INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2017, VOL. 63, NO. 3, PP. 235-240
Manuscript received January 18, 2017; revised June, 2017. DOI: 10.1515/eletel-2017-0031

Graphical Method of Reversible Circuits Synthesis

Andrzej Skorupski

***Abstract*—This paper presents a new approach to designing reversible circuits. Reversible circuits can decrease energy dissipation theoretically to zero. This feature is a base to build quantum computers. The main problem of reversible logic is designing optimal reversible circuits i.e. circuits with minimal gates number implementing the given reversible function. There are many types of reversible gates. Most popular library is a set of three types of gates so called CNT (Control, NOT and Toffoli). The method presented in this paper is based only on the Toffoli gates. A graphical representation of the reversible function called s-maps is introduced in the paper. This representation allows to find optimal reversible circuits. The paper is organized as follows. Section 1 recalls basic concepts of reversible logic. In Section 2 a graphical representation of the reversible functions is presented. Section 3 describes the algorithm whereby all optimal solutions of the given function could be obtained.**

***Keywords*—reversible logic, reversible circuits, reversible gate, Toffoli gate**

I. Introduction

SYNTHESIS of reversible circuits is one of very important problems in new research areas i.e. quantum computing, nanotechnologies as well as in low power computation [1]. A circuit is called reversible if there is a one-to-one correspondence between its inputs and outputs. The idea of such circuits was developed on the base of Landauer principle [2]. Rolf Landauer in 1961 indicated a link between information theory and thermodynamics. He argued that the erasure of information is a dissipative process. Hence circuits without loose information do not need energy. Therefore, reversible logic synthesis has been intensively studied recently. The attention is focused mainly on the synthesis of circuits built from the NCT library of gates, i.e. NOT, CNOT and Toffoli gates [3]. But various researchers uses only Toffoli gates to design reversible circuits. A novel method of synthesis using only Toffoli gates is presented in this paper This method leads to obtaining an optimal solution for the given function. Usually there exist more than one optimal solution and the method allows to find all of them.

The set of *n* balanced Boolean functions of *n* variables is called reversible if each input vector is mapping into a unique output vector. The Boolean function is called balanced if the number of 0's minterms is equal to the number of 1's minterms. From this definition originates the main feature of reversible circuits. It is possible to determine the input vector if the output vector is known. Such circuits fulfil Landauer principle and there is no loss of information.

The example of three variables reversible function is presented in Table I. In this case the reversible function is described by the true table. In this article there will be used two other notations: one as permutation of the output vectors and the other as s-maps introduced in section 3.

TABLE I
EXAMPLE OF THREE VARIABLE REVERSIBLE FUNCTION

No.	$X_2X_1X_0$	$Y_2Y_1Y_0$
0	000	000
1	001	011
2	010	010
3	011	001
4	100	100
5	101	101
6	110	111
7	111	110

This function from Table I could be presented as permutation of output vectors i.e. <0,3,2,1,4,5,7,6>. This is a sequence of the output vectors $Y_2Y_1Y_0$. To implement the reversible function a designer uses reversible gates. A lot of effort has been made to implement reversible gates. Researchers try various technologies. The best known are: semiconductor gates YBS [4], NMR gates [5], ION trap gates [6], quantum dots [7], optical lattice [8].

In this article the technological implementations will be not presented and attention will be focused on logical features of the reversible gate. Many gate libraries have been examined in literature. Very often the NCT library of gates is used. In 1980 they proposed a library with so called generalized Toffoli gate [9]. This gate contains XOR gate on one input line. The two arguments of this gate are:

1. proper variable on this line,
2. AND function of the remaining variables with or without inverter (NOT gate).

As it will be seen each of these gates swaps one pair of input vectors. Thus the sequence of the output vectors differ from the input sequence only in two positions .

For the generalized Toffoli gates some researches proposed a few methods of synthesis of reversible circuits [10]-[13]. They differ in terms of complexity of algorithms. In this paper there will be presented a novel method of an optimal synthesis. This method uses a simple algorithm. For three variables reversible functions a designer can perform synthesis without computer software.

Author is with Institute of Computer Science, Warsaw University of Technology, Poland (e-mail: ask@ii.pw.edu.pl).

Logical features of three variables Toffoli gates are presented in Fig. 1. There are four Toffoli gates with XOR gate located on X_0 line. The symbol of these gates will be begin with T0.

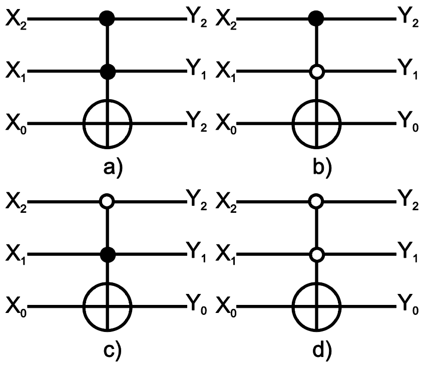


Fig.1. Graphical representations of reversible gates: a) T0, b) T0-1, c) T0-2, d) T0-12.

The gates from Fig. 1 implement the reversible function:

$Y_2 = X_2$

$Y_1 = X_1$

$Y_0 = X_0 \oplus ab$ where $a = X_1$ or $\bar{X}_1$ and $b = X_2$ or $\bar{X}_2$

The AND function arguments are X_i if there is a black dot on the line X_i. The AND function arguments are $\bar{X}_i$ if there is a white dot on the line X_i. Thus the output functions Y_1 and Y_2 have a values of X_1 and X_2. And the output function Y_0 will be as follow:

$Y_0 = X_0 \oplus X_1 X_2$ Fig. 1a

$Y_0 = X_0 \oplus \bar{X}_1 X_2$ Fig. 1b

$Y_0 = X_0 \oplus X_1 \bar{X}_2$ Fig. 1c

$Y_0 = X_0 \oplus \bar{X}_1 \bar{X}_2$ Fig. 1d

In the same manner one should define the other eight gates: four with XOR on line X_2 (T2, T2-0, T2-1, T2-01) and four on line X_1 (T1, T1-0, T1-2, T1 02). Therefore, there exist 12 three variable Toffoli gates.

Lemma 1. The output function of Toffoli gate is the input function with two swapped rows.

Proof. Let be an input function I (identical function i.e. left side of Tab. I). The output function of gate T0 is $Y_0 = X_0 \oplus X_1X_2$. If $X_1X_2=0$ (rows 0,1,2,3,4,5) the output function is the same as the input function. Only if $X_1X_2=1$ (rows 6 and 7) $Y_0= \bar{X}_0$. The output function of gate T0 in row 6 has a value 7 and in row 7 has a value 6. It is a function with two swapped rows 6 and 7.

In the same manner it is possible to show the output functions for all the other gates. The results of this process are presented in Table II. In the first column of this table there are gates symbols and in the second column there are appropriate swapped pair vectors.

To implement the given reversible function a designer can use a proper sequence of reversible gates. All of the 12 gates from Table II can be used. The target of the synthesis of the reversible function is to find a cascade of reversible gates as it is shown in the Fig. 2.

In Fig. 2 there are two complementary solutions of the synthesis problem. The first solution is a cascade with the input vectors F (given function – from the right side of the true table) and the output vectors I (identical function – from the left side of the true table). The second solution is a cascade with the input vectors I (from the left side of the true table) and the output vectors F (from the right side of the true table). Both cascades implement the given reversible function but the first cascade transforms the function F into the function I and the other cascade transforms the function I into the function F. In this paper only the first type of the cascade will be used. The main target for designers is to find an optimal result i.e. cascades with minimal number of gates.

TABLE II
SWAPPED VECTORS BY REVERSIBLE GATES

Gate	Swapped rows
T0	6,7
T0-1	4,5
T0-2	2,3
T0-12	0,1
T1	5,7
T1-0	4,6
T1-2	1,3
T1-20	0,2
T2	3,7
T2-0	2,6
T2-1	1,5
T2-01	0,4

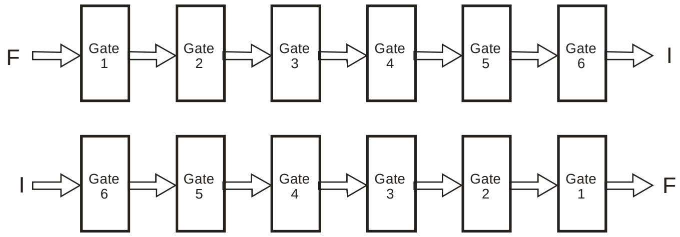


Fig. 2. Two cascades with 6 reversible gates implement the same function: Upper cascade transform given function into identical function. Bottom cascade transform identical function into given function.

II. GRAPHICAL FUNCTION REPRESENTATION

The true table and permutation notation are not convenient during the synthesis process. This is a reason to introduce an easy notation for designers. Because each of the output Boolean functions of reversible functions are balanced, i.e. the number of 1's is equal to the number of 0's, it is convenient to introduce the graphical representation. A set of s-maps composed in two rows is proposed here. In the upper row there are located the vector numbers (minterms) corresponding to 0's of function and in the bottom row the minterms corresponding to 1's of function. The number of the s-maps for the given function is equal to the number of variables. The s-maps for the function I of three variables are shown in the Fig. 3.

X_2

a	b	c	d
0	1	2	3
4	5	6	7

X_1

e	f	g	h
0	1	4	5
2	3	6	7

X_0

i	k	l	m
0	2	4	6
1	3	5	7

Fig. 3. Graphical presentation of I function.

All the columns of s-maps from Fig. 3 are marked by letters from a to m in order to indicate swapping pairs corresponding to the given gate. The graphical representation of the reversible function F from Table I is shown in the Fig. 4.

Y_2				Y_1				Y_0			
a	b	c	d	e	f	g	h	i	k	l	m
0	3	2	1	0	3	4	5	0	2	4	7
4	5	7	6	2	1	7	6	3	1	5	6

Fig. 4. Graphical representation of function <0,3,2,1,4,5,7,6>.

It is easy to note that the output function Y_2 is ordered i.e. all minterms from the upper row of s-map X_2 (Fig. 3) are in the upper row of s-map Y_2 and all minterms from the bottom row of s-map X_2 are in the bottom row of s-map Y_2. The situation with the maps Y_1 and Y_0 is different. Minterms 3 and 1 must be swapped to ordered Y_1 (column *f*) and minterms 6 and 7 must be swapped to ordered Y_0 (column *m*). These improper fields on the s-maps Y_1 and Y_0 are highlighted. It is shown in the Fig. 5.

Y_2				Y_1				Y_0			
a	b	c	d	e	f	g	h	i	k	l	m
0	3	2	1	0	3	4	5	0	2	4	7
4	5	7	6	2	1	7	6	3	1	5	6

Fig. 5. Graphical representation of function <0,3,2,1,4,5,7,6> with highlighted fields.

The design process involves elimination of the improper fields i.e. highlighted fields. For example, for the function from Table I a designer must use the gate T1-2 to swap minterms 3 and 1 (column *f* in Table III) and gate T0 to swap minterms 7 and 6 (column *m* in Table III). The s-maps of the output function after gate T1-2 (so called rest function) are shown in Fig. 6.

Y_2				Y_1				Y_0			
a	b	c	d	e	f	g	h	i	k	l	m
0	1	2	3	0	1	4	5	0	2	4	7
4	5	7	6	2	3	7	6	1	3	5	6

Fig. 6. Graphical representation of rest function after gate T1-2.

The gate T1-2 swaps minterms in column *f* in the s-map Y_1 and it swaps these minterms in rows of the remaining s-maps Y_2 and Y_0. These swaps in rows of the s-maps Y_2 and Y_0 will be called a shift.

The function from Fig. 6 is the input function of gate T0. The rest function after gate T0 is shown in Fig. 7.

Y_2				Y_1				Y_0			
a	b	c	d	e	f	g	h	i	k	l	m
0	1	2	3	0	1	4	5	0	2	4	6
4	5	6	7	2	3	6	7	1	3	5	7

Fig. 7. Graphical representation of rest function after gate T0.

Because the rest function after gate T0 in Fig. 7 is identical to the function I the design process is completed.

These two gates in the cascade are the solution of the optimal synthesis of the reversible function from Table I. In this case the sequence of the gates does not matter. Both cascades are shown in Fig. 8.

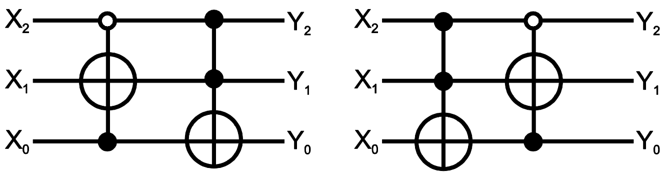


Fig. 8. Reversible cascades implemented of function from Tab. I.

This consideration leads to the design process of the reversible function. The beginning of this process is a selection of the first gate in the cascade. Next the rest function after this gate should be designated. If the rest function is not identical to the function I the process for the rest function should be repeated.

III. Algorithm of Synthesis

The synthesis problem was solved by Yang and others [13]. They used Hamming distance as a criterion to choose a gate in the proper place in the cascade. In this paper there will be presented a novel algorithm based on a graphical representation of the given function.

Let be given a reversible function F. The first step for a designer is to select the first gate in the cascade. To do it is necessary to select the best gate in order to obtain an optimal circuit. As we have seen every gate swaps minterms in one column of one s-map and shifts minterms in the remaining s-maps. Examination of the rest function results in the choice of the best gate. The best choice must lead to the rest function with maximum one over the other highlighted fields. To determine the choice an introduction of two criteria is necessary:

1. in the s-maps of the given function there exist columns with both highlighted fields,
2. the gate will be the best gate if in the s-maps of the rest function a new column with highlighted fields is created.

The best choice of selection is when both the above criteria are fulfilled. We can introduce a measurement of the degree of matching the given gate.

1. If there exists a column with two improper fields the corresponding gate receives +1 point.
2. If swapping one column causes in the other s-maps shifting the improper field under/above the other improper field the corresponding gate receives +1 point.
3. If there exists a column with two improper fields and in the other s-maps shifting of the improper field under/above the other improper field is caused the corresponding gate receive +2 points.
4. If swapping one column causes destruction of the others maps in the column with two improper fields the corresponding gate receive -1 point.

Usually more than one gate fulfilling the criteria of "best" gate is in existence. This means that there exist more than one optimal cascade.

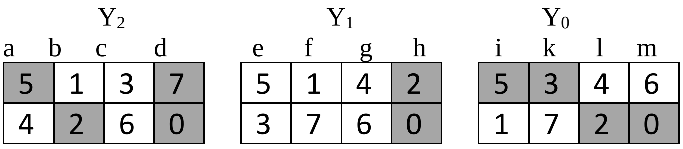
Having chosen the first gate in the cascade we repeat the same algorithm for the rest function of this gate. These steps are performed until the rest function is identical to the function I.

Example:

Consider a reversible function <5,1,3,7,4,2,6,0> from [2].

Step 1

The s-maps of this function is shown in the Fig. 9.



Y_2				Y_1				Y_0			
a	b	c	d	e	f	g	h	i	k	l	m
5	1	3	7	5	1	4	2	5	3	4	6
4	2	6	0	3	7	6	0	1	7	2	0

Fig.9. Graphical representation of function <5,1,3,7,4,2,6,0>.

The ranking of all gates should be determined for the given function. The punctuation for the ranking is collected in Table III.

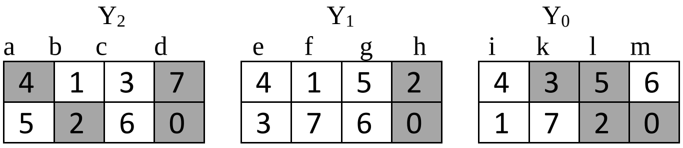
TABLE III
RANKING OF ALL 12 TOFFOLI GATES

Gate	Column	Punctuation $Y_2Y_1Y_0$
T0	m	-1,-1,0
T0-1	l	+1,-1,0
T0-2	k	-1,0,0
T0-12	i	+1,0,0
T1	h	0,+1,0
T1-0	g	0,-1,0
T1-2	f	0,-1,0
T1-20	e	0,-1,0
T2	d	+1,-1,+1
T2-0	c	-1,0,+1
T2-1	b	0,-1,+1
T2-01	a	0,0,+1

From Table III we select four gates as the best gates: T0-12, T1, T2 and T2-01. This choice indicates minimum four optimal solutions. The first gate T2-01 will be chosen for further consideration.

Step 2 for gate T2-01

This step will be repeated three times. Sequentially we will choose the gates: T1, T2 and T0-12. In Fig. 10 is presented the s-maps for the rest function after the gate T2-01.



Y_2				Y_1				Y_0			
a	b	c	d	e	f	g	h	i	k	l	m
4	1	3	7	4	1	5	2	4	3	5	6
5	2	6	0	3	7	6	0	1	7	2	0

Fig. 10. Graphical representation of rest function <4,1,3,7,5,2,6,0> after gate T2-01.

The gate T2-01 causes swap minterms 5 and 4 in column *a*. These minterms in the s-maps Y_1 and Y_0 are shifted. Now must be selected the next gate. The ranking of the gates for rest function (Fig. 10) is collected in Table IV.

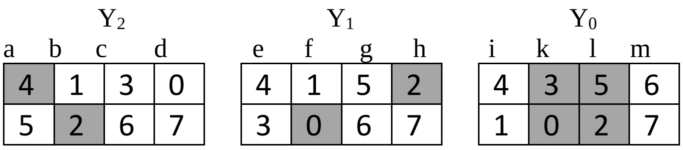
From Table IV we can select three best gates: T0-1, T1, T2. This result indicates three optimal cascades where the gate T2-01 is the first gate in the cascade. Temporarily, we will select the gate T2 and next consider the gates T0-1 and T1.

TABLE IV
RANKING OF ALL 12 TOFFOLI GATES

Gate	Column	Punctuation $Y_2Y_1Y_0$
T0	m	-1,-1,0
T0-1	l	+1,-1,+1
T0-2	k	-1,0,0
T0-12	i	+1,0,-1
T1	h	0,+1,0
T1-0	g	0,-1,0
T1-2	f	0,-1,0
T1-20	e	0,-1,0
T2	d	+1,-1,+1
T2-0	c	-1,0,+1
T2-1	b	0,-1,-1
T2-01	a	0,0,-1

Step 3 for gates T2-10, T2

In Fig. 11 there are presented s-maps of the rest function after the two gates: T2-01 and T2.



Y_2				Y_1				Y_0			
a	b	c	d	e	f	g	h	i	k	l	m
4	1	3	0	4	1	5	2	4	3	5	6
5	2	6	7	3	0	6	7	1	0	2	7

Fig. 11. Graphical representation of rest function after gate T2-10 and T2.

TABLE V
RANKING OF ALL 12 TOFFOLI GATES

Gate	Column	Punctuation $Y_2Y_1Y_0$
T0	m	0,0,-1
T0-1	l	+1,0,+1
T0-2	k	0,0,+1
T0-12	i	+1,0,-1
T1	h	0,0,-1
T1-0	g	0,-1,-1
T1-2	f	0,0,-1
T1-20	e	0,-1,-1
T2	d	-1,+1,-1
T2-0	c	-1,0,-1
T2-1	b	0,+1,-1
T2-01	a	0,0,-1

The minterms 0 and 7 in column *d* are swapped and these minterms in the s-map Y_1 and Y_0 are shifted. The ranking of the gates for this function is collected in Table V. In Table V we can find the unique best gate T0-1 with +2 punctuation.

Step 4 for gates T2-10, T2, T0-1

Fig. 12 contain s-maps of the rest function after gates T2-10, T2 and T0-1.

Y_2				Y_1				Y_0			
a	b	c	d	e	f	g	h	i	k	l	m
4	1	3	0	4	1	2	5	4	3	2	6
2	5	6	7	3	0	6	7	1	0	5	7

Fig. 12. Graphical representation of rest function after gate T2-10, T2 and T0-1.

Without the ranking it is easy to observe two best gates T2-01 (column *a*) and T0-2 (column *k*) in the table. Temporarily the gate T2-01 will be selected.

Step 5 for gates T2-10, T2, T0-1, T2-01

Figure 13 contain s-maps of the rest function after gates T0-12, T2, T0-1 and T2-01. Without the ranking it is easy to observe the unique best gate T0-2 (column *k*) in Fig. 13.

Y_2				Y_1				Y_0			
a	b	c	d	e	f	g	h	i	k	l	m
2	1	3	0	2	1	4	5	2	3	4	6
4	5	6	7	3	0	6	7	1	0	5	7

Fig. 13. Graphical representation of rest function after gate T2-10, T2, T0-1, T2-01.

Step 6 for gates T2-10, T2, T0-1, T2-01, T0-2

Figure 14 contain s-maps of the rest function after gates T2-01, T2, T0-1, T2-01 and T0-2.

Y_2				Y_1				Y_0			
a	b	c	d	e	f	g	h	i	k	l	m
2	1	0	3	2	1	4	5	2	0	4	6
4	5	6	7	0	3	6	7	1	3	5	7

Fig. 14. Graphical representation of rest function after gate T2-10, T2, T0-1, T2-01, T0-2.

Only the gate T1-20 is the best gate. It is easy to notice that the rest function after the gate T1-20 is identical to the function I. It causes a stop of this part of the algorithm. In this manner we obtain a cascade with six gates. It is the cascade with gates:

T2-10, T2, T0-1, T2-01, T0-2, T1-20

This cascade is presented in Fig. 15.

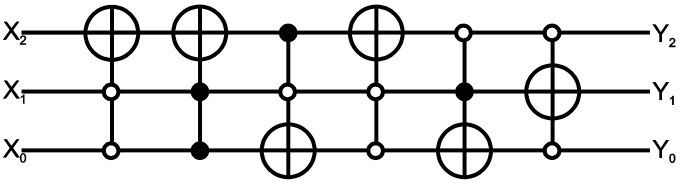


Fig. 15. The one of cascades implemented given function.

Step 5 for gates T2-10, T2, T0-1, T0-2

In step 4 two best gates are chosen. Now the algorithm must come back for the second time to step 5. Must be choose the gate T0-2. It will be exhaust best gates from step 4. Must be determinate the rest function after gates T2-10, T2, T0-1, T0-2. Fig. 16 contains the s-maps of this rest function. In this case also without ranking it is easy to observe the unique best gate T2-01.

Y_2				Y_1				Y_0			
a	b	c	d	e	f	g	h	i	k	l	m
4	1	0	3	4	1	2	5	4	0	2	6
2	5	6	7	0	3	6	7	1	3	5	7

Fig. 16. Graphical representation of rest function after gate T2-10, T2, T0-1, T0-2.

Step 6 for gates T2-10, T2, T0-1, T0-2, T2-01

In this step the best gate is the gate T1-20 and after this gate the rest function is I which indicates a new cascade:

T2-01, T2, T0-1, T0-2, T2-01, T1-20

Now the algorithm comes back to step 2 where there are three best gates T0-1, T1, T2. The algorithm recommends repetition of steps from step 3 for the gates T0-1 and T1.

Step 3 for gates T2-10, T1

The s-maps of the rest function after gates T2-10 and T1 are shown in Fig. 17.

Y_2				Y_1				Y_0			
a	b	c	d	e	f	g	h	i	k	l	m
4	1	3	7	4	1	5	0	4	3	5	6
5	0	6	2	3	7	6	2	1	7	0	2

Fig. 17. The maps of rest function after gate T2-01 and T1.

The ranking punctuation for this function is collected in Table VI.

TABLE VI
RANKING OF ALL 12 TOFFOLI GATES

Gate	Column	Punctuation $Y_2Y_1Y_0$
T0	m	-1,0,0
T0-1	l	+1,0,+1
T0-2	k	-1,0,0
T0-12	i	+1,0,-1
T1	h	0,-1, 0
T1-0	g	0,-1, 0
T1-2	f	0,-1,0
T1-20	e	0,-1, 0
T2	d	+1,0,+1
T2-0	c	-1,0,+1
T2-1	b	0,0,-1
T2-01	a	0,0,-1

From Table VI we can indicate two best gates: T0-1 and T2 with punctuation +2. First will be consider the gate T0-1 and after the gate T2.

Step 4 for gates T2-10, T1, T0-1

The s-maps of the rest function after gates T2-10, T1 and T0-1 are shown on the Fig. 18.

Y_2				Y_1				Y_0			
a	b	c	d	e	f	g	h	i	k	l	m
4	1	3	7	4	1	0	5	4	3	0	6
0	5	6	2	3	7	6	2	1	7	5	2

Fig. 18. The s-maps of rest function after gate T2-10, T1, T0-1.

Without the ranking table we can see that the gate T2 is the unique best gate.

Step 5 for gates T2-10, T1, T0-1, T2

The s-maps of the rest function after gates T2-10, T1, T0-1 and T2 are shown in Fig. 19.

Y_2				Y_1				Y_0			
a	b	c	d	e	f	g	h	i	k	l	m
4	1	3	2	4	1	0	5	4	3	0	6
0	5	6	7	3	2	6	7	1	2	5	7

Fig. 19. The s-maps of rest function after gate T2-10, T1, T0-1 and T2.

Without the ranking we can see that the gate T0-2 and T2-01 are the best gates.

Step 6 for gates T2-10, T1, T0-1, T2, T0-1
Step 6 for gates T2-10, T1, T0-1, T2, T2-01

The s-maps in Fig. 19 indicate further sequence of the two gates T0-2 and T2-01. Therefore, there exist two optimal cascades:

T2-01, T1, T0-1, T2, T0-2, T2-01
T2-01, T1, T0-1, T2, T2-01, T0-2

Subsequent two cascades are found and the algorithm comes back to step 3 where, except gate T0-1 was gate T2. Steps 4, 5 and 6 give below result

T2-01, T1, T2, T0-1, T0-2, T2-01
T2-01, T1, T2, T0-1, T2-01, T0-2

Now the algorithm comes back to step 2 where, except gate T2 and T1 the gate T0-1 are found.

Step 3 for gate T2-01 and T0-1

The s-maps of the rest function after gates T2-01 and T0-1 are shown on Fig. 20.

Y_2				Y_1				Y_0			
a	b	c	d	e	f	g	h	i	k	l	m
4	1	3	7	4	1	2	5	4	3	2	6
2	5	6	0	3	7	6	0	1	7	5	0

Fig. 20. Graphical representation of rest function <4,1,3,7,2,5,6,0> after gate T2-01 and T0-1.

The best gate for s-maps in Fig. 20 is the gate T2. Using further steps of the algorithm two more cascades can be found:

T2-01, T0-1, T2, T2-01, T0-2, T1-20
T2-01, T0-1, T2, T0-2, T2-01, T1-20

These eight solutions finish the repetition of any further steps of the algorithm after step 2.

The last loop of the algorithm begins again in step 2 because in step 1 we find the gates T1, T2 and T0-12 except the gate T2-01. Repeating the steps from 2 to 6 can help to obtain more optimal solutions. Beginning from the gate T0-12 we obtain eight optimal cascades and beginning from the gate T1 the result will be the same. But beginning from the gate T2 we obtain six optimal cascades. This algorithm allows to find 30 optimal cascades, everyone with 6 gates.

IV. Conclusions

The main aim of this paper is a design of optimal reversible cascades which enables implementation of the given function. The presented algorithm for the synthesis of three variable reversible functions allows to design optimal reversible circuits and find optimal solutions by "manual" process. But for more variables it is possible to transform this algorithm into software algorithm. This algorithm is also scalable to a greater number of variables. Other component of this work is transformation this method into other gate sets, especially NCT set. In this case will be more difficult to find optimal solutions by executable on paper algorithm.

References

[1] De Vos, "Reversible Computing. Fundamentals, Quantum Computing, and Applications", Wiley-VCH, Berlin 2010.

[2] R. Landauer, "Irreversibility and heat generation in the computing process", *IBM Journal of Research and Development*," vol. 5, 1961, pp. 183-191.

[3] O. Golubitsky and D. Maslov, "A study of optimal 4-bit reversible Toffoli circuits and their synthesis," *IEEE Transactions on Computers*, vol. 61, no. 9, 2012,. pp. 1341-1353.

[4] E. Forsberg, "Reversible Logic Based on Electron Waveguide Y-branch Switches", *Nanotechnology*, March 2004, vol. 15, no. 4

[5] R. Marx, A. F. Fahmy, John M. Myers, W. Bermel, and S. J. Glaser, "Approaching five-bit NMR quantum computing", *Phys. Rev.* A 62, June 2000

[6] C. Monroe, J. Bollinger, "Atomic physics in ion traps", *Physics World*, March 1997

[7] R. Akter, N. Islam, S. Waheed, "Implementation of Reversible Logic Gate in Quantum Dot Cellular Automata", *International Journal of Computer Applications*, Volume 109, No. 1, January 2015

[8] H. Deutsch, G. K. Brennen, P. S. Jessen, "Quantum computing with neutral atoms in an optical lattice", *Special Issue on Physical Implementations of Quantum Computing –Fortschritte der Physik* 48, 2000

[9] T. Toffoli, J. W. d. Bakker, J. v. Leeuwen, "Reversible computing: MIT LCS TM-151", 1980.

[10] Y. Zheng, C. Huang, "A novel Toffoli network synthesis algorithm for reversible logic," IEEE ASP-DAC, Yokohama, January 2009.

[11] I. M. Tsai S. Y. Kuo, "An algorithm for minimum space quantum boolean circuits construction", *J. Circuit Syst. Comp.*, vol. 15, pp. 719-738, October 2006.

[12] M. Saeedi, M. Sedighi, M. S. Zamani, "A novel synthesis algorithm for reversible circuits", IEEE/ACM ICCAD, California, USA, November 2007.

[13] Y. Yang, H. Chen, S. Kuo, G. Zeng, Y. Chou, "A Novel Efficient Optimal Reversible Circuit Synthesis Algorithm", IEEE International Conference on Systems, Man and Cybernetics, Hong Kong, 2015. pp. 68-73.