Quantum Circuit Optimization by Hadamard Gate Reduction

Nabila Abdessaied[2], Mathias Soeken[1,2], and Rolf Drechsler[1,2]

[1] Institute of Computer Science, University of Bremen, Germany
[2] Cyber-Physical Systems, DFKI GmbH, Bremen, Germany
{nabila,msoeken,drechsle}@informatik.uni-bremen.de

Abstract. Due to its fault-tolerant gates, the Clifford+T library consisting of Hadamard (denoted by H), T, and CNOT gates has attracted interest in the synthesis of quantum circuits. Since the implementation of T gates is expensive, recent research is aiming at minimizing the use of such gates. It has been shown that T-*depth* optimizations can be implemented efficiently for circuits consisting only of T and CNOT gates and that H gates impede the optimization significantly.
In this paper, we investigate the role of H gates in reducing the T-*count* and T-*depth* for quantum circuits. To reduce the number of H gates, we propose several algorithms targeting different steps in the synthesis of reversible functions as quantum circuits.
Experiments show the effect of H gate reductions on the costs for T-*count* and T-*depth*. Our approach yields a significant improvement of up to 88% in the final T-*depth* compared to the best known T-*depth* optimization technique.

1 Introduction

Quantum computing has shown promising results, e.g., for solving certain problems that require exponential running time in classical computers. Quantum computers exploit quantum mechanical effects and their underlying model makes use of qubits. In contrast to Boolean logic, qubits do not only represent the classical 0 and 1 states but also a superposition of both leading to a theoretically enormous speed-up in computing. The Deutsch-Jozsa algorithm [1] as well as the Shor's factorization algorithm [2] from Shor are the famous examples.

As a result, the synthesis of quantum circuits has become an active research area and many theoretical implementations for this kind of circuits have been presented [3]. To that end, since quantum operations are reversible, as a first step a reversible circuit is synthesized for the desired Boolean function after which, the resulting circuit is mapped to a functionally equivalent quantum circuit. It is also possible to build the quantum circuit for the requested Boolean function directly without going through the reversible circuit synthesis stage [4]. For the synthesis of, or mapping to, quantum circuits, several universal quantum gate libraries were introduced. One of the most used libraries is the Clifford+T library which is particularly interesting due to its fault-tolerant implementation [5].

After designing the quantum circuit, optimization techniques are often applied in order to produce a cheaper equivalent circuit. These optimization methods for the resulting Clifford+T circuits mainly focus on reducing the number of T gates and hence the T-*depth* on the resulting circuit because fault-tolerant implementations of T gates are considerably more expensive than those of the Clifford gates [6]. Thus, a couple of optimization techniques [7, 8] targeting the T-*depth* minimization were introduced. The major obstacle facing the T-*depth* minimization techniques is the H gates since T gates cannot commute across such gates. For that reason, attempts for tackling this problem were either to reduce the T-*depth* for quantum circuits over the gate library $\{CNOT,\ T\}$ or to extend the same approach for the Clifford+T library circuits by optimizing the T-*depth* of subcircuits between the H gates boundaries [9]. To the best of our knowledge, no one has studied the effect of minimizing H gates for reducing the T-*depth* so far.

In this paper, we study the characteristics of H gates and show how they significantly restrain the movement of T gates and hence limit the ability to better parallelize T gates. We prove that reducing the not needed H gates leads to a more efficient minimization of the T-*depth*. To do so, we introduce a new methodology which aims to eliminate the H gates as a preprocessing step for improving the T-*depth* optimization results of quantum circuits.

The remainder of the paper is structured as follows: first the basics on reversible and quantum circuits are introduced in Sect. 2. The next section outlines the general idea. Section 4 gives a detailed description of the implementation of the presented approach, and experimental results are evaluated and interpreted in Sect. 5. The paper is concluded in Sect. 6.

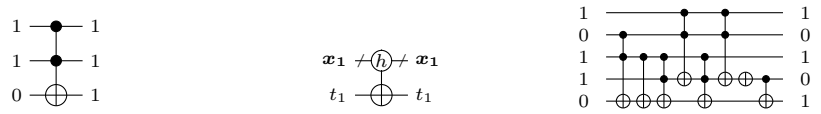
2 Background

To keep the remainder of this paper self-contained, this section briefly introduces the basics on reversible circuits, quantum circuits, and the corresponding mapping from reversible to quantum circuits.

2.1 Reversible Circuits

A Boolean function $f : \mathbb{B}^n \to \mathbb{B}^n$ is said to be *reversible* if it is bijective, i.e., if each input pattern is uniquely mapped to a corresponding output pattern, and vice versa. Reversible functions can be realized by reversible circuits that consist of at least n lines. Reversible circuits are cascades of reversible gates that belong to a gate library. One gate library that is often used consists of multiple control Toffoli gates [10].

Definition 1. *Given a set of variables* $X = \{x_1, \ldots, x_n\}$*, a* multiple control Toffoli *gate* $\mathrm{T}(C, t)$ *has* control lines $C = \{x_{j_1}, x_{j_2}, \ldots, x_{j_l}\} \subset X$ *and a* target line $t \in X \setminus C$*. The gate maps* $t \mapsto t \oplus h(x_{j_1}, x_{j_2}, \ldots, x_{j_l})$ *where* h *is defined as* $h : (x_{j_1}, x_{j_2}, \cdots, x_{j_l}) \mapsto (x_{j_1} \wedge x_{j_2} \wedge \cdots \wedge x_{j_l})$*. All remaining other lines are passed through unaltered.*



(a) Toffoli gate (b) Multiple control Toffoli gate (c) Reversible circuit

Fig. 1: Reversible circuitry

In [11] it has been shown that any reversible function $f : \mathbb{B}^n \to \mathbb{B}^n$ can be realized by a reversible circuit with n lines when using Toffoli gates.

Example 1. Figure 1(a) shows a Toffoli gate with two control lines. The control lines are either denoted by ● as depicted in Fig. 1(a) or represented by a Boolean function $h : \boldsymbol{x_1} = (x_{1_1}, x_{1_2}, \cdots, x_{1_l}) \mapsto (x_{1_1} \wedge x_{1_2} \wedge \cdots \wedge x_{1_l})$ as sketched in Fig. 1(b). The target line is denoted by $\bigoplus$. Figure 1(c) shows different Toffoli gates in a cascade forming a reversible circuit.

2.2 Quantum Circuits

Instead of bits, quantum circuits manipulate qubits which can represent the classical Boolean values but also a superposition of them. A *qubit* $|\varphi\rangle$ is a vector $\begin{pmatrix} a \\ b \end{pmatrix}$ where $a, b \in \mathbb{C}$ such that $|a|^2 + |b|^2 = 1$. If $a = 1$, then $|\varphi\rangle$ represents the classical 0, denoted $|0\rangle$, and if $b = 1$, then $|\varphi\rangle$ represents the classical 1, denoted $|1\rangle$.

In general, a quantum gate acting on n qubits represents a $2^n \times 2^n$ unitary matrix [12]. A matrix U is unitary if $U^\dagger U = UU^\dagger = I$ where $U^\dagger = (U^*)^T$ is the conjugate transpose of U. Using this gate definition many quantum mechanical effects such as superposition and entanglement can be formulated. Although Toffoli gates represent a unitary matrix, they are too general and thus not suitable for realizing quantum circuits [13]. In this paper, we make use of a gate library that is universal for quantum computation as well as its gates can be implemented in a fault-tolerant way.

Definition 2. *We consider the gate library* $\{H, Z, S, T, CNOT\}$ *with*

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},\ Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},\ S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix},\ T = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{pmatrix},\ \text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (1)$$

as the universal gate set. Note that the $S^\dagger$, $T^\dagger$, *and* NOT *gates can be implemented with* SSS, $SSST$, *and* HZH, *respectively. The* S *and* $S^\dagger$ *gates are square roots of the* Z *gate (given by the matrix in (1)). Similarly, the* T *and* $T^\dagger$ *gates are given by matrices that are the fourth root of the* Z *gate.*
A single qubit gate $G(t)$ *over the inputs* $X = \{x_1, \ldots, x_n\}$ *consists of a single target line* $t \in X$, *while a* CNOT *gate* $G(c, t)$ *comprises, in addition, a single control line* $c \in X$ *with* $t \neq c$.

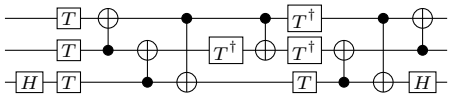


Fig. 2: Quantum circuit realizing a Toffoli gate

The above gate library is often referred to as the Clifford+T library. The so-called T-*depth* refers to the number of T-stages where each stage consists of one or more T or $T^\dagger$ gates that can be performed concurrently on separate qubits. The total number of incorporated T or $T^\dagger$ gates in the whole circuit is denoted by T-*count* while the total number of H gates is denoted by H-*count*. A root of Z gate denotes Z, S, T, $S^\dagger$, or $T^\dagger$.

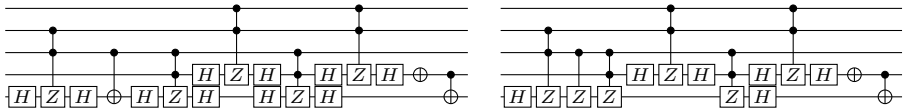
Example 2. Figure 2 shows a quantum circuit consisting of sixteen Clifford+T gates. This circuit represent one of the optimal realization of a Toffoli gate as depicted in [7, Fig. 13]. The circuit has a T-*count* of 7, a T-*depth* of 3, and an H-*count* of 2.

3 General Idea

In this work, we propose an optimization approach that aims for reducing the T-*depth* in a given quantum circuit as a main goal. In this section we motivate the impact of H gates on minimizing the T-*depth* in quantum circuits, afterwards we outline the proposed approach.

Mapping reversible circuits to quantum circuits can be done with different quantum library gates such as the *NCV* [12] or *NCV*-$|v_1\rangle$ [14] libraries. But recently the Clifford+T gate library has attracted most attention since it is composed of fault-tolerant logical gates [5]. Because it has been demonstrated that the fault-tolerant implementation of the T gates is surpassing the cost of the Clifford gates [6], many works have addressed the optimization of quantum circuits by minimizing the T-*count* [8] and the T-*depth* [9, 15].

The algorithm presented in [8] describes a method that performs an exhaustive search for a circuit that implements an n-qubit unitary matrix U using the minimal number of T gates. The work introduced in [7] addressed the optimization of T-*depth* for small circuits composed of four qubits at maximum. This is done by applying an exhaustive search algorithm to find the optimal T-*depth* realization. Another approach proposed a polynomial run-time algorithm for reducing the T-*depth* and the T-*count* of quantum circuits over the gate library $\{CNOT,\ T\}$ [9]. The algorithm deletes redundant T gates by computing the total phase and parallelizing the T gates through Matroid partitioning. The idea is based on decomposing a given function into minimal number of linear Boolean functions and then resynthesize each one with an optimal T-*depth* realization. This algorithm is extended to circuits built with the Clifford+T library. In this case, the same approach is applied for the subcircuits between the H gates, af-



(a) Quantum circuit (b) Symplified quantum circuit

Fig. 3: Equivalent quantum circuit for Fig. 1(c)

terwards an optimization process is applied which detects the identical gates and deletes them.

So far, as it is explained above, all the optimization techniques proposed algorithms for improving the *T-depth* for quantum circuits in the absence of, or locally between, H gates. But no work has introduced an approach for optimizing quantum circuits including H gates since they present the bottleneck for finding the optimal *T-depth*, i.e., they cannot interact with neighbouring gates and thus block the movement of any other gates across them. This restricts possible rearrangements of T gates and hence reduces the ability to perform T gates in parallel or apply possible reduction rules to a target circuit. Also when considering the algorithm [9] explained above and taking into account that the H gates are reduced before, this will allow to have larger subcircuits to resynthesize comparing to the first subcircuits: thus we get a bigger chance to get more parallel T gates and hence lower *T-depth*. Following the previous observations, therefore we present an approach that minimizes the *H-count* of quantum circuits, enabling better optimization results for the *T-depth* and the *T-count*.

Example 3. Figure 3(a) depicts the equivalent quantum circuit for the reversible circuit drawn in Fig. 1(c) according to [9]. Following their algorithm, the circuit is partitioned into a set of subcircuits located between the H gates. As it is shown in Fig. 3(a) we have 7 subcircuits. Next step, each subcircuit is resynthesized with an optimal *T-depth* quantum circuit. The application of the algorithm to the quantum circuit gives a circuit with a *T-depth* equal to 11 and a *T-count* equal to 27. However, one can reduce the *H-count* which yields an equivalent circuit depicted in Fig. 3(b). Applying the same algorithm to this circuit, which has 5 subcircuits, results in a circuit with a *T-depth* of 9 and a *T-count* of 23.

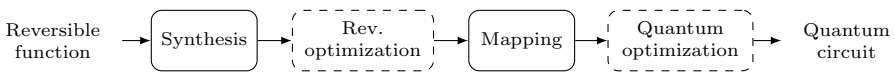


Fig. 4: Design flow for quantum circuits

4 Optimization Approaches

Motivated by the idea outlined in the previous section, we propose a design flow (depicted in Fig. 4) for the synthesis of a reversible function realized using gates from the Clifford+T library. First, the desired function is realized as a reversible circuit with Toffoli gates by applying existing synthesis methods such as [16–18]. To achieve better H gate reductions, we have taken the benefits of existing work aiming to optimize reversible circuits; for instance [19, 20], and applied the template matching technique introduced in [20] to the reversible circuit. Afterwards, we have incorporated an alternative mapping technique that yields circuits which are particularly suitable for H gate reductions. Finally, the obtained quantum circuit is optimized by applying an algorithm that aims at the T-*depth* optimization based on H gate minimizations. To resume, although we have employed the existing optimization algorithms at the reversible optimization level, our main contributions are on the mapping and quantum optimization steps.

4.1 Optimizations at the Reversible Level

In order to enhance the obtained circuit from a synthesis approach, a post synthesis process, also called reversible optimization stage, is applied. There are many existing methods targeting the optimization of reversible circuits for either reducing the number of lines [21], number of gates [20], depth [22], or quantum cost [23]. We are interested in techniques that lead to lower quantum cost. Smaller circuits are likely to have less H-*count* and therefore would have better T-*depth* but this is not always guaranteed.

Among the interesting work that focus on reducing the quantum cost for a given reversible circuit are template matching algorithms as described in [20], the window optimization introduced in [23], and finally the algorithm outlined in [24], that is similar to the template matching algorithm but includes better gate movement properties in the whole circuit. For this work, one can apply all of these approaches along with any other method leading to optimized quantum cost. For our experiments, we have included only the template matching approach [20].

4.2 Optimizations in the Mapping

After realizing and optimizing the reversible circuit for a given reversible function, each reversible gate is mapped to its equivalent quantum circuit as described in [13]. This mapping strategy is optimized with respect to quantum cost [25]. Afterwards, a second mapping technique that leads to an even lower quantum cost was described in [26]. Another functional mapping algorithm was presented in [27]: the described method searches for gates that have the same controls but different targets and decomposes them with a special decomposition.

According to Lemma 7.3 in [13], a reversible Toffoli gate with c controls (where $c \geq 3$) can be mapped to a network consisting of two identical gates with m controls and two other identical gates with $c - m + 1$ controls, where

$m \in \{2, \cdots, c-2\}$ and each of them are placed alternately. One has a lot of freedom on how to choose the controls and the order for each gate. As an example, Fig. 5(b) presents a possible mapping for the circuit depicted in Fig. 5(a) where the partitioning of controls is done with respect to their order in the original gates. However an alternative application of Lemma 7.3 [13] results in a circuit with two identical adjacent gates which can be removed as shown in Fig. 5(c). By removing these gates, at least two H gates are eliminated.

Hence our approach aims to apply a special mapping technique that is particularly suitable for circuits in which H gates cancel. This technique, instead of mapping reversible gates one by one and each on his own side, gathers gates as shown in the schemas in Fig. 6 and finds a suitable partitioning of the controls that leads to reversible gates that cancel and thus reduces the *H-count*.

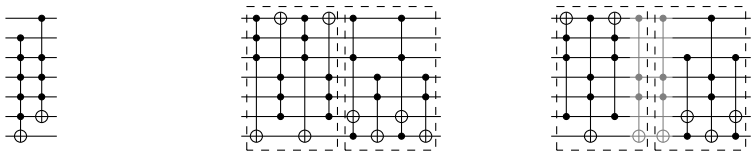
This mapping technique can be applied when a pair of reversible gates have one of the structures explained as below:

- Gates having a structure similar to the Peres gates as sketched in Fig. 6(a), i.e., a control line of the first is a target line of the other, besides they share one or more controls and the first gate has its target in a non shared line.
- Gates having their targets in the same line and sharing one or more control lines as depicted in Fig. 6(b).
- Gates having their target in non shared lines as described in Fig. 6(c). Also they have one or more control lines in common.
- Gates having a structure similar to the swap gates but also they share one or more control lines as outlined in Fig. 6.(d).

Example 4. Consider the case of the pair of gates depicted in Fig. 5(a). Using the new mapping scheme, we obtain the circuit drawn in Fig. 5(c) that contains 2 identical reversible gates with 3 controls each. The removal of these gates will lead to a reduction of 16 H gates compared to the classical mapping algorithm.

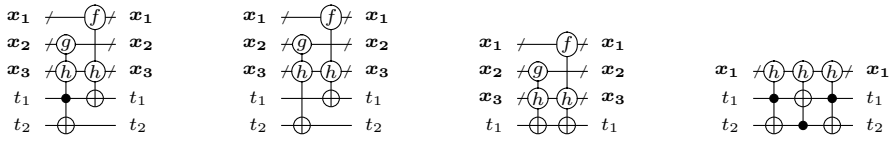
4.3 Optimizations at the Quantum Level

There are many optimization schemes that aim for quantum cost reduction for circuits based on the *NCV* library. In particular, the application of quantum template matching [28] or the merging and deletion rules together with functional



(a) Original circuit (b) Possible mapping (c) Better mapping

Fig. 5: Reversible circuit mapping



(a) First case (b) Second case (c) Third case (d) Forth case

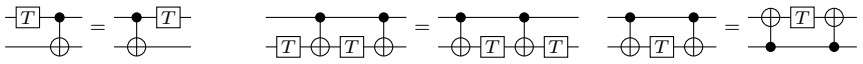
Fig. 6: Functional mapping

moving rules as explained in [24] are beneficial for decreasing the quantum cost. However, post mapping optimization techniques designed for quantum circuits based on Clifford+T gates are limited to the reduction of identical gates as described in [9] and the identities shown in [29].

The optimization approach that we introduce is based on a greedy algorithm that traverses repeatedly the circuit and looks for any possible cascade replacement with a cheaper equivalent cascade or any identity deletion. These two operations are known as merging and deleting rules. This scheme can additionally be improved by applying the moving rules for quantum circuits. In fact, in the Clifford+T library, additionally to the moving rules defined in [28], a CNOT gate $G(c, t_1)$ and a root of Z gate *(*Z*,* S*,* T*,* $S^\dagger$*,* $T^\dagger$*)* $G(t_2)$ can be interchanged if $t_1 \neq t_2$ as it is sketched with all other possible moving rules in Fig. 7. Furthermore, the following moving and deletion rules can be exploited for the Clifford+T circuits:

Hadamard gates reduction. The circuit is mapped in order to locate identical H gates or one of the cascades sketched in Fig. 8. Identical gates that could be moved together are deleted from the circuit and other identified templates are replaced by its equivalent cascade that do not contain any H gate.

Merging and deleting gates reduction. Taking the benefits of the moving properties for Clifford+T gates depicted on Fig. 7, additional reductions are possible for the remaining gates of the library. The algorithm searches for the templates shown in Fig. 10 and replaces these by their cheaper realization.



(a) First moving rule (b) Second moving rule (c) Third moving rule

Fig. 7: Moving rules for the Clifford+T gates

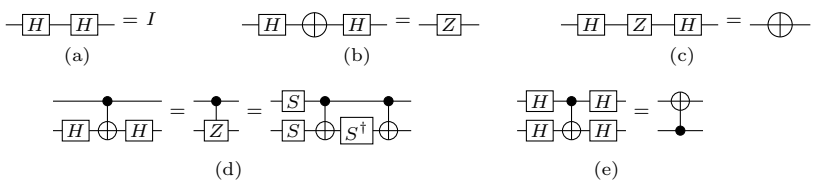


Fig. 8: Reduction rules for the H gates

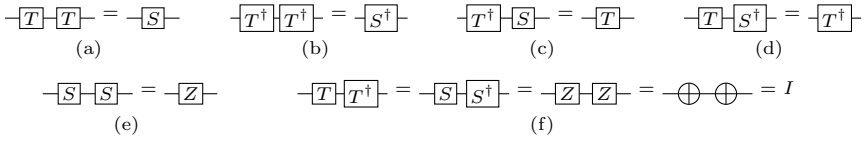


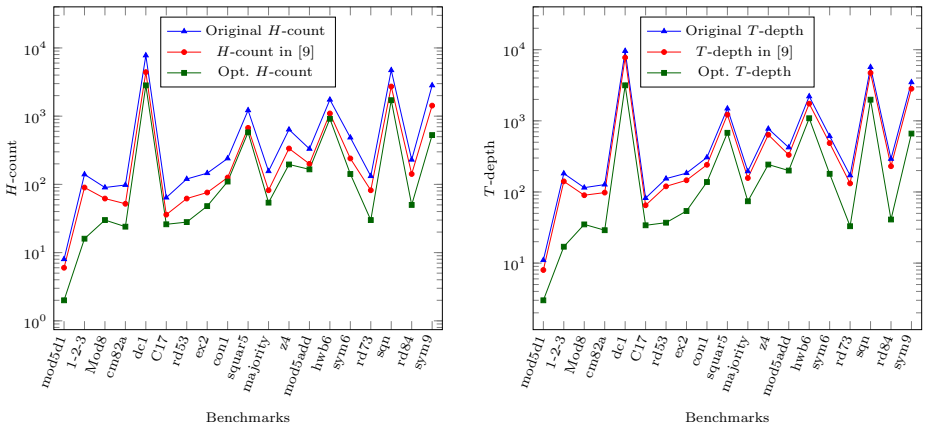
Fig. 9: Reduction rules for the remaining gates

5 Experimental Results

In this work, we proposed considerations of H gate minimizations to optimize quantum circuits build using the Clifford+T library. We have observed that eliminating H gates often leads to quantum circuits with a much smaller T-*depth*. Motivated by this, we introduced an improved design flow that aims at having lower H-*count* when generating the corresponding quantum gate cascades. The proposed idea described above has been implemented in the open source toolkit *RevKit* [30]. The experimental evaluation has been carried out on an Intel Core i5 Processor with 4 GB of main memory using the benchmarks taken from [31,32] database.

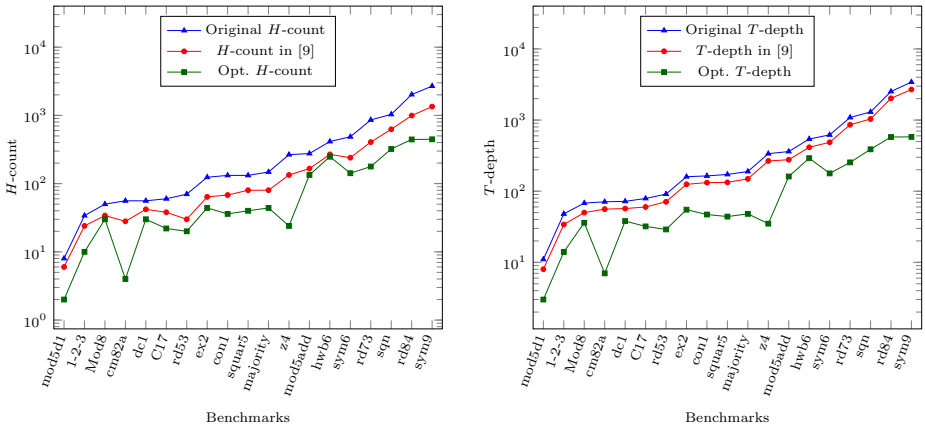
To determine the best synthesis approach with respect to T-*depth*, we have generated for each benchmark its corresponding circuits utilizing the following synthesis approaches: the transformation based synthesis approach (TBS [16]), the Reed-Muller synthesis approach (RMS [33]), the Young subgroups based synthesis approach (YSG [34]), and the ESOP based synthesis approach [17]. Due to space constraints, we have not detailed results for the ESOP based synthesis approach.

The experimental results are shown graphically in the plots in Fig. 10. The values of x-axis and the y-axis (logarithmic scale) denote the benchmark and the T-*depth*, respectively. Each plot contains three different scenarios: the T-*depth* of the original quantum circuits, the T-*depth* of the optimized circuits based on [9], and the T-*depth* of the optimized circuits based on our technique. One can clearly see that the T-*depth* and the H-*count* related to each other. Besides, most of the cases, the Reed-Muller synthesis approach [33] outperforms the other synthesis techniques in terms of producing lower T-*depth* circuits. The same observations are found for the H-*count* as it is shown in Fig. 10. In the rest of the paper we consider only the results of the Reed-Muller synthesis approach.



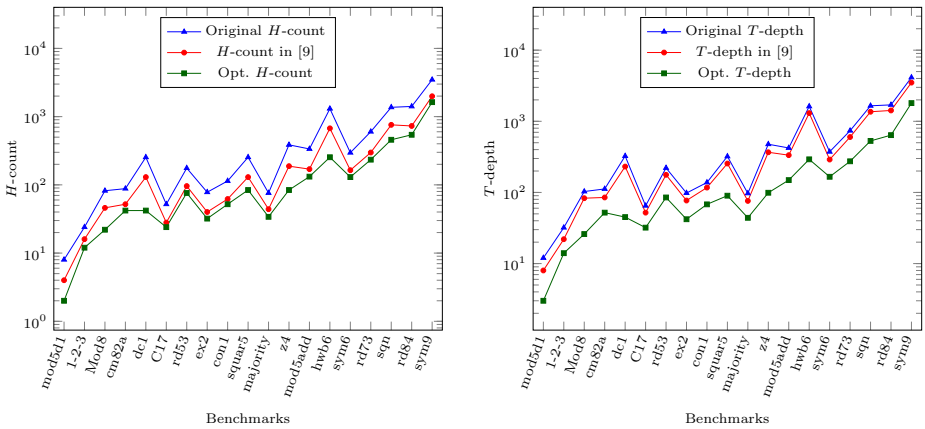
(a) *H-count* for TBS

(b) *T-depth* for TBS



(c) *H-count* for RMS

(d) *T-depth* for RMS



(e) *H-count* for YSG

(f) *T-depth* for YSG

Fig. 10: *H-count* and *T-depth* for original benchmark, optimized *T-depth* benchmarks, optimized *H-count* and *T-depth* benchmarks

Table 1 summarizes the obtained result. All benchmarks are listed in the first column. Then, the number of lines (L), the quantum costs (QC), the *H-count* (HC), the *T-count* (TC), and the *T-depth* (TD) of the respective circuit realizations as well as the needed run-times ($Time$) are provided.
The H gate reductions and the relative *T-depth* improvement of the circuits obtained by the proposed technique with respect to the optimized circuits based on the approach presented in [9] are provided in the columns denoted by ΔH and *TD Imp.*, respectively.

In total three different aspects are studied: (1) the results of circuits generated from the Reed-Muller synthesis approach, (2) the results of optimized circuits using the algorithm introduced in [9], and (3) the results of the optimized circuits using the technique reviewed in Sect. 4 in addition to the approach. in [9].

Applying the *T-depth* optimization approach described in [9] reduces the *T-depth* significantly. However, it is clearly observed that these results can be improved when applying the approach based on H gate reductions. Our proposed approach leads to additional *T-depth* reductions of 10% in average. The results confirm the impact of eliminating H gates on the *T-depth*.

As can be seen, in particular our scheme leads to significant *H-count* reductions. Over all circuits, reductions up to 900 H gates can be obtained, therefore, this enables further improvements of the overall *T-depth* as it is shown in Table 1. The *T-depth* is reduced by 60% on average and in the best case (*cm82*) by 88%.

H-count and *T-depth* are related to each other and more the H gates are reduced, more the *T-depth* is lower. This explains the variation of the *T-depth* improvement for each benchmark. For example, when a circuit contains many Toffoli gates that have their targets in the same line, then after quantum mapping the majority of H gates will be at the same line and many will cancel. Therefore the *T-depth* is reduced significantly (*Z4*, *Sym9*). Whereas, when Toffoli gates have their targets in different lines then the *H-count* cannot be much improved, hence the *T-depth* is not much decreased (*Mod8*).

6 Conclusion

In this paper we introduced a scheme for optimizing the *T-depth* of quantum circuits based on H gate reductions. To that end, we incorporated a 3-*level* strategy targeting the optimization of circuits at the reversible, mapping and quantum level to achieve better *H-count* reductions and hence possible further *T-depth* improvements. Experimental results have shown significant *T-depth* reductions which reach over 80% for quantum circuits.

Table 1: Experimental evaluation for Reed-Muller synthesized circuits where L, QC, HC, TC, TD, Time, ΔH, and TD Imp. refer to the number of lines, the quantum costs, the *H-count*, the *T-count*, the *T-depth*, the needed run-time, the H gate reductions, and the *T-depth* improvement with respect to the optimized circuits based on [9], respectively.

Benchmark		**Original circuit**				***T-depth* opt. in [9]**					***H-count*+*T-depth* opt.**					ΔH	**TD**
Id	L	QC	HC	TC	TD	QC	HC	TC	TD	Time	QC	HC	TC	TD	Time		**Imp.**
cm82a	8	456	56	196	71	462	28	142	56	0.004	108	4	23	7	0.000	-24	88%
z4	11	2140	266	931	337	2222	134	639	266	0.056	467	24	93	35	0.031	-110	87%
sym9	10	21510	2688	9408	3419	22423	1346	6364	2688	4.061	7087	446	1470	580	0.044	-900	78%
rd84	15	16188	2020	7070	2520	16803	994	4756	2021	5.537	7062	444	1402	578	0.011	-550	71%
rd73	10	6885	858	3003	1086	7091	406	2031	859	0.637	3027	178	607	254	0.071	-228	70%
majority	6	1188	148	518	189	1220	80	364	149	0.019	644	44	114	48	3.701	-36	68%
squar5	13	1082	132	462	172	1209	80	322	133	0.021	675	40	136	44	0.001	-40	67%
con1	9	1062	132	462	164	1066	68	330	132	0.015	685	36	126	47	0.124	-32	64%
sym6	7	3888	486	1701	619	3924	240	1143	487	0.187	2224	142	422	178	0.125	-98	63%
sqn	10	8411	1034	3619	1301	8734	626	2523	1035	1.067	4519	322	947	388	35.478	-304	63%
mod5d1	5	68	8	28	11	88	6	20	8	0.001	31	2	7	3	0.104	-4	63%
rd53	7	569	70	245	91	606	30	171	71	0.005	427	20	73	29	0.681	-10	59%
1-2-3	5	285	34	119	48	313	24	87	34	0.002	149	10	37	14	2.090	-14	59%
ex2	6	997	124	434	160	1027	64	300	125	0.015	665	44	130	55	0.076	-20	56%
C17	7	486	60	210	79	520	38	152	60	0.004	344	22	77	32	0.205	-16	47%
mod5add	6	2251	276	966	361	2317	166	670	277	0.055	1863	134	390	161	2.108	-32	42%
dc1	11	503	56	196	72	691	42	142	57	0.003	525	30	86	38	11.806	-12	33%
hwb6	6	3489	414	1449	541	3868	268	1035	414	0.135	3630	248	698	291	6.870	-20	30%
Mod8	5	406	50	175	68	413	34	131	50	0.003	408	30	91	36	0.032	-4	28%

References

1. Deutsch, D., Jozsa, R.: Rapid solution of problems by quantum computation. Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences **439**(1907) (1992) 553–558
2. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. Foundations of Computer Science (1994) 124–134
3. Shende, V.V., Bullock, S.S., Markov, I.L.: Synthesis of quantum-logic circuits. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on **25**(6) (2006) 1000–1010
4. Kliuchnikov, V., Maslov, D., Mosca, M.: Fast and efficient exact synthesis of single-qubit unitaries generated by Clifford and T gates. Quantum Information & Computation **13**(7-8) (2013) 607–630
5. Jones, N.C.: Logic synthesis for fault-tolerant quantum computers. arXiv preprint arXiv:1310.7290 (2013)
6. Fowler, A.G., Stephens, A.M., Groszkowski, P.: High-threshold universal quantum computation on the surface code. Physical Review A **80**(5) (2009) 052312
7. Amy, M., Maslov, D., Mosca, M., Roetteler, M.: A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. IEEE Trans. on CAD of Integrated Circuits and Systems **32**(6) (2013) 818–830
8. Gosset, D., Kliuchnikov, V., Mosca, M., Russo, V.: An algorithm for the T-count. arXiv preprint arXiv:1308.4134 (2013)
9. Amy, M., Maslov, D., Mosca, M.: Polynomial-time T-depth optimization of Clifford+T circuits via matroid partitioning. arXiv preprint arXiv:1303.2042 (2013)
10. Toffoli, T. In de Bakker, W., van Leeuwen, J., eds.: Automata, Languages and Programming. (1980)
11. Shende, V.V., Prasad, A.K., Markov, I.L., Hayes, J.P.: Synthesis of reversible logic circuits. **22**(6) (2003) 710–722
12. Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information. Cambridge Univ. Press (2000)
13. Barenco, A., Bennett, C.H., Cleve, R., DiVinchenzo, D., Margolus, N., Shor, P., Sleator, T., Smolin, J., Weinfurter, H.: Elementary gates for quantum computation. Physical Review A **52** (1995) 3457–3467
14. Sasanian, Z., Wille, R., Miller, D.M., Drechsler, R.: Realizing reversible circuits using a new class of quantum gates. In: Design Automation Conference. (2012) 36–41
15. Selinger, P.: Quantum circuits of T-depth one. Physical Review A **87**(4) (2013) 042302
16. Miller, D.M., Maslov, D., Dueck, G.W.: A transformation based algorithm for reversible logic synthesis. In: Design Automation Conference. (2003) 318–323
17. Fazel, K., Thornton, M., Rice, J.: ESOP-based Toffoli gate cascade generation. In: Communications, Computers and Signal Processing. PacRim 2007. IEEE Pacific Rim Conference on. (2007) 206–209
18. Soeken, M., Wille, R., Hilken, C., Przigoda, N., Drechsler, R.: Synthesis of reversible circuits with minimal lines for large functions. In: Asia and South Pacific Design Automation Conference. (2012) 85–92
19. Wille, R., Große, D., Dueck, G., Drechsler, R.: Reversible logic synthesis with output permutation. In: VLSI Design, 2009 22nd International Conference on. (2009) 189–194

20. Maslov, D., Dueck, G., Miller, D.: Simplification of Toffoli networks via templates. In: Integrated Circuits and Systems Design. Proceedings. 16th Symposium on. (2003) 53–58
21. Wille, R., Soeken, M., Drechsler, R.: Reducing the number of lines in reversible circuits. In: Design Automation Conference, IEEE (2010) 647–652
22. Abdessaied, N., Wille, R., Soeken, M., Drechsler, R.: Reducing the depth of quantum circuits using additional circuit lines. In: Reversible Computation. Springer (2013) 221–233
23. Soeken, M., Wille, R., Dueck, G., Drechsler, R.: Window optimization of reversible and quantum circuits. In: Design and Diagnostics of Electronic Circuits and Systems, 2010 IEEE 13th International Symposium on. (2010) 341–345
24. Sasanian, Z., Miller, D.M.: Reversible and quantum circuit optimization: A functional approach. In: Reversible Computation. Springer (2013) 112–124
25. Maslov, D., Dueck, G.: Improved quantum cost for n-bit Toffoli gates. Electronics Letters **39** (2003) 1790
26. Miller, D.M., Wille, R., Sasanian, Z.: Elementary quantum gate realizations for multiple-control Toffoli gates. In: Multiple-Valued Logic, 41nd IEEE International Symposium on. (2011) 217–222
27. Wille, R., Soeken, M., Otterstedt, C., Drechsler, R.: Improving the mapping of reversible circuits to quantum circuits using multiple target lines. In: Asia and South Pacific Design Automation Conference. (2013) 145–150
28. Maslov, D., Dueck, G., Miller, D., Negrevergne, C.: Quantum circuit simplification and level compaction. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on **27**(3) (2008) 436–444
29. Soeken, M., Miller, D.M., Drechsler, R.: Quantum circuits employing roots of the Pauli matrices. Physical Review A **88** (Oct 2013) 042322
30. Soeken, M., Frehse, S., Wille, R., Drechsler, R.: Revkit: A toolkit for reversible circuit design. Journal of Multiple-Valued Logic & Soft Computing **18**(1) (2012) RevKit is available at http://www.revkit.org.
31. Wille, R., Große, D., Teuber, L., Dueck, G.W., Drechsler, R.: RevLib: an online resource for reversible functions and reversible circuits. In: Multiple-Valued Logic, 38nd IEEE International Symposium on. (2008) 220–225 RevLib is available at http://www.revlib.org.
32. Maslov, D.: Reversible logic synthesis benchmarks page available at http://webhome.cs.uvic.ca dmaslov/, last accessed January 2011.
33. Maslov, D., Dueck, G.W., Miller, D.M.: Techniques for the synthesis of reversible Toffoli networks. ACM Transactions on Design Automation of Electronic Systems (TODAES) **12**(4) (2007) 42
34. De Vos, A., Van Rentergem, Y.: Young subgroups for reversible computers. Advances in Mathematics of Communications **2**(2) (2008) 183–200