Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 1137 – 1146
© TÜBİTAK
doi:10.3906/elk-1505-223

Optimized designs of reversible arithmetic logic unit

Ali BOLHASSANI[1], **Majid HAGHPARAST**[2,*]
[1]Department of Computer Engineering, Arak Branch, Islamic Azad University, Arak, Iran
[2]Department of Computer Engineering, Yadegar -e- Imam Khomeini (RAH) Shahre Rey Branch,
Islamic Azad University, Tehran, Iran

Received: 26.05.2015 • **Accepted/Published Online:** 27.03.2016 • **Final Version:** 10.04.2017

Abstract: Reversible logic has emerged as a promising paradigm in various domains, such as low power VLSI design, quantum cellular automata, and nanotechnology-based systems. The arithmetic logic unit (ALU) is one of the main components of any central processing unit. In this paper we propose two new reversible ALUs using elementary quantum gates, with more functions compared to the existing designs. The results show that the proposed designs are better than the existing counterparts in terms of cost-metrics and the number of functions generated. The proposed reversible ALUs can be used in the implementation of quantum computers. All the designs are with nanometric scales.

Key words: Reversible logic, quantum computing, quantum logic gates, arithmetic logic unit, reversible arithmetic logic unit

1. Introduction

Nanotechnology is a branch of engineering that has appeared promising due to the design and manufacturing of extremely small electronic devices built at the molecular level of matter. Nanodevices create a platform for microelectronic devices that have novel properties because of their small size and the ability to control or manipulate matters on an atomic scale. Based on this definition, the implementation of logic circuits using the properties of elementary particles is introduced. One of the most fundamental obstacles in the development of smaller and more powerful circuits is the shrinkage of transistor sizes and power dissipation [1]. Irreversible hardware computation inherently results in power dissipation. It is due to information loss. However, in reversible operations information is neither lost nor erased between two computation stages. Thus, we have zero power dissipation in reversible circuits. Landauer demonstrated that in irreversible circuits the amount of power dissipated due to the loss of each bit of information, regardless of its realization technique, is at least KTln2 J, where K is the Boltzmann constant (1.38×10^{-23} J K^{-1}) and T is the absolute temperature at which the operation is being implemented [1–5]. In 1973, Bennett [6] showed that to avoid the amount of energy wasted in the form of heat, the computation must be carried out in a reversible way [1,4–8] and all quantum computations should be reversible. Reversible computation models are both forward and backward deterministic, i.e. reversible logic allows the reproduction of inputs from observed outputs. A reversible arithmetic logic unit (ALU) is a vital portion of a quantum computer [9–18]. This paper targets two approaches for designing reversible ALUs. The presented circuits are evaluated over the existing designs and found to be cost-effective. The remainder of this paper is divided into four sections. In Section 2, we outline the basic conceptions of reversible logic,

[*]Correspondence: haghparast@iausr.ac.ir

quantum computing, and some fundamental quantum gates. Section 3 presents two designs for reversible ALUs using elementary quantum gates. In Section 4, the evaluation of the proposed designs and a comparison to the existing works, subjected to all possible defects, are summarized. Section 5 gives the conclusions.

2. Basic concepts

In this section, we have presented the basic principles pertaining to reversible logic, quantum computing, elementary quantum logic gates, optimization parameters, and arithmetic logic unit.

2.1. Reversible logic gate

Reversible logic is considered as a newly developed field of study for reduction of the physical entropy gain that holds promise for quantum computing [2,19–21]. In reversible logic, every input pattern has a unique output pattern and the numbers of inputs and outputs in reversible circuits are the same [3–6]. If the function F is reversible then there is a unique one-to-one mapping between an n-input vector and a corresponding n-output vector [4,7–9]. An $n \times n$reversible logic gate can be denoted as $Iv \leftrightarrow Ov$, where $\mathrm{I}_v = (\mathrm{I}_0, \mathrm{I}_1, \ldots, \mathrm{I}_{n-1})$ is the input vector and $\mathrm{O}_v = (\mathrm{O}_{0,}\ \mathrm{O}_{1,}\ \ldots, \mathrm{O}_{n-1})$ is the output vector. Neither fan-out nor feedback is permitted in reversible circuits [2–6].

2.2. Optimization parameters

To evaluate reversible logic circuits, the researchers considered different cost-metrics such as the number of garbage outputs, constant inputs, quantum cost, and hardware complexity [2–4,7–9,21]. The reductions of the number of garbage outputs, constant inputs, quantum cost, and hardware complexity are required to improve the circuits [7–9,19–21]. In our proposed ALUs, these parameters have been partly improved using an efficient optimization methodology that is elaborated on in this section.

Quantum cost: Quantum cost is the number of elementary 1 × 1 and 2 × 2 quantum logic gates [2–4,7,8].

Constant inputs: Constant inputs refer to the number of inputs that are to be maintained constant either at logical value 0 or 1 in order to serve the desired function of the circuit [2–4,7–9,21].

Garbage output: Garbage outputs are just added to the circuits to maintain reversibility [2,7–9]. The garbage outputs are primary output lines, which do not perform any useful operations for further calculations [7–9,21]. We try to use a minimum number of constant inputs and garbage outputs.

Hardware complexity: This refers to the total number of the logical calculations in a circuit that is determined by counting the number of AND, NOT, and XOR operations [8–10,14–18]. To compute the hardware complexity of the reversible circuits we assume that:

$\alpha =$ Number of two input EX-OR gates.

$\beta =$ Number of two input AND gates.

$\delta =$ Number of NOT gates.

2.3. Quantum computing

Every quantum circuit works on QUBIT states, described by a two-dimensional complex Hilbert space of spinors-states, with the assumption of a two-level quantum system [1,4,18–21]. Eq. (1) represents the Boolean values

0 and 1 for states of quantum bits.

$$|1\rangle \equiv \left(\frac{1}{0}\right) \quad \text{و} \quad |1\rangle \equiv \left(\frac{0}{1}\right) \tag{1}$$

Any state of a QUBIT may be written as $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$, where a and b are complex numbers such that $|a|^2 + |b|^2 = 1$. Quantum logic gates are inherently reversible constructs based on reversible logic [1,11–13,18–21]. These are described in forms of matrices, which are efficient methods for the formulation of quantum computers [11,12,18,19]. A quantum circuit Q is a cascade of quantum gates q_i, i.e. $\text{Q} = (\text{q}_0,\ \text{q}_1,\ \ldots,\ \text{q}_{n-1})$. The most common quantum gates act on the spaces of one or two QUBITS [1,4,18–21]. The states of a particle on one dimension can be represented by QUBITS instead of bits as in quantum electrodynamics [1,4,8,11,18]. In quantum mechanics, the operations on QUBITS are described by matrix multiplications specified by fundamental quantum logic gates [1,4,11,16–21]. The unitary operations on these QUBITS are NOT, CNOT, Controlled-V, and Controlled-V+ gates [1,4–8,11,18]. In general, a quantum n × n gate is defined by a $2^n \times 2^n$ matrix, so the quantum 1 × 1 NOT gate can be shown by the following 2 × 2 matrix. It is the most used basic quantum gate (as a single QUBIT that is inverted) [8,11–13,18–20].

$$NOT: \begin{array}{l} |0\rangle \to |1\rangle \equiv \left(\tfrac{0}{1}\right) \\ |1\rangle \to |0\rangle \equiv \left(\tfrac{1}{0}\right) \end{array} \Longrightarrow \; NOT \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{2}$$

To apply the NOT gate to a state $\left(\tfrac{a}{b}\right)$ vector, we multiply the corresponding state vector by the matrix representation of the gate as follows.

$$NOT\left(\frac{a}{b}\right) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \left(\frac{a}{b}\right) = \left(\frac{b}{a}\right) \tag{3}$$

The CNOT gate can be represented by the following 4 × 4 matrix.

$$\begin{array}{l} CNOT|01\rangle = |01\rangle \\ CNOT|01\rangle = |01\rangle \\ CNOT|10\rangle = |11\rangle \\ CNOT|11\rangle = |10\rangle \end{array} \Longrightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \tag{4}$$

The CNOT gate acts on two QUBITS, labeled as the control and the target bit [1,11–13,16–19]. The target QUBIT is inverted if the control QUBIT is 1.

2.4. Fundamental quantum gates

Any reversible circuit can be realized using 1 × 1 and 2 × 2 quantum reversible gates. We can compare reversible circuits by counting their fundamental quantum gates such as NOT, Controlled-V, Controlled-V+, and CNOT [1,4–8,11–13,18]. The primitive quantum gates can be applied on a QUBIT state as basic operations in a quantum computer. The 2 × 2 quantum gate can be constructed using one of the inputs as a control line. Each of these primitive 2 × 2 quantum reversible gates has the quantum cost of 1.

Quantum NOT gate: A NOT gate is a 1 × 1 quantum gate as shown in Figure 1.

Quantum CNOT gate: The CNOT transforms the computational basic states, which acts on two bits: one control bit and another target bit. The second bit is negated if the first bit is in state $|1\rangle$, and it leaves the target bit unchanged otherwise [1,8,11,18]. The target bit is inverted when the control bit is $|1\rangle$. The CNOT gate is shown in Figure 2.

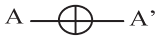

Figure 1. The symbol of 1 × 1 elementary quantum NOT gate.

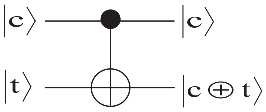

Figure 2. The symbol of 2 × 2 elementary quantum CNOT gate.

Quantum Controlled-V and Controlled-V+ gates: The Controlled-V and the Controlled-V+ gates are the other fundamental 2 × 2 primitive quantum gates [8,11–13,16–19]. The second input is propagated to the output in both Controlled-V and V+ gates when the control input is zero [4,8,11]. If the control line is 1, the values on their target lines will be changed using the transformation given by the matrix $V = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^{\frac{1}{2}} = \frac{1+i}{2}\begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix}$ and $v^{+} = \frac{1-i}{2}\begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix}$ [1,4,8,11–13,18–21]. The quantum configurations of the Controlled-V and the Controlled-V+ gates are shown in Figures 3 and 4. The V gate is also known as the square root of NOT. The V+ gate is the Hermitian of V with properties presented in Eq. (5) [1,4,8,11,18].

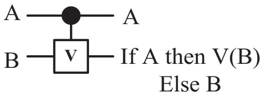

Figure 3. Quantum equivalent circuit of 2 × 2 Controlled-V gate.

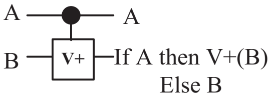

Figure 4. Quantum equivalent circuit of 2 × 2 Controlled-V+ gate.

$$V \times V = NOT,\ V \times V^{+} = V^{+} \times V = I,\ V^{+} \times V^{+} = NOT \qquad (5)$$

Integrated qubit gates: Another type of fundamental 2 × 2 quantum gates are known as integrated qubit gates [8,11–13,16–19]. When the control input is zero, then the second input is propagated to the output in both Controlled-V and V+ gates [4,8,11]. The quantum representations of both patterns for these gates are shown in Figures 5 and 6.

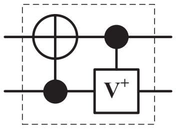

Figure 5. Quantum representation of the first pattern for integrated QUBIT gate.

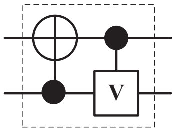

Figure 6. Quantum representation of the second pattern for integrated QUBIT gate.

2.5. Arithmetic logic unit

The ALU is an important part of every CPU and programmable computing devices [8–12]. An ALU performs one of several possible functions on two operands, A and B, depending on control inputs [9–12]. The ALU can perform various kinds of arithmetic and logic operations on data such as addition (ADD), subtraction, multiplication (not including integer division), carry output (C_{out}), increment, decrement, AND, OR, NAND, NOR, XOR, NOT (logic inversion or complement), shift and rotation, multiplexing (MUX), and NO operation (buffer, transfer, pass-through) [9–13,16–18].

3. Proposed designs

This paper focuses on the design of reversible ALUs that can be part of a programmable reversible computer. In order to design an effective reversible ALU, the cost-metrics and the number of logical and arithmetic operations must be considered [10–18]. In this study, we try to construct new reversible ALUs with more arithmetic operations compared to the existing ALUs. We propose two designs of cost-effective reversible ALUs, which can handle different elementary arithmetic and logical operations. The proposed reversible ALU circuits produce the greatest number of calculations at the lowest quantum cost and hardware complexity compared with the existing counterparts. Thus, our proposed designs are better than the existing designs. The comparisons between the proposed designs and the existing designs are reported.

In the first approach, the proposed reversible ALU has 10 inputs and 10 outputs. The inputs consist of three data inputs (A, B, and C_{in}), five select lines (S_0, S_1, S_2, S_3, and S_4), and two constant inputs (set at zero). Three control inputs, S_1, S_2, and C_{in}, are not propagated to the outputs. The inputs of A, S_0, S_3, and S_4 propagated to the outputs. The other outputs are $A \oplus B \oplus S_0$, ADD, Cout/Borrow, Result (main output), and two garbage outputs (G_1, G_2). The first proposed design can produce 10 operations (3 arithmetic and 7 logical calculations). The first design for a 1-bit reversible ALU is depicted in Figure 7. It requires one dotted rectangle, eight V, four V+, and eight CNOT gates. The dotted rectangle is equivalent to a CNOT gate with a quantum cost of unity. The first design has a total quantum cost of 21 and the hardware complexity of this circuit is $17\alpha + 7\beta + 1\delta$. The functions of the first approach are shown in Table 1. According to this table, the circuit produces three functions consisting of ADD, SUB, and Carry out, which are arithmetic functions, and also seven basic operations including AND, NAND, OR, NOR, XOR, XNOR, and Transfer, which are logical functions in output dependent on the existence values in control inputs. The first proposed reversible ALU has three fixed outputs consisting of ADD, C_{out}/Borrow, and A, which are independent of select inputs (S_0, S_1, S_2, S_3, and S_4). Two logical functions, labeled XOR and XNOR, are produced by select line S_0, i.e. if the value in S_0 is set to zero the XOR function ($A \oplus B$) will be produced, or else XNOR is displayed in output.

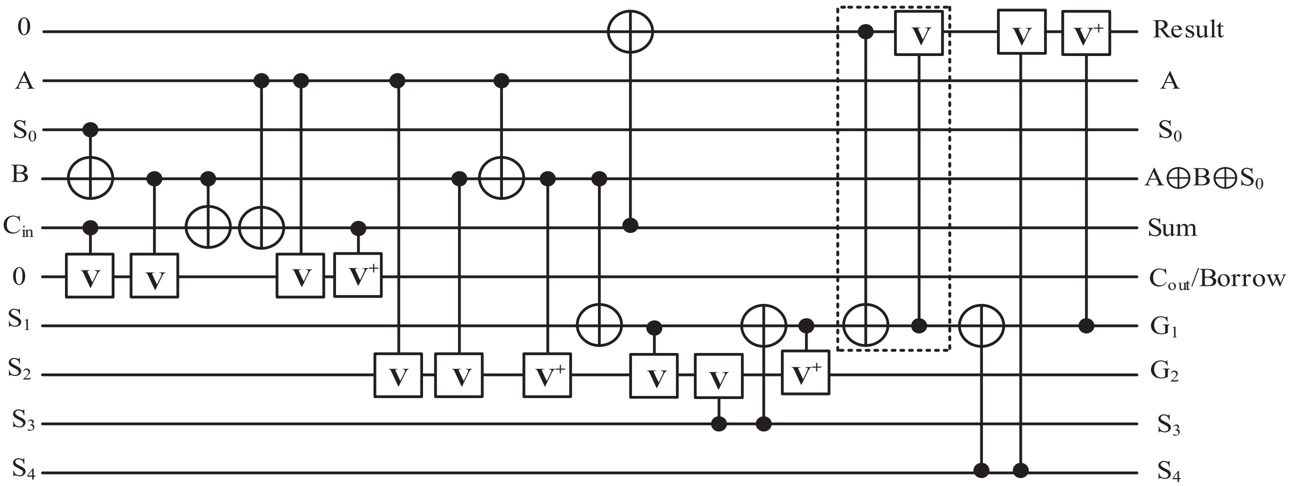

Figure 7. The proposed reversible ALU using elementary quantum gates (Design # 1).

Table 1. Logical and arithmetic operations performed by the first proposed reversible ALU.

C_{in}	S_4	S_3	S_2	S_1	S_0	Outputs	Operations
C_{in}	0	0	0	0	0	$A \oplus B \oplus Cin$	ADD
C_{in}	0	0	0	0	1	$A \oplus B \oplus Cin$	SUB
×	1	0	0	0	0	AB	AND
×	1	0	1	0	0	$(AB)^{'}$	NAND
×	1	1	0	0	0	$(A + B)$	OR
×	1	1	0	1	0	$(A + B)^{'}$	NOR
×	×	×	×	×	0	$A \oplus B$	XOR
×	×	×	×	×	1	$(A \oplus B)^{'}$	XNOR
×	×	×	×	×	×	A	transfer
×	×	×	×	×	×	$(A \oplus B)Cin \oplus AB$	C_{out}

In the second approach, the proposed reversible ALU has 10 circuit lines (10 inputs and 10 outputs). The second proposed design of the 1-bit reversible ALU is shown in Figure 8. We have designed this reversible ALU using primitive quantum logic gates. In this approach we used one dotted rectangle, ten V, five V+, and eight CNOT gates. From this reversible ALU we get three input values (A, B, C_{in}), three select lines (S_0, S_1, S_2), and four constant inputs (kept at logical value 0). The main outputs are the product of F (as a final result output) and C_{out}. The other outputs are garbage outputs: G_1, $G_2(= S_0 \oplus S_1)$, $G_3(= B \oplus S_0 \oplus S_1)$, $G_4(= S_0 \oplus S_2)$, $G_5(= S_2 \oplus C_{in})$, G_6, G_7, and $G_8(= S_1 \oplus B)$. The data inputs A, B, and C_{in} and the selection input lines S_0, S_1, and S_2 may be propagated to the output. In the second design, the total quantum cost is 24. It has 4 constant inputs and produces 8 garbage outputs. The hardware complexity of this circuit is $16\alpha + 7\beta + 2\delta$. We can construct an n-bit ALU by cascading N numbers of this 1-bit ALU circuit. We see that the inputs A, B, and C_{in} can be controlled depending on the values of the selection lines S_0, S_1, and S_2 and carry input (C_{in}). It produces 12 different operations, either arithmetic or logical, depending on the input values of A, B, and C_{in} (as carry input). This circuit has three select control signals (S_0, S_1, S_2) with a provision for realizing different logical and arithmetic operations. Different operations generated by the second design are given in Table 2. The particular function can be selected through S_0, S_1, S_2, and C_{in}. The second presented ALU circuit can implement 12 universal operations. It produces six arithmetic functions consisting

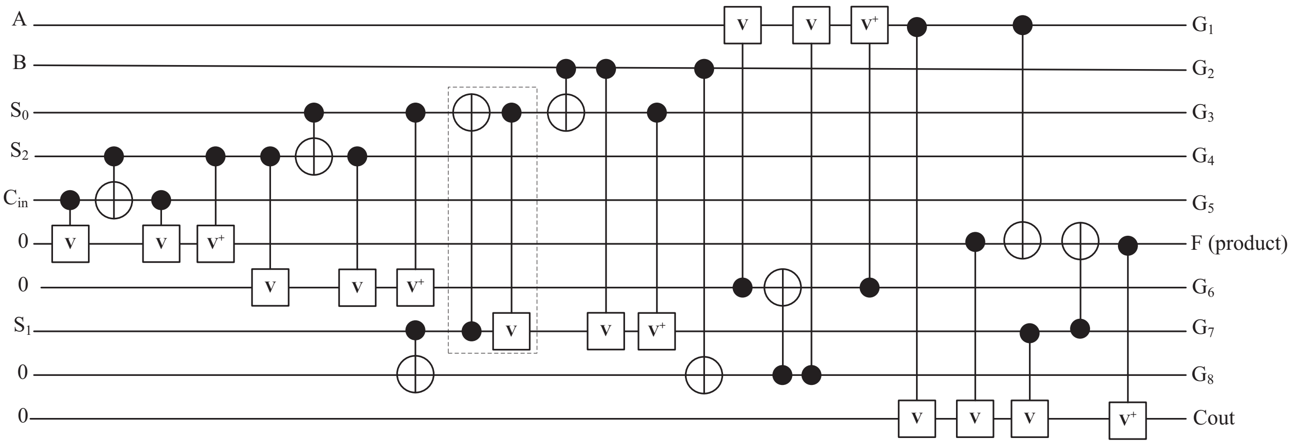

Figure 8. The proposed reversible ALU using elementary quantum gates (Design # 2).

of ADD, SUB, ADD with Carry, SUB with Borrow, Increment, and Decrement, as well as six logical operations including AND, OR, XOR, XNOR, Complement, and Transfer.

Table 2. Logical and arithmetic operations performed by the second proposed reversible ALU.

C_{in}	S_0	S_1	S_2	Output equals	Operations
0	0	0	0	$F = A$	Transfer A
1	0	0	0	$F = A + 1$	Increment A
0	1	0	0	$F = A + B$	Addition
1	1	0	0	$F = A + B + 1$	Add with Carry
0	0	1	0	$F = A + B'$	Subtract with Borrow
1	0	1	0	$F = A+B' +1$	Subtraction
0	1	1	0	$F = A - 1$	Decrement A
1	1	1	0	$F = A$, B	Transfer B , A
×	0	0	1	$F = A \| B$	OR
×	1	0	1	$F = A \oplus B$	XOR
×	0	1	1	$F = A \cap B$	AND
1	1	1	1	$F = A'$	Complement

4. Results and discussion

In this section we present the evaluation results of our proposed designs. Table 3 shows the comparison of two proposed reversible ALUs with the existing counterparts in [13–18,22]. The presented reversible ALUs have several major advantages compared to the previous works. The proposed designs have the lowest quantum cost and hardware complexity, which are the important parameters in evaluation of quantum circuits. The constant inputs and garbage output bits of the proposed circuits are significantly lower. The other advantage of our proposed designs is the number of produced functions.

Table 3. Comparative experimental results of different one-bit reversible ALU circuits.

Designs	Select inputs	Constant inputs	Garbage outputs	Quantum cost	Hardware complexity complexity	Logical and arithmetic* operations
Design #1	5	2	2	21	$17\alpha + 7\beta + 1\delta$	7 + 3* = 10
Design #2	3	4	8	24	$16\alpha + 7\beta + 2\delta$	6 + 6* = 12
Design in [13]	5	2	2	24	$19\alpha + 11\beta + 2\delta$	4 + 2* = 6
Design in [14]	4	4	8	29	$14\alpha + 13\beta + 8\delta$	4 + 8* = 12
Design in [15]	3	5	6	30	$19\alpha + 16\beta + 3\delta$	3 + 1* = 4
Design in [16]	7	2	4	35	$22\alpha + 20\beta + 8\delta$	6 + 2* = 8
Design in [17]	4	10	12	53	$28\alpha + 23\beta + 9\delta$	8 + 4* = 12
Design in [18]	3	6	11	27	$18\alpha + 9\beta + 2\delta$	4 + 8* = 12
Design #1 in [22]	3	2	3	19	N/A	4 + 2* = 6
Design #2 in [22]	5	1	2	20	N/A	4 + 8* = 12
Design #3 in [22]	5	0	0	12	N/A	5 + 5* = 10

As shown in Table 1 our first approach produces 10 elementary operations, and according to Table 2 the second proposed circuit produces 12 elementary operations, which are more than the number of functions in the existing works in [13,15,16] and the first approach in [22]. Parameters with a star ($_*$) symbol in Table 2 indicate the number of arithmetic functions.

In [16] and [17], the number of lines is 13, while our design has 10 lines. It is equal to [13–15]. Morrison et al. in [16] presented a reversible ALU, which produces 8 functions with 7 select inputs, while we reduced the number of select inputs to 5 lines with 9 functions. In our first approach the numbers of garbage outputs and constant inputs are better compared to the existing circuits in [14–18] and are equal to existing circuits in [13].

Apart from more logical and arithmetic operations, the cost-metrics offered by our proposed reversible ALUs in this study are also much better than the existing counterparts in [13–18,22].

Three approaches for reversible ALU are suggested [22]. The total quantum cost of the first existing 1-bit ALU in [22] is 19. The constant inputs and garbage outputs are 2 and 3, respectively. The first existing design in [22] uses 3 select inputs and produces only 6 operations (4 logical and 2 arithmetic functions). The second existing ALU in [22] employs 1 constant input and requires 5 select inputs. It produces 2 garbage outputs. It has quantum cost of 20 and produces 12 operations (4 logical and 8 arithmetic functions). The third ALU in [22] has no garbage outputs and constant inputs. Its quantum cost and number of select inputs are 12 and 5, respectively. The third ALU in [22] produces 10 operations (5 logical and 5 arithmetic functions).

In this paper the first proposed design is better than the first design in [22] in terms of number of garbage outputs and the number of logical and arithmetic operations. Our second proposed design is better than the first design in [22] in terms of the number of generated logical and arithmetic operations. The second proposed design is better than the second design in [22] in terms of the number of select inputs. It is better than the third design in [22] in terms of the number of select inputs and the number of produced logical and arithmetic operations.

We have introduced two approaches for designing the reversible ALU, both aimed at reducing the number of garbage outputs, constant inputs, quantum cost, hardware complexity, and select input lines and arithmetic/logical operations.

From Table 3 we can see that the first proposed design is better than the designs of [13–18] in terms of quantum cost and hardware complexity. The first proposed design is better than [14,15,17,18] in terms of constant inputs. It produces 2 garbage outputs, which is better than the designs of [14–18] and the first design of [22].

The second proposed design required only 3 select inputs to generate the target outputs while the designs of [13,14,16,17] and the second and third approaches in [22] need more than three select inputs. The second proposed design is better than [13–18] in terms of hardware complexity. It produces 12 logical and arithmetic operations, which is better than the existing designs of [13,15,16] and the first and third approaches in [22].

If the number of produced operations is important for a designer, then five different designs in Table 3 have 12 operations. Between them our proposed design #2 is the best choice. If the designers want to have at least 8 operations, then 8 different designs in Table 3 have at least 8 operations. Among them our proposed design #1 is the best choice.

5. Conclusions

In this paper, we put forward two novel approaches for 1-bit reversible ALU design, which are constructed by the combination of elementary quantum gates. The proposed ALUs can handle many logical and arithmetic operations. The first design fundamentally uses 5 control inputs and 3 data inputs in order to select one of the 7 logical functions and 3 arithmetic operations. The second design has 3 control inputs and 3 data inputs in order to realize 6 logical calculations and 6 basic arithmetic operations. Design of the ALU requires thorough optimization in terms of its cost-metrics and architectural complexity. The proposed designs are compared with the existing counterparts. The improvements are reported. If the designers want to have at least 8 operations, then 8 different designs exist. Between them our first proposed design is the best choice. If the number of produced operations is important for a designer, five different designs exist, which have 12 operations. Among them our second proposed design is the best choice.

References

[1] Nielson MA, Chuang IL. Quantum Computation and Quantum Information. Cambridge, UK: Cambridge University Press, 2000.

[2] Haghparast M, Navi K. A novel fault tolerant reversible gate for nanotechnology based systems. American Journal of Applied Sciences 2008; 5: 519-523.

[3] Haghparast M, Jassbi SJ, Navi K, Hashemipour O. Design of a novel reversible multiplier circuit using HNG gate in nanotechnology. World Applied Sciences Journal 2008; 3: 974-978.

[4] Haghparast M, Mohammadi M, Navi K, Eshghi M. Optimized reversible multiplier circuit. J Circuit Syst Comp 2009; 18: 311-323.

[5] Landauer R. Irreversibility and heat generation in the computing process. IBM J Res Develop 19961; 5: 183-191.

[6] Bennett C. Logical reversibility of computation. IBM J Res Develop 1973; 17: 525-532.

[7] Haghparast M, Navi K. Design of a novel fault tolerant reversible full adder for nanotechnology based systems. World Applied Sciences Journal 2008; 3: 114-118.

[8] Mohammadi M, Haghparast M, Eshghi M, Navi K. Minimization and optimization of reversible BCD-full adder/subtractor using genetic algorithm and don't care concept. International Journal of Quantum Information Processing 2009; 7: 969-989.

[9] Safari P, Haghparast M, Azari A. A design of fault tolerant reversible arithmetic logic unit. Life Science Journal 2012; 9: 643-646.

[10] Babazadeh S, Haghparast M. Design of a nanometric fault tolerant reversible multiplier circuit. Journal of Basic and Applied Scientific Research 2012; 2: 1355-1361.

[11] Thomsen MK, Gluck R, Axelsen HB. Reversible arithmetic logic unit for quantum arithmetic. J Phys A-Math Theor 2010; 43: 382002.

[12] Thapliyal H, Srinivas MB. Novel reversible TSG gate and its application for designing components of primitive reversible/quantum ALU. In: Fifth International Conference on Information, Communications and Signal Processing; December 2005; Bangkok, Thailand. New York, NY, USA: IEEE. pp. 1425-1429.

[13] Morrison M, Ranganathan N. Design of a reversible ALU based on novel programmable reversible logic gate structures. In: 2011 IEEE Computer Society Annual Symposium on VLSI; 4–6 July 2011; Chennai, India. New York, NY, USA: IEEE. pp. 126-131.

[14] Dixit A, Kapse V. Arithmetic & logic unit (ALU) design using reversible control unit. International Journal of Engineering and Innovative Technology 2012; 6: 55-60.

[15] Syamala Y, Tilak AVN. Reversible arithmetic logic unit. In: 3rd International Conference on Electronics Computer Technology; 8–10 April 2011; Kanyakumari, India. New York, NY, USA: IEEE. pp. 1-4.

[16] Morrison M, Lewandowski M, Meana R, Ranganathan R. Design of a novel reversible ALU using an enhanced carry look ahead adder. In: 11th IEEE International Conferences on Nanotechnology; 15–18 August 2011; Portland OR, USA. New York, NY, USA: IEEE. pp. 1436-1440.

[17] Guan ZH, Li W, Ding W, Hang Y, Ni L. An arithmetic logic unit design based on reversible logic gates. In: IEEE Pacific Rim Conference on Communications, Computers and Signal Processing; 23–26 August 2011; Victoria, BC, Canada. New York, NY, USA: IEEE. pp. 925-931.

[18] Shefali MM, Biswajit D, Anurima R. An optimized realization of ALU for 12 operations by using a control unit of reversible gates. International Journal of Advanced Research in Computer Science and Software Engineering 2014; 4: 496-502.

[19] Kaye PH, Laflamme RA, Mosca M. An Introduction to Quantum Computing. New York, NY, USA: IEEE, 2007.

[20] Wille R, Drechsler R. Towards a Design Flow for Reversible Logic. New York, NY, USA: Springer Science & Business Media, 2010.

[21] Wille R, Soeken M, Drechsler R. Reducing the number of lines in reversible circuits. In: 47th ACM/IEEE Design Automation Conference; 13–18 June 2010; Anaheim, CA, USA. pp. 647-652.

[22] Haghparast M, Bolhassani A. Optimization approaches for designing quantum reversible arithmetic logic unit. Int J Theor Phys 2015; 54: 3027-3442.