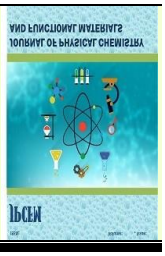
Journal of Physical Chemistry and Functional Materials

Home Page of Journal: https://dergipark.org.tr/jphcfum



Simulation of 4-Qubit Full-Adder Circuit by Mathematica

[a,b] *Shakhawan Salih Abdullah*

[a] *Erbil Polytechnic University, Soran Technical Institute, IT Department, Erbil, IRAQ*

[b] *Gaziantep University, Engineering Faculty, Department of Physics, Gaziantep, Turkey*

Revised: 21-May-2019, **Accepted**: 24-May-2019
ISSN: 2651-3080

ABSTRACT

A correct simulation of a quantum circuit on a classical computer is more important because of their future use. The main purpose of this work is to illustrate a full adder circuit by using a standard Mathematica add-on package. The circuit can be constructed by using CNOT-based quantum gates. The program provides a curriculum unit, to generate the basic elements that make up quantum circuit. This paper shows effective computational design by using analogy of classical circuits. We presented an explicit example to show efficiency of the 4 qubit full adder circuit on classical computer. The method given in this paper can be used to design various quantum circuits.

ARTICLE INFO

Keywords:

Quantum Bit
Qubit
Full adder
Mathematica
Simulation

1. Introduction

There are plenty theoretical [1-5] and experimental study [6-8] that work for buildup quantum computer based on quantum mechanical postulates. The study in the same problems that faced traditional computers through quantum system technology is one of the significant issues that hope technologies for future computing systems.

It has been a great deal of interest for simulating a quantum algorithm on a classical computer. A basic quantum computer has already been built but practically quantum computer has not yet been built. Scientists need to simulate quantum algorithms on classical computers. Consequently, various general-purpose quantum computer simulators have been developed. Recently, several Mathematica-based [9] quantum algorithm simulators have appeared, including Quantum: Mathematica add-on for simulating quantum algorithms [10]. Although running time for such simulators increases exponentially by increasing the number of qubits, many quantum algorithms including few qubits can be simulated efficiently on a classical computer.

Theoretically, quantum computing allows solving problems much faster than classical computing, e.g. N steps need to search an unstructured database for solving one problem with a classical algorithm, but in the quantum Grover algorithm needs only $\sqrt{N}$ steps [11].

In this study simulation a four qubit full adder circuit was simulated by using a Mathematica package developed by J. L. Gómez-Muñoz and F. Delgado [12]. A full adder is an essential component of a classical computer and also an asset component for quantum computers [13]. Although a considerable attention has been paid to present quantum algorithm for full adder circuit, the study of this problem from different point of view leads to the progress of quantum algorithm and simulation techniques. Among various quantum circuits, CNOT-based circuits have attracted widely attentions in the literature [14], and likewise, this study algorithm include CNOT-based quantum circuits. Note that as a part of the development of quantum computing, it is necessary to find efficient ways to design a quantum circuit. According to the quantum theory, quantum logic circuit represents unitary transformations of the state of one or more qubits over time or space. These circuits are modeled as a cascade of one or more quantum logic gates represented by a unitary transformation matrix [15, 16].

The paper is organized such that basics of quantum computation and quantum gates used to construct full adder circuits is briefly reviewed, and a simple quantum circuit and its unitary matrix was represented. In addition, it is devoted to present an algorithm for addition of four qubit numbers using a quantum computer. Classical full adder circuit is briefly summarized and quantum algorithm based on analogy of the classical algorithm is constructed. A Wolfram Mathematica program (WMP) is prepared using Quantum: Mathematica add-on for simulating quantum algorithms [12] is used to simulate four qubit full adder circuit.

* Corresponding author:
E-mail (shakhys74@gmail.com),
Phone number (+964750-448-9550),

2. Theoretical Background

Quantum computation is based on principles of quantum mechanics. In quantum mechanics a quantum state (or qubit) can be typically obtained from the state of a two-level quantum system. As an example ground state and excited state of an atom or the vertical and horizontal polarizations of a single photon are represented as qubits. The qubits are denoted by using Dirac notation such as one of these states as $|0\rangle$ and the other as $|1\rangle$ [17].

According to the theory of quantum mechanics the states can be written as linear combinations of these pure states, which is called superposition, and it is the most significant property that speed up computation based on quantum. In other words, the state of a qubit ψ can be written as $\psi=\alpha|0\rangle+\beta|1\rangle$, where α and β are complex numbers and satisfy $\alpha^2 + \beta^2 = 1$. This implies that by performing a single operation, on the state (ψ), both qubits will be affected at the same time. Similarly, a two-qubit system can perform operation on a four-qubit input, three qubit system can perform operation on eight qubits, and consequently, an n qubit system can perform operation on 2^n qubits. This is known as quantum parallelism [18] and by a sufficient algorithm one can use this property to speed up quantum computer exponentially compared to a classical computer.

There are various quantum gates with different functionalities that can be used for constructing a quantum circuit, including identity (I), NOT, CNOT, C^2NOT and SWAP gates. Icons of the gates are given in the Figure 1, where each symbol •, ⊕ and | are used for control, target and contact qubits, respectively. The operation of each gates are as follows:

- Identity gate (I) with matrix M_I that does not act on the qubits. Its icon is a horizontal wire.
- NOT gate inverts the working qubit and its action is given by the matrix M_{NOT}.
- CNOT gate work such that if the control qubit is $|1\rangle$, then the target qubit is inverted, otherwise it remains constant. It is action on qubits can be obtained by using the matrix M_{CNOT}.
- SWAP gate exchanges the values of input qubits.

C^2NOT gate is controlled-CNOT gate, also known as Toffoli gate, can be described as: if both control qubits are $|1\rangle$, the target is inverted; otherwise, it remains the same.

As aforementioned before, quantum gates are represented by unitary matrices and the circuits are also represented by unitary matrices. Such circuits are called unitary stabilizer circuits [20]. For example, in Figure 2, NOT gate combined with identity gate. Matrix representation of combined gates can be obtained by direct product of M_I and M_{NOT}. In addition, Figure 3 shows the Cascading quantum gates to construct a quantum circuit.

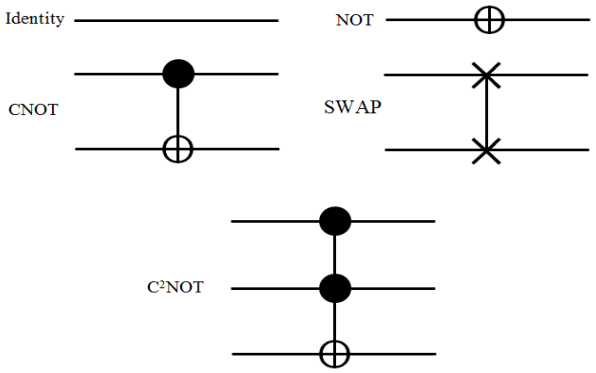


Figure 1. Basic quantum gates [19].

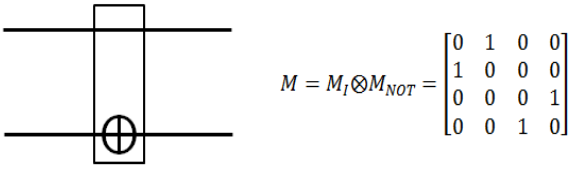


Figure 2. A compound gate constructed from an identity and a NOT gate [14].

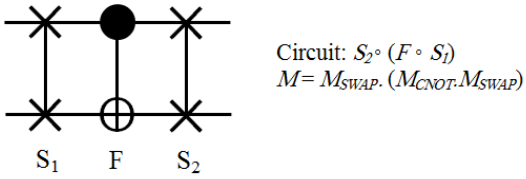


Figure 3. Cascading quantum gates to construct a quantum circuit and its QMatrix [14].

3. Construction of Classical and Quantum Full Adder Circuit

In order to construct a quantum full adder circuit, there is an analog to classical full adder circuits. A classical full adder operates with an input of two addend bits, "*A*" and "*B*", and a carry bit, "C_{in}" (Figure 4), where *S* and C_{out} are the output "sum" and the "carry-over", respectively. The sum (*S*), can be easily expressed as $S = A \oplus B \oplus C_{in}$ with ⊕ is an addition modulo 2. The true table of full-adder is given in Table 1. Also Table 2 gives input combinations that produce the same output combinations in full adder circuit

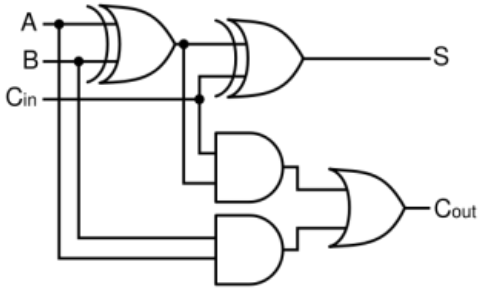


Figure 4. Classical full-adder circuit.

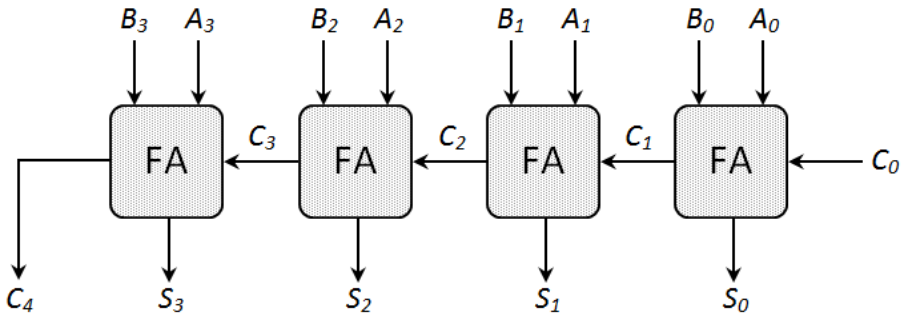


Figure 5. Parallel 4-bit binary Adder [22].

Table 1. Truth Table of classical full-adder.

Input			Output	
A	**B**	**C_{in}**	**S**	**C_{out}**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

C_{out} can be easily obtain by following expression [21]:

$$C_{out} = (A \oplus B)C_{in} + AB \quad (1)$$

In order to construct a full adder circuit for more than 1 binary digit, we connected classical full adder circuit in cascade as shown in Figure 5.

The circuit in Figure 5 performs calculation of two binary numbers of digits ($A_3A_2A_1A_0$ and $B_3B_2B_1B_0$) with initial carry $C_0 = 0$. Therefore, the classical version of full adder circuit does not operating unitary. A classical gate that can perform a unitary transformation on inputted bits are classical CNOT gate [23].

Adder circuits are a key element in any computational logic unit. In order to find an analogy between classical and quantum computation, it is worth to test reversibility of classical circuit [24-26]. Logic equation for CNOT gate is given by:

$$Sum = A \oplus B \text{ and } C_{out} = A\,B \quad (2)$$

A reversible half-adder can be constructed by using two reversible gates (Figure 6). This combination gate corresponds to a Peres gate [27].

A 1-bit full-adder takes two binary digits (A, B) and a carry-in (C_{in}) as input. Its mathematical representation is given as follows:

$$Sum = A \oplus B \oplus C_{in} \quad (3)$$

$$C_{out} = AB \oplus (A \oplus B)C_{in} \quad (4)$$

Figure 7 demonstrates a reversible full-adder circuit that has been made by using four CNOT based gates.

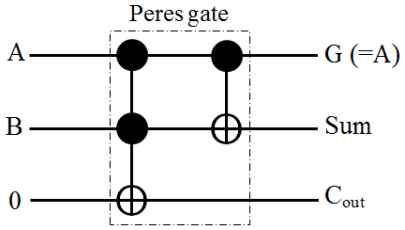


Figure 6. Reversible 1-bit half-adder.

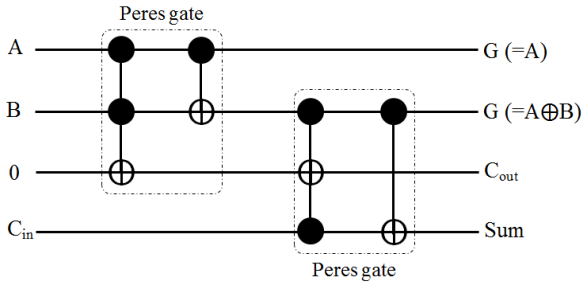


Figure 7. Reversible 1-bit full-adder.

Furthermore, appropriate combinations of the 1-bit half and full-adder, provides a reversible *n*-bit half and full-adders (Figure 8).

We have shown that a reversible full adder circuit can be constructed by using CNOT and Controlled CNOT gates represented by a unitary matrix.

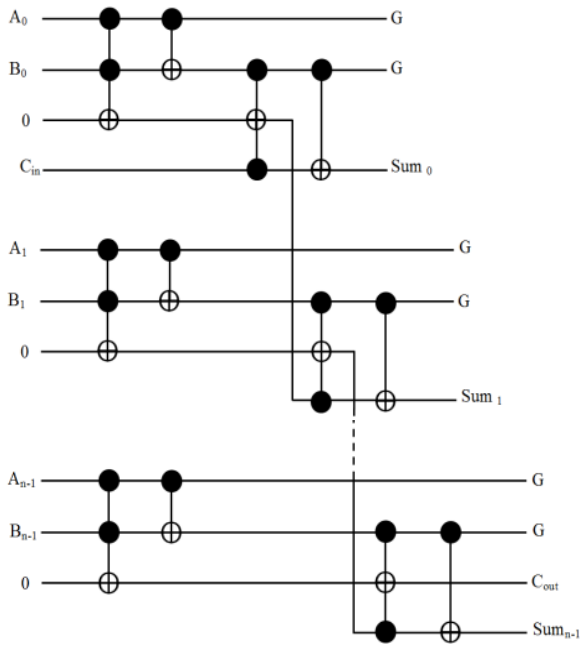


Figure 8. Reversible *n*-bit Full adder.

Table 2. Input combinations that produce the same output combinations in full adder circuit (shown shaded).

Input				Output			
A	B	C_{in}	C1	S	C_{out}	G1	G2
0	0	1	0	1	0	0	0
0	1	0	0	1	0	0	1
1	0	0	0	1	0	1	0

4. The results of Quantum Circuit Simulator

To simulate four bit quantum full adder circuit a programed Mathematica package was used, which is called Mathematica Add-On package, that presented for Dirac Notation, Noncommutative Algebra of Operators and Commutators, Quantum Computing, and Plotting of Quantum Circuits [9]. Also, in order to fully utilize all that quantum circuits, it was necessary to design a circuit simulator that had to be efficient and accurate [23]. Firstly, a half-adder circuit was designed.

4.1 Simulation of Half-Adder

It is easy to use the WMP to construct a unitary circuit. The schematic diagram of the quantum circuit can be drawn by using the command **QuantumPlot[]**, and operation of the circuit on the qubits can be tabulated by using the command **QuantumTableForm[]**.

The half adder circuit and its operation is illustrated in Figure 9, whereby, lines 1, 2 and 3 represents input and output of the circuit. Synthesis of input-output relation of the circuit are summarized in Table 3.

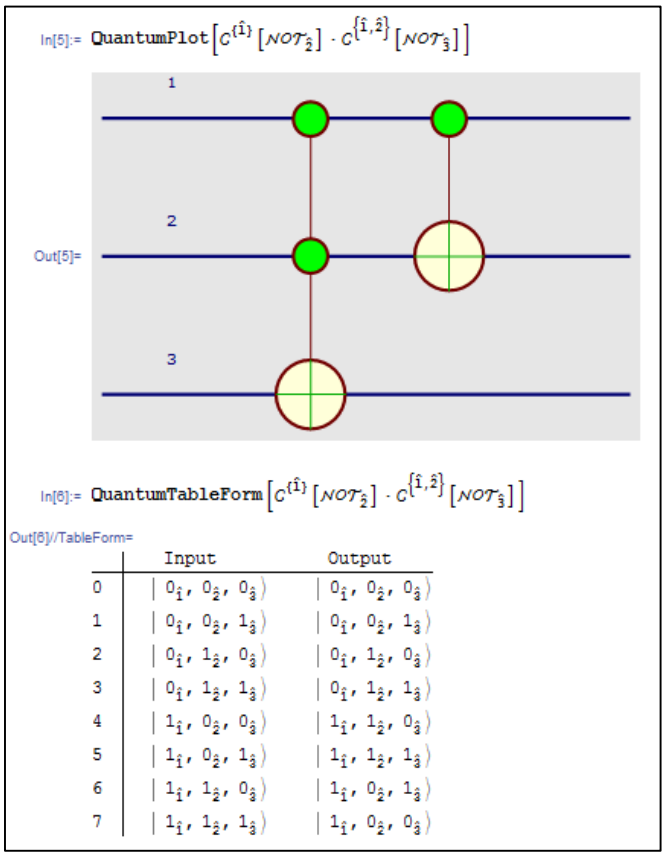


Figure 9. Simulation Quantum Half-Adder and obtained Mathematica results.

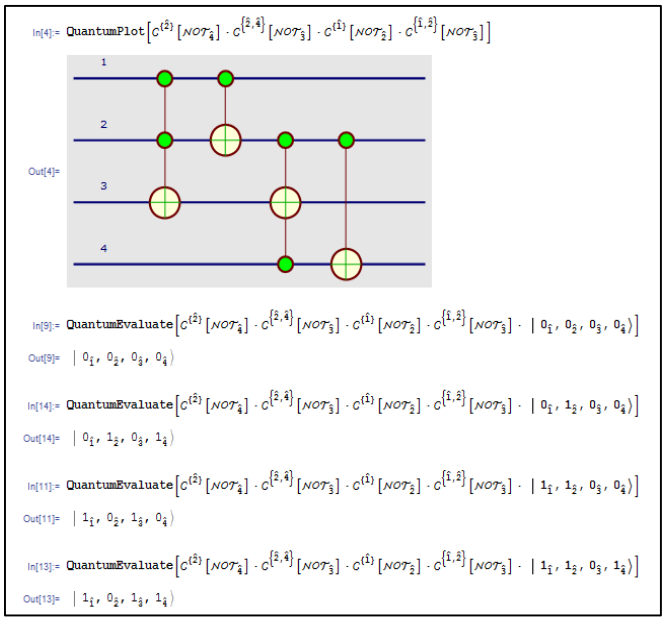


Figure 10. Simulation Quantum Full-adder with result by Mathematica.

Table 3. Synthesis of input and output Quantum Half-Adder.

Input	Output
Line 1 = First Input bit (A)	Line 1 = Garbage
Line 2 = Second Input bit (B)	Line 2 = Sum
Line 3 = 0	Line 3 = C_{out}

Table 4. Synthesis of input and output Quantum Full-Adder.

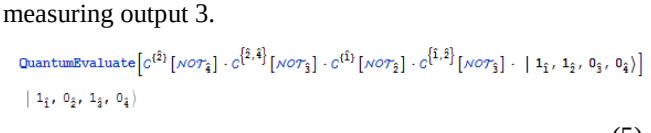
Input	Output
Line 1 = First Input bit (A)	Line 1 = Garbage
Line 2 = Second Input bit (B)	Line 2 = Garbage
Line 3 = 0	Line 3 = C_{out}
Line 4 = C_{in}	Line 4 = Sum

4.2. Simulation Quantum Full-Adder

Similar to the design of half-adder circuit we constructed a full adder circuit. In the circuits input qubits are applied to lines 1 and 2. Input of the line 3 is always 0, while carry input is applied to line 4. Sum of the numbers are appears on output part of line 4 and carry appears on output line 3. The relation between inputs and outputs are given in the Table 4.

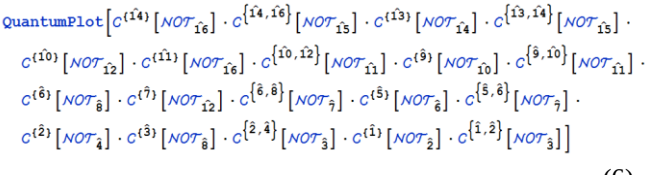
On the other hand, to evaluate action of the circuit on a given input state, one can use the command **QuantumEvaluate[]**. Action of the full adder circuit on various states (qubits) are given in Figure 10.

The following Mathematica line illustrates summation of qubits (1) and (1) with (0) carry input. The sum is obtained by measuring the output 4 and carry can be determined by measuring output 3.



(5)

Using the full adder circuit we can design a 4 qubit quantum full adder circuit by writing the following code in Mathematica Add-On program.



(6)

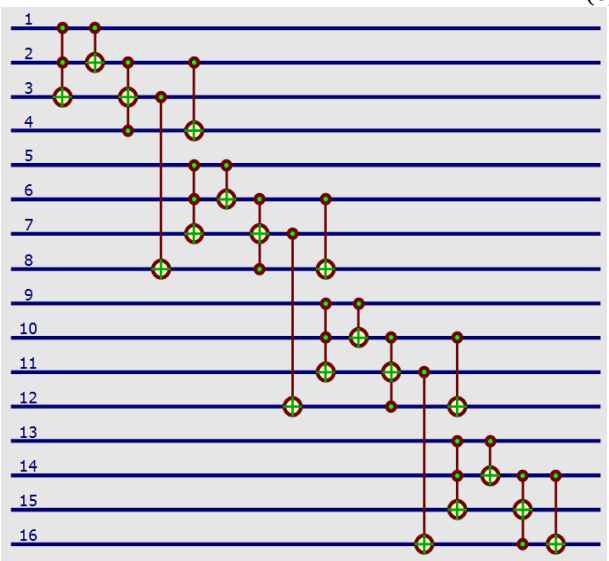


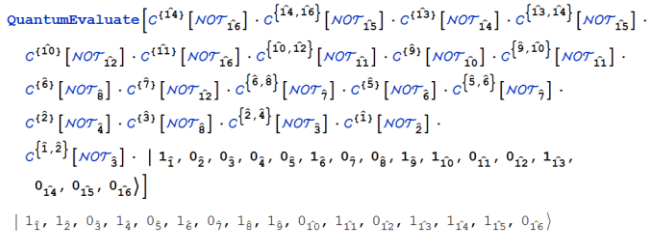
Figure 11. Simulation 4-qbit Adder by Mathematica.

Figure 11 gives the output of the Mathematica code for the quantum full adder circuit. The corresponding action of the circuit on input qubits are summarized in Table 5.

Table 5. Synthesis of input and output 4-qbit Adder.

Input	**Output**
Line 1 = First bit Input (A_0)	Line 1 = Garbage
Line 2 = Second bit Input (B_0)	Line 2 = Garbage
Line 3 = 0	Line 3 = C_{out}
Line 4 = C_{in}	Line 4 = Sum_0
Line 5 = First bit Input (A_1)	Line 5 = Garbage
Line 6 = Second bit Input (B_1)	Line 6 = Garbage
Line 7 = 0+C_{out}(Output line 3)	Line 7 = C_{out}
Line 8 = C_{in}	Line 8 = Sum_1
Line 9 = First bit Input (A_2)	Line 9 = Garbage
Line 10 = Second bit Input (B_2)	Line 10 = Garbage
Line 11 = 0+C_{out}(Output line 7)	Line 11 = C_{out}
Line 12 = C_{in}	Line 12 = Sum_2
Line 13 = First bit Input (A_3)	Line 13 = Garbage
Line 14 = Second bit Input (B_3)	Line 14 = Garbage
Line 15= 0+C_{out}(Output line 11)	Line 15 = C_{out}
Line 16 = C_{in}	Line 16 = Sum_3

As a specific example, the following Mathematica command gives action of the circuit on the input state.



(7)

5. Conclusion

The Mathematica add-on presented in this work utilizes an irreducible form of output decomposition of a general controlled quantum gate with addition conditionals and a highly efficient to simulate complex quantum circuits. Another important application in which large and complex circuit need to be efficiently simulated is in the area of quantum error correction. This demonstrates a part of a general framework for simulation of quantum computers on a classical computer.

Acknowledgements

Many thanks for Erbil Polytechnic University/Soran Technical Institute" for financial support and thanks for Prof. Dr. Ramazan Koç for scientific help. It has been prepared from master thesis of S.S. Abdullah.

Reference

[1] A. Montanaro, Quantum algorithms: an overview, npj Quantum Information, 2 (2016) 15023.

[2] I.N. QADER, R. KOC, Simulation of Controlled Physical Quantum Gates by using Mathematica, International Journal of Computer Science and Network Security, 14 (2014) 59-65.

[3] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, S. Lloyd, Quantum machine learning, Nature, 549 (2017) 195.
[4] I.N. QADER, Physical Realization of Controlled Quantum Gates, in: Engineering Physics, Gaziantep University, 2013.
[5] S.S. Abdullah, Simulation of Quantum Computers on Classical Computers by Using Mathematica, in: Physics, Gaziantep University, Turkey, 2012.
[6] D. Alsina, J.I. Latorre, Experimental test of Mermin inequalities on a five-qubit quantum computer, Physical Review A, 94 (2016) 012314.
[7] B.K. Behera, A. Banerjee, P.K. Panigrahi, Experimental realization of quantum cheque using a five-qubit quantum computer, Quantum Information Processing, 16 (2017) 312.
[8] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J.M. Chow, J.M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature, 549 (2017) 242.
[9] S. Wolfram, Wolfram research, Inc., Mathematica, Version, 8 (2013) 23.
[10] J.G. Munoz, F. Delgado, QUANTUM: A Wolfram Mathematica add-on for Dirac Bra-Ket Notation, Non-Commutative Algebra, and Simulation of Quantum Computing Circuits, in: Journal of Physics: Conference Series, IOP Publishing, 2016, pp. 012019.
[11] A. Tulsi, Faster quantum searching with almost any diffusion operator, Physical Review A, 91 (2015) 052307.
[12] S.E. Venegas-Andraca, Quantum walks: a comprehensive review, Quantum Information Processing, 11 (2012) 1015-1106.
[13] M.H. Khan, A recursive method for synthesizing quantum/reversible quaternary parallel adder/subtractor with look-ahead carry, Journal of Systems Architecture, 54 (2008) 1113-1121.
[14] M. Saeedi, M.S. Zamani, M. Sedighi, Algebraic characterization of CNOT-based quantum circuits with its applications on logic synthesis, in: 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD 2007), IEEE, 2007, pp. 339-346.
[15] D. Goodman, A quantum circuit simulator based on decision diagrams, in, Southern Methodist University, 2007.
[16] S. Maity, A. Pal, T. Roy, S.B. Mandal, A. Chakrabarti, Design of an efficient quantum circuit simulator, in: 2010 International Symposium on Electronic System Design, IEEE, 2010, pp. 50-55.
[17] P.A.M. Dirac, A new notation for quantum mechanics, in: Mathematical Proceedings of the Cambridge Philosophical Society, Cambridge University Press, 1939, pp. 416-418.
[18] D. Deutsch, R. Jozsa, Rapid solution of problems by quantum computation, Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences, 439 (1992) 553-558.
[19] A. Steane, Quantum computing, Reports on Progress in Physics, 61 (1998) 117.
[20] S. Aaronson, D. Gottesman, Improved simulation of stabilizer circuits, Physical Review A, 70 (2004) 052328.
[21] P. Nyman, Representation of Quantum Algorithms with Symbolic Language and Simulation on Classical Computer, in, School of Mathematics and System Engineering, Växjö University, 2008.
[22] S.S. Abdullah, Design and Implementation of a Tutorial Binary Adder/Subtractor, in: Physics, Duhok University, Iraq, 2007.
[23] P. Gossett, Quantum carry-save arithmetic, arXiv preprint quant-ph/9808061, (1998).
[24] T.G. Draper, Addition on a quantum computer, arXiv preprint quant-ph/0008033, (2000).
[25] M.A. Fahdil, A.F. Al-Azawi, S. Said, Operations algorithms on quantum computer, IJCSNS, 10 (2010) 85.
[26] M.S. Islam, M.M. Rahman, Z. Begum, M.Z. Hafiz, Realization of a Novel Fault Tolerant Reversible Full Adder Circuit in Nanotechnology, Int. Arab J. Inf. Technol., 7 (2010) 317-323.
[27] M. Haghparast, M. Mohammadi, K. Navi, M. Eshghi, Optimized reversible multiplier circuit, Journal of Circuits, Systems, and Computers, 18 (2009) 311-323.