

Technology Mapping for Single Target Gate Based Circuits Using Boolean Functional Decomposition

Nabila Abdessaied¹(✉), Mathias Soeken^{1,2}, and Rolf Drechsler^{1,2}

¹ Cyber-Physical Systems, DFKI GmbH, Bremen, Germany

² Institute of Computer Science, University of Bremen, Bremen, Germany
{nabila,msoeken,drechsle}@informatik.uni-bremen.de

Abstract. Quantum computing offers a promising emerging technology due to the potential theoretical capacity of solving many important problems with exponentially less complexity. Since most of the known quantum algorithms include Boolean components, the design of quantum computers is often conducted by a two-stage approach. In a first step, the Boolean component is realized in reversible logic and then mapped to quantum gates in a second step. This paper describes a new mapping flow for determining quantum gate realizations for single-target gates (ST). Since each ST gate contains a Boolean control function, our method attempts to find a decomposition based on its BDD representation. It consists on breaking large ST gate into smaller ones using additional lines. Experiments show that we obtain smaller realizations when comparing to standard mapping.

1 Introduction

Quantum computers are one of the most promising emerging technologies, generating interest from the corporate sector and attracting government investment. Quantum computers exploit the often counter-intuitive rules of quantum physics to perform computations in a substantially different and often much more efficient way than classical computers, enabling computational solutions to problems that are considered intractable for classical systems [1]. The design and fabrication of these machines has progressed rapidly in the past decade, with many research groups now routinely fabricating and operating small quantum computers in multiple physical systems.

Quantum computing does not only provide challenges for physicists but also offers a variety of challenging and interesting problems to the field of computer science. Large parts of quantum computers perform classical computations which can be described in terms of classical Boolean functions instead of arbitrary unitary operations as they are used for general quantum computing. However, since all quantum computers need to be reversible, also the classical computations need to be described in terms of reversible Boolean functions [2]. In order to create a quantum circuit from such a Boolean function, a first intermediate step

synthesizes a reversible circuit description. The most common gate library for this step consists of mixed-polarity multiple-controlled Toffoli gates. Toffoli gates offer a convenient representation to model the functionality of a reversible circuit but are still too abstract to be used as quantum operations. Many aspects, particularly those considering fault tolerance and error correction properties, cannot effectively be considered on that abstraction level. For the latter, quantum gate libraries are used that consist of a few quantum gates that typically act on at most 2 qubits: one of the currently prominent libraries is the Clifford+ T gate library [3]. Technology mapping is performed in order to map Toffoli gates to gates from the quantum gate library and the majority of methods that have been presented so far originate from [4].

Albeit providing a high-level representation for reversible circuits, the lower bound of the size of a reversible circuit consisting of Toffoli gates is exponential [5], i.e., for every number of variables there exists a reversible function for which the size of the minimal circuit is exponential. In order to avoid this complexity when addressing large reversible functions and circuits, recently single-target gates are considered as a representation for reversible circuits. They are a generalization of Toffoli gates and a linear upper bound for reversible circuits composed of these gates has been shown in [6]. Besides that, synthesis approaches presented in [6] and [7] are based on this gate representation. However, for technology mapping into quantum circuits, so far single-target gates are mapped into cascades of Toffoli gates which are then independently mapped using the techniques described in [4].

In this paper, we present a technology mapping approach that is directly based on single-target gates and makes use of Boolean decomposition and a constant number of ancillary lines. Working on the higher level abstraction allows significant cost reductions as shown by our experimental evaluations. In the best case, we were able to reduce the costs of the quantum circuit by 75% and in the average by about 20% for the Clifford+ T gate library.

2 Preliminaries

To keep the paper self-contained, this section reviews definitions and notations from Boolean functions, function decomposition, reversible circuits, and reversible synthesis.

2.1 Boolean Functions

Let $\mathbb{B} \stackrel{\text{def}}{=} \{0, 1\}$ denote the *Boolean values*. Then we refer to $\mathcal{B}_{n,m} \stackrel{\text{def}}{=} \{f \mid f: \mathbb{B}^n \rightarrow \mathbb{B}^m\}$ as the set of all *Boolean multiple-output functions* with n inputs and m outputs. There are 2^{m2^n} such Boolean functions. We write $\mathcal{B}_n \stackrel{\text{def}}{=} \mathcal{B}_{n,1}$ and assume that each $f \in \mathcal{B}_n$ is represented by a propositional formula over the variables x_1, \dots, x_n . Furthermore, we assume that each function $f \in \mathcal{B}_{n,m}$ is represented as a tuple $f = (f_1, \dots, f_m)$ where $f_i \in \mathcal{B}_n$ for each $i \in \{1, \dots, m\}$ and hence $f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$ for each $\mathbf{x} \in \mathbb{B}^n$. If we emphasize on

the domain of the function we write $f(X)$ where X refers to the set of input variables.

2.2 Exclusive Sum of Products

Exclusive sum-of-products (ESOPs, cf. [8]) are two-level descriptions for Boolean functions in which a function is composed of k *product terms* that are combined using the exclusive-OR (XOR, \oplus) operation. A product term is the conjunction of l_i *literals* where a literal is either a propositional variable $x^1 \stackrel{\text{def}}{=} x$ or its negation $x^0 \stackrel{\text{def}}{=} \bar{x}$. ESOPs are the most general form of two-level AND-XOR expressions:

$$f = \bigoplus_{i=1}^k x_{i_1}^{p_{i_1}} \wedge \dots \wedge x_{i_{l_i}}^{p_{i_{l_i}}} \tag{1}$$

Several restricted subclasses have been considered in the past, e.g., *positive polarity Reed-Muller expressions* (PPRM [8]) in which all literals are positive. There are further subclasses and most of them can be defined based on applying one of the following decomposition rules. An arbitrary Boolean function $f(x_1, x_2, \dots, x_n)$ can be expanded with respect to a variable x_i as

$$\begin{aligned} f &= \bar{x}_i f_{\bar{x}_i} \oplus x_i f_{x_i} && \text{(Shannon)} \\ f &= f_{\bar{x}_i} \oplus x_i (f_{\bar{x}_i} \oplus f_{x_i}) && \text{(positive Davio)} \\ f &= f_{x_i} \oplus \bar{x}_i (f_{\bar{x}_i} \oplus f_{x_i}) && \text{(negative Davio)} \end{aligned}$$

with *co-factors* $f_{\bar{x}_i} = f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$ and $f_{x_i} = f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$.

2.3 Boolean Function Decomposition

Boolean function decomposition describes the problem of finding, for a Boolean function, two or more simpler functions that being composed are functionally equivalent. Several types of Boolean function decomposition have been found in the last decades with the most important ones being:

1. Ashenhurst decomposition [9]: A function $f \in \mathcal{B}_n$ is decomposed into $f(X) = h(g(X_1), X_2)$ with $g \in \mathcal{B}_{|X_1|}$, $h \in \mathcal{B}_{|X_2|+1}$, and $X = X_1 \cup X_2$. If $X_1 \cap X_2 = \emptyset$, then the decomposition is called *disjoint*, otherwise it is called a *non-disjoint* decomposition. The set X_1 is called *bound set* and the set X_2 is called *free set*.
2. Curtis decomposition [10] is a generalization of the Ashenhurst decomposition with several inner functions of which each can have multiple outputs, i.e., $f(X) = h(g_1(X_1), g_2(X_2), \dots, g_k(X_k), X_{k+1})$ with $g_i \in \mathcal{B}_{|X_i|, m_i}$, $h \in \mathcal{B}_{|X_{k+1}|+m_1+\dots+m_k}$, and $X = X_1 \cup X_2 \cup \dots \cup X_{k+1}$.
3. Factorization [11]: The function is decomposed as $f(X) = g(X_1) \wedge h(X_2) \vee c(X_3)$, with $g \in \mathcal{B}_{|X_1|}$, $h \in \mathcal{B}_{|X_2|}$, $c \in \mathcal{B}_{|X_3|}$ and $X = X_1 \cup X_2 \cup X_3$.

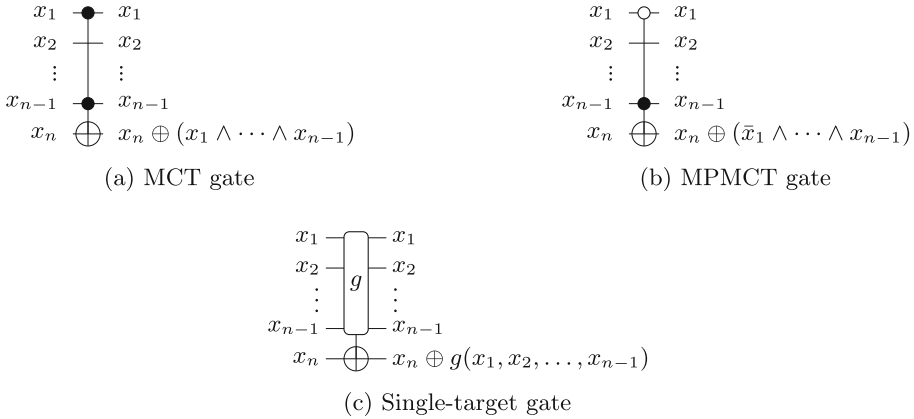


Fig. 1. Examples of reversible gates

4. Bi-decomposition [12] is also known as *simple decomposition*. The function is decomposed into two sub-functions $f(X) = g(X_1) \odot h(X_2)$, with $g \in \mathcal{B}_{|X_1|}$, $h \in \mathcal{B}_{|X_2|}$ and $X = X_1 \cup X_2$. The ‘ \odot ’ is any binary Boolean operation (typically \vee, \wedge, \oplus , or \leftrightarrow).

When X_1, X_2 , and X_3 are disjoint, the decomposition is called *algebraic*, otherwise *Boolean* or *functional*. Functional decomposition is much more powerful because the majority of Boolean functions are likely to have a functional decomposition rather than an algebraic one. Much work has been presented on decomposition algorithms based on truth tables [13] or binary decision diagrams (BDDs) [14–16].

2.4 Reversible Circuits

Reversible functions of n variables can be realized by reversible circuits that consist of at least n lines and are constructed as cascades of reversible gates that belong to a certain universal gate library. Although the Toffoli gate library is the most common gate library, single-target gates are of interest as they can lead to better circuits, e.g., lower quantum cost [7] and better circuit complexity [17].

Definition 1 (Single-Target Gate). *Given a set of variables $X = \{x_1, \dots, x_n\}$, a single-target gate (ST) $T_g(C, t)$ with control lines $C = \{x_{i_1}, \dots, x_{i_k}\} \subset X$, a target line $t \in X \setminus C$, and a control function $g \in \mathcal{B}_k$ inverts the variable on the target line if and only if $g(x_{i_1}, \dots, x_{i_k})$ evaluates to true. All other variables remain unchanged.*

Definition 2 (Toffoli Gate). *Mixed-polarity multiple control Toffoli (MPMCT) gates are a subset of the single-target gates in which the control function g can be represented with one product term or $g = \bigwedge_{k=i}^j x_i^p = 1$. Multiple-control*

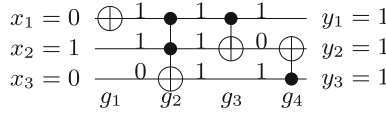


Fig. 2. Reversible circuit

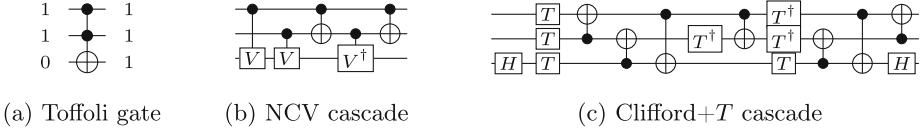


Fig. 3. Quantum mapping of a Toffoli gate

Toffoli gates (MCT) *in turn* are a subset from MPMCT gates in which the product terms can only consist of positive literals.

Following from synthesis algorithm implementations, it can easily be shown that any reversible function $f \in \mathcal{B}_{n,n}$ can be realized by a reversible circuit with n lines when using MCT gates. That is, it is not necessary to add any temporary lines (ancilla) to realize the circuit. This can be the case if the MCT (or MPMCT) gates are restricted to a given size, e.g., three bits. For drawing circuits, we follow the established conventions of using the symbol \oplus to denote the target line, solid black circles to indicate positive controls, and white circles to indicate negative controls.

Example 1. Figure 1a shows a Toffoli gate with positive controls, Figure 1b shows a Toffoli gate with mixed-polarity control lines, and Figure 1c shows the representation of a single-target gate based on Feynman’s notation. Figure 2 shows different Toffoli gates in a cascade forming a reversible circuit. The annotated values demonstrate the computation of the gate for a given input assignment.

2.5 Cost Metrics

To compare quantum circuits, we define metrics which depend on the gate library. For the NCV gate library, the *quantum cost* of a circuit is used while for the Clifford+T gate library, the *T-depth* is used. The motivation for that cost measure originates from the fact that the T gate is significantly larger compared to the other gates in the circuit.

Definition 3 (NCV-Cost). *The NCV-cost is the total number of elementary gates used in a quantum circuit.*

Definition 4 (T-Depth). *The T-depth is the number of T-stages in a quantum circuit 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.*

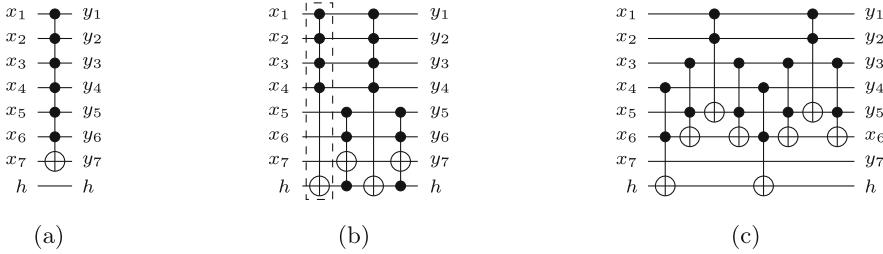


Fig. 4. Mapping from [4]

Example 2. The NCV-cost of the circuit shown in fig. 2 is equal to 8 since the total number of the elementary gates that realize a Toffoli gate is equal to 5. The circuit has a T -count of 7 and T -depth of 3.

2.6 Young Subgroup Synthesis

The young subgroup based synthesis approach makes use of the following property. Given a variable x , every reversible function $f \in \mathcal{B}_{n,n}$ can be decomposed into three functions $f = g_2 \circ f' \circ g_1$ such that $f' \in \mathcal{B}_{n,n}$ is a reversible function that does not change in x , and $g_1, g_2 \in \mathcal{B}_{n,n}$ are reversible functions that can be realized as single-target gates that act on x . By recursively applying the decomposition on the inner function f' , one obtains $2n$ single-target gates that realize f . After at most n recursive applications f' represents the identity function, i.e., it does not change in any variable anymore. Details of the proof can be found in [18]. A synthesis algorithm based on the idea has initially been proposed in [6] that takes as input a reversible function represented by its truth table. The synthesis algorithm has been extended to work symbolically using binary decision diagrams in [7], which allows for handling larger functions.

3 Motivation

As mentioned above, a quantum circuits are described in terms of a reversible Boolean function. In order to derive a quantum circuit for the reversible function, a two step approach is usually applied: first a circuit description in terms of reversible gates is derived, which in the second step is mapped to a quantum circuit composed of gates from a given library. In these steps, reversible gates are very general; e.g., often MCT gates are used for which the number of controls is not restricted. More recently, also the use of MPMCT gates became common practice. Quantum gate libraries are much smaller and usually consist of a few gates which can act on at most 2 qubits. Two prominent quantum gate libraries are the NCV gate library and the Clifford+ T gate library. In particular, the latter library is of significant interest in the design of quantum computers due to its good properties in fault tolerant quantum computing. Minimal quantum

circuit realizations are known for the 2-controlled Toffoli gate and are shown in Figure 3.

For larger Toffoli gates a procedure from [4] is applied which, according to Lemma 7.3 in [4], maps a reversible Toffoli gate with $c \geq 3$ controls to a network consisting of two identical gates with m controls and two other identical gates with $c - m + 1$ controls, where $m \in \{2, \dots, c - 2\}$ and each of them are placed alternately. If no free line is available for the gate, a new helper line must be added to the circuit. Its value is restored and hence can be reused afterwards. Finally, each obtained gate is mapped according to Lemma 7.2 in [4]. As a result, all Toffoli gates have at most 2 control lines. At this point, the mapping given in Figure 3 can be applied.

Example 3. The procedure is illustrated in Figure 4 for a Toffoli gate with six control lines. The first circuit depicts the result after the application of Lemma 7.2 in [4], while the second network sketches the obtained circuit from the decomposition of the first gate in the dashed rectangle after applying Lemma 7.3 in [4].

So far, there is no mapping approach into quantum circuits that directly targets the single-target gates as it is done for the MPMCT gates. To map single-target gates, we aim to decompose them into MPMCT gates so that we can afterwards map each obtained MPMCT gate using the approach explained above.

The mapping of a single-target gate to an MPMCT cascade is so far done by computing the XOR decomposition of its controlling function, then each cube in the obtained expression is represented by an MPMCT gate.

Many other Boolean decompositions do exist and have shown good efficiency [14]. Motivated by this, we want to study the impact of applying different kinds of Boolean decompositions while mapping single-target gates to MPMCT gates, i.e., unlike the standard mapping, we will not restrict the decomposition to XOR but also to bi-decomposition, Ashenhurst, and Curtis decomposition.

4 Mapping of Single-Target Gates

This section describes how Boolean decomposition can be applied to map reversible circuits composed of single-target gates into quantum circuits. Only the Young subgroup synthesis, for both the truth table based variant [6] and the BDD-based variant [7], makes use of single-target gates, however, due to the complexity of reversible circuits based on Toffoli gates (see, e.g., [17]) single-target gates are a preferable choice especially for large circuits.

Figure 5 shows how the functional decomposition of a single-target gate's control function can be used to generate less complex circuits. In the following we assume that the control function of the single-target gate that should be mapped depends on the variables x_1, \dots, x_{n-1} . Figure 5(a) shows the mapping approach for a disjoint Ashenhurst-Curtis decomposition. The variables are partitioned into four sets of variables represented as bit-vectors \mathbf{x}_1 , \mathbf{x}_2 , \mathbf{x}_3 , and \mathbf{x}_4 . First, the inner functions g_1 , g_2 , and g_3 are computed and each of their results is

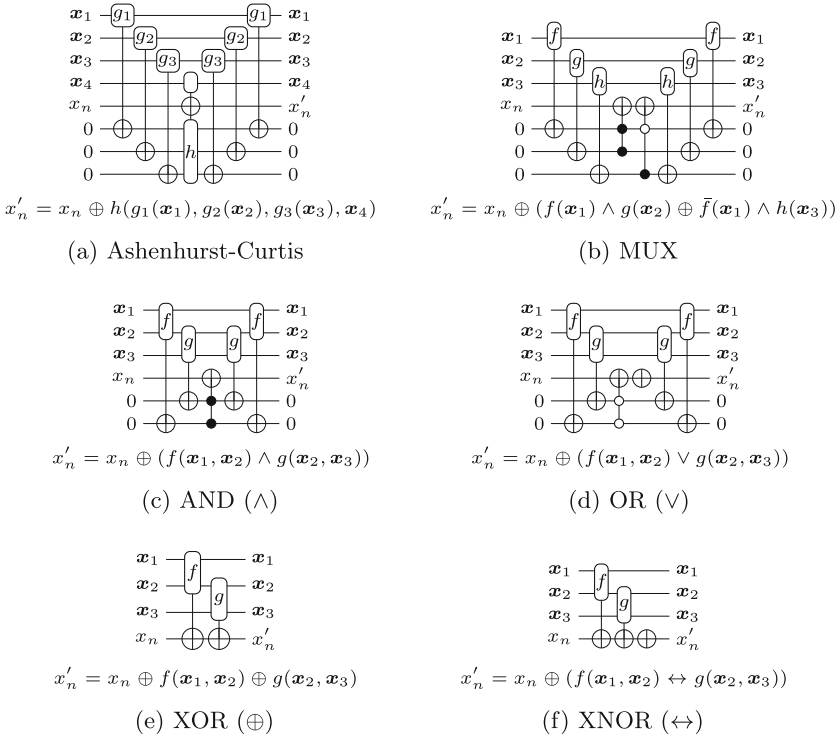


Fig. 5. Different types of decomposition

stored on an additional helper line that is initialized with a constant 0 value. Having the resulting values on these lines the outer function can be computed and afterwards the constant values on the helper lines are restored by reapplying the inner functions.

Figures 5(b) and (c) show non-disjoint bi-decompositions based on the AND and OR operation, respectively. The sub-function f depends on variables in \mathbf{x}_1 and \mathbf{x}_2 and the sub-function g depends on variables in \mathbf{x}_2 and \mathbf{x}_3 . As can be seen, the construction follows the representation of the Ashenhurst-Curtis decomposition in Figure 5(a). Whether a decomposition is disjoint or non-disjoint does not have an effect on the circuit construction but only on the size of the single-target gates in terms of their support. Also, a decomposition based on the MUX operation can analogously be performed by adding an extra helper line (see Figure 5(d)).

When using bi-decomposition based on the XOR and XNOR operator, one can update the target line directly as can be seen in Figures 5(e) and (f).

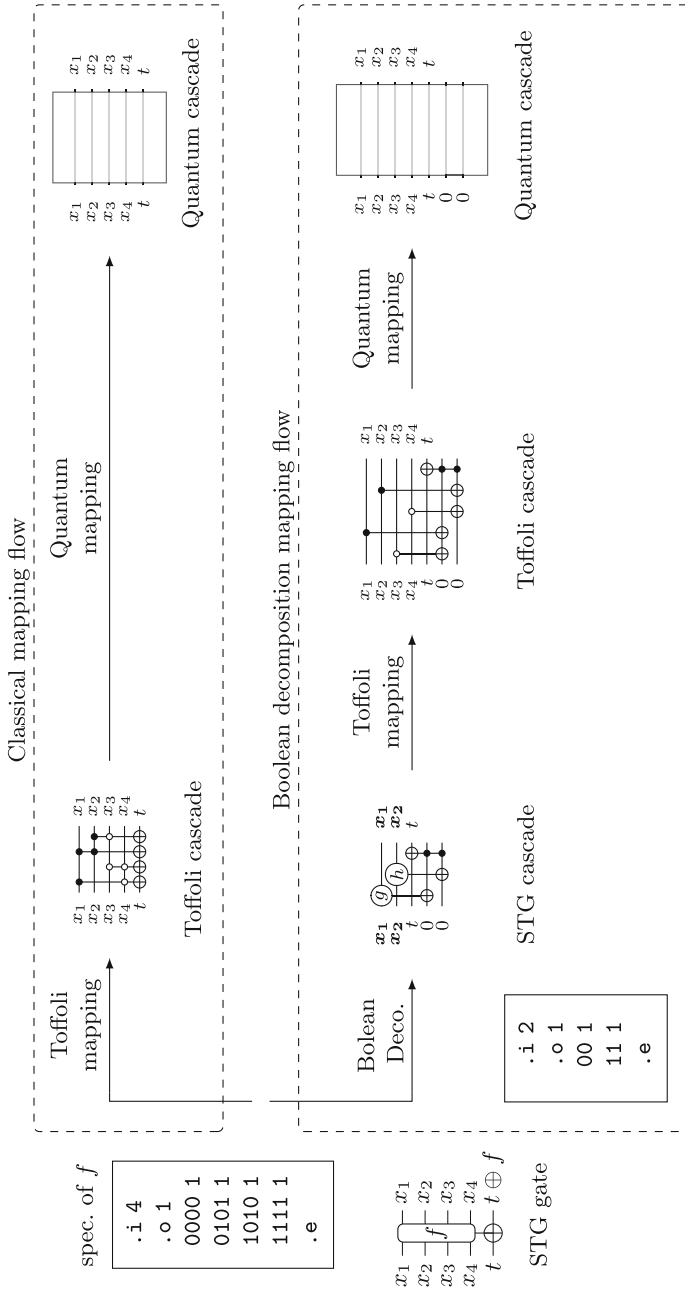


Fig. 6. Example for the mapping flow

The remainder of this section discusses an example application of the approach illustrated in Figure 6. The starting point is a single-target gate that is controlled by a control function

$$f(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 \vee \bar{x}_1x_2\bar{x}_3x_4 \vee x_1\bar{x}_2x_3\bar{x}_4 \vee x_1x_2x_3x_4$$

as also illustrated in its specification.

Decomposing the single-target gate using the standard mapping requires finding an ESOP representation of the function. Since f is given in terms of its minterms, it already resembles an ESOP representation. However, one can obtain a smaller one in terms of literals by applying ESOP minimization techniques finally resulting in the Toffoli gate cascade depicted in the upper box of Figure 6. The circuit consists of 4 Toffoli gates each having 2 controls. Mapping it into quantum circuits using the algorithm presented in [4] gives quantum costs of 21 for the NCV gate library (see Figure 3b and note that a Toffoli gate with two negative controls requires at most 6 NCV gates) and a T -depth of 12 when using the Clifford+ T gate library. Each Toffoli gate has a T -depth of 3 as it is depicted in Figure 3c.

Applying our proposed flow will first find a disjoint bi-decomposition

$$f(x_1, x_2, x_3, x_4) = g(x_1, x_3) \wedge h(x_2, x_4)$$

with $g(x_1, x_3) = x_1 \leftrightarrow x_3$ and $h(x_2, x_4) = x_2 \leftrightarrow x_4$. Next, each of the resulting single-target gates controlled by g and h are mapped to Toffoli cascades as it is shown in the reversible circuit, each ST gate has two Toffoli gates with only one control while the last gate computes the AND of both sub-functions using a Toffoli gate with two controls. Finally, the resulting reversible circuit is mapped to a quantum network with the same algorithm used in the standard flow. The number of NCV gates of the resulting circuit is 9 (compared to 21) and the T -depth is 3 (compared to 12).

5 Experimental Evaluation

In order to confirm the benefits of incorporating the Boolean decomposition technique into the mapping flow of reversible circuits to quantum circuits described in Section 4, we have implemented the proposed idea in the open source toolkit *RevKit* [19]. The starting point is reversible circuits obtained from applying the BDD-based version of the Young subgroup synthesis [7], which creates reversible circuits composed of single-target gates¹. We used the BDS-PGA tool [20] to decompose each control function of a single-target gate to smaller ones. We restricted the decomposition of each single-target gate to at most 3 smaller single-target gates to limit the use of additional lines to at most 3. To map the resulting smaller gates into cascades of Toffoli gates we used the XOR minimization algorithm implemented in EXORCISM [21]. Finally, we applied the quantum

¹ Benchmarks were taken from <http://webhome.cs.uvic.ca/~dmaslov/> and <http://www.revlib.org>

mapping algorithm explained in [4]. The experimental evaluation has been carried out on an Intel Core i5 processor with 4 GB of main memory.

Table 1 summarizes the obtained results. All benchmark names and original lines are listed in the first and second column, respectively. Then, the number of lines (L), the number of gates G, the NCV quantum costs (NCV), the T -depth (TD), the H -count (HC), and the required run-times (TIME) are provided for the synthesized circuits based on standard mapping and the synthesized circuits based on Boolean decomposition as explained in Section 4.

We provide absolute and relative improvement in the last two columns for quantum costs in terms of the NCV and the Clifford+ T gate libraries. The NCV quantum cost reductions and its relative improvement of the circuits obtained by the proposed technique with respect to the realized circuits without taking into account the Boolean decomposition are given in the columns denoted by Δ NCV and IMP.NCV, respectively. The procedure presented above yields circuits with lower NCV quantum cost comparing to circuits obtained by standard mapping. The table shows a percentage improvement in terms of NCV quantum cost by approx. 16%. In the best case improvements of up to 67% are observed (*cycle10_2_61*).

The T -depth cost reductions and its relative improvement are provided in the columns denoted by Δ TD and IMP.TD, respectively. Also for this gate library realizations with fewer T -depth are obtained when our technique is applied. On average, the size of the resulting quantum gate cascades was decreased by 20%. In the best cases, reductions of up to 47916 in the T -depth for the benchmark *bw_116* are obtained.

Remarks and Observations. When applying the BDS-PGA tool to find a decomposition for a Boolean function that controls a single-target gate, it first searches for an algebraic decomposition and only looks for a Boolean decomposition if the first attempt is not successful. This process is done recursively for each resulting sub-function until Boolean functions with at most 2 inputs are reached. We adapted the tool such that recursion stops after maximum three decompositions in order to keep a reasonable number of additional lines.

We refer to the *common set* as the intersection of bound set and free set in Ashenhurst-Curtis decompositions and as the intersection of supports in bi-decompositions. We have observed that for large common sets the results of the standard mapping approach outperforms our approach. To ensure good results, we only decomposed functions with a small common set.

We further noticed that bi-decompositions based on XOR and XNOR are less effective compared to the standard mapping since EXORCISM finds efficient ESOP representations. Consequently, we adapted the BDS-PGA tool such that it does not try to find bi-decompositions based on XOR or XNOR.

Finally, factorization with respect to a single variable can usually not improve the overall result since the incorporation of the variable on the helper line does not minimize the functional support. We have therefore turned off that option in the BDS-PGA tool.

To summarize, we looked for function decompositions with a small common-set and allowed bi-decompositions only for the OR, AND, and MUX operator.

6 Conclusions

In this work, we proposed a mapping approach that starts with single-target gates and therefore significantly differs from the standard mapping approach that has been state-of-the-art for the last two decades. We observed that incorporating Boolean decomposition in the mapping process of single-target gates often leads to better quantum realizations. Motivated by this, we introduced an improved mapping scheme which uses a constant number of ancillary lines and exploits the Boolean decomposition when generating the quantum gate cascades for a given single-target gate. Including our approach results in quantum circuits with a smaller NCV quantum cost as well as a lower T -depth cost comparing to standard mapping results.

References

1. Devitt, S.J.: Classical control of large-scale quantum computers. In: Yamashita, S., Minato, S. (eds.) RC 2014. LNCS, vol. 8507, pp. 26–39. Springer, Heidelberg (2014)
2. Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information. Cambridge Univ. Press (2000)
3. Fowler, A.G., Stephens, A.M., Groszkowski, P.: High-threshold universal quantum computation on the surface code. *Physical Review A* **80**, 052312 (2009)
4. Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D., Margolus, N., Shor, P., Sleator, T., Smolin, J., Weinfurter, H.: Elementary gates for quantum computation. *Physical Review A* **52**, 3457–3467 (1995)
5. Maslov, D., Dueck, G.W., Miller, D.M.: Toffoli network synthesis with templates. *IEEE Trans. on CAD of Integrated Circuits and Systems* **24**(6), 807–817 (2005)
6. De Vos, A., Van Rentergem, Y.: Young subgroups for reversible computers. *Advances in Mathematics of Communications* **2**(2), 183–200 (2008)
7. Soeken, M., Tague, L., Dueck, G.W., Drechsler, R.: Ancilla-free synthesis of large reversible functions using binary decision diagrams (2014). CoRR abs/1408.3955
8. Sasao, T.: AND-EXOR expressions and their optimization. In: Sasao, T., (ed.) *Logic Synthesis and Optimization*. Kluwer Academic Publisher, pp. 287–312 (1993)
9. Ashenurst, R.L.: The decomposition of switching functions. In: *Int'l Symp. on the Theory of Switching*, pp. 74–116 (1957)
10. Curtis, H.A.: *A new approach to the design of switching circuits*. van Nostrand Princeton, NJ (1962)
11. Brayton, R.K.: Factoring logic functions. *IBM Journal of Research and Development* **31**(2), 187–198 (1987)
12. Sasao, T., Matsuura, M.: DECOMPOS: an integrated system for functional decomposition. In: *Int'l Workshop on Logic Synthesis*, pp. 471–477 (1998)
13. Mishchenko, A., Brayton, R.K., Chatterjee, S.: Boolean factoring and decomposition of logic networks. In: *Int'l Conf. on Computer-Aided Design*, pp. 38–44 (2008)

14. Mishchenko, A., Steinbach, B., Perkowski, M.A.: An algorithm for bi-decomposition of logic functions. In: Design Automation Conference, pp. 103–108 (2001)
15. Bertacco, V., Damiani, M.: The disjunctive decomposition of logic functions. In: Int'l Conf. on Computer-Aided Design, pp. 78–82 (1997)
16. Yang, C., Ciesielski, M.J.: BDS: a BDD-based logic optimization system. *IEEE Trans. on CAD of Integrated Circuits and Systems* **21**(7), 866–876 (2002)
17. Abdessaied, N., Soeken, M., Thomsen, M.K., Drechsler, R.: Upper bounds for reversible circuits based on Young subgroups. *Information Processing Letters* **114**(6), 282–286 (2014)
18. Van Rentergem, Y., De Vos, A., Storme, L.: Implementing an arbitrary reversible logic gate. *Journal of Physics A: Mathematical and General* **38**(16), 3555–3577 (2005)
19. 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>
20. Vemuri, N., Kalla, P., Tessier, R.: BDD-based logic synthesis for LUT-based FPGAs. *ACM Trans. Design Autom. Electr. Syst.* **7**(4), 501–525 (2002)
21. Mishchenko, A., Perkowski, M.: Fast heuristic minimization of exclusive-sums-of-products. In: Int'l Workshop on Applications of the Reed-Muller Expansion in Circuit Design, pp. 242–250 (2001)