I.J. Information Technology and Computer Science, **2015, 09, 38-45**
Published Online August 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2015.09.06

Design of a Ternary Reversible/Quantum Adder using Genetic Algorithm

Vitaly G. Deibuk and Andrij V. Biloshytskyi
Department of Computer Systems and Networks, Chernivtsi National University, 2 Kotsubinsky str., Chernivtsi 58012, UKRAINE,
Email: v.deibuk@chnu.edu.ua

Abstract**—**Typical methods of quantum/reversible synthesis are based on using the binary character of quantum computing. However, multi-valued logic is a promising choice for future computer technologies, given a set of advantages when comparing to binary circuits. In this work, we have developed a genetic algorithm-based synthesis of ternary reversible circuits using Muthukrishnan-Stroud gates. The method for chromosomes coding that we present, as well as a judicious choice of algorithm parameters, allowed obtaining circuits for half-adder and full adder which are better than other published methods in terms of cost, delay times and amount of input ancillary bits. A structure of the circuits is analyzed in details, based on their decomposition.

Index Terms**—**Multiple-valued logic, ternary logic, ternary reversible adder, reversible logic.

I. Introduction

Quantum information and quantum computing [1] are very attractive research areas and have made huge advances both theoretically and experimentally in recent years. There are many different proposals for the physical implementation of a quantum computer, all of which are specified by the physics of their qubit systems and the nature of interactions between the qubits. More generally, a lot of attention is attracted to the possibility of realizing a reversible computer based on the laws of quantum mechanics. A reversible circuit maps each input vector into a unique output vector. Quantum algorithms and protocols for quantum communication and cryptography have been studied extensively with the qubit as the information storage and transport medium. Most approaches to quantum computing use two-valued quantum systems (qubits). Recent studies [2-6] have indicated that there are many advantages to expand quantum computers from qubits to multi-valued systems. Three-valued quantum systems, so called qutrits, are the simplest multiple valued systems. It is expected that qutrit-based quantum information processing will be more powerful than qubit implementations [4,5]. But multiple-valued quantum logic (MVL) synthesis is still a developing research area. The crucial issue of choosing the elementary gate set for multi-valued quantum circuits is still not well explored. Synthesis of reversible logic circuits differs significantly from the synthesis of combinational logic circuits. In a reversible circuit, the number of inputs must be equal to the number of outputs, thus, every output can be used only once (i.e., no fan-out is permitted), and must be acyclic. Multiple valued quantum logic synthesis, and ternary logic in particular, has become popular in the recent years [7-11]. Previous works [10,11] used cascades of ternary reversible gates, like Feynman and Toffoli gates, to realize ternary logic functions. The advantage of this approach is that ternary logic functions having many input variables can be easily expressed as a ternary Galois field sum of products and can be realized using a cascade of ternary Feynman and Toffoli gates. Muthukrishnan and Stroud demonstrated the realization of MVL for quantum computing using liquid ion-trap quantum technology [3], referred to as Muthukrishnan and Stroud (M-S) gate. The macro-level ternary Feynman and Toffoli gates can be realized on top of M-S gates. It was shown that arbitrary MVL operations on any number of multiple valued qubits can be decomposed into elementary logic gates that operate only on two qubits at a time. We refer to these elementary two-qutrit gates as M-S gates. The M-S gate is considered to be one of the elementary gates in ternary logic synthesis. Designing the so-called permutative circuits (described by unitary matrices that are in addition permutative) is practically important since all the oracles in algorithms so far are permutative, and the non-permutative quantum blocks are usually standardized (like Hadamard or Fourier transforms). The algorithms designed for these use evolutionary programming principles or are adaptations of some methods used in Galois Field (GF3) classical circuits and classical multiple-valued reversible logic [12-20]. There are very few examples in the literature on synthesizing certain types of practical special circuits, like arithmetic ones, using such quantum CAD tools or by hand.

In this paper, we have used the genetic algorithm (GA) with real valued encoding of the chromosomes using ordered 3-tuples [14]. Genetic algorithms are very popular Soft Computing approaches for solving problems with no identified structure and high level of noise [20-22]. Advantages of such a choice in our opinion are:

(i) a large search space for solving problems,
(ii) the size of the search space can be changed by choosing the parameters settings,

(iii) the possibility of obtaining a large variety of new solutions,
(iv) from this wide variety of solutions one can choose the optimal ones according to the specified conditions.

These advantages make GAs useful for synthesizing ternary reversible/quantum circuits using a cascade of M-S gates. This follows from the fact that the structure of such cascade is still unidentified and the search space itself is exponentially large. We have defined the quantum cost of a ternary reversible gate as the number of M-S gates required in its implementation. The multi-valued reversible logic circuit with minimal number of garbage outputs, minimal quantum cost and minimal number of ancilla bits (ancilla bit is an auxiliary input constant bit needed in the circuit other than the function input bits) is considered as an efficient design. 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 proposed genetic algorithm; Section 4 shows the proposed design of the reversible ternary half-adder, full adder and the analysis of the proposed circuits. Section 5 provides the conclusions.

II. Ternary Galois Field and Basic Gates

The ternary Galois field (GF(3)) consists of a set of elements T = {0, 1, 2} and two basic binary operations – addition modulo 3 (denoted by $\oplus$) and multiplication (denoted by $\cdot$ or absence of any operator) as defined in Table 1, which satisfies the following axioms:

(1) Commutative law for addition

$$x \oplus y = y \oplus x$$

(2) Associative law for addition

$$(x \oplus y) \oplus z = x \oplus (y \oplus z)$$

(3) There is an element 0 (zero) such that

$$x \oplus 0 = x \qquad \text{for all } x$$

(4) There is an element ($-x$) for any x, such that

$$x \oplus (-x) = 0$$

(5) Commutative law for multiplication

$$x \cdot y = y \cdot x$$

(6) Associative law for multiplication

$$(x \cdot y) \cdot z = x \cdot (y \cdot z)$$

(7) There is an element 1 (not equal to 0) such that

$$x \cdot 1 = 1 \cdot x = x \qquad \text{for all } x$$

(8) For any $x \neq 0$, there is an element x^{-1} such that

$$x \cdot x^{-1} = x^{-1} \cdot x = 1$$

(9) Distributive law

$$x \cdot (y \oplus z) = (x \cdot y) \oplus (x \cdot z)$$

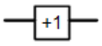
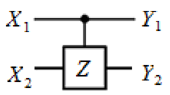
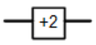
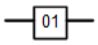
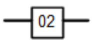
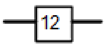
Table 1. GF(3) Operations

$\oplus$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

$\cdot$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Any transformation of the qutrit state represented by a 3×3 unitary matrix specifies a valid 1-qutrit ternary quantum gate. There are many such non-trivial 1-qutrit gates. We use only the permutative transforms as shown by permutative matrices of Table 2.

Table 2. 1- and 2-Qutrits Ternary Permutative Transforms and its Symbolic Representation

1-qutrit gates	1-qutrit transformations	2-qutrit M-S gate
	$Z_3(+1) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$	
	$Z_3(+2) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$	
	$Z_3(01) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	
	$Z_3(02) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$	
	$Z_3(12) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	

The logical equivalent of these 1-qutrit transforms can be expressed using GF3 expressions and truth tables as shown in Table 3. These permutative operations are thus good both technologically and algebraically. The transforms $Z_3(+1)$ and $Z_3(02)$ are referred to as *C*1 and *N*, respectively, in [2,8]. The transforms $Z_3(+2)$, $Z_3(01)$ and $Z_3(12)$ are referred to as *C*2, *D*, and *E*, respectively, in [8]. Using the reasoning similar to [2,8,9,10], we assign the gate costs of these 1-qutrit gates to be 1. 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 elements Z3(+1), Z3(+2), Z3(01), Z3(02), and Z3(12) the inverse elements are Z3(+2), Z3(+1), Z3(01), Z3(02), and Z3(12), respectively.

Table 3. 1-Qutrit Ternary Permutative Transforms

Input A	Output					
	A(0) =A	A(+1)= A+1	A(+2) = A+2	A(01) = 2A+1	A(02) = 2A+2	A(12) = 2A
0	0	1	2	1	2	0
1	1	2	0	0	1	2
2	2	0	1	2	0	1

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 2. 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 [2] and their extensions used by Miller *et al.* [6,8], namely the *CC*1, *CN*, *CC*2, *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 [6,8], the controlled 2-qutrit gates *CC*1, *CN*, and *CC*2 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.

III. Proposed Genetic Algorithm

In this work we propose an improved approach to synthesis of reversible/quantum ternary adders based on genetic algorithm. Such approach to reversible logic synthesis is called for due to necessity of taking into account several additional conditions, namely, forbidding the fan-out and fan-in (no-cloning theorem [1]), and feedback is not allowed. Genetic algorithms belong to adaptive and meta-heuristic algorithms of finding the optimal solution for various problems, based on the idea of natural selection and genetics. They utilize the intelligent random search for solving the problems with a large phase space with many dimensions. Consider that a sought scheme can be represented by a sequence of controlled and uncontrolled logic primitives described above, placed in parallel and/or in series. We will place not more than one gate in each column (Fig.1). Information inputs (controlling and controlled) are placed in the upper part of the network while the constant ternary signals are placed at the bottom. We have chosen the following optimization parameters:

Minimal amount of logic errors on output, according to the truth table of synthesized ternary reversible device;
Minimal amount of constant (ancillary) inputs;
Minimal amount of circuit gates;
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.

A. Chromosome Coding

Chromosome represents a schematic of a device which is coded as a vector of genes which in turn describe a gate of a given column. A gene contains the following information: the number of line which controls the gate, line number of the gate itself, gate type (Fig.1, bottom row). If the number of the controlling line equals the gate line number we'll call it a non-controlled gate. Gate *A*(0) is encoded as "000". Figure 1 presents an example of chromosome coding (for half-adder). To model a half-adder we used four input and output lines. The top two lines receive information signals *A* and *B*, bottom two (*L*0, *L*1) get ancilla signals 2 and 0, respectively. To guarantee the reversibility of the synthesized devices, the number of inputs equals the number of outputs. At output, besides repeating the input signals *A* and *B*, we obtain the sum *S*ab and the output carry *C*ab.

B. Fitness Function

An important feature of the genetic algorithm is the estimation of the fitness function for each chromosome. Successful choice of fitness function is important to achieving convergence in a search for solution. In this work we use a fitness function containing three components:

$$F = k_1F1 + k_2F2 + k_3F3 \qquad (1)$$

*F*1 – a fitness component that minimizes logical errors in output signals according to the truth table of the synthesized device:

$$F1 = \frac{1}{Error + 1}, \qquad (2)$$

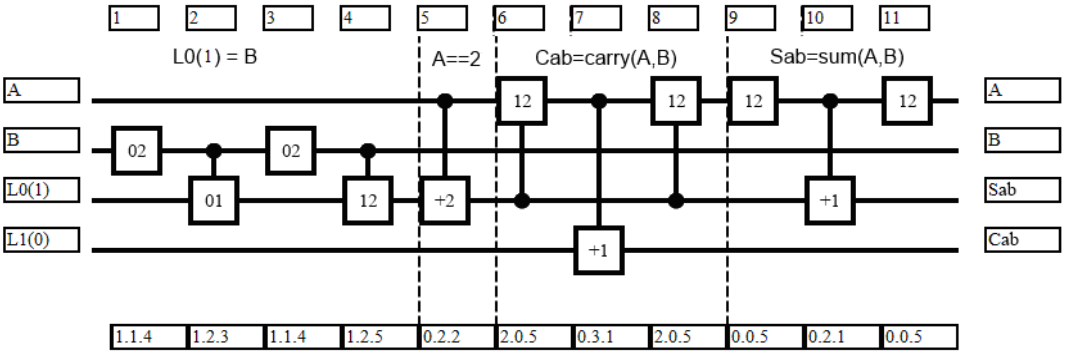


Fig. 1. Ternary reversible half-adder with two ancilla bits ($L0(1)$ and $L1(0)$). Top row shows the column (gene) number, bottom row -chromosome coding.

$F2$ – a component that minimizes the amount of non-zero gates dG in a circuit for a chromosome of length dL:

$$F2 = \frac{dL - dG}{dL}, \tag{3}$$

$F2$ – a component that minimizes the amount of controlled M-S gates:

$$F3 = \frac{dGM}{dG}, \tag{4}$$

where dGM is the number of 1-qutrit gates; k_1, k_2, k_3 are weighting coefficients. In a search for valid logic circuits we always set k_1=1. Other coefficients of the fitness functions were chosen to be less than unity.

C. *Main Steps of the Genetic Algorithm*

Proposed genetic algorithm has the following main steps.

1. Generation of the initial population of chromosomes, each representing a possible solution to the problem of a reversible device synthesis.
2. Initialization of a seed – no duplicates are allowed in the offspring population. Removed chromosomes are replaced with random individuals.
3. Selection operator – panmixis. Panmixis is 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 individuals in the population. These numbers represent the individuals that will participate in the crossover. Crossovers of the individuals with themselves are discarded. Some members of the population will participate in reproduction with many other individuals. Despite the simplicity, this algorithm is universal for many classes of problems.
4. Crossover operation – a uniform crossover, exchanging genes of parents in each locus with a p_C probability. It ensures that offsprings would contain interchanging short chains from parents.
5. Mutation – a random gene change can happen in each locus with a probability p_m.
6. Offspring evaluation according to fitness function. If offsprings are better than the worst parent individuals, then the former replaces the latter and stagnant counter is reset. Otherwise, stagnation counter is incremented.
7. If stagnation counter hasn't reached a predetermined value, return to step 3. Otherwise, start the next cycle.
8. Analyze whether fitness function value have improved for a chosen amount of cycles. If not, run the inner GA which uses the obtained population as input. Inner GA has the features of the main one and is meant to fix at least one error in the truth table of the best chromosomes.
9. If the cycle counter hasn't reached the chosen value, return to step 2.
10. The algorithm stops after a chosen amount of cycles or when chromosomes with a fitness function larger or equal to 1 are obtained.

Post GA reduction is a process of minimization of the obtained circuits with an equivalent one but having smaller length. The following rules are used [14]: i) if there are two 1-qutrit gates connected in series in the same line they are replaced with one, according to the product of their corresponding matrices (Table 1); ii) a similar replacement is made for M-S gates, provided that they have the same controlling line; iii) a group consisting of 1-qutrit gate, M-S gates, 1-qutrit gate connected in series within a single controlled line is replaced with an equivalent 1-qutrit gate, according to Table 1, and M-S gates in series.

By applying the above genetic algorithm we have discovered that it quickly finds circuits with low amount of the output signal logic errors (from 13 down to 0, for various circuits). In case of finding a local maximum, it is easier to run another inner genetic algorithm which will try to fix at least one error and combine its result with the previous one, obtaining thus an individual with less errors and moving out of the local maximum at the same time.

IV. Experimental Results and Discussion

Proposed GA was used to design reversible ternary adder and half-adder. In case of ternary half-adder, the input signals are *A* and *B*, and output signals, apart from *A* and *B* which ensure the reversibility, are the sum S_{ab} and carry C_{ab} according to the truth table of half-adder (Table 4). According to Table 4:

$$S_{ab} = \text{sum}(A, B) = A \oplus B\,, \tag{5}$$

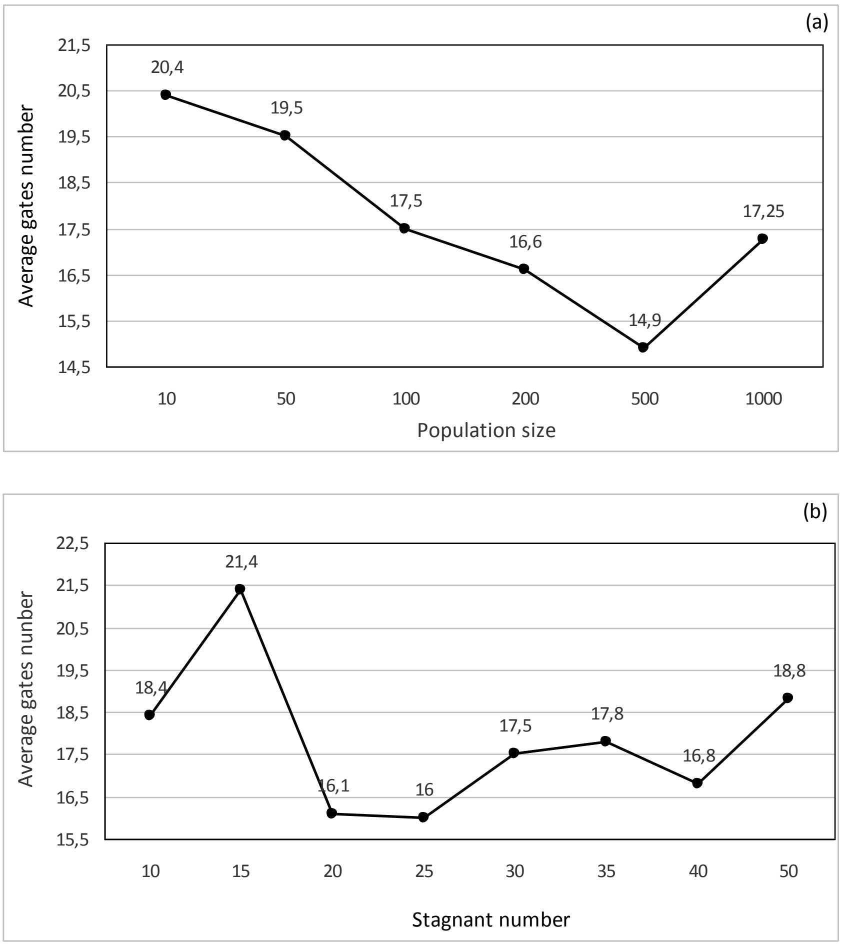
$$C_{ab} = \text{carry}\,(A, B) = \text{int}\left[\frac{A+B}{3}\right]. \tag{6}$$

Table 4. Truth Table of Ternary Half-Adder

A	*B*	C_{ab}	S_{ab}
0	0	0	0
0	1	0	1
0	2	0	2
1	0	0	1
1	1	0	2
1	2	1	0
2	0	0	2
2	1	1	0
2	2	1	1

Parameters of the algorithm were chosen according to the assessment described below and illustrated in Fig. 2. In particular, Fig. 2(a) shows the influence of population size on the average gate amount (chromosome length) in a synthesized half-adder. According to this assessment, the optimal population size is 500. Following the results in Fig. 2(b), the optimal stagnant number was chosen to be 25, whereas crossover probability p_c=0.5, and probability of mutation p_m=0.03 (Fig. 2(c), (d)). The best chromosome satisfying the above criteria is shown in Fig.1. Obtained half-adder circuit (chromosome) consists of the following elements: first, copying of information from line *B* to *L*0 is performed (*L*0=*B*, genes 1-4); next, the condition *A* = 2 is processed, i.e. if line *A* contains signal 2, an M-S gate with two inputs adds 2 to the content of the line (*L*0 = *B* + 2, gene 5); next, a carry in line *L*1 is formed (C_{ab} = carry (*A*,*B*), genes 6 – 8), and finally, a sum is formed in line *L*0 (S_{ab} = sum (*A*,*B*), genes 9–11).

A circuit of the ternary reversible half-adder obtained using our GA has four input lines – two informational and two constant. Obtained circuit contains 11 primitives, thus, assuming that the cost of each primitive is the same and equals 1, the cost (amount of elementary gates) of building the ternary half-adder is 11. The delay time of the obtained half-adder, as shown in Fig.1, equals $11t_0$, where t_0 is the delay time of a single gate.



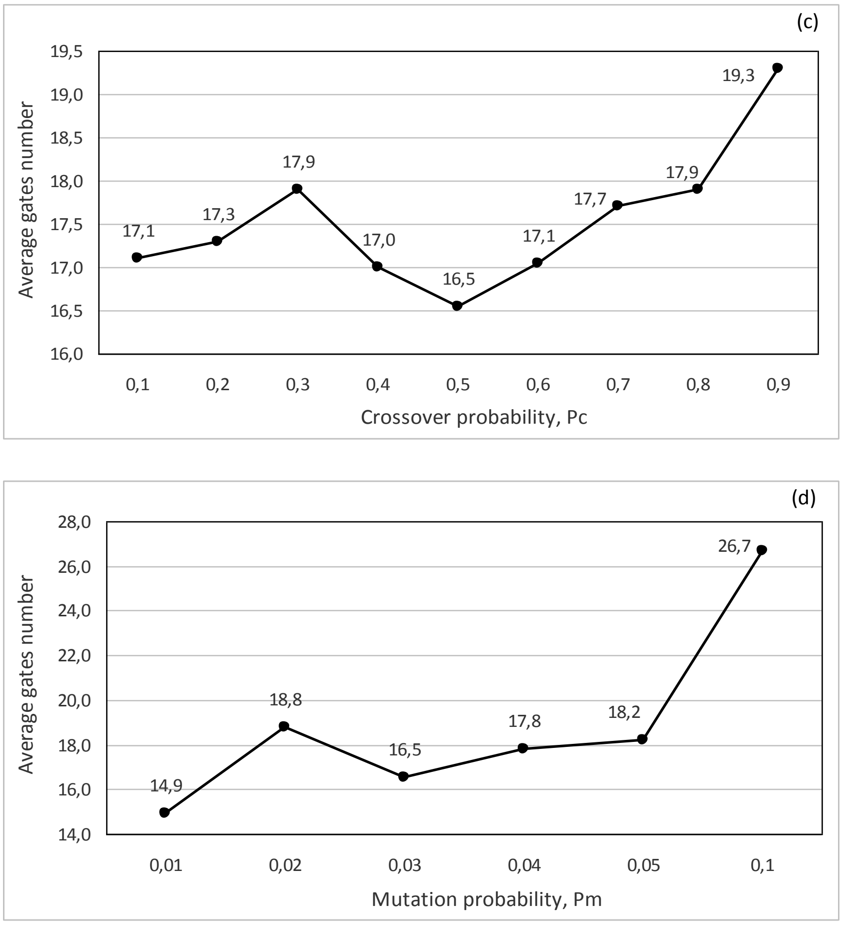


Fig. 2. Dependence of the average gate amount of a ternary reversible half-adder on (a) population size (p_c=0.5, p_m=0.03), (b) stagnant number (p_c=0.5, p_m=0.03, pop-size =500), (c) crossover probability (p_m=0.03, pop-size =500), and (d) mutation probability (p_c=0.5, pop-size =500).

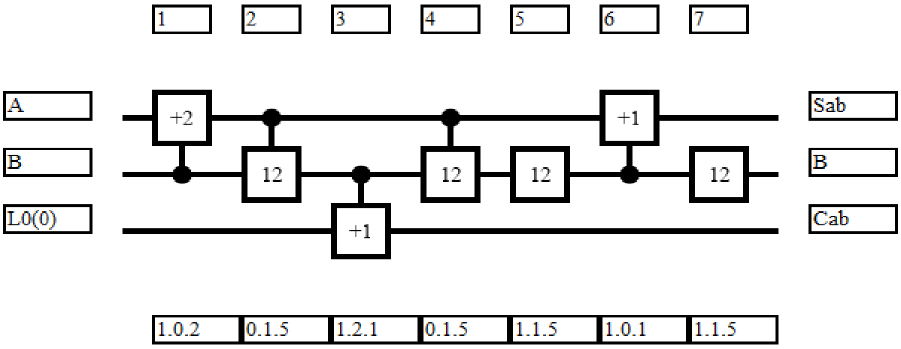


Fig. 3. Ternary reversible half-adder with one ancilla bit L0(0).

Compared to previous work where the cost of the ternary half-adder was 19[14], our ternary half-adder developed using the proposed genetic algorithm contains less elementary gates. Genetic algorithm proposed above also allows performing a search for a circuit of a reversible ternary half-adder with three inputs, one of which is constant (ancilla). In this case, apart from the sum and carry, the output contains a copy of only one input signal. Optimized circuit with minimal amount of logic primitives is shown in Fig.3. The cost of such half-adder is 7 and the delay time is $7t_0$, compared to a cost of 9 for a 3-input reversible half-adder proposed in [15].

Using the half-adder circuits we have also built a ternary full adder. Supplying the input of the adder with three signals A, B, C_i, we obtain a sum S and carry C on output which are defined as

$$S = \mathrm{sum}(A,B,C_i) = A \oplus B \oplus C_i \,, \qquad (7)$$

$$C = \mathrm{carry}\,(A,B,C_i) = \mathrm{int}\left[\frac{A+B+C_i}{3}\right]. \qquad (8)$$

Taking into account (5) and (6), we transform (7) and (8) to the following form [12]:

$$S = \mathrm{sum}(S_{ab}, C_i)\,, \tag{9}$$

$$C = \mathrm{sum}\big(C_{ab}, \mathrm{carry}\,(S_{ab}, C_i)\big)\,. \tag{10}$$

Proposed GA was used to obtain the optimal circuit of a ternary full adder in the basis of 1-input and 2-input M-S gates described above. The best obtained 5-input chromosome (Fig. 4) according to described criteria contains the following elements: first, the sum and carry are formed based on the half-adder circuit (Fig. 1), genes 1 – 11. After that, the output carry of the full adder *C* is formed (genes 12 – 14) using the input carry C_i, sum S_{ab} and carry C_{ab} of the half-adder according to eq.(10). Finally, a sum *S* is formed (genes 15 – 17) using the previously obtained input transfer C_i and sum S_{ab} according to (9). The circuit of the ternary full adder obtained using our genetic algorithm has 5 input lines – three informational (*A*, *B*, and C_i) and two constant (ancilla bits) (*L*0 and *L*1, which receive, similarly to the case of half-adder, constant signals 1 and 0). On output, the three lines repeat the input signals (*A*, *B*, and C_i) and the other two provide the sum and carry to the next rank (*S* and *C*). Obtained circuit contains 17 primitives, corresponding to the cost of realization of 17. The delay time of the obtained full adder, as can be seen from Fig. 4, equals $16t_0$, where t_0 is the single gate delay time. For comparison, the cost of ternary full adder developed by Khan et al. in [12] is 50 and requires 4 ancilla bits. The cost of full-adder realization by Miller et al. [8] is 96. Our approach based on genetic algorithm allowed obtaining ternary reversible adder with a much smaller amount of elementary gates than previous work [8,12] and a shorter delay time.

To optimize the amount of the input lines of the reversible ternary adder we have used the circuit of the half-adder with three inputs (Fig. 3) (one ancilla bit). After replacing the genes 1–11 in the circuit in Fig. 4 with the genes 1-7 of the half-adder (Fig. 3) we obtain 4-input circuit of the reversible ternary full-adder with one ancilla bit (Fig. 5) with a cost of 13, as compared to the cost of 18 presented in previous work [15], i.e. requiring less elementary gates.

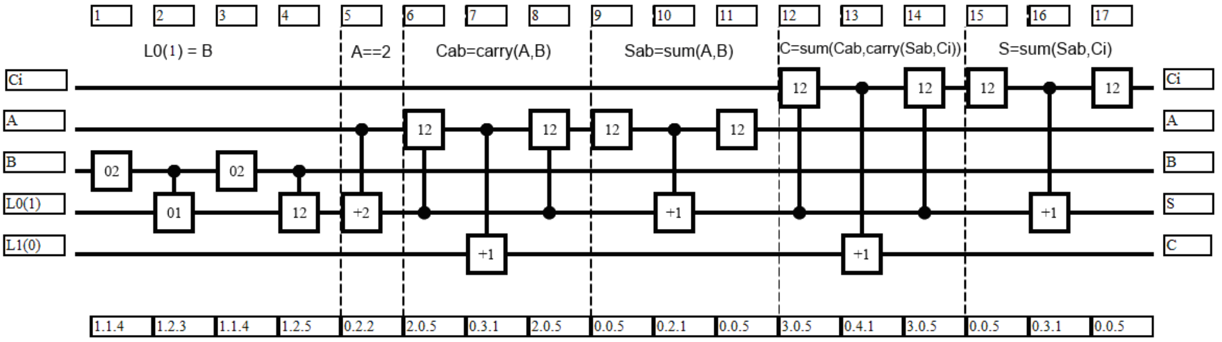


Fig. 4. Circuit of a reversible ternary full adder with two ancilla bits (*L*0(1) and *L*1(0)).

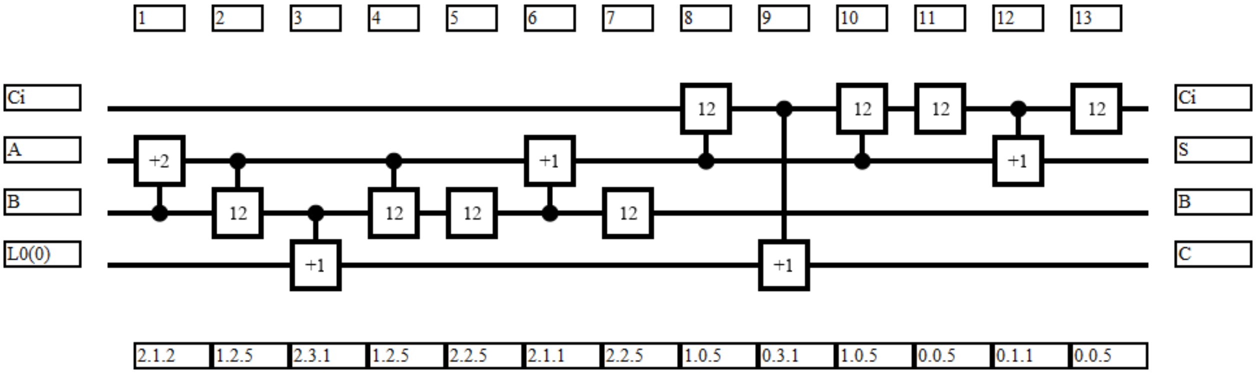


Fig. 5. Circuit of a ternary full adder with one ancilla bit L0(0).

V. Conclusion

We have described the use of genetic algorithm to find the optimal design of ternary reversible/quantum logic devices on the example of half-adder and full adder. Due to complexity of realization of quantum gates with more than two inputs related to the difficulty of controlling them, we used liquid ion-trap realizable 1-qubit gates and 2-qubit Muthukrishnan-Stroud primitive gates [3]. Proposed method for chromosome coding and judicious choice of parameters of the algorithm allowed us to obtain circuits for ternary half-adder and full adder which are better than other available approaches. We have achieved lower-cost reversible devices in terms of elementary gates they are built of. In particular, using our approach we have developed a ternary reversible full adder with two ancilla bits with a cost of 17 and a full adder with one ancilla bit with a cost of 13, as compared to previous works achieving costs of 18 by Zobov et al. [15], 50 by Khan et al. [12], and 96 by Miller et al. [8]. Moreover, the proposed implementation of the genetic algorithm allowed shortening the device delay time and reducing the amount of ancilla bits to 1. A notable feature of the proposed approach is the use of the modeling results for half-adder to build an optimal design of a ternary full adder. We have also addressed a strategy for choosing an effective fitness-function. We suggest using a fitness-function whose main component minimizes the amount of logic errors in the output according to the truth table of the synthesized device. Other components minimize the amount of 1- and 2-input primitives and provide only minor contribution to the fitness-function.

Proposed evolutionary approach to the design of reversible ternary adders allows addressing the design of ternary parallel adders and subtractors with various ways of carry which will be addressed in future work.

ACKNOWLEDGMENTS

We thank Dr. Oleksandr Voznyy for proofreading early versions of the manuscript and for helpful discussions.

REFERENCES

[1] M. Nielsen, I. Chuang, *Quantum Computation and Quantum Information*. Cambridge: Cambridge University Press, 2000.

[2] A. De Vos, Y. van Rentergem, “Multiple-valued reversible logic circuits,” *J. Multiple-Valued Logic and Soft Comput.*, vol.15, no.4-5, pp.489-505, 2009.

[3] A. Muthukrishnan, C. R. Stroud, Jr., “Multivalued logic gates for quantum computation,” *Physical Review A,* vol. 62, no. 5, pp. 052309/1-8, 2000..

[4] A. B. Klimov, R. Guzman, J. C. Retamal, C. Saavedra, “Qutrit quantum computer with trapped ions,” *Physical Review A,* vol. 67, no. 6, 062313/1-7, 2003.

[5] D. McHugh, J. Twamley, “Trapped-ion qutrit spin molecule quantum computer”, *New J. Physics*, vol. 7, no. 1, pp. 174/1-9,2005.

[6] D. M. Miller, M. A. Thornton, *Multiple Valued Logic: Concepts and Representations*. Morgan & Claypool Publishers, 2008.

[7] M. Lukac, M. Perkowski, H. Goi, M. Pivtoraiko, C. H. Yu, K. Chung, H. Jee, B-G. Kim, Y-D. Kim, “Evolutionary approach to Quantum and Reversible Circuits synthesis,” *Artificial Intelligence Review,* vol. 20, no. 3-4, pp. 361-417, 2003.

[8] D.M. Miller, D. Maslov, 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.

[9] D.M. Miller, R. Wille, R. Drechsler, ”Reducing Reversible Circuit Cost by Adding Lines”, *J. Multiple-Valued Logic and Soft Comput.*, vol. 9, no. 1-3, pp. 185-201, 2012.

[10] M. Lukac M. Perkowski and M. Kameyama, “Evolutionary quantum logic synthesis of boolean reversible logic circuits embedded in ternary quantum space using structural restrictions”, *Proceedings of the IEEE Congress on Evolutionary Computation* (CEC 2010), Barcelona, Spain, 18-23 July 2010, pp. 1-8.

[11] 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. 5-6, pp. 567-602, 2005.

[12] M.H.A. Khan, M.A. Perkowski, ”Quantum Ternary Parallel Adder/Subtractor with Partially-Look-Ahead Carry,” *J. Systems Architecture*, vol.53, no.7, pp. 453-464, 2007.

[13] M. H. A. Khan, “Design of Reversible/Quantum Ternary Comparator Circuits,” *Engineering Letters,* vol. 16, no. 2, pp. 178-164, 2008.

[14] R. Khanom, T. Kamal, M. H. A. Khan, ”Genetic Algorithm Based Synthesis of Ternary Reversible/Quantum Circuits,” *Proceedings of 11th IEEE Int. Conf. on Computer and Information Technology* (ICCIT 2008), Khulna, Bangladesh, 25-27 December, 2008, pp. 270-275.

[15] V.E. Zobov, D.I. Pekhterev, “Adder on ternary base elements for a quantum computer,” *JETP Letters,* vol. 89, no. 5, pp. 260-263, 2009.

[16] R. P. Zadeh, M. Haghparast, “A New Reversible/Quantum Ternary Comparator,” *Australian J. Basic and Applied Sciences*, vol. 5, no. 12, pp. 2348-2355, 2011.

[17] T. Satoh, S. Nagayama, R. Van Meter, “A Reversible Ternary Adder for Quantum Computation,” *Asian Conf. on Quantum Information Science* (AQIS-07). Kyoto, Japan, 3-6 September, 2007.

[18] X. Li, G. Yang, D. Zheng, „Logic Synthesis of Ternary Quantum Circuits with Minimal Qutrits,” *Journal of Computers,* vol. 8, no. 8, pp. 1941-1946, 2013.

[19] Y.-M. Di, and H.-R. Wie, “Synthesis of Multivalued Quantum Logic Circuits by Elementary Gates,” *Physical Review A*, vol. 87, no. 1, pp. 012325/1-8, 2013.

[20] R. S. Zebulum, M. C. Pachecco, M. M. Vellasco, *Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms*. CRC Press, 2002.

[21] L. Spector, *Automatic Quantum Computer Programming: A Genetic Programming Approach*. Kluwer Academic Publishers, 2004.

[22] A. E. Eiben, J. E. Smith, *Introduction to Evolutionary Computing*. Springer-Verlag, 2013.

Authors’ Profiles

Vitaly Deibuk: Professor of computer engineering at Faculty of Computer Science in Chernivtsi National University, Ukraine.

Andrij Biloshytskyi: Post-graduate Student for PhD in computer engineering at Faculty of Computer Science in Chernivtsi National University, Ukraine.