Reversible logic-based image steganography using quantum dot cellular automata for secure nanocommunication

Bikash Debnath, Jadav Chandra Das\textsuperscript{\textcopyright}, Debashis De

Department of Computer Science and Engineering, West Bengal University of Technology, BF-142, Salt Lake, Sector I, Kolkata 700064, West Bengal, India
\textsuperscript{\textcopyright} E-mail: jadav2u@gmail.com

Abstract: This study introduces a novel architecture for image steganography using reversible logic based on quantum dot cellular automata (QCA). Feynman gate is used to achieve the reversible encoder and decoder for image steganography. A Nanocommunication circuit for image steganography is shown using proposed encoder/decoder circuit. The proposed QCA circuits have lower quantum cost than traditional designs. It shows the cost effectiveness functionality of the proposed designs. The proposed circuit has 28.33% improvement in terms of area over complementary metal–oxide–semiconductor circuit. To perform image steganography, LSB technique is used; signal-to-noise ratio (SNR), peak SNR and mean squared error (MSE) are also computed. The proposed QCA encoder/decoder circuit is suitable for reversible computing. To establish this, the heat energy dissipation by the proposed encoder/decoder circuit is estimated. The estimation shows that the encoder/decoder circuit has very low energy dissipation. Single missing/additional cell-based defect analysis is also explored in this study. Reliability of the circuit is tested against different temperatures. Implementation and testing of the circuit are achieved using QCADesigner tool. MATLAB is used to produce the input to the proposed circuit.

1 Introduction

In nanotechnology, quantum dot cellular automata (QCA) is an emerging field which has the capability to function as an alternative to complementary metal–oxide–semiconductor (CMOS) [1–5]. Information is stored in QCA cell in terms of the charge possessed by the electrons, due to their electronic configuration within the QCA cell [6–10]. Mutual electrostatic columbic interaction between cells gives raise to bi-stable activity, required to perform the computation. Without any flow of current between the cells, binary information is encoded. In a CMOS circuit, different components are coupled together which give rise to problem during synchronisation [1–3]. The physical limitations of CMOS will be threatening in future for scalability of CMOS circuit. Thus, transition is essential from transistor circuit to transistor less circuit, where QCA is one of the potential solutions. QCA has low power consumption, high density and high clocking frequency [11–14]. Again, on losing a bit of information in irreversible computations, $K_b T_n^2$ joules of heat energy is produced, where $K_b$ and $T$ represents Boltzmann’s constant and absolute temperature, respectively. In case of reversible computation, energy dissipation would not occur [15]. This shows that the reversible logic is necessary to create low power consumption circuits. In Nanocommunication, the most important issue is to resist data against unauthorised access. Through image steganography data can be secured. The challenging aspect in steganography is the design of steganographic hardware at nanoscale with low power dissipation. Thus, by considering the inherent capabilities of QCA [16–24] and to get ideally zero power dissipation, reversible logic and QCA are combined together to achieve the encoder and decoder circuit for steganographic hardware at nanoscale. In this paper, it has been shown that how a reversible circuit can be designed to perform steganographic operation [21] as well as QCA can be used to implement this reversible steganographic architecture at nanoscale level. This steganographic operation is achieved in QCA based on reversible logic. For simplicity, the basic least significant bit (LSB)-based image steganography is considered in this paper.

The contributions are as follows:

(i) A novel reversible encoder and decoder for image steganography are proposed.
(ii) To achieve nanoscale architecture, the proposed encoder/decoder is realised in QCA.
(iii) A secure Nanocommunication system for steganography is achieved.
(iv) The cost of the proposed encoder/decoder and Nanocommunication circuit is estimated.
(v) Defect analysis of the proposed design is explored.
(vi) Power dissipation by the proposed layouts is estimated.
(vii) Reliability of the proposed circuit is also explored.

2 Image steganography using QCA

2.1 Related work

Recently image steganographic architecture is reported by Das et al. [21] based on irreversible logic. In comparison to the irreversible architecture proposed by Das et al. [21], the design of reversible QCA architecture for image steganography is achieved in this paper.

2.2 Feynman gate

Feynman gate is a reversible logic circuit [4]. It consists of two inputs and two outputs as shown in Fig. 1a. Input and output has one-to-one mapping. Quantum cost of Feynman gate is one. The equivalent QCA layout is shown in Fig. 1b.

2.3 Proposed encoder and decoder model for QCA-based image steganography

The procedure is outlined in Section 2.3.1 with an example using the proposed encoding algorithm (Fig. 10) and decoding algorithm (Fig. 11). The overall circuit implementation is performed in Section
2.3.2. For simplicity, LSB-based steganography is used for $8 \times 8$ pixels greyscale image (GI) [25].

2.3.1 Encoding and decoding using image:

(i) A GI of $8 \times 8$ pixels ‘Input_checkbox.bmp’ is considered, as shown in Fig. 2a. Matrix representation of its pixel intensity and the corresponding binary representation are shown in Figs. 2b and c, respectively.

(ii) A secret key of 8 bits ‘10001001’ is considered for encoding.

(iii) A message ‘automata’ is considered for embedding and converted into its corresponding binary value as shown in Fig. 3a.

(iv) Now, to encode and embed the bits of string ‘automata’ within the image, transfer the first seven bits of each pixel of the image through input channel $A_1$–$A_7$, respectively, of the proposed Reversible_Stego_Encoder/Decoder circuit as shown in Fig. 5b. Similarly, at the same time, transfer the binary value of the string ‘automata’ through the input channel ‘DATA’ and the key bits through input channel ‘Key’ of the circuit as shown in Fig. 5b. The XOR operation will produce the encoded bits for the string ‘automata’, i.e. the eighth bit of each pixel of the stego image (SI).

(v) As shown in Fig. 5b, the outputs obtained from output lines $B_1$ to $B_8$, which are the bits of each pixel of the encoded image, i.e. SI. The outputs at the line $B_8$ represents the embedded message bits and the outputs at the output line ‘GAR’ is considered as garbage output.

(vi) The bit stream obtained from output lines $B_1$ to $B_8$ are converted into their corresponding pixel intensities, which are then used to generate the SI ‘Encoded_checkbox.bmp’, as shown in Fig. 3b. Its pixel representation as well as the binary representation of each pixel is shown in Figs. 3c and d, respectively. The changed bits are outlined in Fig. 3d by a shaded rectangular box.

(vii) During decoding process, the SI will be converted into pixel intensities. These pixels are converted into binary values of length 8 bits each. Next, LSB bit of each pixel, i.e. the eighth bit of each bit stream will be extracted to obtain the encoded bits. These encoded bits are further used as input to the proposed encoder/decoder in a bit-stream manner. At the same instance, the secret key bits must be transferred through the channel ‘Key’. In this case, the outputs $B_1$–$B_8$ will be the bits of original message. The values at output lines $B_1$–$B_8$ are then converted to the characters to obtain the message as described in Fig. 4.

Similarly, at the same time, transfer the binary value of the string ‘automata’ through the input channel ‘DATA’ and the key bits through input channel ‘Key’ of the circuit as shown in Fig. 5b. The XOR operation will produce the encoded bits for the string ‘automata’, i.e. the eighth bit of each pixel of the stego image (SI).

(v) As shown in Fig. 5b, the outputs obtained from output lines $B_1$ to $B_8$, which are the bits of each pixel of the encoded image, i.e. SI. The outputs at the line $B_8$ represents the embedded message bits and the outputs at the output line ‘GAR’ is considered as garbage output.

(vi) The bit stream obtained from output lines $B_1$ to $B_8$ are converted into their corresponding pixel intensities, which are then used to generate the SI ‘Encoded_checkbox.bmp’, as shown in Fig. 3b. Its pixel representation as well as the binary representation of each pixel is shown in Figs. 3c and d, respectively. The changed bits are outlined in Fig. 3d by a shaded rectangular box.

(vii) During decoding process, the SI will be converted into pixel intensities. These pixels are converted into binary values of length 8 bits each. Next, LSB bit of each pixel, i.e. the eighth bit of each bit stream will be extracted to obtain the encoded bits. These encoded bits are further used as input to the proposed encoder/decoder in a bit-stream manner. At the same instance, the secret key bits must be transferred through the channel ‘Key’. In this case, the outputs $B_1$–$B_8$ will be the bits of original message. The values at output lines $B_1$–$B_8$ are then converted to the characters to obtain the message as described in Fig. 4.
2.3.2 Implementation of proposed method: The truth table of the proposed ‘Reversible_Stego_Encoder/Decoder’ is shown in Table 1. The shaded area at the input column (\(A_8\)) represents the LSB bits of the cover image. The shaded area at the output column (\(B_8\)) represents the LSB bits of the SI. Table 1 reflects that only the LSB bit is changed. All other input and output bits remain unchanged.

If the input bits are: \(A_1-A_7\) and \(B_1-B_7\) are the corresponding output bits, respectively. Then, from the truth table it can be seen that the output bits \(B_1-B_7\) will always be same as input bits \(A_1-A_7\), respectively, i.e.

\[
B_1 = A_1
\]
As mentioned earlier, the output bit $B_8$ is the bitwise XOR value of the data bits, i.e., bits of the word 'automata' and the secret key bits. Therefore, the expression of $B_8$ can be represented as

$$B_8 = k \oplus A_k \left[ A_8 = \text{data bit to be embedded and } k = \text{key bit} \right] \quad (8)$$

Here $A_8 = 0110100001100001 \ldots 01111001$, i.e., binary value of secret message 'automata' and secret key $(k) = 10001001$ and the corresponding majority gate expression can be written as

$$B_8 = M(M(A_8', k, 0), M(A_8, k', 0), 1) \quad (9)$$

In (9), $M$ stands for majority voter, the basic gate of QCA device. The value of $M$ depends on its inputs majority [7–10]. If one of the input to $M$ is set to '0', the output will be the logic AND value of the given inputs. If one of the input to $M$ is set to '1', the output will be the logic OR value of the given inputs. Thus, in (9), $M(A'_8, k, 0)$ is the logic AND value between complement of $A_8$ and $k$. Similarly, $M(A_8, k', 0)$ is the logic AND value between $A_8$ and complement of $k$. Finally, $M(M(A'_8, k, 0), M(A_8, k', 0), 1)$ is the logic OR value of $M(A'_8, k, 0)$ and $M(A_8, k', 0)$. Therefore, $B_8$ is the logic XOR value of inputs $A_8$ and $k$.

Using all of these expressions (1)–(7) and (9), the design of Reversible_Stego Encoder/Decoder circuit is achieved. The block diagram and layout of the Reversible_Stego Encoder/Decoder are shown in Figs. 5a and b, respectively.

3 Nano communication using proposed encoder/decoder

Fig. 5c shows a secure Nano communication technique using image. At the sender’s end, the encoder embedded the text within image along with secret key. Then, the generated encrypted image is transmitted through communication medium. On receiving the encrypted image at the receiver’s end, decoder retrieves the text using the secret key. The QCA layout of the Nano communication circuit is shown in Fig. 5d.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>$B_1$</td>
</tr>
<tr>
<td>$A_2$</td>
<td>$B_2$</td>
</tr>
<tr>
<td>$A_3$</td>
<td>$B_3$</td>
</tr>
<tr>
<td>$A_4$</td>
<td>$B_4$</td>
</tr>
<tr>
<td>$A_5$</td>
<td>$B_5$</td>
</tr>
<tr>
<td>$A_6$</td>
<td>$B_6$</td>
</tr>
<tr>
<td>$A_7$</td>
<td>$B_7$</td>
</tr>
<tr>
<td>$A_8$</td>
<td>$B_8$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$B_1$</th>
<th>$B_2$</th>
<th>$B_3$</th>
<th>$B_4$</th>
<th>$B_5$</th>
<th>$B_6$</th>
<th>$B_7$</th>
<th>$B_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

4 Result and discussions

Several GI of size $8 \times 8$ pixels are used to observe the efficiency of the proposed method as shown in Fig. 6. It can be observed that the changes in encoded image 'Encoded_checkbox.bmp' after embedding the text 'automata' within cover image 'Image_checkbox.bmp' are not perceived by human eye. Here, only single LSB bit of each pixel is replaced by single bit of secret message and the size of input image is $8 \times 8$ pixels. Thus, size of message that can be embedded within the input image is of maximum 64 bits in length. Messages <64 bits in length can also be embedded easily as shown in fourth row in Fig. 6. However, to embed the message more than 64 bits in length, the size of the input image must be increased accordingly.

The signal-to-noise ratio (SNR), peak SNR (PSNR) [21] values are better for embedding string 'auto' than embedding the string 'automata', because when the string 'auto' is embedded within the image, comparatively less amount of bits are changed within the image compared with embedding the string 'automata'. Since the string 'auto' has length of 32 bits and the input image

Table 1

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>$B_1$</td>
</tr>
<tr>
<td>$A_2$</td>
<td>$B_2$</td>
</tr>
<tr>
<td>$A_3$</td>
<td>$B_3$</td>
</tr>
<tr>
<td>$A_4$</td>
<td>$B_4$</td>
</tr>
<tr>
<td>$A_5$</td>
<td>$B_5$</td>
</tr>
<tr>
<td>$A_6$</td>
<td>$B_6$</td>
</tr>
<tr>
<td>$A_7$</td>
<td>$B_7$</td>
</tr>
<tr>
<td>$A_8$</td>
<td>$B_8$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$B_1$</th>
<th>$B_2$</th>
<th>$B_3$</th>
<th>$B_4$</th>
<th>$B_5$</th>
<th>$B_6$</th>
<th>$B_7$</th>
<th>$B_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Fig. 5  Block diagram and QCA layout of

a Proposed Reversible Stego_Encoder/Decoder circuit
b Proposed reversible Stego_Encoder/Decoder circuit
c Nano-communication circuit using the proposed encoder/decoder
d Nano-communication circuit using the proposed encoder/decoder
‘Image_checkbox.bmp’ has size of 8 × 8, i.e. 64 pixels. Therefore, 32 pixels out of 64 pixels are required to embed the string ‘auto’, caused less number of LSB bits to change. However, the string ‘automata’ has length of 64 bits and it is required to use all the 64 pixels of the input image ‘Image_checkbox.bmp’, results alteration of higher number of LSB bits. So, the total number of pixels will be used for embedding depends on the size of text, i.e. the total number of bits of the text. If the text to be embedded is smaller than the image area, then only a certain area of the image will be encoded. Duplication of the information within the image is not required as smaller text than image area means better SNR, PSNR values. It can be noted that for the proposed technique, the embedding will be started from the first pixel position and will be continued sequentially till the embedding is over.

4.1 Simulation result of proposed layouts

The simulation result of Nanocommunication circuit is shown in Fig. 7b. To simulate the Nanocommunication circuit, the same input column of Table 1 is used. Fig. 7b shows that when the key bit, i.e. Key=1 and the Data Bit=0, then the Encoded Bit (Y8) = 1 and Decoded Bit, i.e. original Data Bit=0, which is shown in the figure by an arrow. When the key bit, i.e. Key=0 and the Data Bit=1, then the Encoded Bit (Y8) = 1 and Decoded Bit, i.e. retrieved Data Bit=1. Similarly, for all inputs, the corresponding outputs are shown by rectangular box.

The results are verified with the truth table as shown in Table 1. This evaluation proves that the proposed QCA circuit works efficiently. The arrows in Fig. 7 indicate the starting point of required outputs. Key and data in Figs. 7a and b are the key bits and the bits of the secret data ‘automata’. Encoded data in Figs. 7a and b denotes the bits formed after XOR-ing the key bits with the bits of the secret message ‘automata’. The ‘decoded data’ in Fig. 7b represents the data obtained after second time XOR-ing with the key bits.

4.2 Circuit complexity of proposed designs

The hardware complexity of the circuits is described in Table 2; it clearly depicts that the proposed QCA Feynman gate acquires 0.037 µm² area and requires 54 QCA cells. The latency is 0.75. Similarly, the circuit complexity of the proposed Reversible_Stego_Encoder/Decoder and Steganographic system are estimated and shown in Table 2.

Traditional circuit for the proposed Reversible_Stego_Encoder/Decoder required 0.18 µm² area whereas QCA circuit required only 0.129 µm² in area as shown in Table 3. Thus, 28.33% improvement in terms of area has been achieved over transistor technology.

4.3 Quantum cost of proposed designs

The quantum cost of the proposed circuit in traditional-based design and in QCA-based design is illustrated in Table 4. It can be seen that QCA circuit of the proposed designs have low quantum cost and cost effectiveness than their quantum gate circuits.
Table 2  Complexity of the proposed encoder/decoder circuit in QCA

<table>
<thead>
<tr>
<th>QCA circuits</th>
<th>Complexity</th>
<th>Total area, µm²</th>
<th>Cell area, µm²</th>
<th>Area usage, %</th>
<th>Latency (clock cycle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feynman gate (Fig. 1)</td>
<td>three majority gates and two inverters</td>
<td>54</td>
<td>0.037</td>
<td>0.017</td>
<td>46.15</td>
</tr>
<tr>
<td>Reversible_Stego_Encoder/Decoder (Fig. 5b)</td>
<td>three majority gates and two inverters</td>
<td>181</td>
<td>0.129</td>
<td>0.058</td>
<td>45.53</td>
</tr>
<tr>
<td>Steganographic system (Fig. 5d)</td>
<td>six majority gates and four inverters</td>
<td>483</td>
<td>0.335</td>
<td>0.16</td>
<td>46.71</td>
</tr>
</tbody>
</table>

Table 3  Comparison of the proposed QCA encoder/decoder circuit with traditional circuit

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Traditional circuit, µm²</th>
<th>Proposed QCA circuit, µm²</th>
<th>Improvement, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reversible_Stego_Encoder/Decoder</td>
<td>~0.18</td>
<td>~0.129</td>
<td>28.33</td>
</tr>
</tbody>
</table>

Table 4  Quantum cost of the proposed reversible circuit

<table>
<thead>
<tr>
<th>Proposed reversible circuit</th>
<th>Quantum cost</th>
<th>Quantum cost of QCA circuit (area latency²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feynman gate (Fig. 1)</td>
<td>1</td>
<td>0.0208</td>
</tr>
<tr>
<td>Reversible_Stego_Encoder/Decoder (Fig. 5b)</td>
<td>1</td>
<td>0.0725</td>
</tr>
<tr>
<td>Steganographic system (Fig. 5d)</td>
<td>2</td>
<td>2.0937</td>
</tr>
</tbody>
</table>
4.4 Defect analysis of QCA circuit of proposed encoder/decoder

This section explores the defect analysis of QCA circuit of the proposed encoder/decoder. The analysis is performed based on two criteria as follows:

(i) Single missing cell defect.
(ii) Additional cell defect.

To perform the analysis, according to different layers all the cells of the circuit is outlined based on grid position as shown in Fig. 8 [15]. For example, C2 grid position describes the cell at Cth row and second column. The defects caused by single missing/additional cell are illustrated in Table 5. QCADesigner tool [15] is used to simulate the circuit for each missing cell and additional cell. The outputs are analysed to select the test vector.

Table 5 shows that if a single cell in between O2 and O6 is deleted, the faulty output will be ‘000101100’. This fault can be detected using test vector ‘000101101’. The expected output is

Fig. 8 Different layers and grid position of cell in QCA layout of the proposed encoder/decoder circuit

a Main cell layer
b Layer 1 and 2
c Layer 3


4.5 Power consumption of proposed encoder/decoder circuit

The power consumption of any QCA circuit depends on majority gate and inverters used in designing the circuit [26]. Table 6 shows the power consumed by the proposed QCA circuits. The estimation is performed at temperature ($T = 2.0$ K) by employing different levels of tunnelling energy ($E_t$) between dots. Here, $K$ represents the unit of temperature in kelvin.

### Table 6 Traditional-based quantum cost versus QCA-based quantum cost of the proposed design

<table>
<thead>
<tr>
<th>Proposed QCA design</th>
<th>Consumption of power at $T = 2.0$ K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\gamma = 0.25E_t$ meV</td>
</tr>
<tr>
<td>Feynman gate (Fig. 1)</td>
<td>132.7 mV</td>
</tr>
<tr>
<td>Stego Encoder/Decoder (Fig. 5b)</td>
<td>331.5 mV</td>
</tr>
<tr>
<td>Steganographic system (Fig. 5d)</td>
<td>663.0 mV</td>
</tr>
</tbody>
</table>

Fig. 9 depicts the effect of polarisation on output of stego encoder/decoder circuit due to temperature variations [27]. Clearly visible from Fig. 9 that little change is within the range of temperature 1–7 K. Thus, the circuit works efficiently within 1–7 K range of temperature. A radical change in the output is observed at temperature above $T = 7$ K and the circuit starts to produce wrong results. QCADesigner [28] is used to observe these effects. Average output polarisation (AOP) is calculated for each output. For instance, the max and min values of polarisation for output $Y_b$ at temperature $T = 1$ K is $9.54 \times 10^{-4}$ and $-9.54 \times 10^{-4}$, respectively. The AOP is therefore $(9.54 \times 10^{-4} + 9.54 \times 10^{-4})/2 = 3.51$ as explored in Fig. 9. The AOP of each output cell is degraded by raising the temperature. As shown in Fig. 5b, all the output cells are in clock zone 2 and all output cells will have same polarisation state. Thus, in between temperature range 1–7 K, at any point of temperature, the changes to polarisation for all output cells have very little difference. Therefore, there is a sharp difference when going from 1 to 7 K. Above the temperature range 1–7 K, though the AOP degrades rapidly, still there is a quiet sharp difference between AOP of output cells.

### 5 Conclusion

In this paper, a reversible encoder/decoder circuit for image steganography is proposed at nanoscale. This is not only effective in terms of low power dissipation but also focuses on the important aspects of a reversible computing. Feynman gate is used as a basic element to achieve the proposed designs. A secure Nanocommunication process for image steganography is clearly depicted in this paper. The proposed encoder/decoder has 0.335 $\mu$m device density that confirms 28.33% reduction over traditional CMOS-based implementation. To reduce the circuit complexity, a GI is considered. Several possible defects are explored to achieve fault free implementation. The proposed encoder/decoder circuit dissipates very low energy. Though the input to the encoder/decoder circuit is considered as a sequence of 8 bits, the circuit can be used for higher number of bits in sequence. The functional efficiency of the circuit is analysed at different temperatures and its accuracy is measured. One of the important features of this steganographic circuit is that it possesses a non-deterministic aspect and inherent resistivity against power analysis attack, which enhances the security of the hidden data.
6 Acknowledgment

The authors were grateful to the University Grants Commission, India, for providing with the grant for accomplishment of the project under the UGC Major Project File No. 41-631/2012(SR).

7 References

22 Roy, D., Maitra, S., De, D., et al.: ‘Analysis of effect of temperature variation on computational faithfulness of a QCA XOR gate’, Int. Conf. on Electronics, Communication and Instrumentation (ICECI), Kolkata, West Bengal, India, January 2014, pp. 1–4

8 Appendix 1

(See Fig. 10)

9 Appendix 2

(See Fig. 11)

Algorithm 1

Input: Gray scale image (GI)
Secret key (K)
Message (M)

Output: Stego image (SI)

1. Convert each alphabet of the message (M) into corresponding ASCII value.
2. Obtain the binary value from ASCII value.
3. Partition the total bit-stream into sub bit-streams of 8-bit long each.
4. Provide a secret key (K) of 8-bit long for encoding.
5. Perform XOR operation between key and each sub bit-stream of the message.
6. Continue step-5 until all the sub bit-streams are XOR-ed with the key, to form the encoded message.
7. Obtain pixel information from the gray scale image (GI).
8. Convert each pixel value into its corresponding binary value.
9. Insert the encoded message bit by bit into the LSB bit of each pixel of the image, until all the message bits are embedded within the pixels.
10. Regenerate the image from the encoded pixels as produced in step-9, to obtain the stego image (SI).

Fig. 10 Data hiding algorithm

Algorithm 2

Input: Stego image (SI)
Secret key (K)

Output: Message (M)

1. Obtain the pixel intensities from the stego image.
2. Generate binary value of each pixel intensities.
3. Extract the LSB bit of each pixel from the binary values, which contains the encoded information.
4. Partition the total LSB bit-stream into sub bit-streams of 8-bit long each.
5. Perform XOR operation between key and each sub bit-stream of the LSB bit-stream.
6. Continue step-5 until all the sub bit-streams are XOR-ed with the key, to decode the bits of original information.
7. Convert the decoded binary values into corresponding ASCII value.
8. Regenerate the character from the ASCII value.

Fig. 11 Data recovery algorithm