Implementing the Toffoli gate in Quantum-dot Cellular Automata

Mayank Dubey[1],Aparajit Shrivastava[2]
[1, 2]Department of Computer Science and Engineering
[1, 2]ShriRam Group Of Colleges, Banmore (M.P)

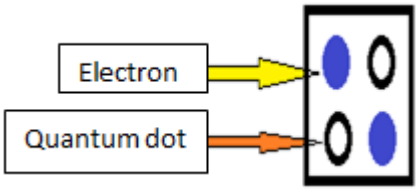
Abstract-*This paper presents results of our research work in course Unconventional information processing method that leads to a new approach of circuit designing in computer technology.*

I. INTRODUCTION

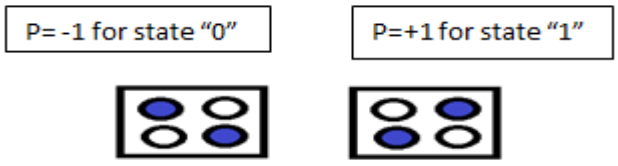
The idea of this study is to build the design of reversible Toffoli logic gate in QCA. For that manner, we created a practical implementation of the gate in the QCA Designer and gave some examples that demonstrate how the gate may be used as a part of more complicated reversible circuits.

1.1 QCA

A QCA is a finite state machine comprising of infinite or a finite grid of quantum-dot cells. A quantum-dot cell is a collection of 4 quantum dots located at corners of the cell in addition to an electron pair diagonally located. By way of provided that tunneling junctions with potential barriers, which are raised to prevent electron crusade and lowered to permitelectron movement. The electrons can localize on any dot and the Null state occurs When barriers are low, but the cell is polarized and other 2 states can occur when the barrier is raised. These 2 states represent the logic 1"and 0". Due to Coulombic interactions, cells which are situated near every other are forced into matching polarizations. The proliferation of polarization provides information transfer.



(a) Quantum-dot cell.



(b) The two different states of a QCA cell.

Fig. 1 QCA cell.

1.1.1 Basic building structures

Basic structure in QCA technology is the wire and the majority gate or majority voter (MV). Other structures like AND and OR can be implemented using the MV by setting one of its inputs to a constant value.

1.1.2 Clock

The proliferation of signals in all QCA circuits is driven by a codlock signal. Each QCA cell belongs into one of four clock zones. The clock signal periodically goes through four phases. There are 2 transitory phases that separate the two main phases, the hold phase when the cell holds onto its value and the release phase when the cell assumes a new value. The clock signals in the four clock zones are shifted from all other by a quarter of a period. The difference from clock in a CMOS circuits is that in QCA the clock is an external electric field that drives each individual cell, while in CMOS this is a signal inside the circuit that is used to synchronize larger structures.

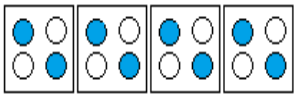
1.1.3 Wire

The simplest practical cell prearrangement is given by placing QCA in sequence, to the side of every further. Such as, Figure 2b (p. 3) shows a 90 degrees wire. When the dots in wire are rotated by 45 degrees this arrangement is called the 45-degree wire. Wire crossing on further hand can be done using two quantum-dot wires (one 90 degrees wire and one 45 degrees wire). The wire composed of one type passes perpendicularly through" a wire of the other type, Figure 2a (p. 3). The first type of wire always propagates the same polarization, but the second type changes the polarization from one cell to the next. But, at crossing point there is no polarization change in either wire, therefore both wires preserve their own information.

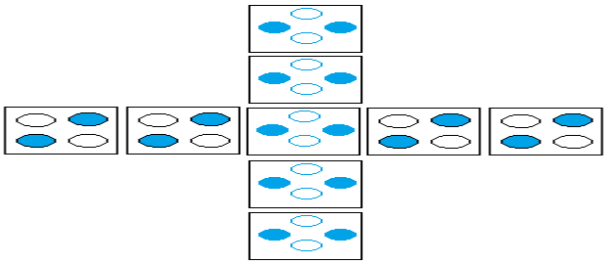
1.1.4 Majority gate

As discussed earlier the most important logic gate in QCA is majority gate. The electrical field result of every input on output is additive and identical, with outcome that whichever input state (\binary 0" or \binary 1") is in mainstream becomes the state of output cell | hence the gate's

name. Majority gate can be used to implement logic conjunction (AND) and disjunction (OR), which together with negation (NOT) gives us a complete logic system in this structure.

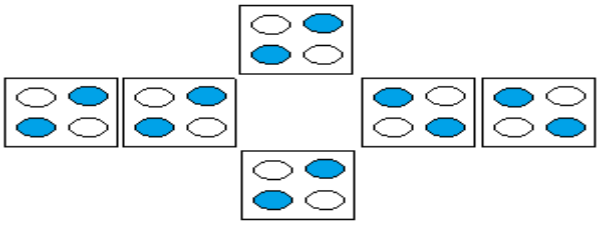


(a) 90-degree wire

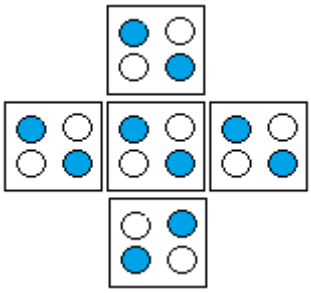


(b) Wire crossing.

Figure 2: Wires



(a) Negation



(b) Majority gate

Figure 3: Negation and majority gate.

1.1.5 Negation

The NOT gate is not constructed using the MV gate. It has a single input and output and it simply returns the opposite of the input. A standard implementation of the NOT gate is given in Figure 3a (p. 3). Negation can be also implemented by having two cells (both either 90-degree or 45-degree) touch at the corner. Another possibility for negation is when a 45-degree wire connects to a 90-degree wire.

1.2 Reversibility

A rescindable logic gate is an n-output n-input logic device with one-to-one mapping. It helps to define outputs from input and the inputs can be uniquely recovered from outputs. The simplest example of a reversible logic function is negation. An example of a nonreversible logic function is logical conjunction or disjunction, since there are several input configurations that produce the same output.
(a; b; c) = Toffoli(Toffoli(a; b; c))

As with the NAND function which is universal there has been a lot of research trying to devise interesting universal reversible gates. Using only one kind of a gate in a circuit simplifies manufacturing process.

1.3 The Toffoli gate

Toffoli gate is an universal rescindable logic gate suggested in by Tommaso Toffoli and can be used in construction of any reversible circuit. Its structure is as follows. The gate has three inputs that can be labeled a, b, c and three outputs y1, y2, y3 described by equations in Figure 4b (p. 5).

The working of Toffoli gate may be also described by equations in Figure 4c (p. 5). The Toffoli gate negates its first input if both the 2nd and 3rd input are equal to logic \1". The 2nd and 3rd input are passed through unchanged. The input values that are passed through unchanged (only to provide reversibility) are sometimes called garbage outputs [1].

1.4 QCADesigner

Before we proceed to implementation of Toffoli Gate, we would like to add a few introductory words on the tool we used for drawing QCA structures. QCA Designer [6] is a simulation computer program and design for QCA circuits developed by Walus Group at the University of British Columbia. It allows to design and simulate QCA circuits. It is written using the GTK2 graphic library and it is published under General Public Licence. The official pages provide source code as well as prebuild binaries for Windows and deb and rpm packages for GNU/Linux.

Input			Output		
a	b	c	Y1=ab xor c	Y2=a	Y3=b
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	0	1	1

Table 1.1(a) Fig. Toffoli gate.

II. IMPLEMENTING THE TOFFOLI GATE

In QCA Designer We implemented Toffoli gate in 3 stages. First we implemented the equation describing the first output, then we added other 2 outputs and finally we modified the design to make it more compact while keeping the functionality unaffected.

We found a circuit implementing the XOR function in paper by Shah et al. [5] that was provided to us by prof. Mraz. Together with basic QCA blocks described earlier we now have all the components that are wanted to implement Toffoli gate.

2.1 Previous work

Before implementing the gate ourselves, we searched the literature for existing implemen- tations. We found figures depicting Toffoli gate in a paper by Chandra and Netam and also in another paper by Mohammadi et al.. Designs in those two papers are identical and do not actually implement Toffoli gate even though both papers claim to be so. Still, the illustrations in the papers proved useful to demonstrate some of the design techniques used together in a larger example. Next we studied the diploma thesis by Rolih [4] which is mainly concerned with three state logic but two state logic is also discussed. A working design of the Toffoli gate is portrayed in this paper in Figure 4.5 (p. 29). Compared to this design, the one we come up with uses more cells (101 compared to 44) and the computation takes longer (1 14 clock cycle compared to 1). On further hand, since outputs and inputs in our design are easily accessible, it can be more easily integrated as a component into a larger circuit.

2.2 Designing the first output

The function for the first output was obtained by connecting inputs a and b to an AND gate, output of which was connected to a XOR gate, together with input c.

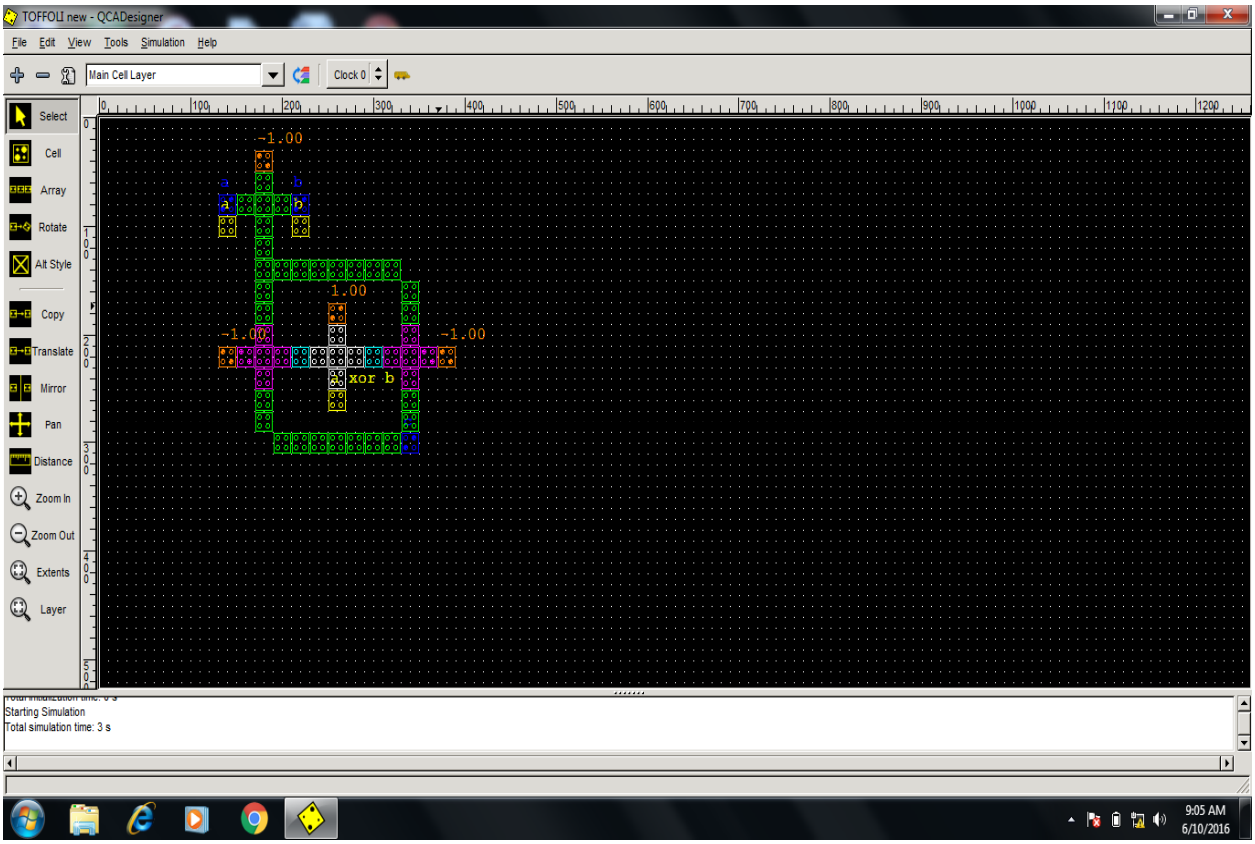


Fig. Toffoli gate using QCA designer

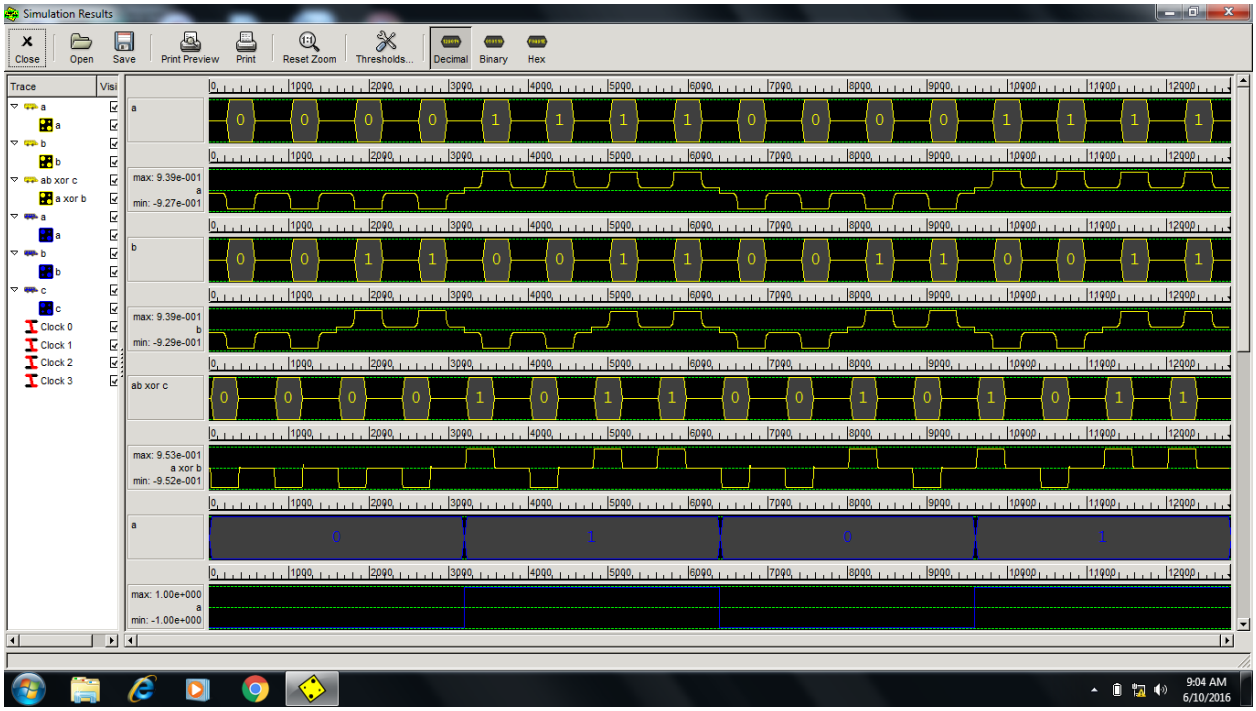


Fig. Output of toffoli gate using QCA designer

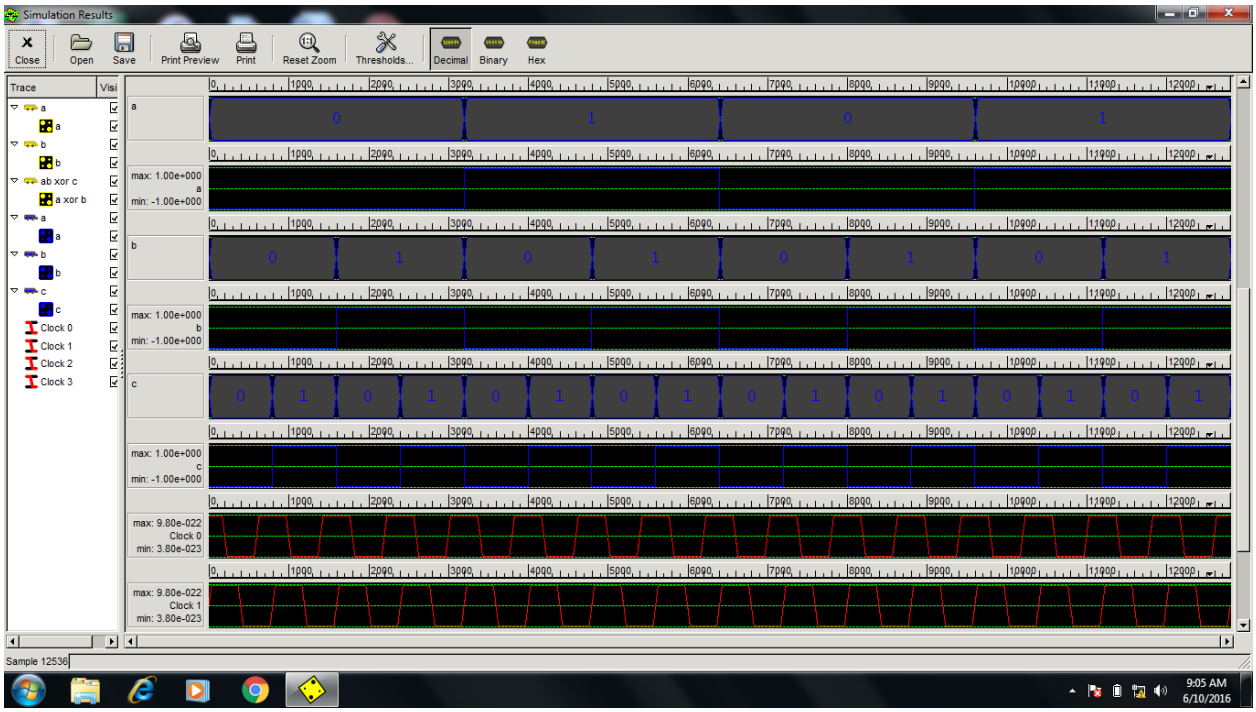


Fig. Output showing inputs a,b,c using QCA designer.

2.3 Adding the other two outputs

To make the design look intuitive and to simplify connecting gates we decided to have all inputs on left side of our circuit and outputs on the right side. We haven't used any wire crossing to make the design simpler.

2.4 Problems

We have not succeeded in running QCADesigner in Linux. The binary packages used out of date shared libraries and the source distribution did not compile successfully. Even though we eventually managed to get it compile, when executed it crashed immediately after the splash screen. We decided to use it on Windows XP as well as Windows 7 machines. While designing the circuit, we did not have to debug it per se. Rather we sometimes needed to correct our

misconceptions about how the QCA works. We describe one of them in the following subsection.

2.4.1 Effects of QCA clocking

Even though QCADesigner does not visually differentiate it by color, the output elements belong to and are affected by clock zones. One of our problems we had was that a single wrong clock on any cell can cause the distorted signal in result which totally changes the output. Because it was one clock zone after the right one, the signal looked distorted. So clocks are specially considered on each cell.

III. USE OF TOFFOLI GATE

Toffoli gate is universal logic gate, meaning any logic function may be constructed using only Toffoli gates. This is easily proven by constructing a NAND gate using only a Toffoli gate or alternatively by constructing AND, OR and NOT gates, another complete logic system. To simplify the equations, we define a supposed Toffoli gate which is a Toffoli gate without the two garbage outputs.

Toffoli(a; b; c) =(:a if a = b = 1, otherwise a.
With our new Toffoli' gate, NAND(a; b) can be constructed as
NAND(a; b) = Toffoli(1; a; b);
NOT(x) becomes
NOT(x) = Toffoli(x; 1; 1);
finally the AND(a; b) gate is
AND(a; b) = Toffoli(0; a; b):

Having constructed these gates, the OR(a; b) gate can be then constructed from the previous gates using one of the De Morgan rules
OR(a; b) = :(:a ^ :b)
= Toffoli'(Toffoli'(Toffoli'(a; 1; 1);Toffoli'(b; 1; 1); 0); 1; 1):

IV. CONCLUSION

This paper presents the precise design of Toffoli gate using quantum dots in QCA designer. The total transmission of information requires minimum a lots of cells (only 56 cells) to perform operation with minimum area (0.10um2) required by the cells. The design shows the less distorted and more precisely obtained signals. The proposed design is energy efficient and maximizes the possibility of increasing functionality of the circuit. When reversible circuits are designed, we usually used the standard gates only as a first step and develop modified reversible gates that allowed us to minimize amount of garbage outputs in precise problem they are solving and use less cells. In future this design can be compared with the other efficient designs when developed. To check the functionality difference we can increase many cells and analyze the performance.

REFERENCES

[1] MdSelim Al Mamun, IndraniManda, and MdHasanuzzaman. Design of universal shift register using reversible logic. 2012.

[2] Saroj Kumar Chandra and Deepak Kant Netam. Exploring quantum dot cellular automata based reversible circuit. interaction, 2(1March), 2012.

[3] Zahra Mohammadi, MajidMohammadi, and Mahdi Hasani. Designing of testable reversible qca circuits using a new reversible mux 2_.

[4] Mark Rolih. Analizamo_znostirealizacijelogi_cnihreverzibilnihvrat v trostanjskemkvantnemceli_cnemavtomatu. PhD thesis, Univerza v Ljubljani, 2013.

[5] NA Shah, FA Khanday, and J Iqbal. Quantum-dot cellular automata (qca) design of multi-function reversible logic gate. Communications in Information Science and Management Engineering.

[6] KonradWalus, Timothy J Dysart, Graham A Jullien, and R ArietBudiman. Qcade- signer: A rapid design and simulation tool for quantum-dot cellular automata. Nano technology, IEEE Transactions on, 3(1):26{31, 2004.