An Efficient Algorithm to Synthesize Quantum Circuits and Optimization

Ömercan Susam and Mustafa Altun

ECE Department, Istanbul Technical University, Istanbul, Turkey

susam@itu.edu.tr, altunmus@itu.edu.tr

Abstract— **Quantum computers, more specifically quantum circuits, take on the eyes with their computational promises such as reversibility. In this paper, we perform synthesis and optimization of quantum circuits. In the first part, we propose a fast synthesis algorithm that implements any given reversible Boolean function with quantum gates. Instead of an exhaustive search on every given function, our algorithm creates a library of essential functions and performs sorting. As an example, to implement 4 bit circuits we only use 120 essential functions out of all 20922789888000 functions. In the second part, we optimize our circuits by using new templates. The proposed templates mainly consist of Toffoli gates with negative and positive controlling lines. These templates also show us that optimum area solutions proposed in the literature are not actually optimum; they can be improved.**

Keywords—quantum computing; circuit synthesis; reversible circuits; optimization

I. INTRODUCTION

The idea of using quantum computers is first proposed by Richard Feynman at 1982 [1]. Exploiting quantum mechanical phenomena and its features is the main power of this idea which is leaning on the concept that accepts information as a physical item. Theoretical advantages of quantum computing over classical computing are proved by researchers and scientist for certain cases [2]. In addition, quantum mechanical unitarity brings reversibility for quantum algorithms based on quantum circuits. This is creating ideal application area for reversible computing which is influentially motivating for scientist and researchers for years to achieve energy efficient computation, due to Landauer's principle [3].

Although experimental applications are still in crawling stage [4], studies show that quantum computation is being applicable. At this point, reversible quantum circuit design is coming forward that is the core of this computation. Studies aim to synthesize optimal circuits with using minimum number of gates [5][6][7]. Optimized circuits will boost the security [8] on one hand and decrease the run time on the other hand. Nevertheless, optimal circuit synthesis can be achieved only up to four bits in the literature [9]. When we consider recent experiments [10] where 84 qubits are used, optimal circuit synthesis can take very long time that is not practical at all. This is the main motivation why we aim at a fast synthesis algorithm in this study.

In our synthesis process, we face a decision of whether or not to use garbage outputs. Few academic works use garbage outputs, especially those synthesizing functions with high number of bits [11]. Garbage outputs are additional bits to use when they are needed. In most cases they are not considered favorable because of the problems in energy efficiency [12]. In this study, we do not use garbage outputs. In the literature it is well known that all reversible functions can be implemented without a need of a garbage bit by using the CNT (**C**NOT, **N**ot, **T**offoli) gate library. This library includes gates with only positive control lines. In this study, we improve the CNT library by taking negative control lines into account which provides important circuit cost reduction (presented in experimental results). We call this new library as PN-CNT that stands for "**P**ositively and **N**egatively Controlled CNOT, **N**ot, **T**offoli". Fig. 1 illustrates CNT and PN-CNT libraries.

Our synthesis method efficiently implements any desired reversible Boolean function with quantum gates. Our method has two steps. In the first step, we use permutation based algorithm to find "essential functions" with optimum gate usage for selected bit size. Instead of using optimal circuits for essential functions, one could use circuits that are not necessarily optimal that would result in a faster algorithm, but also a larger circuit. Circuit synthesis is followed by a sorting process to obtain desired functions. This is the key part of our algorithm, where we gain our synthesis speed. Although optimum circuit synthesis is a time consuming process, because of the essential functions' fewness and sorting algorithms' speed, total synthesis time is still very short compared to the studies in the literature. In the second step, we perform optimization by constructing our templates. Templates are reversible circuits realizing the same function with different gate combinations that results in different circuit area costs. We construct our templates in two ways that are by directly using the gates from our PN-CNT library and by considering the inner quantum structure of these gates. The proposed templates not only optimize our synthesized circuits but also show us that optimum area solutions proposed in the literature are not actually optimum; they can be improved.

The paper is organized as follows. In Section II, we introduce background information for reversible circuits and reversible gates. In Section III, we present our synthesis algorithm based on essential functions and sorting. In Section IV, we present our optimization method based on template matching mechanism. In Section V and VI, we present experimental results and conclusions, respectively.
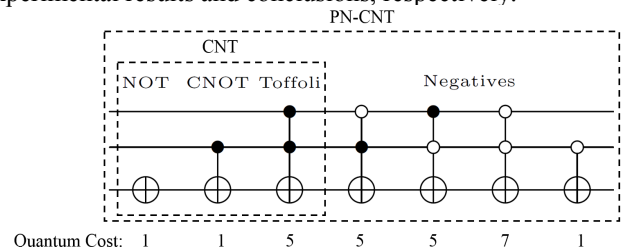

Fig. 1. NOT, CNOT, Toffoli and Toffoli with negative control lines.

II. Preliminaries

While the output of a conventional Boolean function always carries a one bit information (0 or 1) that is independent of the number of input bits, the number of input and output bits should be same for a reversible Boolean function. For reversible functions, each input bit combination results in a unique output bit combination; the reverse of this is also true because of the reversibility. In mathematics, reversible functions can be considered as bijection functions with input and output sets having the same number of elements.

By contrast with digital circuits, feedback and fan-out are not allowed in reversible circuits [7]. Reversible gates are used to accomplish given functions by changing their logical values. CNT (CNOT, Not, Toffoli) is the well-known and mostly accepted gate library for this purpose. Toffoli gates are formed by control lines and a target bit. When all of the control lines are designating to 1, target bit shifts to its inverse value. C-Not and Not gates can also be thinkable as Toffoli gates with one control bit and without control bits, respectively. The illustration of the gates are shown in Fig. 1.

Toffoli gates with negative control lines are also used in studies. These gates are based on the same principle unless, they accept 0 instead of 1 for its negative control line (s) to invert the target value. Number of sub-gates called as quantum operations, used to form a gate, gives the quantum cost of the gate. Quantum costs of negatively and positively controlled gates are given in Fig.1 [13].

III. Synthesis

In this part, we propose a fast synthesis algorithm that implements any given reversible Boolean function with quantum gates. Instead of an exhaustive search on every given function, our algorithm creates a library of **essential functions** and performs **sorting**.

A. Essential Functions

For a certain bit size n, a reversible Boolean function has a truth table with 2^n rows that results in $2^n!$ possible functions. Instead of implementing each of these functions separately, we consider very small amount of the total functions called as essential functions. To implement an essential function, we first consider a truth table such that input and output bit values are always identical. Then we switch any 2 rows (elements) without changing others. The result is an essential function. Examples of such essential functions can be seen in Table II. The total number of essential functions for a specific bit size n is $C(2^n,2) = 2^{n-1} * (2^n-1)$. Table I visualizes this formula. The ratio of the number of total and essential functions increases exponentially with the bit size. This means that the more bits we have, the more beneficial our approach is.

To synthesize essential functions, we develop an algorithm that results in optimal circuit sizes in terms of the quantum gate costs. Our algorithm is based on permutation trials. Since essential functions are mainly identity functions with an exception of two switched rows, they are considerably easy to be synthesized. This is the main reason why we use an optimal synthesis algorithm among non-optimal algorithms based on exact methods, constructive approaches, decision diagrams and exclusive sum of products [14]. The run time of our algorithm is slightly worse than that of a non-optimal algorithm, but it results in optimal circuits.

TABLE I. ESSENTIAL FUNCTION NUMBER ACCORDING TO BIT SIZE

Bit Size	Functions	
	# of Essential Functions	# of Total Functions
2	6	24
3	28	40320
4	120	20922789888000
5	469	2.613308e + 35
6	2016	1.268869e + 89

B. Sorting

After achieving essential functions we need an efficient sorting algorithm to implement any given function. Table II shows an example of this; f_1 and f_2 are essential functions used to obtain F. Bold lines are indicating the swapped rows. Fig. 2 shows the circuit realization of F.

Sorting algorithms are well studied in the literature. Among many different choices we use a selection sort algorithm. The reason is that it checks the equivalence of the input and output bits row by row. If input and output bits are equivalent the algorithm goes to the next row; if they are not equivalent the algorithm makes them equivalent using essential functions. This process uses essential functions effectively. However, operations like sliding or dividing used in other sorting algorithms necessitates relatively larger amount of essential function usage. For example, merge sort divides the set into subsets, and sorts these subsets first. After subset sorting is completed, equal sized subsets join together as pairs and these new subsets are sorted again that makes extra usage of essential functions. Also sliding process used in insertion sort method has the same handicap, each slide requires an extra essential function.

Selection sort can be improved when correct sequence is applied to the function. Table III shows an example for this situation; bolds represent mismatched lines. Four rows are not in their correct line for a given function. This creates 12 possible ways and two of them are presented in this table. Table IIIa shows a selection algorithm that starts at the first row and progresses to the bottom one by one. Table IIIb shows the optimal sorting algorithm for this function that is obtained after a trial of all possible sequences.

If we call mismatching line number as m for a function, all possible sequences can be calculated with $m!/2$. For the higher mismatched line numbers, total possible ways increases. In this paper we use two sequence methods, top-to-bottom, shown in Table IIIa, and bottom-up. Optimal sequence synthesis method is in our future work interests.

TABLE II. REALIZATION OF A FUNCTION WITH ESSENTIAL FUNCTIONS

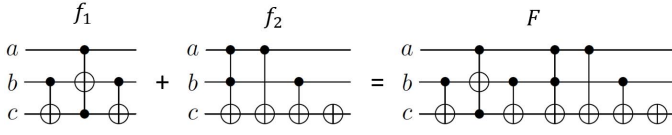| f₁			f₂			F	
In	**Out**		**In**	**Out**		**In**	**Out**
cba	cba		cba	cba		cba	cba
000	000		000	**100**		000	**100**
001	001		001	001		001	001
010	010	+	010	010	=	010	010
011	**101**		011	011		011	**101**
100	100		100	**000**		100	**000**
101	**011**		101	101		101	**011**
110	110		110	110		110	110
111	111		111	111		111	111

Fig. 2. Essential functions f_1 and f_2 used to obtain F function.

TABLE III.   TWO DIFFERENT SEQUENCE OF SELECTION ALGORITHM WITH DIFFERENT COST SIZE

(a)

f				
000	0	0	0	0
111	**7**	1	1	1
001	**1**	7	2	2
010	**2**	2	7	3
100	4	4	4	4
101	5	5	5	5
110	6	6	6	6
011	**3**	3	3	7
e.f.[a]	1-7	2-7	3-7	✓
Cost	4	4	1	**9**

(b)

f				
000	0	0	0	0
111	**7**	3	1	1
001	**1**	1	3	2
010	**2**	2	2	3
100	4	4	4	4
101	5	5	5	5
110	6	6	6	6
011	**3**	7	7	7
e.f.	7-3	1-3	2-3	✓
Cost	1	2	2	**5**

a. Essential functions

IV. OPTIMIZATION

We optimize our synthesized circuits by exploiting templates. We construct our templates in two ways that are by directly using the gates from our PN-CNT library and by considering the inner quantum structure of these gates.

A. Templates Using PN-CNT Library

We look for pre-defined templates in our sorting based synthesis and replace them with their optimal equivalent to reduce total quantum cost. Because of the sorting sequence, essential functions used one after the other that sometimes forms identical neighbor gates. This results at least two or more gate reduction in the final circuit, shown in Fig. 3. Additionally we create 12 more templates for negative gates. One of them shown in Fig. 4.
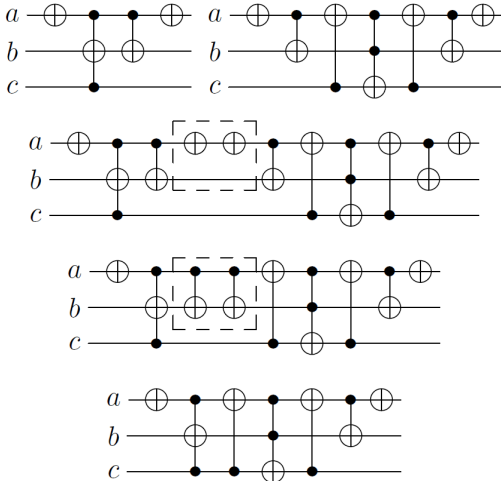

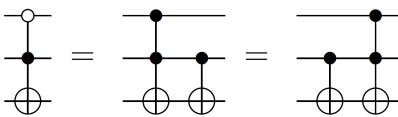Fig. 3. Sorting sequence with identical neighbor gates; 4 gates removed from the circuit.

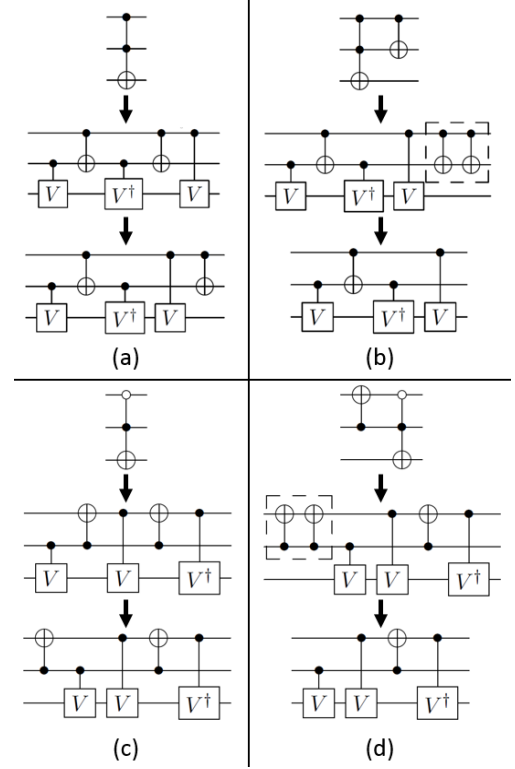Fig. 4. Templates for gates with a negative control line.

Fig. 5. (a) Toffoli gate with its inner quantum realization. (b) Template for positively controlled Toffoli. (c) Single negatively controlled Toffoli gate its quantum realization. (d) Template with single negative controlled Toffoli.

B. Templates Using Quantum Structure of Gates

After optimizing our circuits using PN-CNT library based templates, here we perform a second optimization process using quantum inner structure of the gates. In this part, Toffoli gates are expanded to their quantum circuit structure (Fig. 5a, Fig. 5c). Rescan process begins to find new identical neighbors. If a Toffoli gate has an appropriate CNOT gate neighbor (Fig. 5b, Fig. 5d), second reduction is applied. There are 12 different situations for Toffoli-CNOT case in 3 bit reversible circuits.

V. EXPERIMENTAL RESULTS

Implementations are realized in C. All experiments run on a 3.20-GHz Intel Core i5 CPU (only single core used) with 4.00 GB memory.

Table IV represents results of our proposed approach and optimal methods in the literature. Run times are in seconds. Reversible costs are calculated by counting each gate cost as one. Quantum costs are calculated by considering previous studies [17] presented in Fig. 1. Numbers in "size" column represents the number of gates used. Numbers in other columns represents the number of functions implemented with the corresponding gate number in size column. For example, the first row tells that 28 gates are used to implement 4 different functions based on the SSS approach. "Proposed Approach" column shows the results of our algorithm. CNT library based synthesis results are shown under CNT column. While SSS approach synthesizes circuits by using only top-to-bottom sequence selection, DSS approach uses both top-to-bottom and bottom-to-top sequence selections and take the minimum cost value of them. Comparing these two approaches, DSS overwhelms SSS with a 5.13% improvement in reversible circuit cost.

TABLE IV. REVERSIBLE FUNCTIONS OBTAINED WITH SPECIFIC GATE NUMBER ACCORDING TO 3 BITS

Size	Proposed Approach			Optimal	
	CNT		PN-CNT	CNT	PN-CNT
	SSS[b]	DSS[b]			
28	4				
27	29	1			
26	90	11			
25	207	55			
24	436	149			
23	791	327			
22	1252	712			
21	1954	1362			
20	2523	1936	1		
19	3349	2938	27		
18	3772	3477	105		
17	4125	4140	381		
16	4211	4388	956		
15	3842	4202	1827		
14	3522	3974	2965		
13	2835	3313	4260		
12	2308	2771	4910		
11	1706	2169	5659		
10	1239	1529	5288		
9	843	1150	4479		
8	547	682	3793	577	
7	340	495	2420	10253	
6	194	248	1619	17049	3236
5	111	167	928	8921	20480
4	52	63	423	2780	13282
3	25	41	198	625	2925
2	9	10	68	102	369
1	3	9	12	12	27
0	1	1	1	1	1
R.C.[c] W.A.	15.97	15.15	10.58	5.86	4.57
Q.C.[c] W.A.	34.26	32.57	30.43	13.88	-
Time	8	13	14	40 [15]	? [16]

[b]. SSS: Single Selection Sequence (top-to-bottom), DSS: Double Selection Sequence (top-to-bottom + bottom-to-top)

[c]. Reversible Cost- Quantum Cost

Our PN-CNT approach uses the library of PN-CNT presented in Fig.1. Note that this approach gives the best result in both reversible and quantum costs. Column named Optimal, shows optimal results that are derived using synthesis techniques for reversible gates. We convert reversible gate based circuits to quantum form and apply our quantum optimization method illustrated in Fig. 5. Thus, we reduce the optimal CNT quantum cost by 5.61%, from 13.88 to 13.10.

Comparing our synthesis approach with the optimal ones, our run times are always better at the cost of the circuit size. This is an expected result for 3 bit circuits. Here, an important point is that our synthesis approach can effectively work in higher bits. However, the optimal synthesis method is not practically applicable even for 5 bit circuits. If the optimal method is used to synthesize 5 bit circuits, it will take 447×10^{18} years that is justified using the numbers in Table I.

VI. CONCLUSION

In this paper, we present a new method for synthesis and optimization of quantum circuits. We propose a fast synthesis algorithm that implements any given reversible Boolean function with quantum gates. Instead of an exhaustive search on every given function, our algorithm creates a library of essential functions and performs sorting. Our synthesis algorithm is considerably faster than the optimal ones presented in the literature, especially for large bit sizes.

We optimize our circuits by using new templates. The proposed templates not only optimize our synthesized circuits but also show us that optimum area solutions proposed in the literature are not actually optimum; they can be improved. We reduce the previously proposed optimal CNT synthesis cost by 5.61%.

We are currently working on sequence selection algorithms used in the proposed sorting approach to find optimal sequences. Additionally, low level circuit synthesis is one of our primary aims to reduce quantum operations in quantum computers. As a future work we will seek deeper optimization methods with using elementary gates and pulse signals.

REFERENCES

[1] R.P. Feynman, "Simulating physics with computers." International journal of theoretical physics, pp. 467-488, June 1982.

[2] P.W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer." SIAM journal on computing, pp. 1484-1509, 1997.

[3] R. Landauer, "Irreversibility and heat generation in the computing process." IBM journal of research and development, pp. 183-191, July 1961.

[4] E. Lucero, et al. "Computing prime factors with a Josephson phase qubit quantum processor." Nature Physics,pp. 719-723, October 2012.

[5] P. Gupta, A. Agrawal, N.K. Jha. "An algorithm for synthesis of reversible logic circuits." Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions, pp. 2317-2330, November 2006.

[6] Maslov, Dmitri, Gerhard W. Dueck, and D. Michael Miller. "Toffoli network synthesis with templates." Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions, pp. 807-817, November 2005.

[7] M.A. Nielsen, I.L. Chuang. Quantum computation and quantum information. Cambridge University press, 2010.

[8] P.W. Shor, "Scheme for reducing decoherence in quantum computer memory." Physical review A 52, October 1995.

[9] O. Golubitsky, D. Maslov, "A study of optimal 4-bit reversible toffoli circuits and their synthesis." Computers, IEEE Transactions, pp. 1341-1353, September 2012.

[10] Z. Bian, F. Chudak, W.G. Macready, L. Clark, F. Gaitan "Experimental determination of Ramsey numbers." Physical review letters 111, 25 September 2013.

[11] R. Wille, R. Drechsler. "BDD-based synthesis of reversible logic for large functions." Proceedings of the 46th Annual Design Automation Conference. ACM, July 2009.

[12] T. Toffoli, Reversible computing. Springer Berlin Heidelberg, 1980.

[13] D. Maslov, C. Young, D.M. Miller, G.W. Dueck, "Quantum circuit simplification using templates." Proceedings of Design, Automation and Test in Europe, IEEE, Proceedings IEEE, pp. 1208-1213, March 2005.

[14] Schönborn, Eleonora, et al. "Optimizing DD-based Synthesis of Reversible Circuits using Negative Control Lines," unpublished.

[15] V.V. Shende, A.K. Prasad, I.L. Markov, J.P. Hayes, "Synthesis of reversible logic circuits." Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions, pp. 710-722, June 2003.

[16] R. Wille, M. Soeken, N. Przigoda, R. Drechsler, "Exact synthesis of Toffoli gate circuits with negative control lines." Multiple-Valued Logic (ISMVL), 2012 42nd IEEE International Symposium on. IEEE, pp. 69-74, 2012.

[17] A. Barenco, et al. "Elementary gates for quantum computation." Physical Review A 52, November 1995.