performance of reported code converters designs and highlights its benefits.

## 2 Background

### 2.1 QCA basics

QCA is a zero dimensional cell which contains 4 quantum dots and two free electrons. These two electrons occupy two diagonally opposite quantum dots. According to the electrons position, the polarization of the cell is determined. Fig. 1a shows



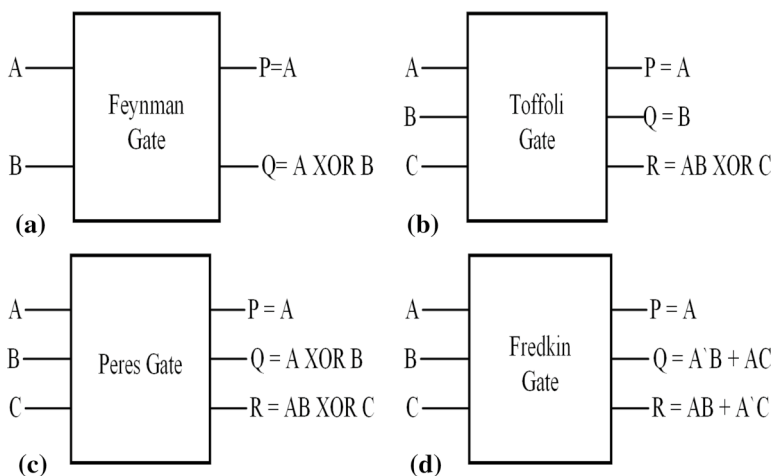
**Fig. 1** **a** QCA cells for different considered polarization state [8], **b** clocking technique [11], **c** three input majority gates [12], and **d** inverters [12]

two different polarizations of the QCA cell and their corresponding electron positions [8–10]. The binary data is stored in a cell in terms of the polarization of the cell. To change the polarization of a cell, electrons move from one dot to another by quantum tunnelling process. The clocking technique of QCA controls this tunnelling phenomena [11]. One clock cycle is divided into four clocking zones with 90° phase

delay and each clocking zone has four clocking states, i.e. switch, hold, release and relax. Figure 1b shows the clocking technique of QCA. In this four state the barrier between the quantum dots becomes high and low with the clock and electrons tunnel through it. There is no direct connection between the QCA cells, so for data transmission these cells depends on the coulomb's interaction process. As there is no direct connection between them so no leakage is present, which results very less power requirement. Moreover, to design any circuit, majority gates and inverters are the basic building blocks [12]. Two types of majority gates are reported for QCA circuits, three input and five input majority gates. Two input AND gate and OR gate can be designed using the three input majority gate. The logic of three input majority gate is  $M(A, B, C) = AB + BC + CA$ . QCA based majority gate, AND gate and OR gate implementations are shown in Fig. 1c. Figure 1(d) shows the QCA based inverter design.

## 2.2 Reversible gate overview

To reduce the heat energy dissipation due to the loss of each data bit, the circuits are designed using reversible logic. To realize the reversible logic, reversible gates are designed which have bijective relation between input and output, that means each of the output of the logic gate can be determined from input and vice versa. To produce same number of input and output, it generates some additional inputs and some garbage outputs. The performance of the reversible gate is analysed in terms of quantum cost, latency, number of garbage outputs etc. Different types of reversible logic gates have already designed namely, Feynman gate [22, 23], Toffoli gate [24], Peres gate [25], Fredkin gate [26] etc. Figure. 2 shows the block diagram of the following reversible gates. Feynman gate is a  $2 \times 2$  reversible gate, that means it has 2 inputs and 2 outputs with bijective relation. The block diagram shows that A and B are 2



**Fig. 2** Four different reversible gates **a** Feynman gate [22, 23], **b** Toffoli gate [24], **c** Peres gate [25], and **d** Fredkin gate [26]

inputs of Feynman gate and  $P$  and  $Q$  are 2 outputs, where  $P = A$  and  $Q = A \oplus B$ . Peres gate is a  $3 \times 3$  reversible gate.  $A$ ,  $B$  and  $C$  are three inputs of Peres gate and  $P$ ,  $Q$  and  $R$  are three outputs as shown in the block diagram in Fig. 2, where  $P = A$ ,  $Q = A \oplus B$  and  $R = AB \oplus C$ . Feynman gate and Peres gate are used to design four code converter circuits in our reported design. Here, Peres gate is analysed using 2 Feynman gate. Figure 3 shows the block diagram of Peres gate that is designed using Feynman gates. The code converter circuits binary-to-gray code converter and gray-to-binary code converters are designed by using Feynman gates. Binary-to-excess-3 code converter and excess-3-to-binary code converters are designed using Peres gate, inverters and majority gates.

### 3 Proposed code-converters designs

#### 3.1 Binary-to-gray code converter

Digital systems have different types of code for data transmission. Binary code is the easiest and mostly used code in digital system. But in binary number system when the value of each binary number is increased by one, then one or multiple bits are changed from 1 to 0 or 0–1. More number of transition of bits causes more energy dissipation. If the binary code is converted to gray code, then energy dissipation can be reduced. Because in gray code, while traversing from one step to another step only one bit in that code number changes. That means two adjacent code numbers differ from each other by only one bit in gray code this results lower energy consumption. The logic circuit used to convert the binary code to gray code, is known

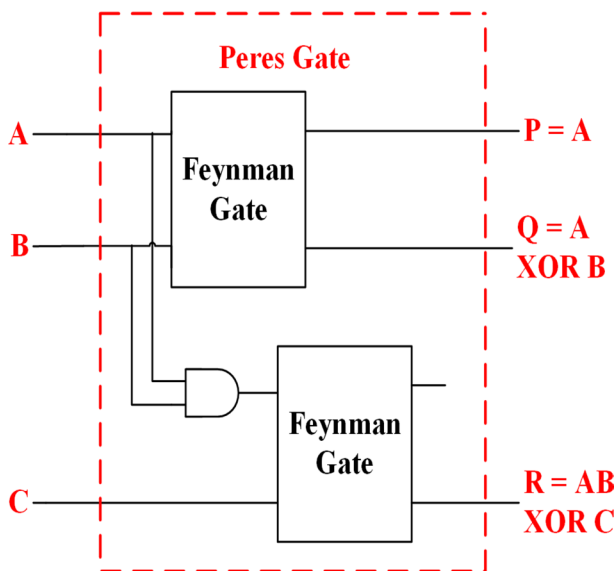


Fig. 3 QCA cell based Peres gate using Feynman gate

as binary-to-gray code converter. These code converter circuits are not only used to reduce the energy dissipation, this binary to gray code converter is also used in digital communication for error correcting. This binary-to-gray code conversion technique has following steps:

1. The MSB (Most significant bit) of the gray code is same as the MSB of the converting binary code.
2. If we assume the MSB is the first bit, then the second bit of the gray code will be the XOR of first and second bit of binary code.
3. The third bit of the gray code will be the XOR of second and third bit of the binary code. Thus the conversion process goes on.

In the present work, we have designed the binary-to-gray code converter of 4-bit data. The truth table of 4-bit binary-to-gray code converter is shown in Table 1. B0, B1, B2 and B3 are four binary input bits and G0, G1, G2 and G3 are four corresponding gray code output bits. Figure 4 depicts the block diagram of binary-to-gray code converter using Feynman reversible gates. The logic expressions for the gray code output bits are:

$$G3 = B3 \quad (1)$$

$$G2 = B3 \oplus B2 \quad (2)$$

$$G1 = B2 \oplus B1 \quad (3)$$

$$G0 = B1 \oplus B0 \quad (4)$$

Here  $\oplus$  represents the XOR gate. To design this 4-bit reversible binary-to-gray code converter circuit three Feynman reversible logic gates are required. The QCA based design, input/output waveforms and bus layout of this circuit are shown in Figs. 5, 6 and 7 respectively.

### 3.2 Gray-to-binary code converter

As it is evident that the binary number system is the simplest number system, so in digital system all other codes are finally converted to binary system for better

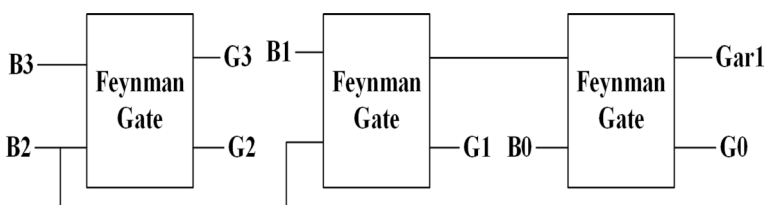


Fig. 4 Block diagram of reported reversible binary-to-gray code converter

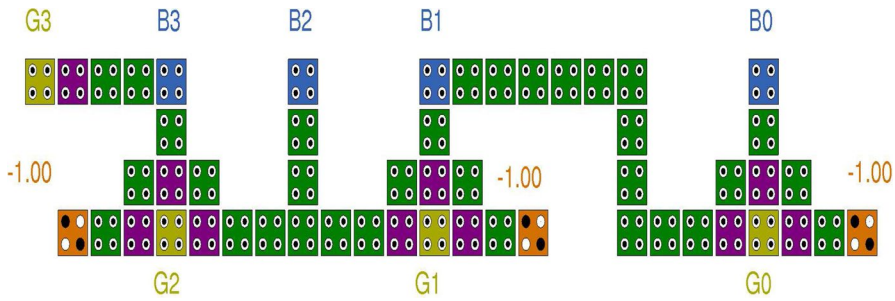


Fig. 5 QCA cell based reported reversible binary-to-gray code converter

understanding. Moreover, the binary to decimal number conversion is very easy as compared to other code conversion. For some purposes gray codes are also useful but to get the final digital output data, these gray codes need to be converted to binary code. Gray-to-binary code conversion requires some easy and simple steps as follows:

1. MSB of the equivalent binary number is same as the MSB of the given gray code.
2. If we consider MSB as the first bit of both the code, then the second bit of the binary number will be the XOR of MSB of Binary number and second bit of the gray code.
3. The third bit of the binary number will be the XOR of second bit of the binary code and third bit of the gray code. This process goes on to calculate the other bits of the binary number.

Here, a 4-bit QCA based gray-to-binary converter is designed using Feynman reversible gate. The truth table of this converter is shown in Table 2. A detailed block diagram of this converter circuit is designed in Fig. 8. Here, G0, G1, G2, G3 are input gray code bits and B0, B1, B2, B3 are output binary bits. The logic expressions for the binary output bit are:

$$B3 = G3 \tag{5}$$

$$B2 = B3 \oplus G2 \tag{6}$$

$$B1 = B2 \oplus G1 \tag{7}$$

$$B0 = B1 \oplus G0 \tag{8}$$

Where  $\oplus$  represents the XOR gate. Binary output bits are achieved from the gray input bits by cascading three reversible Feynman gates. Figures 9, 10 and 11 show the QCA based gray-to-binary code converter circuit, its input/output waveforms and bus layout respectively.



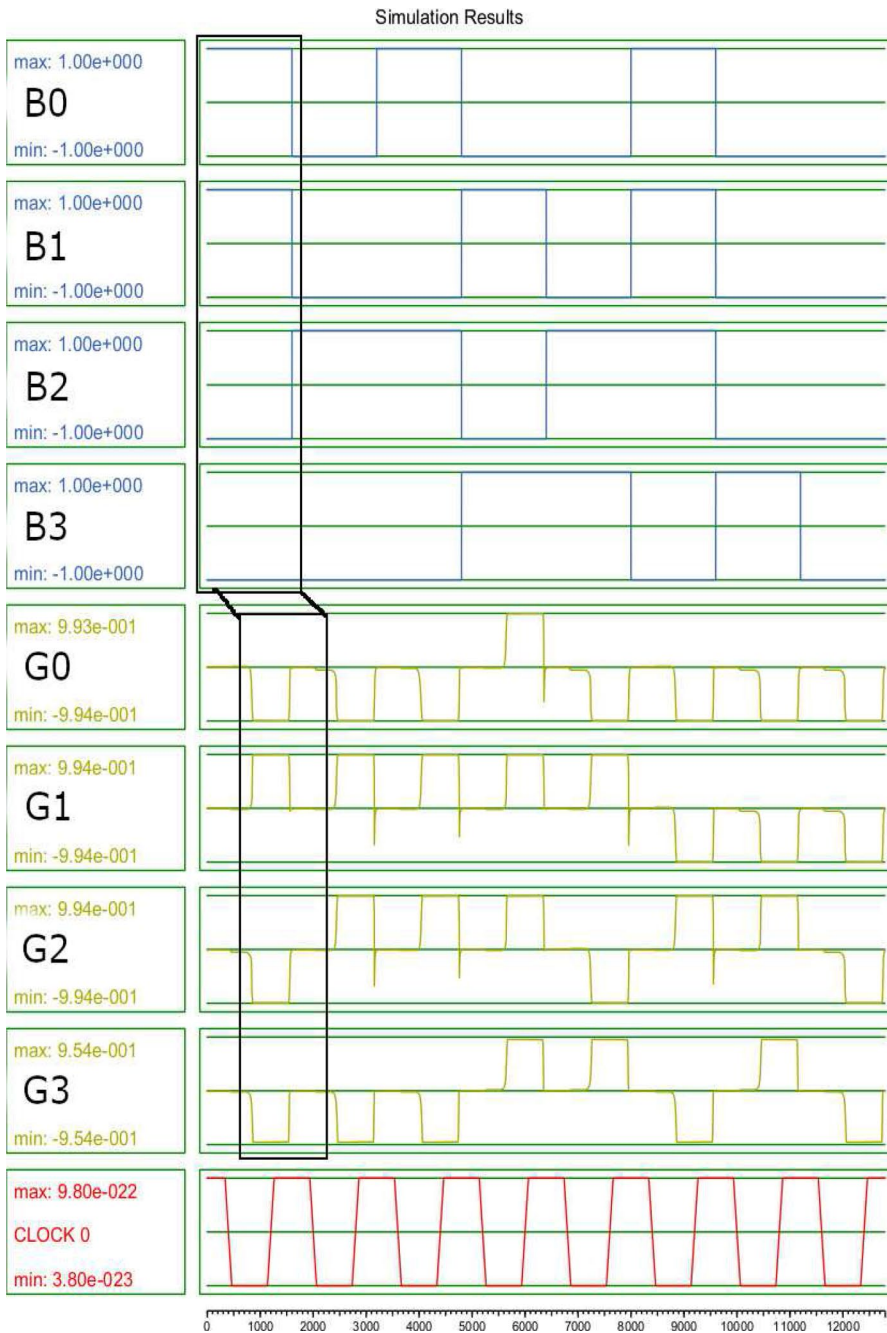


Fig. 6 Input/output waveform of reversible binary-to-gray code converter

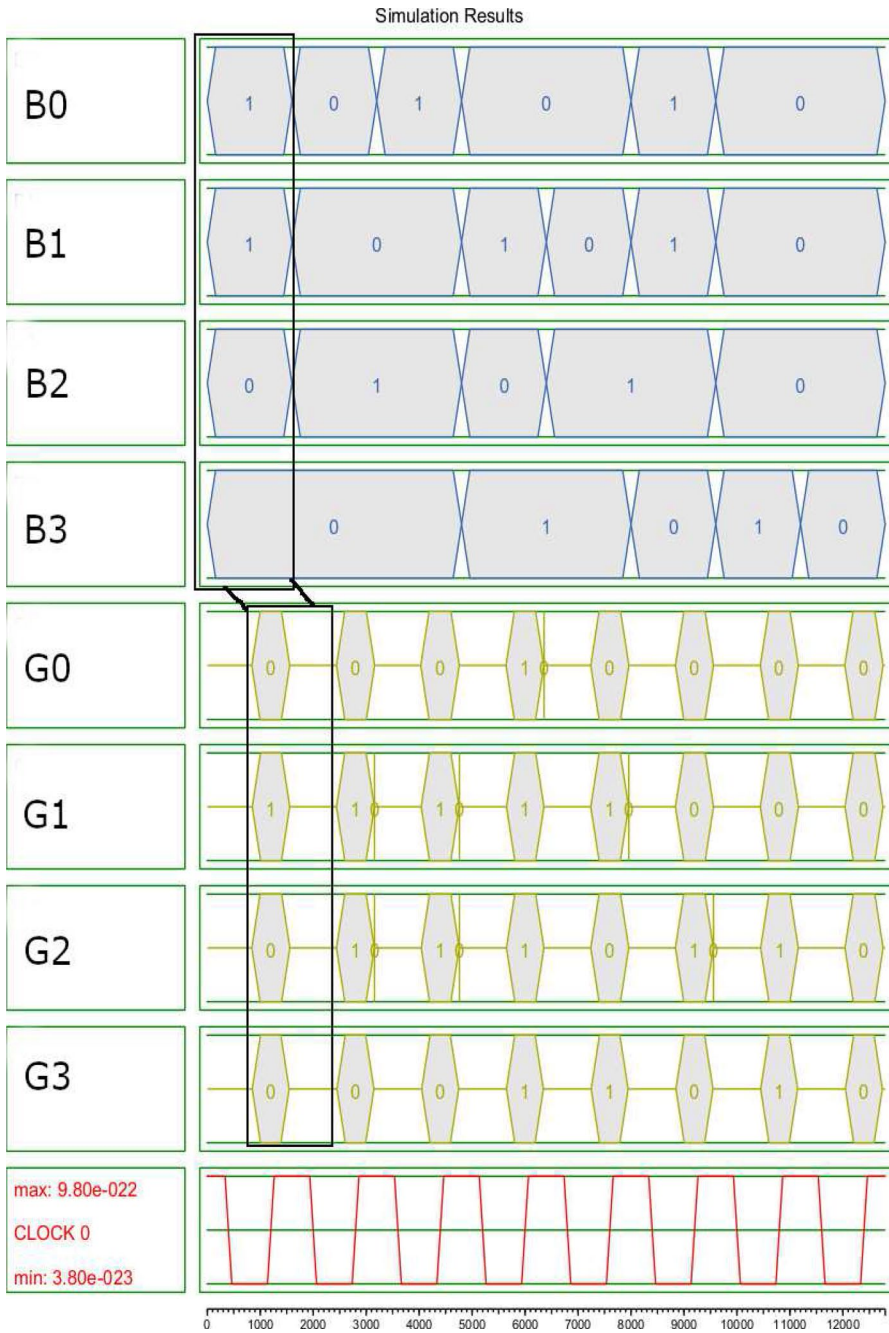


Fig. 7 Bus layout of reversible binary-to-gray code converter

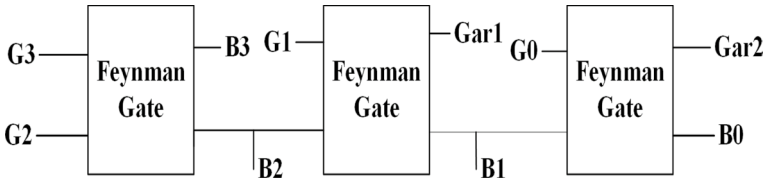


Fig. 8 Block diagram of reversible gray-to-binary code converter

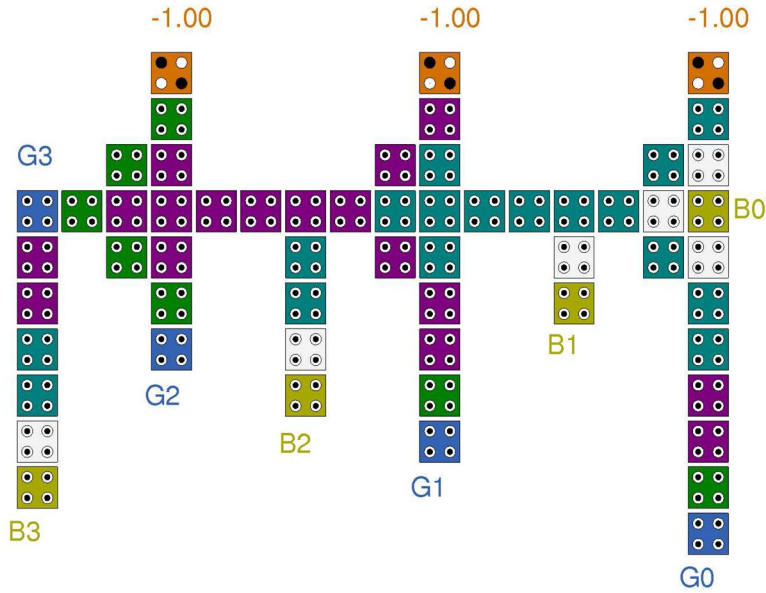


Fig. 9 QCA cell based reversible gray-to-binary code converter

### 3.3 Binary-to-excess-3 code converter

In digital number system excess-3 code is an another type of digital code like binary code and gray code. Excess-3 code is an unweighted self-complementary binary-coded-decimal (BCD) code. If we consider one decimal code, then the 1's complement of the excess-3 code of that decimal number will be equal to the excess-3 code of the 9's complement of the corresponding decimal number, this property of excess-3 code is known as self-complementary property. That means, the 9's complement calculation of decimal number system is equivalent to the 1's complement calculation of the excess-3 number system. For this property, excess-3 code is very useful for calculating the complement of a binary number or for subtraction purpose. The conversion of binary code to excess-3 code is very simple; it requires just to add binary 3 with the converting binary number. Here, we considered only 4-bit binary-to-excess-3 converter circuit. The truth table of

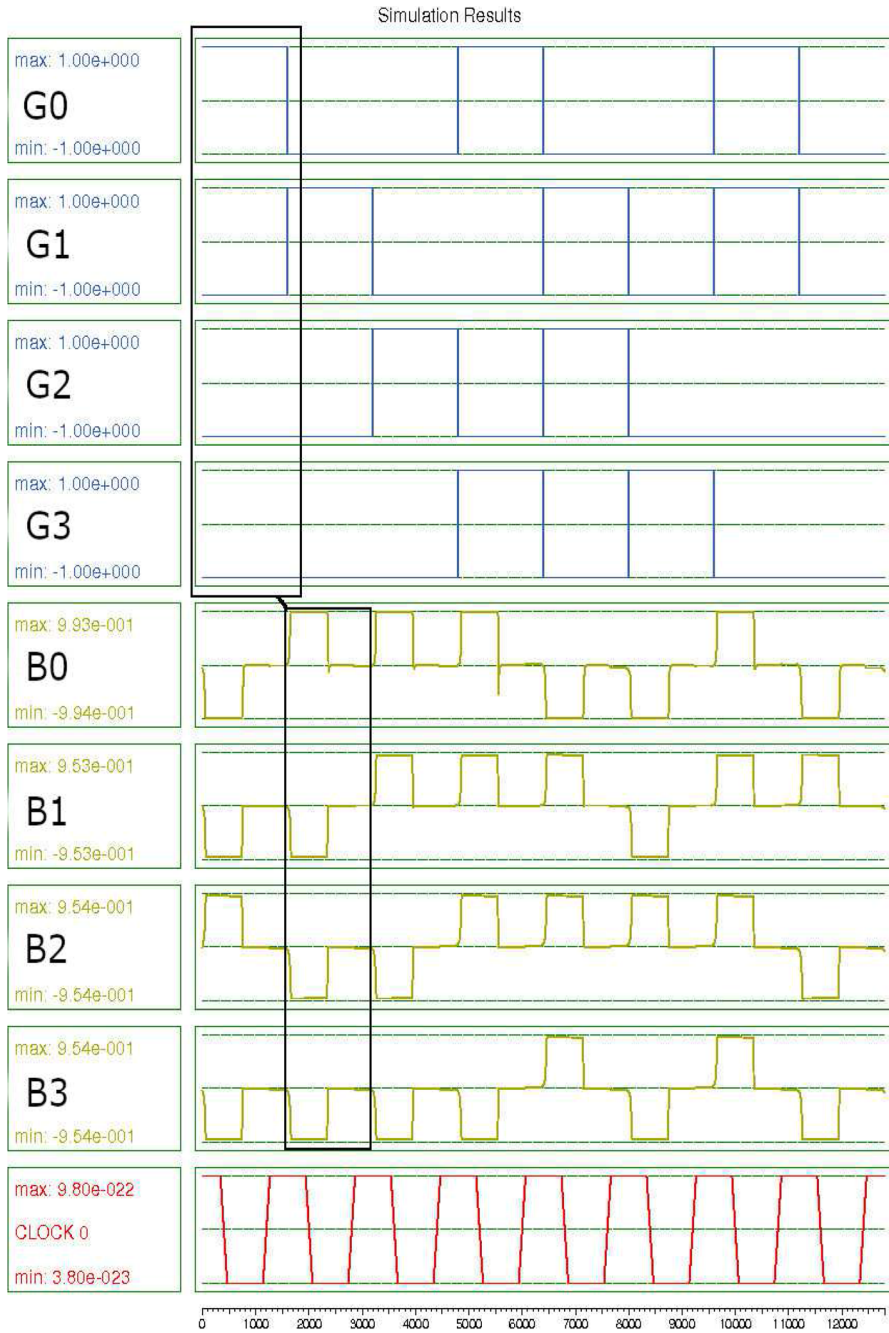


Fig. 10 Input/output waveform of reversible gray-to-binary code converter

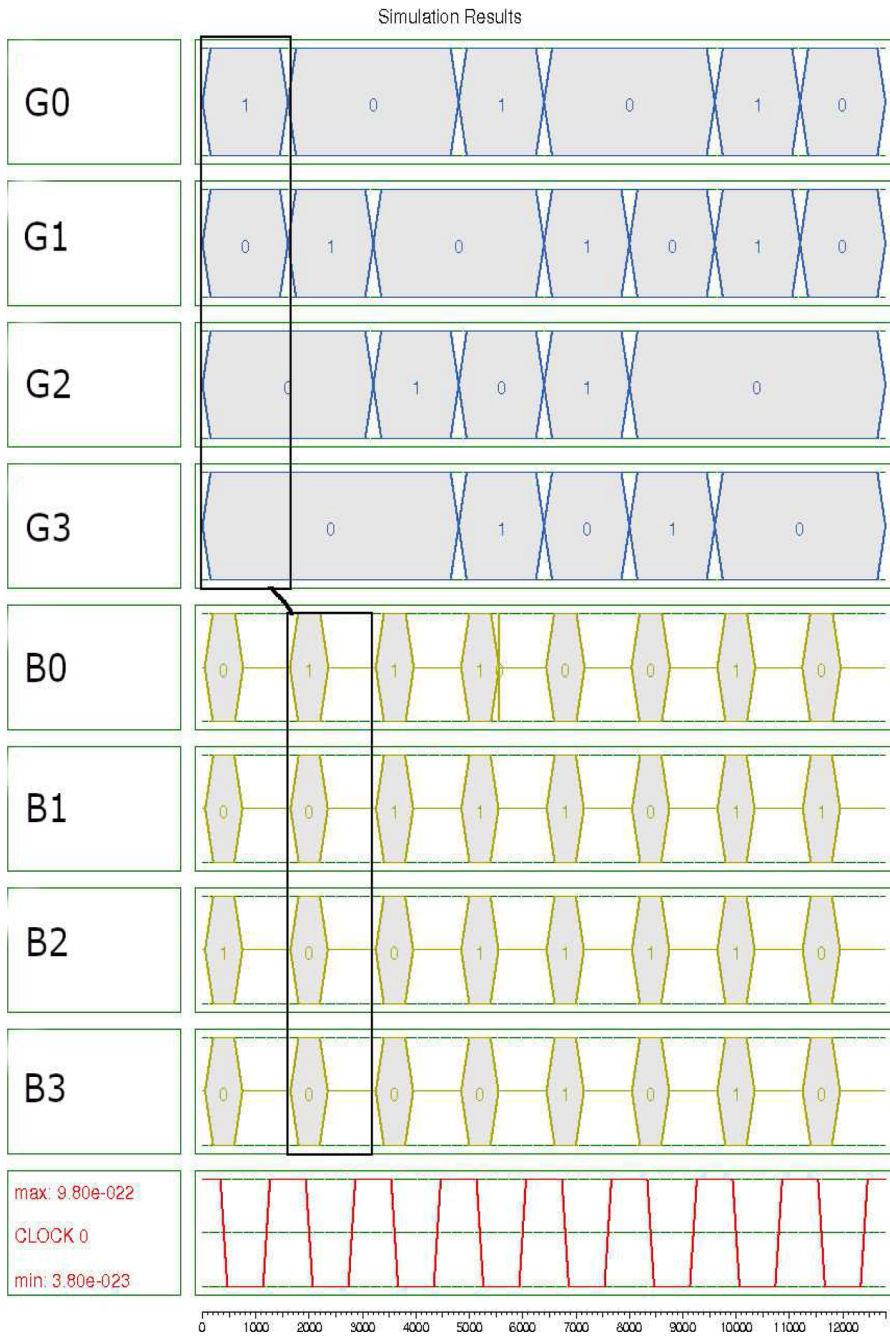


Fig. 11 Bus layout of reversible gray-to-binary code converter

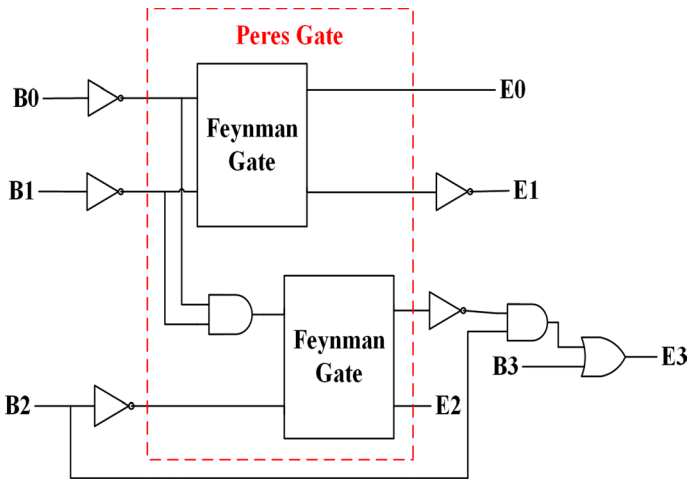


Fig. 12 Block diagram of reversible binary-to-excess-3 code converter

Table 1 Truth table of binary-to-gray code converter

| B3 | B2 | B1 | B0 | G3 | G2 | G1 | G0 |
|----|----|----|----|----|----|----|----|
| 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0  | 0  | 0  | 1  | 0  | 0  | 0  | 1  |
| 0  | 0  | 1  | 0  | 0  | 0  | 1  | 1  |
| 0  | 0  | 1  | 1  | 0  | 0  | 1  | 0  |
| 0  | 1  | 0  | 0  | 0  | 1  | 1  | 0  |
| 0  | 1  | 0  | 1  | 0  | 1  | 1  | 1  |
| 0  | 1  | 1  | 0  | 0  | 1  | 0  | 1  |
| 0  | 1  | 1  | 1  | 0  | 1  | 0  | 0  |
| 1  | 0  | 0  | 0  | 1  | 1  | 0  | 0  |
| 1  | 0  | 0  | 1  | 1  | 1  | 0  | 1  |
| 1  | 0  | 1  | 0  | 1  | 1  | 1  | 1  |
| 1  | 0  | 1  | 1  | 1  | 1  | 1  | 0  |
| 1  | 1  | 0  | 0  | 1  | 0  | 1  | 0  |
| 1  | 1  | 0  | 1  | 1  | 0  | 1  | 1  |
| 1  | 1  | 1  | 0  | 1  | 0  | 0  | 1  |
| 1  | 1  | 1  | 1  | 1  | 0  | 0  | 0  |

binary-to-excess-3 code converter is mentioned in Table 3. B0, B1, B2, B3 are four input binary bits and E0, E1, E2, E3 are four output excess-3 bits.

The logic expression for the output excess-3 bits can be drawn as:

$$E3 = B2 \cdot (B0 + B1) + B3 \tag{9}$$

$$E2 = \overline{B2} \cdot (B0 + B1) + B2 \cdot \overline{B0} \cdot \overline{B1} \tag{10}$$

**Table 2** Truth table of gray-to-binary code converter

| G3 | G2 | G1 | G0 | B3 | B2 | B1 | B0 |
|----|----|----|----|----|----|----|----|
| 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0  | 0  | 0  | 1  | 0  | 0  | 0  | 1  |
| 0  | 0  | 1  | 0  | 0  | 0  | 1  | 1  |
| 0  | 0  | 1  | 1  | 0  | 0  | 1  | 0  |
| 0  | 1  | 0  | 0  | 0  | 1  | 1  | 1  |
| 0  | 1  | 0  | 1  | 0  | 1  | 1  | 0  |
| 0  | 1  | 1  | 0  | 0  | 1  | 0  | 0  |
| 0  | 1  | 1  | 1  | 0  | 1  | 0  | 1  |
| 1  | 0  | 0  | 0  | 1  | 1  | 1  | 1  |
| 1  | 0  | 0  | 1  | 1  | 1  | 1  | 0  |
| 1  | 0  | 1  | 0  | 1  | 1  | 0  | 0  |
| 1  | 0  | 1  | 1  | 1  | 1  | 0  | 1  |
| 1  | 1  | 0  | 0  | 1  | 0  | 0  | 0  |
| 1  | 1  | 0  | 1  | 1  | 0  | 0  | 1  |
| 1  | 1  | 1  | 0  | 1  | 0  | 1  | 1  |
| 1  | 1  | 1  | 1  | 1  | 0  | 1  | 0  |

**Table 3** Truth table of binary-to-excess-3 code converter

| B3 | B2 | B1 | B0 | E3 | E2 | E1 | E0 |
|----|----|----|----|----|----|----|----|
| 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  |
| 0  | 0  | 0  | 1  | 0  | 1  | 0  | 0  |
| 0  | 0  | 1  | 0  | 0  | 1  | 0  | 1  |
| 0  | 0  | 1  | 1  | 0  | 1  | 1  | 0  |
| 0  | 1  | 0  | 0  | 0  | 1  | 1  | 1  |
| 0  | 1  | 0  | 1  | 1  | 0  | 0  | 0  |
| 0  | 1  | 1  | 0  | 1  | 0  | 0  | 1  |
| 0  | 1  | 1  | 1  | 1  | 0  | 1  | 0  |
| 1  | 0  | 0  | 0  | 1  | 0  | 1  | 1  |
| 1  | 0  | 0  | 1  | 1  | 1  | 0  | 0  |
| 1  | 0  | 1  | 0  | 1  | 1  | 0  | 1  |
| 1  | 0  | 1  | 1  | 1  | 1  | 1  | 0  |
| 1  | 1  | 0  | 0  | 1  | 1  | 1  | 1  |
| 1  | 1  | 0  | 1  | X  | X  | X  | X  |
| 1  | 1  | 1  | 0  | X  | X  | X  | X  |
| 1  | 1  | 1  | 1  | X  | X  | X  | X  |

$$E1 = B0 \odot B1 \quad (11)$$

$$E0 = \overline{B0} \quad (12)$$

Where,  $\bullet$  denotes the logical AND operation,  $+$  denotes the logical OR operation and  $\odot$  denotes logical XNOR operation. Including one Peres reversible gate, this

circuit consists of five inverters, one AND gate and one OR gate. Again this Peres gate is designed using two Feynman gates and one AND gate. Figure 12 shows the block diagram of the circuit and in Figs. 13, 14 and 15 the QCA based binary-to-excess-3 code converter circuit, input/output waveforms and bus layout of this circuit are shown respectively.

### 3.4 Excess-3-to-binary code converter

To get the final output data all the codes in digital number system are needed to convert in binary code. Excess-3 code is converted to binary code by subtracting binary 3 from the corresponding excess-3 code. After the mathematical operation the excess-3 code is converted to the binary code to get final output data, so the excess-3-to-binary code converter is needed to be designed. In current work, a 4-bit excess-3-to-binary converter is considered. The truth table for this following converter is shown in Table 4. The proposed block diagram for this converter circuit is designed in Fig. 16, where E0, E1, E2, E3 are four input excess-3 bits and B0, B1, B2, B3 are four output binary bits. The logic expression for converted output binary bits should be:

$$B3 = E3 \cdot (E2 + E1 \cdot E0) \tag{13}$$

$$B2 = \overline{E2} \cdot (\overline{E1} + \overline{E0}) + E0 \cdot E1 \cdot E2 \tag{14}$$

$$B1 = E1 \oplus E0 \tag{15}$$

**Table 4** Truth table of Excess-3-to-binary code converter

| E3 | E2 | E1 | E0 | B3 | B2 | B1 | B0 |
|----|----|----|----|----|----|----|----|
| 0  | 0  | 0  | 0  | X  | X  | X  | X  |
| 0  | 0  | 0  | 1  | X  | X  | X  | X  |
| 0  | 0  | 1  | 0  | X  | X  | X  | X  |
| 0  | 0  | 1  | 1  | 0  | 0  | 0  | 0  |
| 0  | 1  | 0  | 0  | 0  | 0  | 0  | 1  |
| 0  | 1  | 0  | 1  | 0  | 0  | 1  | 0  |
| 0  | 1  | 1  | 0  | 0  | 0  | 1  | 1  |
| 0  | 1  | 1  | 1  | 0  | 1  | 0  | 0  |
| 1  | 0  | 0  | 0  | 0  | 1  | 0  | 1  |
| 1  | 0  | 0  | 1  | 0  | 1  | 1  | 0  |
| 1  | 0  | 1  | 0  | 0  | 1  | 1  | 1  |
| 1  | 0  | 1  | 1  | 1  | 0  | 0  | 0  |
| 1  | 1  | 0  | 0  | 1  | 0  | 0  | 1  |
| 1  | 1  | 0  | 1  | 1  | 0  | 1  | 0  |
| 1  | 1  | 1  | 0  | 1  | 0  | 1  | 1  |
| 1  | 1  | 1  | 1  | 1  | 1  | 0  | 0  |



$$B0 = \overline{E0} \quad (16)$$

Here,  $\bullet$  is the AND operation,  $+$  is the OR operation and  $\oplus$  is the logical XOR operation. To design this circuit one Peres gate is used along with two inverters, an AND gate and an OR gate. This Peres gate is designed using two Feynman gate and one AND gate. QCA based converter circuit, input/output waveforms and bus layout for the excess-3-to-binary code converter are shown in Figs. 17, 18 and 19 respectively.

## 4 Simulation framework

The QCA based code converter circuits are designed using QCA Designer tool v 2.0.3 to analyse the circuit complexity, area requirement, latency of the circuits, quantum cost, robustness of the circuits etc. To check the accuracy of the proposed design coherence vector simulation engine is used. The simulation parameters of this engine are mentioned in Table 5. By using this simulation engine the circuits operation in form of input/output waveforms and bus layout can be obtained. All these proposed circuits consist of reversible logic gates, so it consumes less energy. The energy dissipation of these circuits are analysed in an extension of QCA Designer tool, QCA Designer-E tool.

## 5 Results and discussion

### 5.1 Circuit complexity

The circuit complexity of a QCA based circuit includes cell count, total area, cell area, area coverage and latency of the circuit. Here, the circuit complexity of Feynman gate, Peres gate, binary-to gray code converter, gray-to-binary code converter,

**Table 5** The coherence vector parameters considered for the circuit realization in QCA designer tool v2.0.3

| Parameters             | Coherence vector considered |
|------------------------|-----------------------------|
| Cell size              | 18 nm × 18 nm               |
| Dot diameter           | 5 nm                        |
| Radius effect          | 80 nm                       |
| Relative permittivity  | 12.9                        |
| Clock high             | $9.8 \times 10^{-22}$ J     |
| Clock low              | $3.8 \times 10^{-23}$ J     |
| Clock amplitude factor | 2.0000                      |
| Relaxation time        | $4.135 \times 10^{-14}$ s   |
| Time step              | $7 \times 10^{-16}$ s       |
| Layer separation       | 11.5 nm                     |
| Cell separation        | 2 nm                        |

**Table 6** Complexity analysis of proposed circuits and performance comparison with previous literature

| QCA circuit   | Cell count | Total area<br>( $\mu\text{m}^2$ ) | Cell area<br>( $\mu\text{m}^2$ ) | Area<br>coverage | Latency<br>(clock periods) |
|---|------------|-----------------------------------|----------------------------------|------------------|----------------------------|
| Feynman Gate [23]   | 14         | 0.0045                            | 0.02                             | 22.5%            | 0.5                        |
| 4-bit irreversible Binary-to-gray code converter [37]     | 126        | 0.04                              | 0.15                             | 26.6%            | 0.75                       |
| 3-bit reversible Binary-to-gray code converter [38]       | 118        | 0.038                             | 0.092                            | 41%              | 0.75                       |
| 4-bit reversible Binary-to-gray code converter [39]       | 108        | 0.034                             | 0.11                             | 30.9%            | 0.75                       |
| Proposed Binary-to-gray code converter                    | 53         | 0.017                             | 0.07                             | 24.285%          | 0.5                        |
| 3-bit reversible Gray-to-binary code converter [38]       | 112        | 0.036                             | 0.139                            | 25.82 %          | 3                          |
| 4-bit reversible Gray-to-binary code converter [39]       | 77         | 0.024                             | 0.1                              | 24%              | 0.75                       |
| Proposed Gray-to-binary code converter                    | 58         | 0.018                             | 0.09                             | 20%              | 1                          |
| 4-bit irreversible Binary-to-excess-3 code converter [40] | 413        | 0.133                             | –                                | –                | 8                          |
| 4-bit irreversible Binary-to-excess-3 code converter [18] | 143        | 0.046                             | 0.22                             | 21.1%            | 6                          |
| Proposed Binary-to-excess-3 code converter                | 110        | 0.035                             | 0.17                             | 20.588%          | 1.5                        |
| Proposed Excess-3-to-binary code converter                | 98         | 0.031                             | 0.17                             | 18.235%          | 1.5                        |

binary-to-excess-3 converter and excess-3-to-binary code converter circuits are analysed. The circuit complexity of all the above mentioned circuits are mentioned in Table 6. The major circuit complexity is the cell count, which is the number of QCA cells required to design any circuit. The reversible code converters are designed using Feynman reversible gate. This gate consists of 14 QCA cells. Thus the minimum cell count of code converter circuit is 53 and maximum cell count is 110 for 4-bit codes. By comparing the cell count of the 4-bit binary-to-gray code converter with the published literature [39] this work reports 50.9% reduction. Moreover, the proposed circuit requires even much less number of cells than the 3-bit binary-to-gray code converter in published literature [38]. Cell count of gray-to-binary code converter also reduced by 25% as compared to [39]. Cell requirement of binary-to-gray code converter also reduce by 23% than [18]. Next parameter of circuit complexity is cell area. This parameter is important to calculate the area usage of any circuit. Cell area can be calculated by multiplying the number of cells with the area of the cell ( $18\text{nm} \times 18\text{nm}$ ). That implies the circuit which has lesser cell count, that requires lesser cell area. In case of binary-to-gray code converter the cell area is also reduced by 50%. Total area of the circuit is the minimum rectangular area within which the QCA cell based circuit can be designed. It is required to determine the circuit density in a chip. This total area requirement depends on the arrangement of QCA cells to design any circuits. In proposed circuits, the minimum total area requirement is  $0.07\text{um}^2$  for binary-to-gray code converter. In both the circuits,

binary-to-excess-3 and excess-3-to-binary converter the total area requirement is same though these two circuits have different cell count. As the cell number is reduced, so the total area also reduces by significant amount as compared to the previous literature. Area coverage of any circuit is determined from the ratio of cell area and total area of that circuit. This parameter is important to analyse the area of a chip is used for QCA cells. Latency denotes the delay arises in the circuit between input and output. The main objective of a circuit is to reduce the delay for fast response. The minimum and maximum latency of the proposed circuits are 0.5 s and 1.5 s, respectively. The latency of all the proposed circuits are less as compared to published literature. The detailed comparison of circuit complexity of all the proposed code converter circuits with the previous published literature are shown in Table 6.

## 5.2 Quantum cost

Quantum cost of the circuit is calculated by multiplying the total area of the circuit with the latency. This results represent the efficiency of a circuit in terms of area and delay. Here, the reported maximum quantum cost is 0.255 and minimum quantum cost is 0.035. Moreover, QCA circuit cost of any circuit is the product of quantum cost and latency. The QCA circuit cost is required to be as minimum as possible. Garbage output calculation is another important factor for QCA reversible logic gate based circuits. In the proposed circuit the maximum number of garbage output generated is 2. Thus it is clear that the circuits are designed with minimum number of garbage outputs. Table 7 mentions the quantum cost, QCA circuit cost and number of garbage outputs of all the proposed code converter circuits.

## 5.3 Energy dissipation

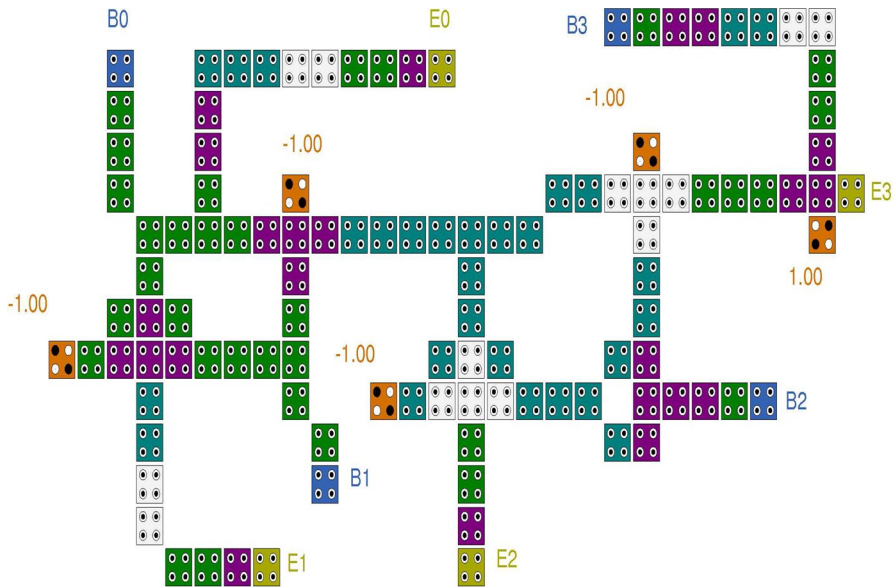
Energy calculation of a circuit is a very important factor. The circuits should dissipate minimum energy for better performance. It is theoretically proven that irreversible logic circuits dissipates more energy as compared to reversible logic circuits, so researchers pay more attention to design logic circuits using reversible gates [43–45]. Here also reversible logic is used to design the proposed circuits to minimize the energy requirement. For these four circuits the maximum total energy

**Table 7** Quantum cost analysis of proposed circuits

| Proposed Circuits                 | Quantum cost | QCA circuit cost<br>(total area $\times$ latency <sup>2</sup> ) | Garbage outputs | Number<br>of iterations |
|-----------------------------------|--------------|---|-----------------|-------------------------|
| Binary-to-gray code converter     | 0.035        | 0.0175  | 1               | 6                       |
| Gray-to-binary code converter     | 0.09         | 0.09  | 2               | 11                      |
| Binary-to-excess-3 code converter | 0.255        | 0.382   | 0               | 10                      |
| Excess-3-to-binary code converter | 0.255        | 0.382   | 0               | 10                      |

**Table 8** Energy efficiency analysis of proposed circuits

| Proposed Circuits                 | Total energy dissipation (eV) | Average energy dissipation per cycle (eV) |
|-----------------------------------|-------------------------------|---|
| Binary-to-gray code converter     | $4.13 \times 10^{-2}$         | $3.76 \times 10^{-3}$                     |
| Gray-to-binary code converter     | $4.21 \times 10^{-2}$         | $3.82 \times 10^{-3}$                     |
| Binary-to-excess-3 code converter | $3.79 \times 10^{-2}$         | $3.44 \times 10^{-3}$                     |
| Excess-3-to-binary code converter | $5.88 \times 10^{-2}$         | $5.35 \times 10^{-3}$                     |



**Fig. 13** QCA cell based reversible binary-to-excess-3 code converter

dissipation is  $5.88 \times 10^{-2}$  eV and minimum energy dissipation is  $3.79 \times 10^{-2}$  eV. Table 8 shows the total energy dissipation and average energy dissipation per cycle of all the proposed reversible code converter circuits.

## 6 Conclusion

A compact, optimized, and area efficient quantum dot cellular automata (QCA) based reversible 4-bit binary-to-excess-3 code converter and reversible excess-3-to-binary code converter circuits are reported for the first time. Post this an area efficient binary-to-gray code converter and gray-to-binary code converter circuits are also designed. The area coverage of all the circuits are close to the ideal value of 34%, which verifies the compact designing. As compared to the previous

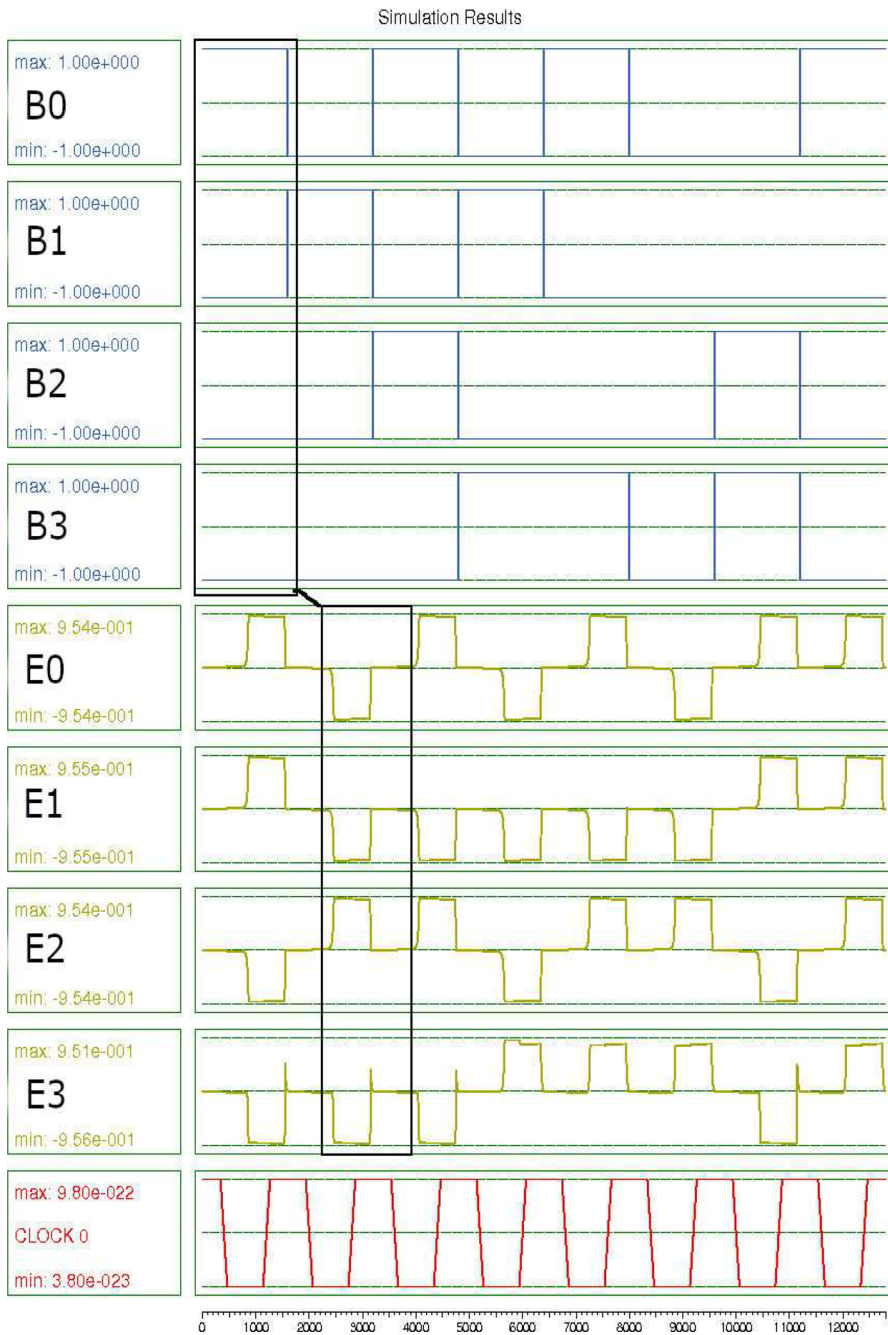


Fig. 14 Input/output waveform of reversible binary-to-excess-3 code converter

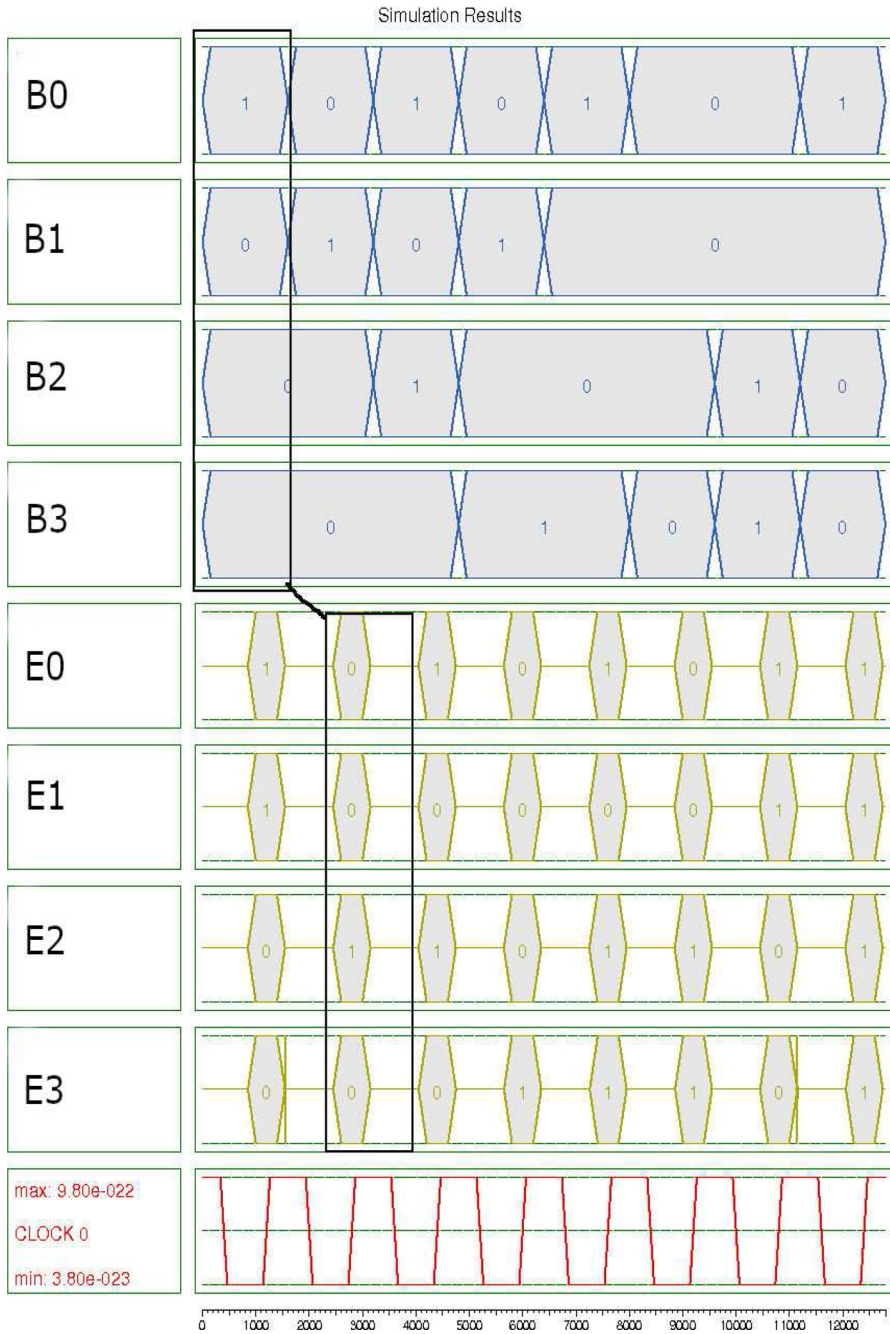


Fig. 15 Bus layout of reversible binary-to-excess-3 code converter

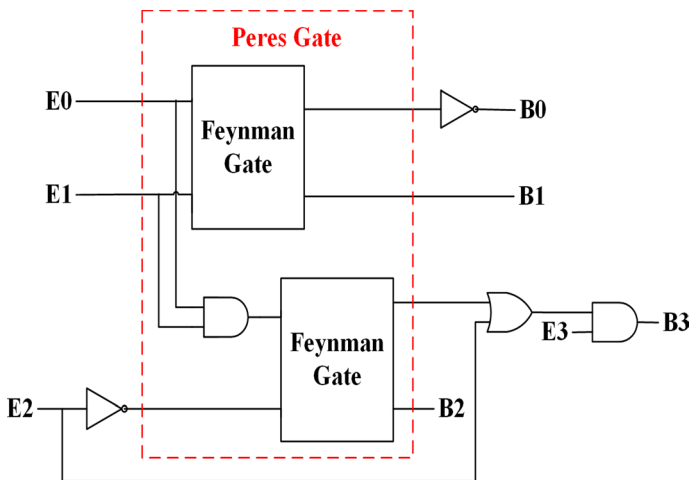


Fig. 16 Block diagram of reversible excess-3-to-binary code converter

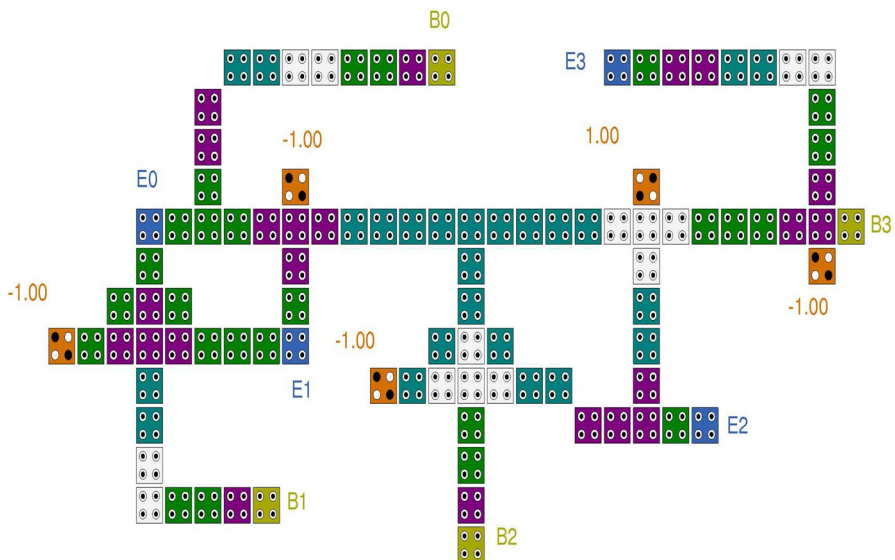


Fig. 17 QCA cell based reversible excess-3-to-binary code converter

literature the cell count and area of the circuits are reduced by 50% and latency of all the code converter circuits are also reduced by a significant amount. As the area and the latency are minimum thereby the quantum cost of the circuits are also minimum. Code converter circuits are based on reversible gates, which generates only two garbage outputs. Overall, these reversible code converter circuits can be most suited for quantum computing circuits in future.

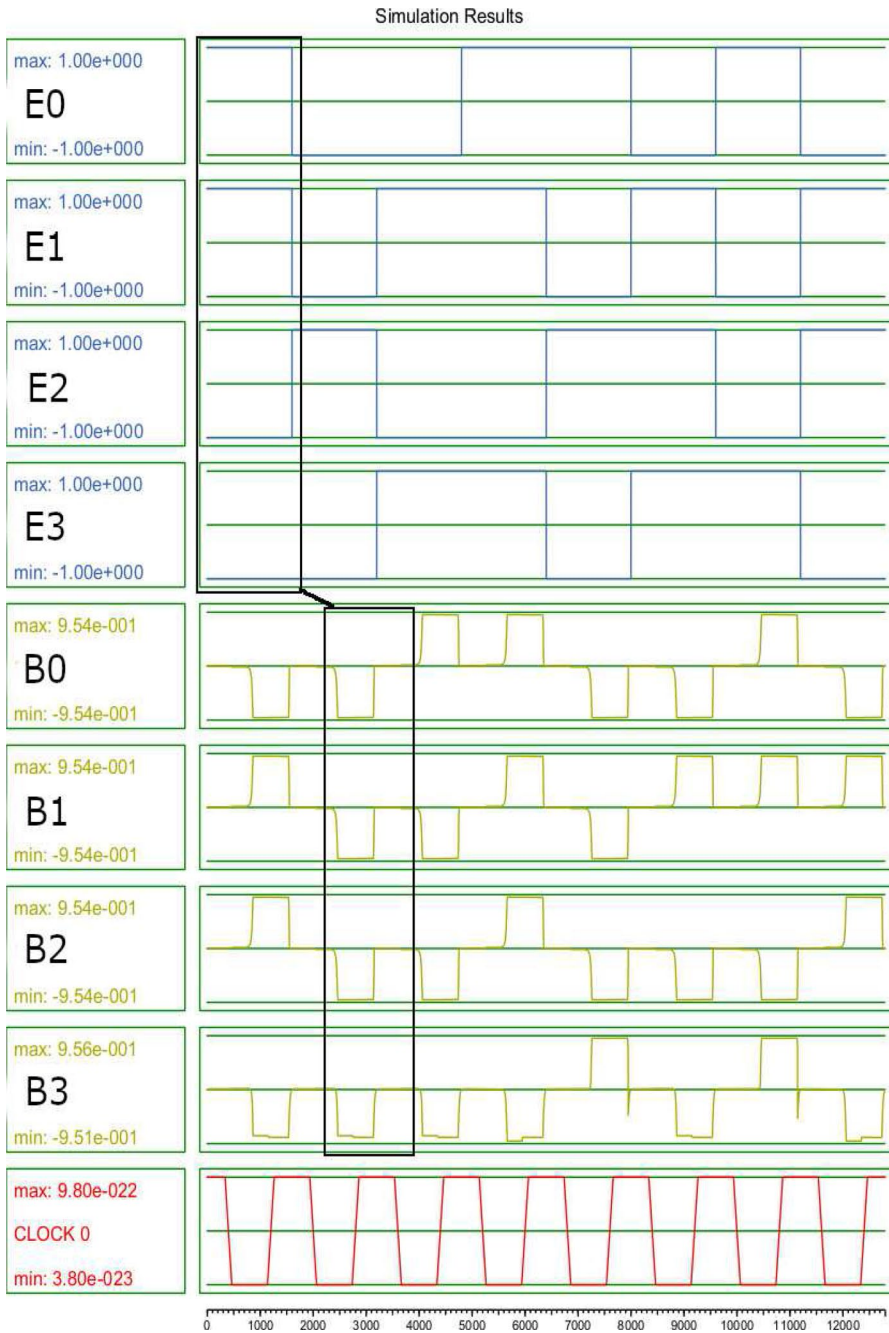


Fig. 18 Input/output waveform of reversible excess-3-to-binary code converter



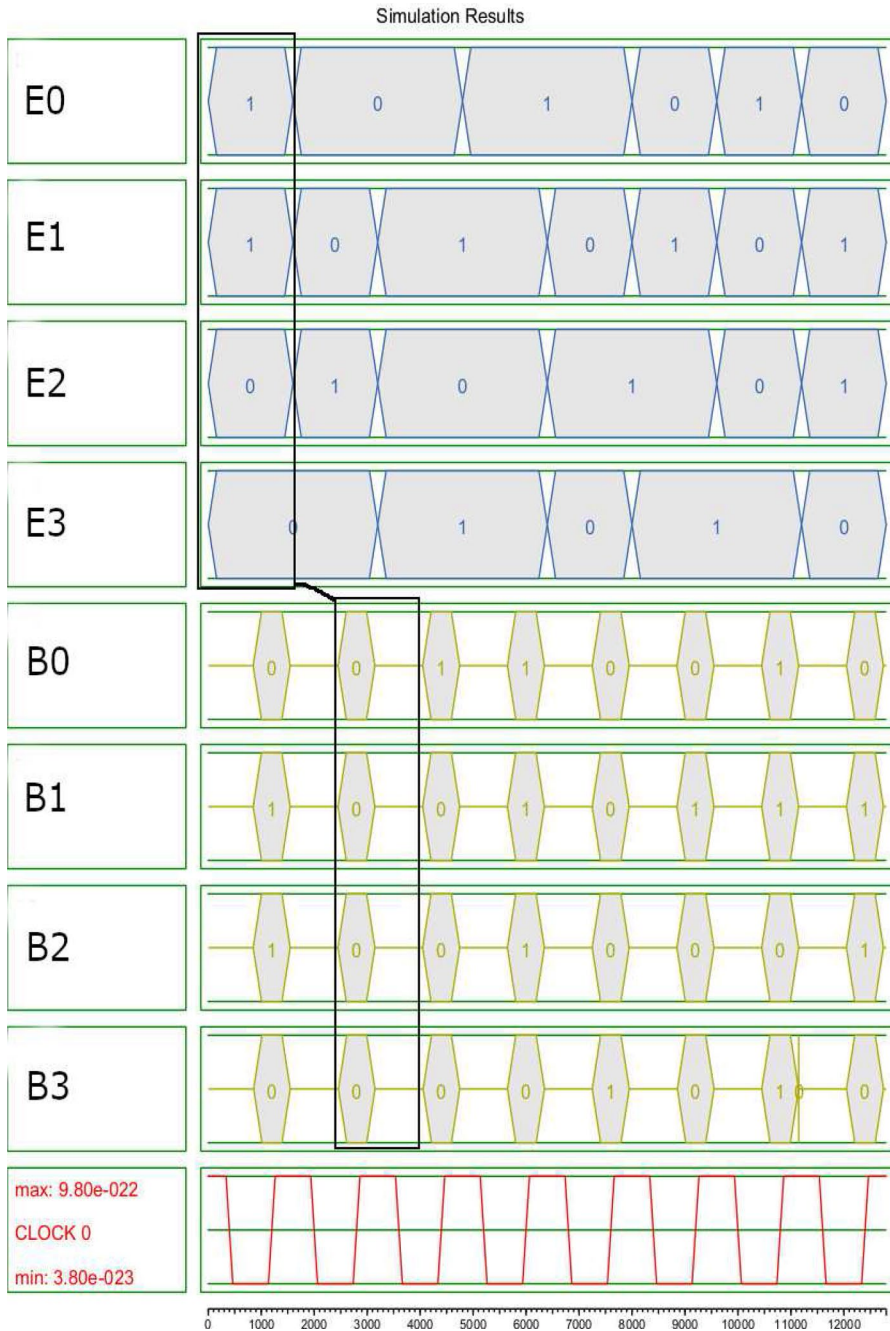


Fig. 19 Bus layout of reversible excess-3-to-binary code converter

## References

1. Shalf J (2020) The future of computing beyond Moore's law. *Philos Trans R Soc A* 378(2166):20190061
2. Yang JH, Li GF, Liu HL (2008) Edge direct tunneling current in nano- scale MOSFET with high- K dielectrics. *Microelectron Int* 25:30–33
3. Kaity A, Singh S, Kondekar PN (2020) Silicon-on-nothing electrostatically doped junctionless tunnel field effect transistor (SON-ED-JLTFET): a short channel effect resilient design. *Silicon* 6:1–15
4. Kastner MA (1992) The single-electron transistor. *Rev Mod Phys* 64(3):849
5. Ohshima T (1996) Stability of binary logic tunneling phase states in dc- biased and ac- pumped single-electron tunnel junctions. *Appl Phys Lett* 69(26):4059–4061
6. Javey A, Guo J, Wang Q, Lundstrom M, Dai H (2003) Ballistic carbon nanotube field-effect transistors. *Nature* 424(6949):654–657
7. Lent CS, Tougaw PD, Porod W, Bernstein GH (1993) Quantum cellular automata. *Nanotechnology* 4:49–57
8. Orlov AO, Amlani I, Bernstein GH, Lent CS, Sinder GL (1997) Realization of a functional cell for quantum dot cellular automata. *Science* 277:928–930
9. Porod W (1997) Quantum-dot devices and quantum-dot cellular automata. *J Frankl Inst* 334(5–6):1147–1175
10. Lent CS, Tougaw P (1997) A device architecture for computing with quantum dots. *Proc IEEE* 85(4):541–557
11. Porod W, Lent CS, Bernstein GH, Orlov AO, Hamlani I, Snider GL, Merz JL (1999) Quantum-dot cellular automata: computing with coupled quantum dots. *Int J Electron* 86(5):549–590
12. Tougaw PD, Lent CS (1994) Logical devices implemented using quantum cellular automata. *J Appl Phys* 75(3):1818–1825
13. Wang W, Walus K, Jullien GA (2003) August. Quantum-dot cellular automata adders. *Third IEEE Conf Nanotechnol* 1:461–464
14. Lakshmi SK, Athisha G, Karthikeyan M, Ganesh C (2010) October. Design of subtractor using nanotechnology based QCA. *International conference on communication control and computing technologies*, pp. 384–388
15. Hashemi S, Azghadi MR, Zakerolhosseini A (2008) A novel QCA multiplexer design. *International symposium on telecommunications*, pp. 692–696
16. Kianpour M, Sabbaghi-Nadooshan R (2011) December. A novel modular decoder implementation in quantum-dot cellular automata (QCA). *International conference on nanoscience, technology and societal implications*, pp. 1–5
17. Ghosh B, Gupta S, Kumari S, Salimath A (2013) Novel design of combinational and sequential logical structures in quantum dot cellular automata. *J Nanostruct Chem* 3(1):15
18. Ahmad F, Bhat G (2014) Novel code converters based on quantum-dot cellular automata (QCA). *Int J Sci Res* 3(5):364–371
19. Huang J, Momenzadeh M, Lombardi F (2007) Design of sequential circuits by quantum-dot cellular automata. *Microelectron J* 38(4–5):525–537
20. Landauer R (1961) Irreversibility and heat generation in the computing process. *IBM J Res Dev* 5:183–191
21. Bennett CH (1973) Logical reversibility of computation. *IBM J Res Dev* 17:525–532
22. Feynman RP (1985) Quantum mechanical computers. *Opt News* 11(2):11–20
23. Biswas PK, Bahar AN, Habib MdA, Shafi Md A-A, (2017) Efficient design of Feynman and Toffoli gate in quantum dot cellular automata (QCA) with energy dissipation analysis. *Nanosci Nanotechnol* 7(2):27–33
24. Toffoli T (1980) Reversible computing. *International colloquium on automata, languages, and programming*. Springer, Berlin, pp 632–644
25. Peres A (1985) Reversible logic and quantum computers. *Phys Rev A* 32(6):3266
26. Fredkin E, Toffoli T (1982) Conservative logic. *Int J theor Phys* 21(3–4):219–253
27. Das K, De D (2010) Characterization, test and logic synthesis of novel conservative and reversible logic gates for QCA. *Int J Nanosci* 9(3):201–214
28. Kumar P, Singh S (2019) Optimization of the area efficiency and robustness of a QCA-based reversible full adder. *J Comput Electron* 18(4):1478–1489

29. Thapliyal H, Ranganathan N (2009) Design of efficient reversible binary subtractors based on a new reversible gate. *IEEE computer society annual symposium on VLSI*, pp. 229–234
30. Sen B, Dutta M, Goswami M, Sikdar BK (2014) Modular design of testable reversible ALU by QCA multiplexer with increase in programmability. *Microelectron J* 45(11):1522–1532
31. Das JC, De D, Sadhu T (2016) A novel low power nanoscale reversible decoder using quantum-dot cellular automata for nanocommunication. *3rd international conference on devices, circuits and systems (ICDCS)*, pp. 220–224
32. Das JC, De D (2019) Novel design of reversible priority encoder in quantum dot cellular automata based on Toffoli gate and Feynman gate. *J Supercomput* 75:6882–6903
33. Thapliyal H, Ranganathan N (2010) Reversible logic-based concurrently testable latches for molecular QCA. *IEEE Trans Nanotechnol* 9(1):62–69
34. Thapliyal H, Ranganathan N, Kotiyal S (2013) Design of testable reversible sequential circuits. *IEEE Trans Very Large Scale Integr (VLSI) Syst* 21(7):1201–1209
35. Lu L, Liu W, O'Neill M, Swartzlander EE (2011) QCA systolic array design. *IEEE Trans Comput* 62(3):548–560
36. Hashemi S, Navi K (2014) Designing quantum-dot cellular automata circuits using a robust one layer crossover scheme. *J Eng* 3:93–97
37. Abdullah-Al-Shafi M, Bahar AN (2016) Novel binary to gray code converters in QCA with power dissipation analysis. *Int J Multimed Ubiquitous Eng* 11(8):379–396
38. Das JC, De D (2015) Reversible binary to grey and grey to binary code converter using QCA. *IETE J Res* 61(3):223–229
39. Karkaj ET, Heikalabad SR (2017) Binary to gray and gray to binary converter in quantum-dot cellular automata. *Optik* 130:981–989
40. Iqbal J, Khanday FA, Shah NA (2013) Efficient quantum dot cellular automata (QCA) implementation of code converters. *Commun Inf Sci Manag Eng* 3(10):504–515
41. Sasamal TN, Singh AK, Mohan A (2020) Quantum-dot cellular automata based digital logic circuits: a design perspective, *studies in computational intelligence*, vol 879. Springer, Singapore
42. Sasamal TN, Singh AK, Mohan A (2016) An optimal design of full adder based on 5-input majority gate in co-planar quantum-dot cellular automata. *Optik* 127(20):8576–8591
43. Bahar AN, Wahid KA (2019) Design of QCA-serial parallel multiplier (QSPM) with energy dissipation analysis. *IEEE Trans Circ Syst II Exp Briefs* 67(10):1939–1943
44. Chaves JF, Ribeiro MA, Silva LM, de Assis LM, Torres MS, Neto OPV (2018) Energy efficient QCA circuits design: simulating and analyzing partially reversible pipelines. *J Comput Electron* 17(1):479–489
45. Kaity A, Singh S, Hossain K (2020) Quantum dot cellular automata-based encoder and priority encoder circuits: low latency and area efficient design. *Int J Numer Model Electron Netw Devices Fields* 1:e2850

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.