

REVERSIBLE/QUANTUM TERNARY ARITHMETIC LOGIC UNIT DESIGN

VITALY DEIBUK

Department of Computer Systems and Networks
Chernivtsi National University
No. 2, Kotsjubynskyi Str., Chernivtsi 58012, Ukraine
v.deibuk@chnu.edu.ua

Received April 2016; revised August 2016

ABSTRACT. *Multiple-valued logic is a promising choice for future computer technologies which provides a set of advantages compared to binary circuits. We have developed an adaptive genetic algorithm for ternary reversible circuits using Muthukrishnan-Stroud gates. The method for chromosomes coding, as well as a reasonable choice of algorithm parameters, allowed obtaining circuits for ternary arithmetic logic unit which are better than other known methods in terms of quantum cost, delay time and amount of input ancillary and output garbage qutrits. On the basis of designed ternary full-adder, we have synthesized reversible ternary parallel adder/subtractor which has better parameters over the previously reported devices.*

Keywords: Quantum computing, Ternary logic, Ternary reversible arithmetic logic unit

1. **Introduction.** Significant development of new methods for quantum computing and quantum information has led to the emergence of new quantum devices and circuits for their implementation, both theoretical and experimental [1-4]. Various physical mechanisms have been proposed for their implementation that allowed the development of quantum network to check fundamentally new quantum algorithms [4]. These circuits (or gates) have a number of features that are due to their reversibility, i.e., bijective mapping of the input states onto the set of output states. In such circuits the power is not dissipated [5] and gates can be described by the permutation matrices. In addition, the quantum mechanical nature of such systems also requires that circuits are acyclic and gates fan out equals unity. Reversible computing is a good alternative of classical computing when a logical operation does not cause information loss. Reversible circuits are used already in low power CMOS devices, quantum and optical computing, digital signal processing and cryptography [6]. Most approaches for the synthesis of such reversible circuits have been based so far on the binary representation of the information and work only for small circuits. However, the recent progress in the study of reversible networks ternary logic shows a number of advantages [7-12]. Therefore, the synthesis of reversible ternary logic devices is one of the challenging problems of modern computer circuitry in recent years. It includes the cascades of ternary reversible gates, such as Feynman and Toffoli ones, used to implement ternary logic functions [12]. Performance of ternary logic for quantum/reversible computing with the use of liquid ion-trap quantum technology [13] led to the proposal of the Muthukrishnan and Stroud (M-S) gates. The following section provides a brief description of these gates, as well as the one-input permutative ternary gates that are the complete basis for the synthesis of the combinational reversible circuits.

The problem of reversible arithmetic logic unit (ALU) design in multiple-valued logic today is not solved. ALU is a major part of the CPU and is designed to perform some arithmetic and logic operations. The design of the ALU used arithmetic and logic devices and multiplexers not yet been developed for reversible ternary logic. The paper is an attempt to synthesize these devices, as well as the overall structure of the proposed construction of a reversible ternary ALU. In contrast to the binary logic, CAD tools for the automated synthesis of ternary logic devices have not yet been created. To solve the problems associated with an unidentified structure, many researchers lately increasingly use the soft computing methods [11,15-19]. Among them there are genetic algorithms (GAs), which are based on the principles of natural selection and evolution [20,21]. In this paper we use an adaptive genetic algorithm to find the optimal design of the reversible devices. This algorithm gives a possibility to implement main ternary arithmetic and logic operations on the basis of one- and two-qutrits M-S gates. Since these operations are the main components of ALU, their quantum cost should be minimal. The quantum cost of a ternary reversible circuit is defined as the number of primitive gates required in its implementation. The multiple-valued reversible logic circuit with minimal number of garbage outputs, minimal quantum cost, minimal number of ancillary qutrits and time delay is considered as an efficient design. The presented adaptive algorithm extends the genetic algorithm described in [19]. This algorithm is tested for the synthesis of the ternary reversible full adder.

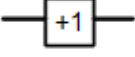
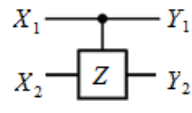
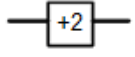

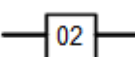
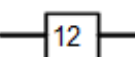
The structure of the paper is as follows: Section 2 explains the ternary Galois field and basic ternary reversible gates; Section 3 presents the details of the used adaptive genetic algorithm; Section 4 shows the proposed design of the ternary adder and parallel adder/subtractor; Section 5 describes the design of ternary logic unit. The main structure of reversible ternary ALU as well as 3×1 ternary multiplexer are shown in Section 6; Section 7 provides the conclusions.

2. Ternary Galois Field and Basic Gates. The unit of information in ternary quantum logic is quantum ternary unit (qutrit). Qutrit states can be described with the ternary Galois field GF (3) which is an algebraic structure on the set $\{0, 1, 2\}$ with two binary operations – addition and multiplication by modulo 3. These operations satisfy the following axioms: commutative and associative laws for addition and multiplication. Besides, multiplication is distributive over addition. Quantum gates manipulate quantum information. The output quantum state of the gate is determined by multiplying the input qutrit state to the unitary 3×3 matrix which specifies a 1-qutrit ternary gate. Qutrit states in quantum ternary logic can be presented as

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad |2\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \quad (1)$$

In this article we use 1-qutrit gates that match permutative unitary 3×3 matrices as shown in Table 1. The elements $Z_3(+1)$ and $Z_3(+2)$ shift the input state of qutrit by 1 and 2, respectively. The elements $Z_3(01)$, $Z_3(02)$, and $Z_3(12)$ permute the corresponding input states ($|0\rangle$ and $|1\rangle$; $|0\rangle$ and $|2\rangle$; $|1\rangle$ and $|2\rangle$, respectively) of single qutrit. We assign the gate costs of these 1-qutrit gates to be 1, using the reasoning similar to [3,8,11]. The 1-qutrit gate B is said to be the inverse gate of a 1-qutrit gate A if gates A and B in cascade have the resultant effect that the input signal to gate A is restored at the output of gate B . In particular, for the gates $Z_3(+1)$, $Z_3(+2)$, $Z_3(01)$, $Z_3(02)$, and $Z_3(12)$ the inverse elements are $Z_3(+2)$, $Z_3(+1)$, $Z_3(01)$, $Z_3(02)$, and $Z_3(12)$, respectively.

TABLE 1. Ternary permutative transforms and its symbolic representation

1-qutrite gates	1-qutrite transformations	2-qutrite M-S gates
	$Z_3(+1) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$	 $Y_1 = X_1$ $Y_2 = \begin{cases} X_2, & \text{if } X_1 = 0, 1 \\ Z, & \text{if } X_1 = 2 \end{cases}$ $Z \in \{+1, +2, 01, 02, 12\}$
	$Z_3(+2) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$	
	$Z_3(01) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	
	$Z_3(02) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$	
	$Z_3(12) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	

Important gates for designing ternary quantum circuits are the ternary M-S gates. The diagram of a ternary M-S gate is shown in Table 1. Here, input X_1 is controlling, and input X_2 is controlled. The output Y_1 is equal to the input X_1 . If $X_1 = 2$, the other output Y_2 is the Z -transform of the input X_2 , otherwise $Y_2 = X_2$. The M-S gates are similar to the controlled 2-qutrit De Vos gates [3] and their extensions used by Miller et al. [7,9], namely the $CC1$, CN , $CC2$, CD , and CE gates. The only exception is that in the M-S gates the controlling value is 2 and in the De Vos gates (including the extensions) the controlling value is 1. In [7,9], the controlled 2-qutrit gates $CC1$, CN , and $CC2$ are considered as elementary gates and their cost is assumed to be 1. Using similar reasoning, we assign the gate cost of M-S gates to be one.

3. Proposed Adaptive Genetic Algorithm. An improved approach to synthesis of reversible/quantum ternary devices which is based on adaptive genetic algorithm has been used. Such approach to reversible logic synthesis is designed to take into account several additional conditions, namely, forbidding the fan-out (no-cloning theorem [4]), and no feedback. We consider that the desired circuit can be represented by a sequence of described above controlled and uncontrolled logic primitives placed in parallel and/or in series. We placed not more than one gate in each column (Figure 1). Information inputs (controlling and controlled) are placed in the upper part of the network while the constant ternary signals (ancillaries) are placed at the bottom. We have chosen the following optimization parameters: a) minimal amount of logic errors on output, according to the truth table of synthesized ternary reversible device; b) minimal amount of constant (ancillary) inputs; c) minimal amount of circuit gates; d) minimal circuit delay time. Genetic algorithm was used to automatically search for a sequence of primitive connections (chromosome) that satisfy a given truth table and imposed optimization conditions. The parameter space contains almost all possible chromosomes. Chromosome is a scheme of

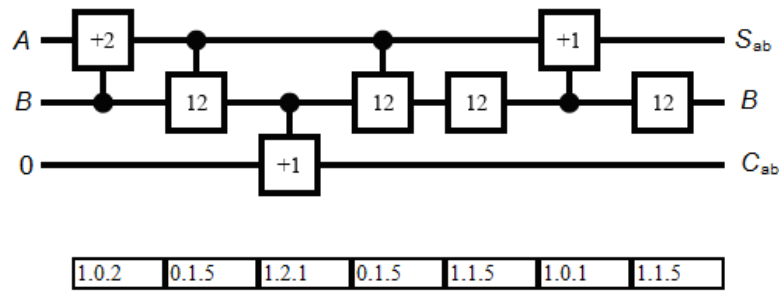


FIGURE 1. Realization of ternary reversible half-adder with one ancilla input (0)

the device, and is encoded as the ordered 3-tuples of genes, which correspond to the gate of the column. The gene contains the following information: the number of controlling lines, the line number of a gate location, the type of a gate (Figure 1, bottom row). In Figure 1, an example of chromosome coding is presented.

Fitness-function F used in this paper consists of three fitness components as a weighted sum:

$$F = k_1 F_1 + k_2 F_2 + k_3 F_3. \quad (2)$$

F_1 – fitness component that minimizes the number of logical errors (*Error*) of output signals according to the truth table of the synthesized device:

$$F_1 = (Error + 1)^{-1}. \quad (3)$$

F_2 – fitness component that minimizes the number of gates dG not 0-type of the circuit, if the length of the chromosome, i.e., the number of genes in the chromosome, is dL :

$$F_2 = \frac{dL - dG}{dL}, \quad (4)$$

F_3 – fitness component that minimizes the number of controlled M-S gates:

$$F_3 = \frac{dGM}{dG}, \quad (5)$$

where dGM – number of 2-qutrit gates; k_1 , k_2 , k_3 – weights coefficients. To find the correct logic circuits, the coefficient k_1 is always taken as 1. Other coefficients of fitness function are assumed to be less than one. We have used adaptive genetic algorithm where k_i changes dynamically as [14]:

$$k_i(itr + 1) = k_i(itr) \left(1 - \overline{F_i(itr)}\right) \quad (6)$$

where $k_i(itr)$ is the weight of the i -th fitness component ($i = 2, 3$) in the itr generation.

We chose as an operator selection a simple panmixis, i.e., random mating. The simplest selection operator assigning each member of the population a random integer number from the range $[1, n]$ where n is the amount of members in the population. These numbers represent the individuals that will participate in the crossover. Some members of the population will participate in reproduction with many other individuals. A panmictic population is one where all individuals are potential partners. This assumes that there are no mating restrictions, neither genetic nor behavioral, upon the population, and that therefore all recombination is possible. Despite the simplicity, this algorithm is universal for many classes of problems. The proposed GA uses one-point crossover operation or uniform crossover with probability 0.5. In this way, the offspring is ensured to have alternate short lines of individuals-parents. The mutation occurs in each locus with a

certain probability p_m and means random change of gene. When reaching a given number of stagnation GA re-runs with a new seed. A feature of the approach is to use PostGA [19], i.e., a process of minimizing genetic algorithm circuits obtained by replacing certain parts of chromosomes on their equivalents, shorter in length. The algorithm terminates when the specified number of working cycles is done or if chromosomes with a fitness-function equal to or greater than one is obtained.

4. Design of Ternary Arithmetic Unit. The major components of ternary reversible ALU and oracles are adders and subtractors [7]. They can also be used for building multipliers and dividers. The main problem in design of reversible adder and subtractor is that the target functions are not bijective. It is important to find the proper reversible presentation of target functions with the minimum of extra signals (ancillaries inputs). We used the proposed adaptive genetic algorithm for realization of two basic reversible arithmetic operations – addition and subtraction of ternary numbers.

4.1. Ternary full-adder circuit. For the ternary full-adder the inputs are A , B and C_i (input carry). The outputs S_{ab} (sum) and C_{ab} (output carry) can be expressed as below:

$$S_{ab} = \text{sum}(A, B, C_i) = A \oplus B \oplus C_i, \tag{7}$$

$$C_{ab} = \text{carry}(A, B, C_i) = \text{int} \left[\frac{A + B + C_i}{3} \right]. \tag{8}$$

Here we have introduced the following notations: $\text{sum}(A, B, C)$ – the sum modulo three of input ternary terms A , B and input carry C ; the $\text{carry}(A, B, C)$ – output carry; $\text{int}(A)$ – the integer part of A . Realization of ternary reversible full-adder with one ancilla input (0) is shown in Figure 2. The obtained circuit contains 13 primitives, corresponding to the quantum cost of realization of 13. The delay time of the obtained full adder, as can be seen from Figure 2, equals $12t_0$, where t_0 is the single gate delay time. For instance, the cost of ternary full adder developed by Zobov and Pekhterev [25] is 18 and requires one ancilla input. Proposed full-adder circuit coincides with the circuit obtained in [19], but gives the result in twice less time at the same conditions. This task can be seen as a test of the use of adaptive genetic algorithm proposed in the previous chapter.

In Table 2, it can be seen a comparison of the experimental results of this work with some other works on the basic parameters – the number of ancilla inputs, delay time and

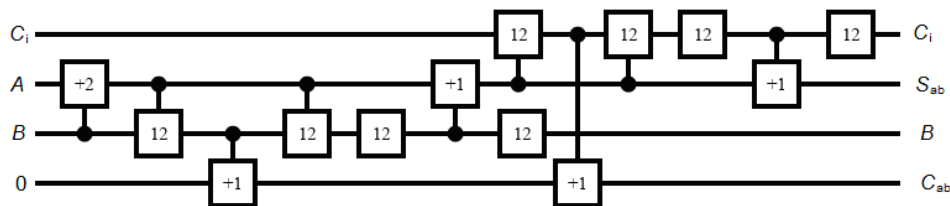


FIGURE 2. Realization of ternary reversible full-adder with one ancilla input (0) (Population size – 500, $p_c = 0.3$, $p_m = 0.02$)

TABLE 2. Comparative results of different reversible ternary full adder circuits

	Ancilla inputs	Delay time, t_0	Quantum cost
M. H. A. Khan & M. Perkowski [26]	2	42	50
A. T. Monfared & M. Haghparast [28]	1	30	34
V. E. Zobov & D. I. Pekhterev [25]	1	18	18
This work	1	12	13

the quantum cost of the full adder. Furthermore, it should be noted that the best results of other authors [25,26,28], usually derived heuristically. Thus, we can conclude that we have received ternary circuit of reversible full adder with better performance.

4.2. Control ternary adder/subtractor circuit. The synthesized above full-adder circuit can be used to build a control ternary reversible adder/subtractor. The idea of using the adder to perform a subtraction of numbers $A - B$ is implemented as $A + B' + 1$, where B' is 2's complement ternary digit. 2's complement just gets in the circuit, if applied to the digit of permutation 02 (Table 2). Control one-digit adder/subtractor is shown in Figure 3, where the controlling block is marked by dashed line. According to the above algorithm, if the control signal $A/S = 0$ or 1, the input signal B does not change the output control unit and performs the operation of addition. When the input of controlling unit $A/S = 2$, then subtrahend digit B will be converted into their 2's complement as can be seen in Figure 3. The value of the input signal C_{in} is equal to 1 and after the addition of 2's complement of B giving 3's complement of B . Thus, it is possible to perform a subtraction operation using the previously obtained full adder circuit (Figure 2). In Figure 3 $C_{in}(B_{in})$ – input carry (borrow), $C_{out}(B_{out})$ – output carry (borrow), $S(D)$ – sum (difference). The main parameters of realized control ternary reversible one-digit adder/subtractor are allowed: quantum cost – 15, ancilla input – 1, garbage outputs – 3, delay time – $13t_0$. Synthesized circuit of the control ternary one-digit adder/subtractor has significantly better parameters in comparison with similar circuits [26,28] (see Table 3).

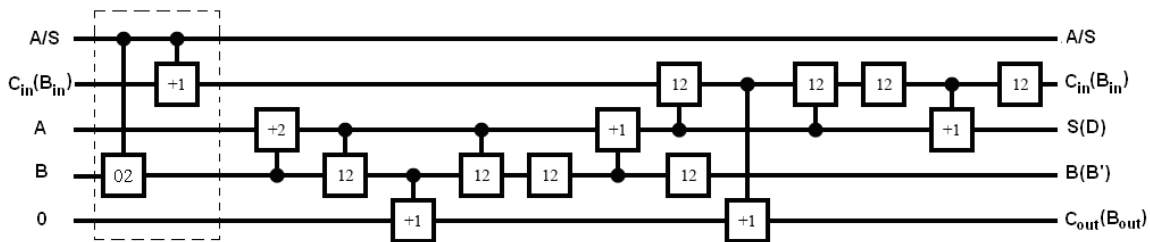


FIGURE 3. Realization of ternary control (A/S) one-digit adder/subtractor with one ancilla input (0)

TABLE 3. Comparison of different reversible ternary parallel one-digit adder/subtractor circuits

	Ancilla inputs	Delay time, t_0	Total cost
M. H. A. Khan & M. Perkowski [26]	3	50	52
A. T. Monfared & M. Haghparast [28]	2	35	36
This work	1	13	15

5. Design of Ternary Logic Unit. Basic logical operations in classical design are AND, OR, NOT and XOR, that are irreversible. The reversible ternary equivalents of these logical operations are TAND: $(A, B, 0 \rightarrow A, B, \text{MIN}(AB))$, ternary TOR: $(A, B, 0 \rightarrow A, B, \text{MAX}(AB))$, standard ternary inversion STI: $(0, 1, 2 \rightarrow 2, 1, 0)$, and ternary TXOR: $(A, B \rightarrow A, A \oplus B)$. From the truth table (Table 4), the standard ternary inversion may obviously be represented by a permutational gate $Z_3(02)$ (Figure 4).

For the synthesis of other logical operations, the developed adaptive genetic algorithm has been used. As can be seen from Figure 5, for implementation of TXOR operation

TABLE 4. Truth table of basic ternary logic operations

A	B	STI (A)	TAND (A, B)	TOR (A, B)	TXOR (A, B)
0	0	2	0	0	0
0	1	2	0	1	1
0	2	2	0	2	2
1	0	1	0	1	1
1	1	1	1	1	2
1	2	1	1	2	0
2	0	0	0	2	2
2	1	0	1	2	0
2	2	0	2	2	1

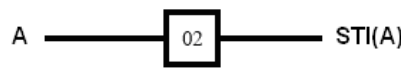


FIGURE 4. Realization of ternary standard inverter (STI)

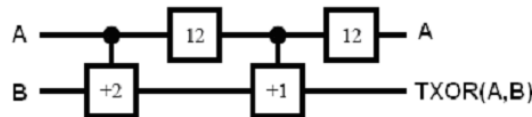


FIGURE 5. Reversible implementation of ternary TXOR(A, B)

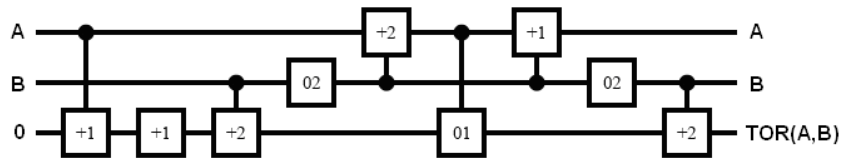


FIGURE 6. Reversible implementation of ternary TOR(A, B) with one ancilla input (0)

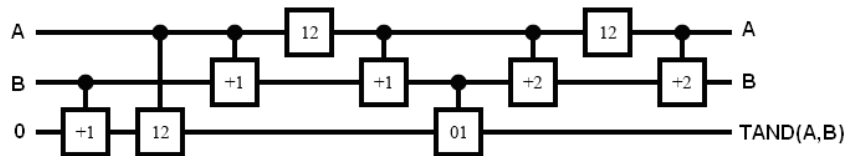


FIGURE 7. Reversible implementation of ternary TAND(A, B) with one ancilla input (0)

constant lines are not required, while reversible logical operations TOR and TAND led to the use of one input ancilla signal (0) (Figures 6 and 7). Our reversible implementation of ternary TXOR operation is entirely consistent with the previously obtained [27].

The synthesized in MS-gates base TOR and TAND circuits have a quantum cost equal to 9, the delay time – $9t_0$, ancilla inputs – 1, garbage outputs – 2. These logic circuits have been obtained for the first time, so there is no way to compare them with others. The resulting circuits of the ternary logic operations can be easily modified for TXNOR, TNOR and TNAND operations, which increase the applicability and flexibility of proposed ternary ALU.

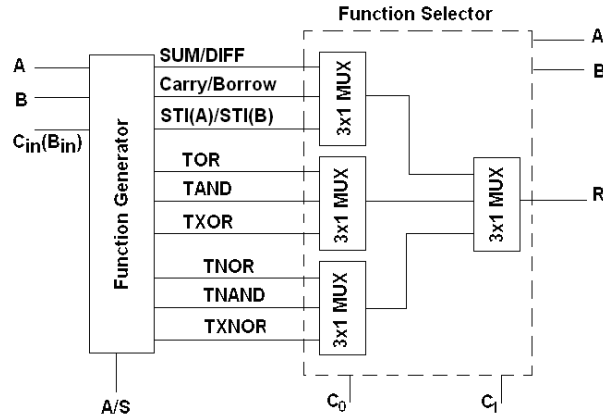


FIGURE 8. Block diagram of ternary reversible ALU

6. Reversible Ternary Arithmetic Logic Unit. Arithmetic logic unit (ALU) is the main part of the central processor unit (CPU) and is intended for one of a few of arithmetic and logical functions of two input operands A and B . This choice depends on the additional control signals. Required resulting function is selected by the multiplexer. Architecture reversible ALU has been considered in detail [22-24]. Serial, parallel and V-shape designs have both advantages and disadvantages. In this paper, it was proposed for the first time the design of the ternary reversible ALU consisting of two modules. The first module is the function generator, outputs of which perform arithmetic and logic ternary functions. In particular, the outputs of functional generator synthesize eight logical operations ($STI(A)$, $STI(B)$, $TOR(A, B)$, $TXOR(A, B)$, $TAND(A, B)$, $TXNOR(A, B)$, $TNOR(A, B)$ and $TNAND(A, B)$), as well as four arithmetic operations ($sum(A, B, C)$, $carry(A, B, C)$, $diff(A, B, C)$ and $borrow(A, B, C)$) depending on the control signal A/S , which determines addition ($A/S = 0$) or subtraction ($A/S = 2$) ternary operation. Then, signals generated in the first module, input to the second module – function selector. Figure 8 shows the structure of the proposed ALU.

The function selector selects a desired function and transmits a corresponding signal to the output of the ALU. C_0 and C_1 are the selection lines, whereas A/S control line selects addition ($A/S = 0$) or subtraction ($A/S = 2$) operation of the input A and B (Table 5). When $A/S = 2$, then B will be converted into their 2's complement B' . If the control signal $A/S = 0$, the input signal B is transmitted to the output unchanged. This parallel architecture is quite easy to verify, cascading and size expanding of ternary reversible ALU.

Implementation of the function selector module is a reversible multiplexer 9×1 , which is performed through two stages 3×1 multiplexers. We have synthesized reversible ternary 3×1 multiplexer using above described adaptive genetic algorithm (Figure 9). Proposed multiplexer has three information (X, Y, Z), a selective (C) and four ancilla inputs. One of the three input signals is passed to the output depending on the controlling signal. The structure of the multiplexer coincides with the results of [27]. The quantum cost of our implementation is 72, delay time is $24t_0$, while the cost of the 9×1 multiplexer proposed by Khan [27] is 102, and delay time is $90t_0$. Therefore, the proposed two-stage multiplexer 9×1 has a lower quantum cost and delay time in comparison with realization [27].

Proposed adaptive genetic algorithm has been successfully used for the synthesis of the function generator as described above. Figure 10 shows the ternary M-S gate realization of function generator. The quantum cost of the realization is 64 and it requires 5 ancilla

TABLE 5. Function table for ternary reversible ALU

A/S	C ₁	C ₀	Operation
0	0	0	SUM(<i>A</i> , <i>B</i> , <i>C_{in}</i>)
2	0	0	DIFF(<i>A</i> , <i>B</i> , <i>B_{in}</i>)
0	0	1	Carry(<i>A</i> , <i>B</i> , <i>C_{in}</i>)
2	0	1	Borrow(<i>A</i> , <i>B</i> , <i>B_{in}</i>)
0	0	2	STI(<i>A</i>)
2	0	2	STI(<i>B</i>)
0	1	0	TOR(<i>A</i> , <i>B</i>)
0	1	1	TAND(<i>A</i> , <i>B</i>)
0	1	2	TXOR(<i>A</i> , <i>B</i>)
0	2	0	TNOR(<i>A</i> , <i>B</i>)
0	2	1	TNAND(<i>A</i> , <i>B</i>)
0	2	2	TXNOR(<i>A</i> , <i>B</i>)

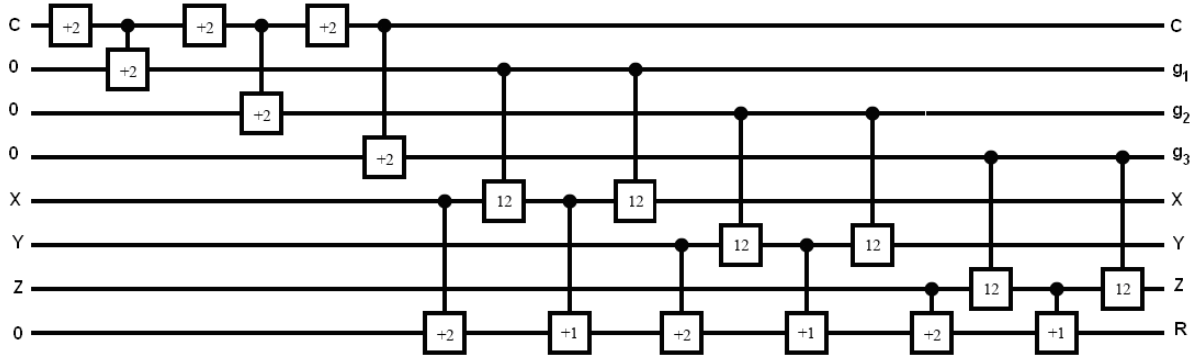


FIGURE 9. Realization of a 3×1 ternary reversible multiplexer (cost is 18, delay time is $12t_0$)

bits, delay time is $14t_0$. Copies of the input signals *A* and *B* can be easily implemented using the TXOR(*A*, *B*) operations (Figure 5) at $A = 0$ or $B = 0$.

Therefore, proposed ALU is characterized by the following parameters: quantum cost is 136, ancillary inputs $6 + 4 \times 4 = 22$, the delay time $14t_0 + 24t_0 = 38t_0$. The obtained ternary reversible ALU device cannot be compared with similar devices, because it has been implemented in this paper for the first time.

7. Conclusions. We present the new design of reversible ternary arithmetic logic unit based on adaptive genetic algorithm. Proposed circuit was synthesized for the first time on the basis of the full set of permutational one- and two-input Muthukrishnan-Stroud gates which can be physically implemented and have a minimum quantum cost. Based on our proposed adaptive genetic algorithm, we got a better implementation of the basic arithmetic and logic ternary devices. The obtained reversible ternary full-adder, control adder/subtractor and two-stage 9×1 multiplexer circuits have better parameters (quantum cost, ancilla inputs, garbage outputs and delay time) in comparison with those obtained in other studies. These results allowed us to propose for the first time a design of the reversible ternary ALU. An advantage of proposed circuits is the possibility of their potential implementation in NMR technology.

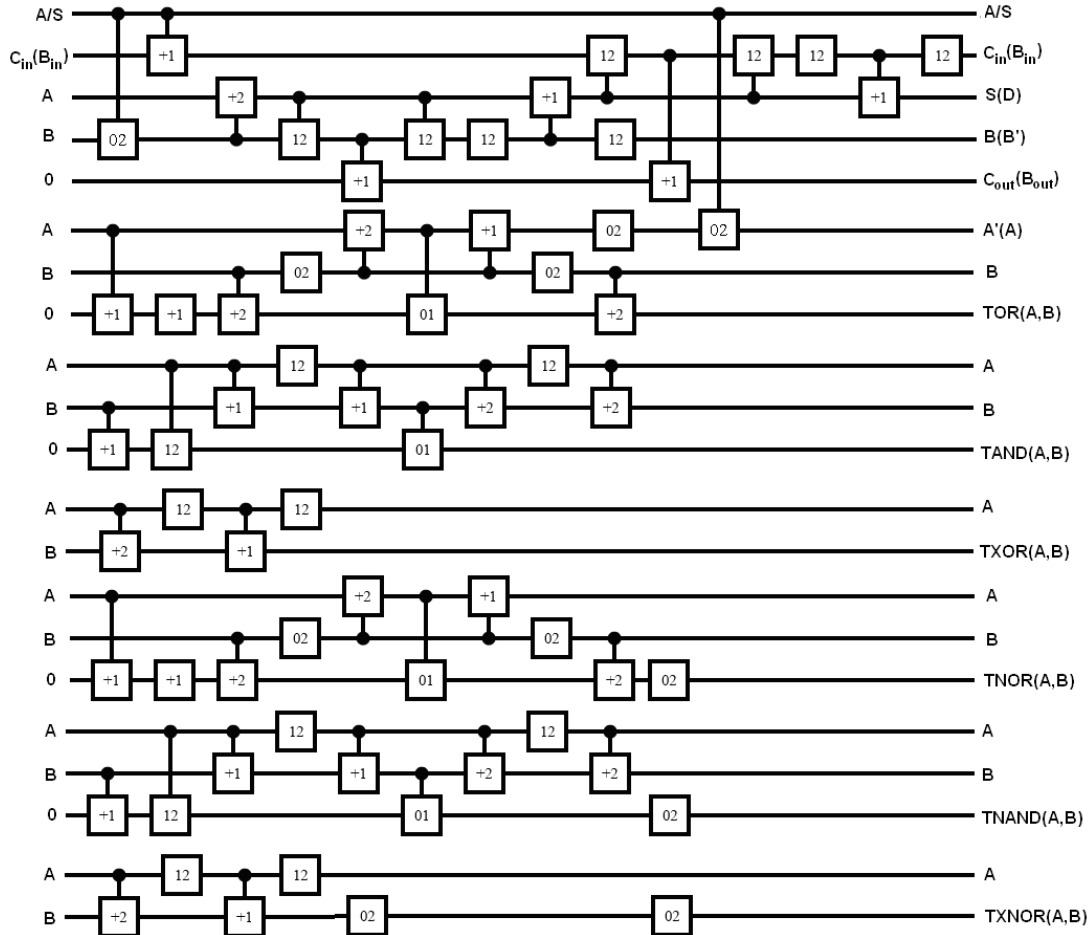


FIGURE 10. Realization of a function generator (cost is 64, delay time is $14t_0$)

Acknowledgment. This work is partially supported by the e-COST Action IC1405. The author also gratefully acknowledges the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

- [1] A. B. Klimov, R. Guzman, J. C. Retamal and C. Saavedra, Qutrit quantum computer with trapped ions, *Phys. Rev. A*, vol.67, no.6, pp.062313/1-7, 2003.
- [2] D. McHugh and J. Twamley, Trapped-ion qutrit spin molecule quantum computer, *New J. Physics*, vol.7, no.1, pp.174/1-9, 2005.
- [3] A. De Vos and Y. van Rentergem, Multiple-valued reversible logic circuits, *J. Multiple-Valued Logic and Soft Comput.*, vol.15, nos.4-5, pp.489-505, 2009.
- [4] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, NY, 2000.
- [5] C. H. Bennett, Logical reversibility of computation, *IBM J. Res. Develop.*, vol.17, no.6, pp.525-532, 1973.
- [6] R. Drechsler and R. Wille, Reversible circuits: Recent accomplishments and future challenges for an emerging technology, *Progress in VLSI Design and Test*, Berlin, Heidelberg, pp.383-392, 2012.
- [7] D. M. Miller and M. A. Thornton, *Multiple Valued Logic: Concepts and Representations*, Morgan & Claypool Publishers, 2008.
- [8] C. Moraga, On some basic aspects of ternary reversible and quantum computing, *Proc. of the 44th IEEE Int. Symp. Multiple-Valued Logic*, Bremen, pp.178-183, 2014.
- [9] D. M. Miller, D. Maslov and G. W. Dueck, Synthesis of quantum multiple-valued circuits, *J. Multiple-Valued Logic and Soft Comput.*, vol.12, no.5/6, pp.431-450, 2006.

- [10] D. M. Miller, R. Wille and R. Drechsler, Reducing reversible circuit cost by adding lines, *J. Multiple-Valued Logic and Soft Comput.*, vol.9, nos.1-3, pp.185-201, 2012.
- [11] M. Lukac, M. Perkowski and M. Kameyama, Evolutionary quantum logic synthesis of Boolean reversible logic circuits embedded in ternary quantum space using structural restrictions, *Proc. of IEEE Congress on Evolutionary Comput.*, Barcelona, pp.1-8, 2010.
- [12] M. H. A. Khan, M. A. Perkowski, M. R. Khan and P. Kerntopf, Ternary GFSOP minimization using Kronecker decision diagrams and their synthesis with quantum cascades, *J. Multiple-Valued Logic and Soft Comput.*, vol.11, no.5, pp.567-602, 2005.
- [13] A. Muthukrishnan and C. R. Stroud Jr., Multivalued logic gates for quantum computation, *Phys. Rev. A*, vol.62, no.5, pp.052309/1-8, 2000.
- [14] T. N. Sasamal, A. K. Singh and A. Mohan, Reversible logic circuit synthesis and optimization using adaptive genetic algorithm, *Procedia Computer Science*, vol.70, pp.407-413, 2015.
- [15] F. Z. Hadjam and C. Moraga, RIMEP2: Evolutionary design of reversible digital circuits, *ACM J. Emerg. Technol. Comput. Systems*, vol.11, no.3, pp.2348-2355, 2014.
- [16] V. Deibuk and A. Biloshytskyi, Genetic synthesis of new reversible/quantum ternary comparator, *Adv. Electr. Comp. Eng.*, vol.15, no.3, pp.147-152, 2015.
- [17] V. Deibuk, I. Turchenko and V. Shults, Optimized design of the universal ternary gates for quantum/reversible computing, *Proc. of the 8th IEEE Int. Conf. Intelligent Data Acquisition and Adv. Comp. Systems*, Warsaw, vol.2, pp.987-991, 2015.
- [18] X. Wang, L. Jiao, Y. Li, Y. Qi and J. Wu, A variable-length chromosome evolutionary algorithm for reversible circuit synthesis, *J. Multiple-Valued Logic and Soft Comput.*, vol.25, no.6, pp.643-671, 2015.
- [19] V. G. Deibuk and A. V. Biloshytskyi, Design of a ternary reversible/quantum adder using genetic algorithm, *Int. J. Inform. Technol. Comp. Sci.*, vol.7, no.9, pp.38-45, 2015.
- [20] R. S. Zebulum, M. C. Pacheco and M. M. Vellasco, *Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms*, CRC Press, 2002.
- [21] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Springer, Heidelberg, 2013.
- [22] S. Sultana and K. Radecka, Reversible architecture of computer arithmetic, *Int. J. Computer Applications*, vol.93, pp.6-14, 2014.
- [23] M. K. Thomsen, R. Glück and H. B. Axelsen, Reversible arithmetic logic unit for quantum arithmetic, *J. Physics A: Math. and Theor.*, vol.43, no.38, 2010.
- [24] L. Gopal, M. Mahayadin, N. Syahira, A. K. Chowdhury, A. A. Gopalai and A. K. Singh, Design and synthesis of reversible arithmetic and logic unit (ALU), *Proc. of the 14th IEEE Int. Conf. Computer, Communications, and Control Technology*, Batu Ferringhi, pp.289-293, 2014.
- [25] V. E. Zobov and D. I. Pekhterev, Adder on ternary base elements for a quantum computer, *JETP Letters*, vol.89, no.5, pp.260-263, 2009.
- [26] M. H. A. Khan and M. Perkowski, Quantum ternary parallel adder/subtractor with partially-look-ahead carry, *J. Syst. Architect.*, vol.53, no.7, pp.453-464, 2007.
- [27] M. H. Khan, Design of reversible/quantum ternary multiplexer and demultiplexer, *Engineering Letters*, vol.13, no.2, pp.65-69, 2006.
- [28] A. T. Monfared and M. Haghparast, Design of novel quantum/reversible ternary adder circuits, *Int. J. Electronics Letters*, 2016.