A Novel and Systematic Approach to Implement Reversible Gates in Quantum Dot Cellular Automata

[1]P. SARAVANAN [2]P. KALPANA
[1]Assistant Professor [2]Professor
Electronics and Communication Engineering,
PSG College of Technology, Coimbatore, Tamil Nadu.
INDIA
dpsaravanan@yahoo.com

Abstract: - Quantum dot cellular automata (QCA) is one of the emerging technologies in the area of nano-electronics and is found to be an attractive alternative to conventional CMOS technology for several reasons. However, there have been no reports in the literature so far on QCA implementation of conventional reversible gates. In this work, we propose a novel and systematic approach for the QCA implementation of conventional reversible gates. The proposed method utilizes universal nature of the majority gate for its operation. The combination of reversible logic synthesis and its QCA implementation proves to be a superior solution for side channel attack based on power analysis in security applications. This is mainly due to the negligible amount of power consumption in both reversible logic and QCA implementation. Hence in this work, a dual-field adder, which plays a vital role in unified architectures of public-key crypto system, has been synthesized in reversible logic and implemented in QCA.

Key-Words: - Quantum dot cellular automata, Reversible logic, Side channel attack, Nano-technology, Cryptography, Reversible gates.

1 Introduction

In the recent past, VLSI technology has achieved tremendous progress and technology nodes have been continuously shrinking down. As the typical feature size of CMOS VLSI has shrunk into deep submicron domain, nanotechnology is the next step in order to maintain Moore's law for several more decades. Nanotechnology not only improves the resolution in traditional photolithography process, but also introduces many brand-new fabrication strategies such as bottom-up molecular self-assembly. Nanotechnology is also enabling many novel devices and circuit architectures which are totally different from current microelectronics circuits such as quantum computing, nano-wire crossbar circuits, and spin electronics, etc.

Quantum-dot cellular automata (QCA) is one of the promising and alternative technologies to replace CMOS technology. A new paradigm for computing with cellular automata has been formulated in [1]. The conventional logical operations such as AND and OR are performed by QCA based gates in [2]. Implementation of digital logic architectures such as latches and shift registers have been discussed in [3]. Recently several adder structures have been proposed in QCA [4][5][6]. Ultra-low-power binary multiplier design based on QCA is reported in [7]. Reversible computing has emerged as a promising technology in low power applications.

The idea of reversible computing was discussed by Landauer [8] and Bennett [9], and further refined by Toffoli [10]. According to Landauer's research [8], the change in entropy associated with the loss of one bit of information is $\ln 2$, which thermodynamically, corresponds to an energy increase of $k_B T \ln 2$, where k_B is Boltzman's constant and T is the temperature. The heat dissipated during a process is usually taken to be a sign of physical irreversibility, that the microscopic physical state of the system cannot be restored exactly as it was before the process had taken place. This classical computation can be done reversibly with no energy dissipated per computational step was discovered by Bennett in 1973 [9]. Several reversible logic circuits have been proposed in the literature [11][12].

Reversible logic synthesis has been done in an Arithmetic and Logic Unit (ALU) of a crypto-processor to prevent differential power analysis attacks [13]. Montgomery multiplier has been synthesized efficiently in reversible logic with the same intention of preventing power analysis attacks [14]. All these works have shown how to synthesize a circuit in reversible logic and their final results were given in gate level. However, none of them have given any physical level implementation for reversible logic. In this work, a novel and straightforward approach is introduced for implementation of reversible logic gates in QCA. The proposed implementation makes use of the universal nature of majority gates.

2 Reversible Logic Gates

The traditional logic gates such as AND, OR and NAND are "irreversible" in the sense that their input cannot be retrieved from the output, since these gates realize logic functions that are not one-to-one. On the other hand, NOT is reversible since it realizes a one-to-one function. A reversible gate is one that realizes a one-to-one logic function, hence it is in general, a many-input many-output gate [15].

A multiple-control Toffoli gate [10] $C^m\ NOT\ (x_1, x_2, \cdots, x_{m+1})$ passes the first m lines, control lines, unchanged. This gate flips the $(m+1)^{th}$ line, target line, if and only if each positive (negative) control line carries the 1 (0) value. For $m = 0, 1, 2$ the gates are named $NOT(N), CNOT(C)$ and $Toffoli(T)$ respectively. These three gates compose the universal NCT library.

A multiple-control Fredkin gate [16] $Fred\,(x_1, x_2, \cdots, x_{m+2})$ has two target lines x_{m+1}, x_{m+2} and m control lines $x_1, x_2, \cdots, x_m$. The gate interchanges the values of the targets if the conjunction of all m positive (negative) controls evaluates to 1 (0). For $m = 0,1$ the gates are called $SWAP(S)$ and $Fredkin(F)$ respectively.

A Peres gate [17] $P(x_1, x_2, x_3)$ has one control line x_1 and two target lines x_2 and x_3. It represents a $C^2NOT\,(x_1, x_2, x_3)$ and a $CNOT(x_1, x_2)$ in a cascade.

2.1 Ancilla and Garbage lines

As mentioned earlier, the important condition for a reversible gate is that the number of input and output lines should be equal to each other. To make the specification reversible, input/output should be added. The added lines are called ancillae and typically start out with 0 or 1 constant. An output line that is needed to maintain reversibility is known as a garbage line [18].

2.2 Quantum Cost

Quantum cost denotes the amount of effort needed to transform a reversible circuit to a quantum circuit. Table 1 shows the quantum cost for a selection of Toffoli and Fredkin gate configurations as introduced in [19] and further optimized in [20]. As can be seen, from Table 1, gates of larger size are considerably more expensive than gates of smaller size [21].

Table 1. Quantum Cost for Toffoli and Fredkin Gates

No. of Control Lines	Quantum cost of Toffoli Gate	Quantum Cost of Fredkin Gate
0	1	3
1	1	7
2	5	15

3 Quantum-Dot Cellular Automata

Quantum-dots are semiconductors or conductors in nanosize [22]. These quantum-dots consist of a few to several hundreds of atoms, which are spatially located in some arrangement. QCA offers a novel alternative to the transistor paradigm [23,24]. Quantum cells are the basic elements of QCA circuits. A simple quantum cell consists of four quantum-dots and two loaded electrons [25], as shown in Fig. 1. It is clear that the dots are the places where the charge can be localized. The two electrons will tend to occupy antipodal sites as a result of their mutual electrostatic repulsion. However, they can change their positions within dots as a result of tunnelling effect. This phenomenon takes place when the potential barrier separating the two quantum dots is low. However, tunnelling process into or out of a cell will be blocked severely. Consequently, two configurations are possible, which can be used to encode binary information, as shown in Fig. 2.

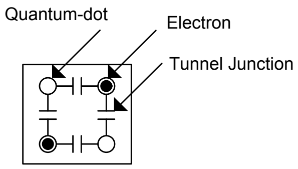

Fig. 1. QCA Cell

The numbering of quantum dots starts clockwise from the top-right dot. Polarization (P), which represents the distribution form of electric charge in the four dots in each cell is defined as follows:

$$P = \frac{(\rho_1 + \rho_3) - (\rho_2 + \rho_4)}{\rho_1 + \rho_2 + \rho_3 + \rho_4} \qquad (1)$$

where ρ_i indicates electric charge density at dot i. Because of Coulombic repulsion, electrons rest on the two extremes of diagonals in each cell. Considering ρ_i values in Eq. (1), it is concluded that P can only take values P = 1 and P = -1, which represent binary values "0" and "1" respectively. These two states are used for encoding binary data. When a polarized cell is placed in a line close to another cell, Coulomb repulsion between them makes the second cell to be in the same state as the first one and this causes electrostatic energy to get minimized in the charge configuration of the cells. This is how the state is propagating in a line of cells. Based on Coulombic interactions between the cells, basic QCA devices can be developed.

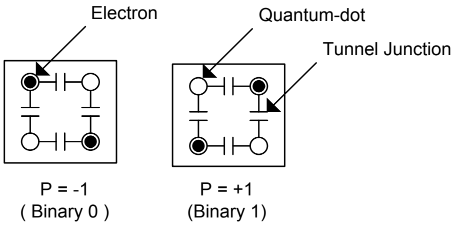

Fig. 2. Representation of QCA cells in binary form

3.1 Basic QCA Gate

A very basic QCA gate is called majority gate or majority voter shown in Fig. 3. It can be proved that all other logical gates like AND, OR are implementable by majority gate [25,23]. The truth table of this gate is shown in Table 2. Fig. 4 and Fig. 5 shows how AND, OR gates are created by majority gate. Fig. 6 shows the structure of a QCA implemented NOT gate. But this basic QCA gate is irreversible in nature since it takes three inputs and produces only a single output.

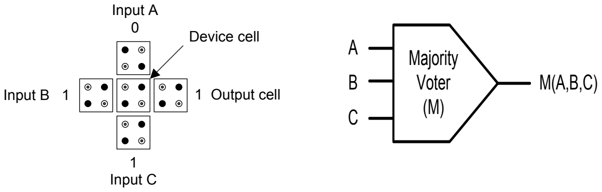

Fig. 3. The basic QCA Majority gate

3.2 QCA clocking scheme

An important thing about QCA information flow is the clocking scheme; that is to say for adjacent cells, in order to control the polarization reactions and effects, one should hold the polarization of the first cell fixed and lower the potential barrier of its adjacent cell in order to let the electrons of the adjacent cell relocate. This phenomenon should repeat over and over again to pass the information through cells. It has been shown that for a QCA circuit to function correctly, only four clocking zones are necessary. Each clock signal lags 90° in phase with respect to the previous clocking [26]. The four clock zones are shown in Fig. 7. These four clock zones are also called as one clock sequence.

In QCA implementations, the power consumption of QCA circuits are mainly affected by clock signals. Therefore, in practice using a smoother clock, the power consumption will be very less but still it is data dependent. In order to remove the data dependency of power traces, Bennett clocking scheme can be used. As compared to its counterpart Landauer-clocked QCA circuit, Bennett clocking produces very low and very similar power traces for different inputs [27]. Therefore, by using Bennett clocking, the power dependence of QCA circuits on the inputs can be effectively removed making it impossible to perform power analysis attack [28].

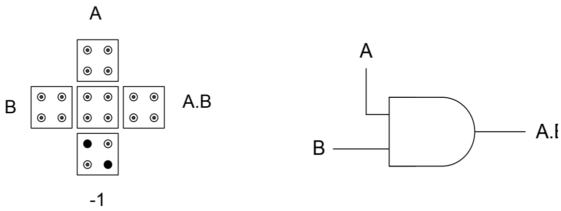

Fig. 4. AND gate by using basic QCA Majority gate

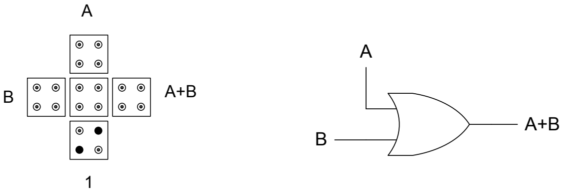

Fig. 5. OR gate by using QCA Majority gate

Table 2. Truth Table of Basic QCA Majority Gate.

A	B	C	M(A,B,C)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

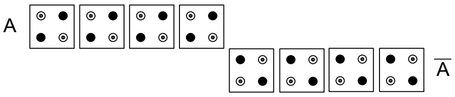

Fig. 6. QCA NOT gate

4 Proposed Method

This section describes a family of reversible majority gates from Miller's gate [29]. As shown in Table 2, the output of the basic QCA majority gate M becomes high when majority of the inputs A, B and C are in high level. The boolean function of the basic QCA majority gate is given in Eq. (2). Since the number of inputs and outputs are not equal, this basic majority gate is irreversible in nature.

4.1 Reversible Majority Gate 1

In order to make the basic majority gate reversible, along with the present output, two more outputs are added in the basic majority gate. The boolean equation for the first added output Y is obtained by complementing the input signal A in the basic majority gate Eq. (2). Similarly the boolean equation for the second added output Z is obtained by complementing the input signal B in the basic majority gate Eq. (2). After adding two outputs, the number of inputs and outputs are equal and unique mapping is also achieved between inputs and outputs which makes the basic majority gate as a reversible majority gate [30] as given in Eq. (3). Hereafter this reversible majority gate will be referred as reversible majority gate 1. The truth table of the reversible majority gate 1 is shown in Table 3. From the truth table, it can be inferred that two different outputs are available at a time. If AND operation is taken from output X, then OR operation can be obtained through output Y and the vice versa is also true.

4.2 Reversible Majority Gate 2

The reversible majority gate 2 can be obtained by complementing inputs A and C as given in Eq. (4). In this gate also unique mapping is achieved between inputs and outputs shown in Table 4. From the truth table, it can be inferred that two different outputs are available at a time. If AND operation is taken from output X, then OR operation can be obtained through output Y and the vice versa is also true. The reversible majority gate 3 can be obtained by complementing inputs B and C as given in Eq. (5). In this gate also unique mapping is achieved between the inputs and outputs but only one output, either AND or OR can be used at a time, as shown in Table 5. Hence this gate is ignored for our discussion.

$$M(A,B,C) = A.B + B.C + A.C \quad (2)$$

$$\left.\begin{aligned} X &= A.B + B.C + A.C \\ Y &= \overline{A}.B + B.C + \overline{A}.C \\ Z &= A.\overline{B} + \overline{B}.C + A.C \end{aligned}\right\} \quad (3)$$

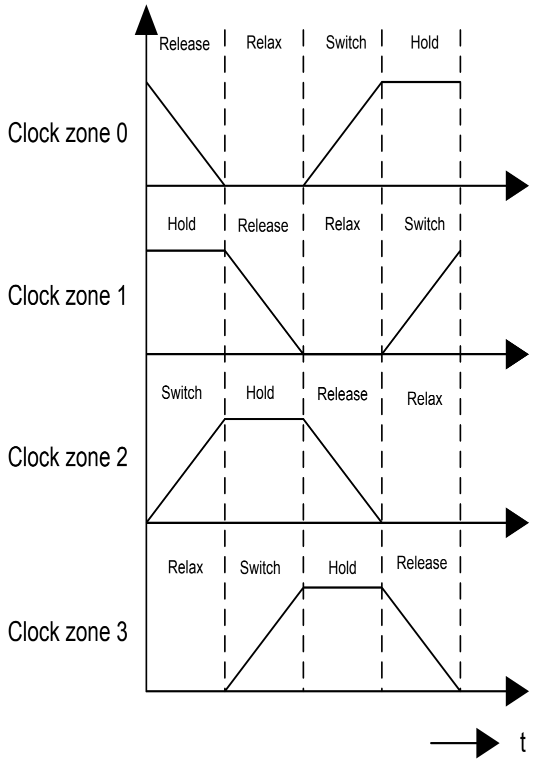

Fig. 7. Clocking scheme of QCA circuits

Table 3. Truth Table of Reversible Majority Gate 1

A	B	C	X	Y	Z	Unique Mapping
0	0	0	0	0	0	0
0	0	1	0	1	1	3
0	1	0	0	1	0	2
0	1	1	1	1	0	6
1	0	0	0	0	1	1
1	0	1	1	0	1	5
1	1	0	1	0	0	4
1	1	1	1	1	1	7

$$\left.\begin{aligned} X &= A.B + B.C + A.C \\ Y &= \overline{A}.B + B.C + \overline{A}.C \\ Z &= A.B + B.\overline{C} + A.\overline{C} \end{aligned}\right\} \quad (4)$$

$$\left.\begin{aligned} X &= A.B + B.C + A.C \\ Y &= A.\overline{B} + \overline{B}.C + A.C \\ Z &= A.B + B.\overline{C} + A.\overline{C} \end{aligned}\right\} \quad (5)$$

Table 4. Truth Table of Reversible Majority Gate 2

A	B	C	X	Y	Z	Unique Mapping
0	0	0	0	0	0	0
0	0	1	0	1	0	2
0	1	0	0	1	1	3
0	1	1	1	1	0	6
1	0	0	0	0	1	1
1	0	1	1	0	0	4
1	1	0	1	0	1	5
1	1	1	1	1	1	7

Table 5. Truth Table of Reversible Majority Gate 3

A	B	C	X	Y	Z	Unique Mapping
0	0	0	0	0	0	0
0	0	1	0	1	0	2
0	1	0	0	0	1	1
0	1	1	1	0	0	4
1	0	0	0	1	1	3
1	0	1	1	1	0	6
1	1	0	1	0	1	5
1	1	1	1	1	1	7

The reversible majority gate 1 is functionally similar to reversible majority gate 2 in terms of outputs X and Y. Only the third output Z is different in both gates and this output can be ignored since it does not give any useful logic functions. Since both reversible majority gate 1 and 2 are functionally similar, in this work, reversible majority gate 1 is considered for all our discussions.

The reversible majority gate 1 has got three inputs and two useful outputs ie. AND output and OR output. In order to make it as a two input reversible gate, the third input A is kept as a constant value and is pulled up to a high level. By doing so, a two input reversible AND gate and a two input reversible OR gate can be obtained. The QCA implementation of the reversible majority gate 1 is given in the next section.

4.3 QCA Implementation

4.3.1 Instrumentation and Parameters

The QCA cell placement and simulation for all circuits discussed in this paper have been carried out in QCADesigner, version 2.0.3 [31]. The default parameters were used for simulation of QCA cells. Each cell has a height of 18nm and width of 18nm while the quantum dots have a diameter of 5 nm .Since all the cells are placed on a single grid, they have a cell centre-to-centre distance of 20 nm.

4.3.2 Majority based reversible logic gate

The basic majority gate can be implemented in QCA by placing nine cells as shown in Fig. 8. Out of four sides, three sides can be used for inputs and the fourth side gives the output. In each side, two cells are placed adjacent to each other in order to improve the stability of the basic majority gate. Three basic majority gates are cascaded to get the proposed QCA implementation of the reversible majority gate 1 as shown in Fig. 9. The polarization between the cells in all the three basic majority gates are synchronized by using a sequence of four clocks.

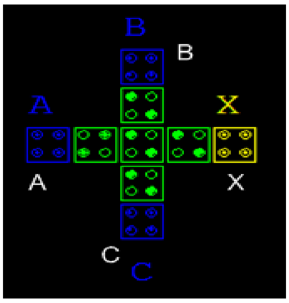

Fig. 8. QCA implementation of basic majority gate

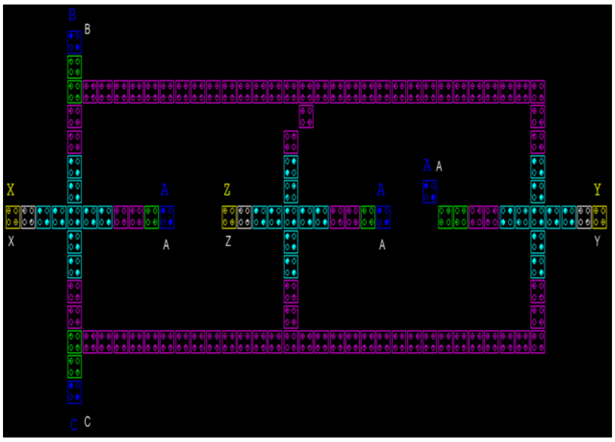

Fig. 9. QCA implementation of proposed reversible majority gate

Input signals for all gates are applied at clock 0. In clock 1 and clock 2, signals are processed and the final output of all the three gates are available at clock 3.

The proposed QCA implementation of reversible majority gate takes 4 clock cycles to generate its final output. Hence the latency is one clock sequence. The simulated output of the reversible majority gate is shown in Fig. 10. From the simulation output, we can infer that the proposed QCA implementation generates outputs of two different gates in output X and Y. The third output Z can be ignored since it does not generate any useful logic function.

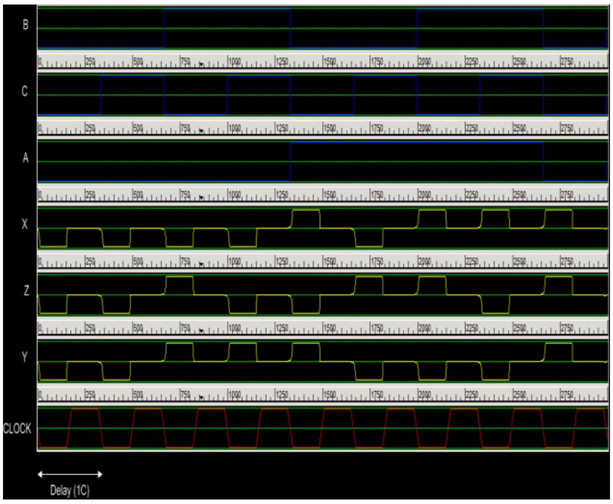

Fig. 10. Simulation response of proposed reversible majority gate

When input A is fixed at a constant value such as logic 0 then output X generates AND gate output and output Y generates OR gate output. Similarly when input A is fixed at a constant value of logic 1, then output X generates OR gate output and output Y generates AND gate output. By keeping one of the inputs at a fixed constant value, the proposed QCA implementation generates a reversible AND gate and a reversible OR gate output. Hence any one of the gate outputs can be used depending upon the requirement.

5 QCA implementation of reversible gates

5.1 CNOT gate

The reversible CNOT gate takes two inputs say A and B and produces two outputs P and Q out of which one output P follows the input A and the other output Q is the XOR of the two inputs A and B. The XOR operation can be performed by the logic function AB' + A'B which has two AND gates and one OR gate. In order to implement this gate in QCA, two reversible two input AND gates and one reversible two input OR gate is required as shown in Fig. 11. The reversible gate implementation of CNOT gate in QCA takes 3 ancilla inputs, 6 garbage outputs and has a latency of 3 clock cycles. The simulated waveform is shown in Fig. 12 where the valid output comes from the fourth clock cycle onwards.

5.2 CCNOT gate

The reversible CCNOT gate which is also known as Toffoli gate takes three inputs A, B, C and produces three outputs P, Q, R out of which two outputs P, Q follow the inputs A, B respectively and the third output is the conditional XOR of its inputs ie. R = A.B XOR C. In order to implement this gate in QCA, three reversible two input AND gates and one reversible two input OR gate is required as shown in Fig. 13. The reversible gate implementation of CCNOT gate in QCA takes 4 ancilla inputs, 8 garbage outputs and has a latency of 5 clock cycles. The simulated waveform is shown in Fig. 14, where the valid output comes from the sixth clock cycle onwards.

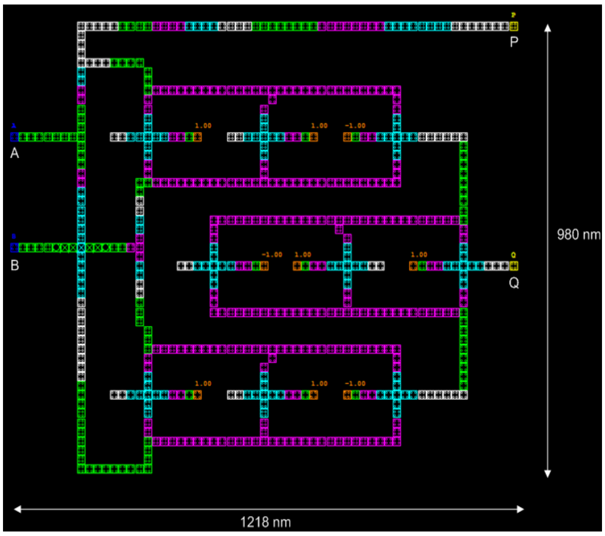

Fig. 11. QCA implementation of CNOT gate

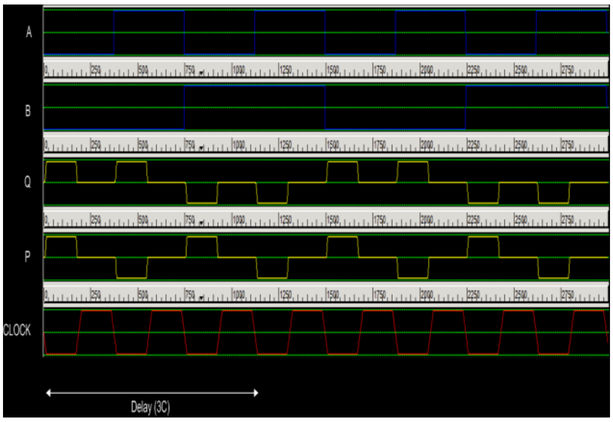

Fig. 12. Simulation response of CNOT gate

5.3 FREDKIN gate

The reversible FREDKIN gate also known as conditional SWAP gate takes three inputs A, B, C and produces three outputs P, Q, R out of which one output P follows the input A and the other two outputs Q, R are nothing but the conditional SWAP of the inputs B, C based on the logic value in input A. In order to implement this gate in QCA, four reversible two input AND gates and two reversible two input OR gates are required as shown in Fig. 15. The reversible gate implementation of FREDKIN gate in QCA takes 6 ancilla inputs, 12 garbage outputs and has a latency of 5 clock cycles. The simulated waveform is shown in Fig. 16, where the valid output comes from the sixth clock cycle onwards.

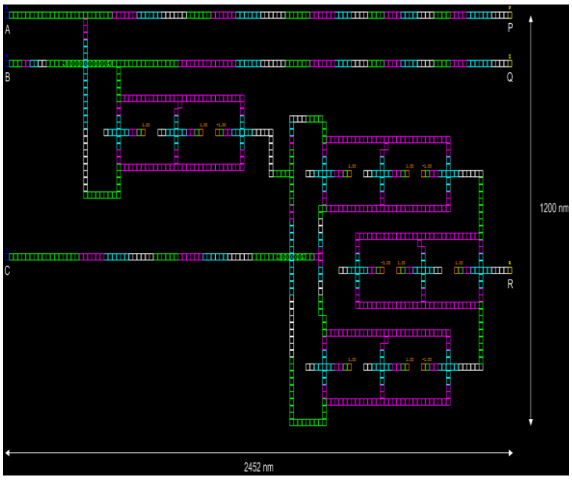

Fig. 13. QCA implementation of CCNOT gate

5.4 SWAP gate

The reversible SWAP gate takes two inputs A, B and produces two outputs P, Q which are nothing but the direct swap of its inputs ie. the input A is swapped to output Q and input B to output P respectively. In order to implement this gate in QCA, three CNOT gates are cascaded. Hence six reversible two input AND gates and three reversible two input OR gates are required as shown in Fig. 17. The reversible gate implementation of SWAP gate in QCA takes 9 ancilla inputs, 18 garbage outputs and has a latency of 9 clock cycles. The simulated waveform is shown in Fig. 18, where the valid output comes from the tenth clock cycle onwards.

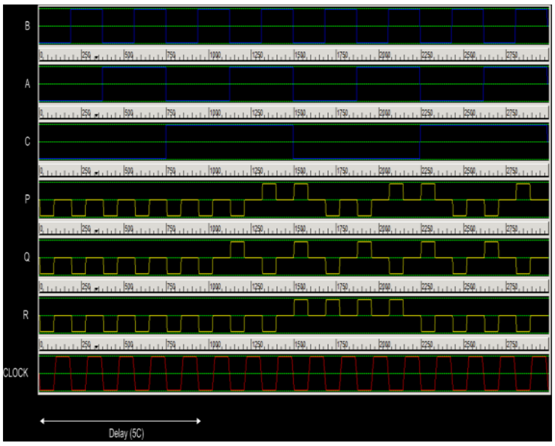

Fig. 14. Simulation response of CCNOT gate

5.5 PERES gate

The reversible PERES gate takes three inputs A, B, C and produces three outputs P, Q, R out of which one output P follows the input A and the second output Q is the XOR of its inputs A, B and the third output is the conditional XOR of its inputs ie. R = A.B XOR C. In order to implement this gate in QCA, one CNOT gate and one CCNOT gate connected in parallel hence five reversible two input AND gates and two reversible two input OR gates are required as shown in Fig. 19. The reversible gate implementation of PERES gate in QCA takes 7 ancilla inputs, 14 garbage outputs and has a latency of 5 clock cycles. The simulated waveform is shown in Fig. 20, where the valid output comes from the sixth clock cycle onwards.

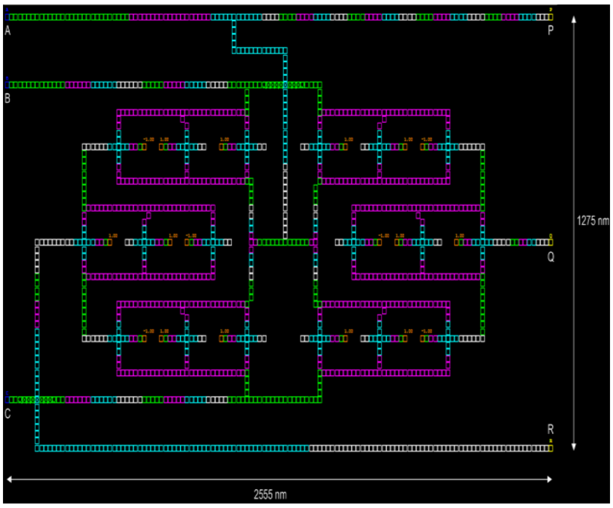

Fig. 15. QCA implementation of FREDKIN gate

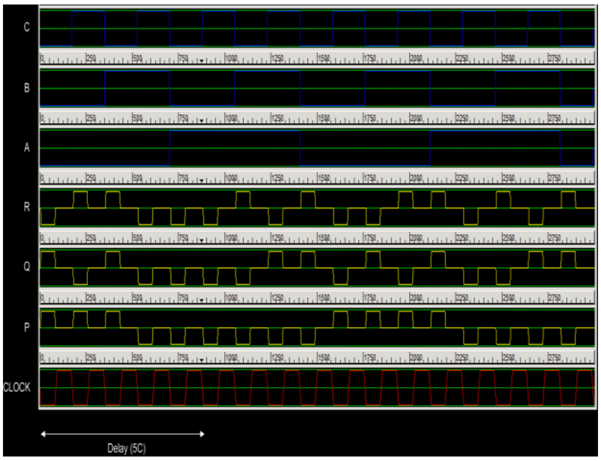

Fig. 16. Simulation response of FREDKIN gate

5.6. Dual-Field Adder

Dual-field adder (DFA) is capable of performing addition, with and without carry. Hence it requires a normal full-adder with controlled carry generator. Addition with carry corresponds to the addition operation in the field $GF(p)$ while addition without carry corresponds to the addition operation in the field of $GF(2^m)$. It has an input called FSEL (field select) that enables the selection of the field. When FSEL = 1, the DFA performs the bitwise addition with carry which enables the prime field $GF(p)$ to do its arithmetic operations.

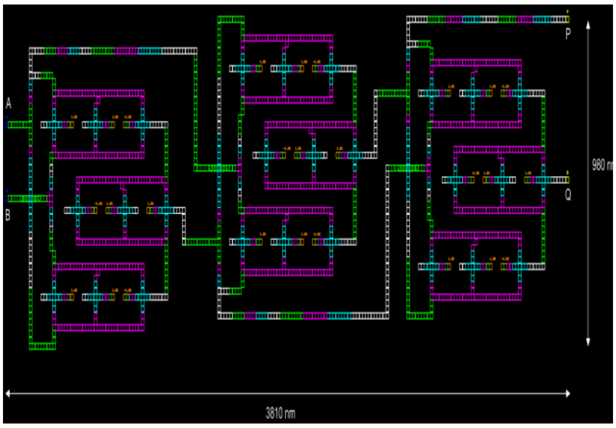

Fig. 17. QCA implementation of SWAP gate

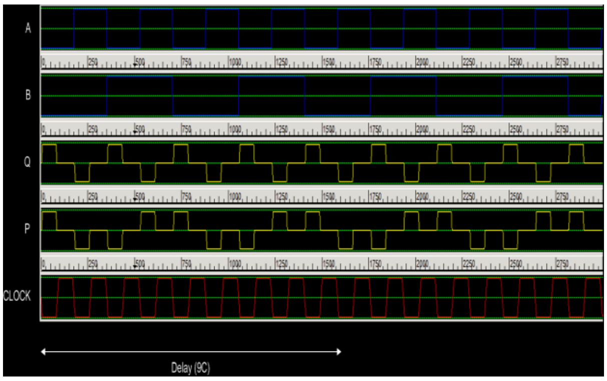

Fig. 18. Simulation response of SWAP gate

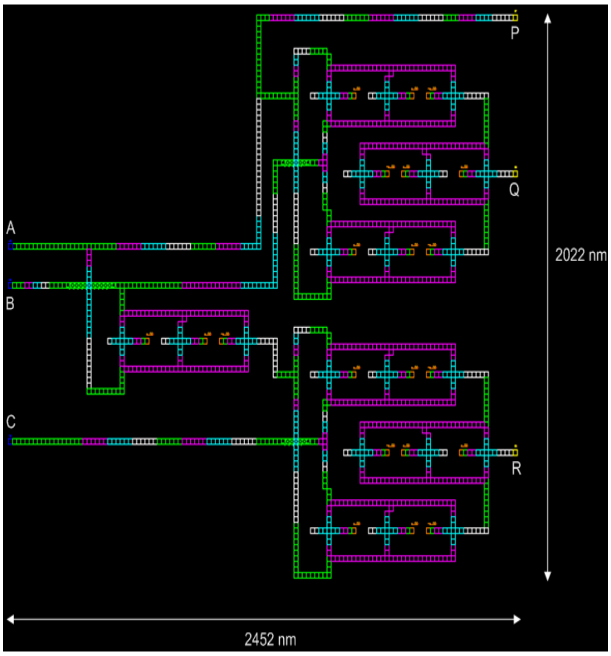

Fig. 19. QCA implementation of PERES gate

On the other hand, when FSEL = 0, the output C_{out} is forced to 0, regardless of the values of the inputs. The output S produces the result of bitwise modulo-2 addition of three input values which is nothing but the arithmetic in binary field $GF(2^m)$. Since the Dual-field adder performs arithmetic operations both in prime field $GF(p)$ and binary field $GF(2^m)$, it can be used in the unified architectures of Galois field multipliers of crypto processors.

The Dual-field adder can be implemented in reversible gates by cascading two peres gates and one reversible two input AND gate as shown in Fig. 21. The reversible gate implementation of Dual-field adder in QCA takes 15 ancilla inputs, 30 garbage outputs and has a latency of 11 clock cycles. The simulated waveform is shown in Fig. 22, where the valid output comes from the twelfth clock cycle onwards.

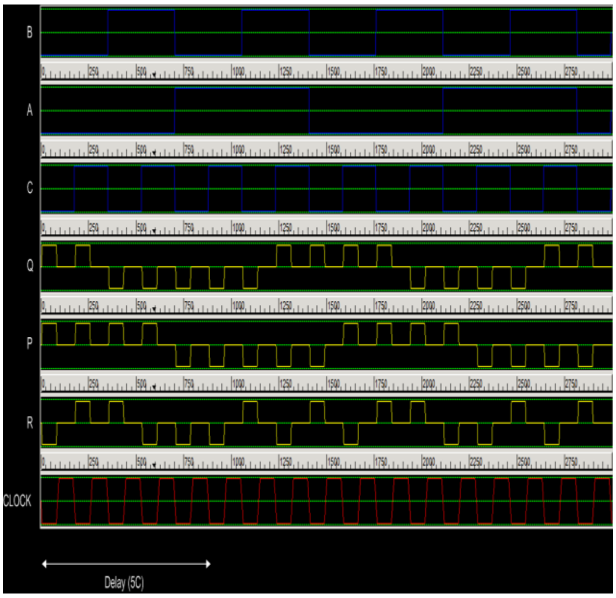

Fig. 20. Simulation response of PERES gate

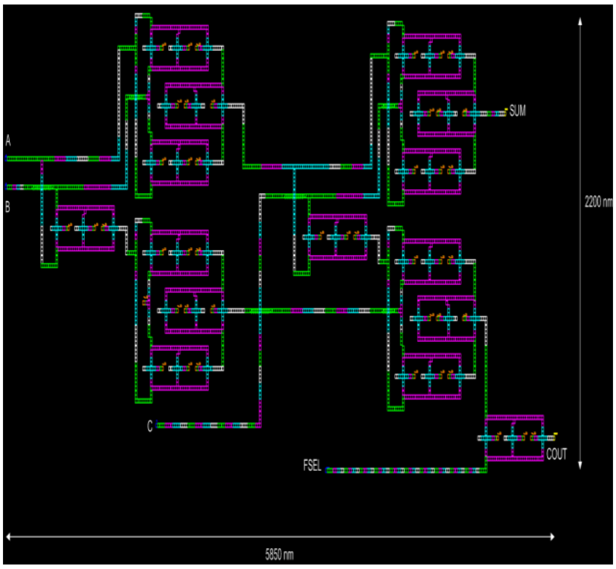

Fig. 21. QCA implementation of Dual-field adder

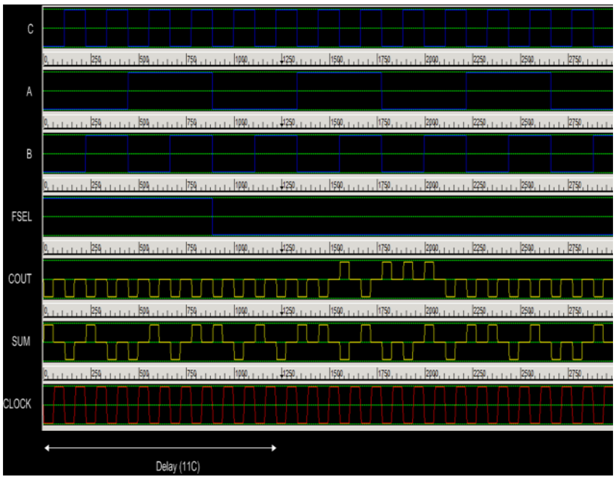

Fig. 22. Simulation response of Dual-field adder

6 Performance Analysis

The performance metrics of the QCA implementation are assessed in terms of Complexity (number of cells), required area, propagation delay (latency), number of ancilla inputs and garbage outputs. Table 6. gives the complete analysis results of all the QCA implemented reversible logic gates.

Table 6. Performance Analysis of QCA implemented Reversible Gates

Type	Cell Complexity	Area (nm)	Latency (Clocks)	Ancilla Inputs	Garbage Outputs
CNOT	552	1218x980	3	3	6
CCNOT	976	2452x1200	5	4	8
FREDKIN	1329	2555x1275	5	6	12
SWAP	1730	3810x980	9	9	18
PERES	1474	2452x2022	5	7	14
DF ADDER	3110	5850x2200	11	15	30

Considering the universal nature of reversible gates, CCNOT and FREDKIN gates are most often realized and used in reversible logic synthesis. But from the QCA implementation point of view, complexity of CCNOT gate is far better than FREDKIN gate in all aspects. So CCNOT gate is found to be the best candidate for reversible logic synthesis compared to all other gates. An example circuit named Dual-field adder has been synthesized using family of CCNOT gates and its performance has been analyzed and tabulated.

7 Conclusion

For the first time in literature, a novel QCA implementation has been proposed for reversible gates in this paper. All conventional reversible gates such as CNOT, CCNOT, FREDKIN, SWAP and PERES gates have been implemented in QCA and the simulation results indicate that CCNOT gate is more efficient in all aspects such as cell complexity, area, number of ancilla inputs and garbage outputs when compared to its counterpart FREDKIN gate. Hence CCNOT gate is found to be the best candidate for reversible logic synthesis in QCA implementation.

The combination of reversible logic synthesis and its QCA implementation can be a good countermeasure for side channel attack based on power analysis in security applications. A dual-field adder which plays a vital role in public-key crypto architectures has been synthesized in reversible logic and implemented in QCA. The proposed QCA implementation of dual-field adder takes 15 ancilla inputs, 30 garbage outputs and has a latency of 11 clock cycles. Currently, we are working on the QCA implementation of a complete crypto-processor in reversible logic.

References:

[1] C. S. Lent, P. D. Tougaw, Porod, et al., "Quantum cellular automata," *Nanotechnology*, vol.4, no. 1, pp. 49, 1993.

[2] Amlani, I. Orlov, A. O. Toth, et al., "Digital logic gate using quantum-dot cellular automata," *Science*, vol. 284, no. 5412, pp. 289-291, 1999.

[3] Kummamuru, R. K. Orlov, A. O. Ramasubramaniam, et al., "Operation of a quantum-dot cellular automata (QCA) shift register and analysis of errors," *IEEE Transactions on Electron Devices*, vol. 50, no.9, pp. 1906-1913, 2003.

[4] Vikramkumar Pudi, K. Sridharan, "Efficient Design of a Hybrid Adder in Quantum-Dot Cellular Automata," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 9, pp. 1535-1548, 2011.

[5] Vikramkumar Pudi, K. Sridharan, "Low Complexity Design of Ripple Carry and Brent–Kung Adders in QCA," *IEEE Transactions on Nanotechnology*, vol. 11, no. 1, pp. 105-119, 2012.

[6] Mostafa Rahimi Azghadi, O. Kavehie, K. Navi, "A novel design for quantum-dot cellular automata cells and full adders," *Journal of Applied Sciences*, vol. 7, no. 22, pp. 3460-3468, 2007.

[7] Hänninen, Ismo, Jarmo Takala, "Binary multipliers on quantum-dot cellular automata," *Facta universitatis-series: Electronics and Energetics*, vol. 20, no. 3, pp. 541- 560, 2007.

[8] R. Landauer, "Irreversibility and heat generation in the computing process," *IBM journal of research and development*, vol. 5, no. 3, pp. 183-191, 1961.

[9] Bennett, H. Charles, "Logical reversibility of computation," *IBM journal of Research and Development*, vol. 17, no. 6, pp. 525-532, 1973.

[10] Tommaso Toffoli, "Reversible computing," *Springer*, Berlin, Heidelberg, pp. 632-644, 1980.

[11] Himanshu Thapliyal, Ranganathan Nagarajan, "Design of reversible sequential circuits optimizing quantum cost, delay, and garbage outputs," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 6, no. 4, pp. 14, 2010.

[12] Majid Mohammadi, Aliakbar Niknafs, Mohammad Eshghi, et al, "Design and optimization of single and multiple-loop reversible and quantum feedback circuits," *Journal of Circuits, Systems and Computers*, vol. 21, no. 3, pp. 17, 2012.

[13] Himanshu Thapliyal, M. Zwolinski, "Reversible logic to cryptographic hardware: A new paradigm," *49th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS'06)*, San Juan PR, pp. 342-346, 2006.

[14] N. M. Nayeem, L. Jamal, H. M. H. Babu, "Efficient reversible Montgomery multiplier and its application to hardware cryptography," *Journal of computer science*, vol. 5, no. 1, pp. 49, 2009.

[15] Morita, Kenichi, "Reversible computing and cellular automata—A survey," *Theoretical Computer Science*, vol. 395, no. 1, pp. 101-131, 2008.

[16] Fredkin, Edward, Tommaso Toffoli, "Conservative logic, " *International Journal of Theoretical Physics*, vol. 21, no. 3,pp. 219-253, 1982.

[17] A. Peres, "Reversible logic and quantum computers," *Physical Review A*, vol. 6, no. 32, pp. 3266–3276, 1985.

[18] Mehdi Saeedi, Igor L. Markov, "Synthesis and optimization of reversible circuits-a survey," *Journal of ACM Computing Surveys (CSUR)*, vol. 45, no. 2, 2013.

[19] Barenco, A. Bennett, C. H. Cleve, et al., "Elementary gates for quantum computation," *Physical Review A*, vol. 52, no. 5, pp. 3457, 1995.

[20] D. Maslov, C. Young, D. M. Miller, et al., "Quantum circuit simplification using templates," *Design, Automation and Test*, Europe, pp. 1208-1213, 2005.

[21] Robert Wille, "An introduction to reversible circuit design," *Electronics, Communications and Photonics Conference (SIECPC)*, Riyadh, Saudi Arabia, pp. 1-4, 2011.

[22] Anwar Sohail, M. Yasin Akhtar Raja, Qazi Salahuddin, et al., "*Nanotechnology for telecommunications*," CRC Press, 2010.

[23] C. S. Lent, P. d. Tougaw, "A device architecture for computing with quantum dots," *Proceedings of the IEEE*, vol.85, no. 4, pp. 541-557, 1997.

[24] S. C. Benjamin, N. F. Johnson, "A possible nanometer-scale computing device based on an adding cellular automation," *Applied Physics Letters*, vol. 70, no.17,pp. 2321-2323, 1997.

[25] Wei Wang, Konrad Walus, G. A. Jullien, "Quantum-dot cellular automata adders," *Third IEEE Conference on Nanotechnology*, pp. 461-464, 2003.

[26] G. Toth, C. S. Lent, "Quasiadiabatic switching for metal-island quantum-dot cellular automata," *Journal of Applied Physics*, vol. 85, no. 5, pp. 2977-2984, 1995.

[27] Lent, Craig S., Mo Liu, and Yuhui Lu., "Bennett clocking of quantum-dot cellular automata and the limits to binary logic scaling," *Nanotechnology*, vol. 17, no. 16, pp. 4240, 2006.

[28] Liu, Weiqiang, et al., "Are QCA Cryptographic Circuits Resistant to Power Analysis Attack?," *IEEE transactions on nanotechnology*, vol. 11, no. 6, pp. 1239-1251, 2012.

[29] M. Miller, "Spectral and Two-Place Decomposition Techniques in Reversible Logic," *Proceedings of the IEEE Midwest Symposium on Circuits and Systems*, vol. 45, no. 2, pp. 493-496, 2002.

[30] Guowu Yang, William N. N. Hung, Xiaoyu Song, et al., "Majority-based reversible logic gates," *Theoretical Computer Science*, vol. 334, no. 1, pp. 259-274, 2005.

[31] K. Walus, T. J. Dysart, G. A. Jullien, et al., "QCADesigner: A Rapid Design and Simulation Tool for Quantum-Dot Cellular Automata," *IEEE Transactions on Nanotechnology*, vol. 3, no. 1, pp. 26-31, 2004.