# Improving the Mapping of Reversible Circuits to Quantum Circuits Using Multiple Target Lines

Robert Wille[1] Mathias Soeken[1,2] Christian Otterstedt[1] Rolf Drechsler[1,2]
[1]Institute of Computer Science
University of Bremen
28359 Bremen, Germany
[2]Cyber Physical Systems
DFKI GmbH
28359 Bremen, Germany
e-mail: {rwille,msoeken,ostedt,drechsle}@informatik.uni-bremen.de

**Abstract — The efficient synthesis of quantum circuits is an active research area. Since many of the known quantum algorithms include a large Boolean component (e.g. the database in the Grover search algorithm), quantum circuits are commonly synthesized in a two-stage approach. First, the desired function is realized as a reversible circuit making use of existing synthesis methods for this domain. Afterwards, each reversible gate is mapped to a functionally equivalent quantum gate cascade.**

**In this paper, we propose an improved mapping of reversible circuits to quantum circuits which exploits a certain structure of many reversible circuits. In fact, it can be observed that reversible circuits are often composed of similar gates which only differ in the position of their target lines. We introduce an extension of reversible gates which allow multiple target lines in a single gate. This enables a significantly cheaper mapping to quantum circuits. Experiments show that considering multiple target lines leads to improvements of up to 85% in the resulting quantum cost.**

## I. Introduction

Using quantum mechanical phenomena such as superposition and entanglement, quantum computation [1] allows for breaching complexity bounds which are valid for computing devices based on conventional mechanics. The Grover search [2] and the factorization algorithm by Shor [3] rank among the most famous examples for quantum algorithms that solve problems in time complexities which cannot be achieved using conventional computing.

For example, the Grover algorithm addresses the search of an item in an unsorted database with $N$ elements in time $O(\sqrt{N})$, whereas conventional methods cannot be performed using less than linear time. Shor's algorithm performs prime factorization in polynomial time by exploiting quantum computation methods such as the quantum Fourier transform. Thus, the algorithm is exponentially faster than its best known conventional counterpart, the general number field sieve, which solves the problem in sub-exponential time [4].

Motivated by these theoretical concepts and their recent prototypical implementations (e.g. see [5] or, more recently, [6]), the design of such devices became an active research area. The quantum algorithms can be represented in terms of quantum circuits that often inhibit a pre-defined structure (e.g. quantum gates putting the respective quantum bits into superposition followed by the actual computation and, finally, the evaluation in terms of measurement). While most of these structures are already given by the respective quantum algorithms (e.g. the Grover iteration in the search algorithm or the quantum Fourier transform in the factorization algorithm), Boolean components often depend on the given input. For example, the respectively considered database or the provided product are not fix and need to be designed for each application.

In order to design these Boolean components, the property that all quantum circuits are inherently reversible is exploited. First, they are designed as a reversible circuit that works on conventional bits rather than on quantum bits. For this purpose, existing methods such as [7, 8, 9, 10, 11] are used. Afterwards, mapping techniques are applied that transform the reversible circuit into a functionally equivalent quantum circuit [12, 13, 14]. This is described in more detail later in Section II.C. While this is an established solution to design quantum circuits, mapping reversible circuits to quantum circuits has mainly been considered in a local manner so far, i.e. each reversible gate is solely been mapped to corresponding quantum gates.

In this paper, we propose an improved mapping of reversible circuits to quantum circuits which exploits a certain structure of many reversible circuits. In fact, it can be observed that reversible circuits are often composed of similar gates which only differ in the position of their target lines. Allowing instead multiple target lines in reversible gates enables much cheaper mappings and, thus, cheaper quantum circuit designs. We introduce corresponding mappings and show their application and benefits for both, existing reversible circuits and existing synthesis approaches. As confirmed by an experimental evaluation, improvements of up to 66% in the first case and up to 85% in the latter case can be observed.

The paper is structured as follows. The next section provides the necessary background, while the general idea is given in Section III. Section IV introduces the extension of reversible gates that consider more than one target line. Its application is outlined in Section V. Experimental results follow in Section VI before the paper concludes in Section VII.

## II. 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. For a more detailed treatment, we refer the reader to [1].

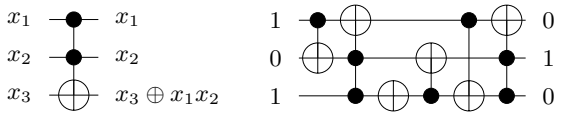
### A. Reversible Circuits

*Reversible circuits* are digital circuits with the same number of input signals and output signals. Furthermore, reversible circuits realize bijections, i.e. each input assignment maps to a unique output assignment. Accordingly, computations can not only be performed from the inputs to the outputs but also in the other direction.

Reversible circuits are composed as cascades of reversible gates. The *Toffoli gate* [15] is widely used in the literature and also considered in this paper.

**Definition 1** *Given a set of variables or signals* $X = \{x_1, \ldots, x_n\}$*, a* Toffoli gate $g(C, t)$ *is a tuple of a possibly empty set* $C \subset X$ *of* control lines *and a single* target line $t \in X \setminus C$*. The Toffoli gate inverts the value on the target line if all values on the control lines are set to* 1 *or if* $C = \emptyset$*. All remaining values are passed through unaltered.*

**Example 1** *Fig. 1(a) shows a Toffoli gate drawn in standard notation, i.e. control lines are denoted by* $\bullet$ *while the target line is denoted by* $\oplus$*. A circuit composed of several Toffoli gates is depicted in Fig. 1(b). This circuit maps e.g. the input* 101 *to the output* 010 *and vice versa.*




(a) Toffoli gate (b) Toffoli circuit

Fig. 1. Toffoli gate and Toffoli circuit

### B. Quantum Circuits

Quantum computation [1] is a promising application of reversible logic. The corresponding *quantum circuits* are very similar to reversible circuits but work on quantum bits (*qubits*) instead of bits. In contrast to Boolean logic, qubits do not only allow to represent the conventional Boolean states 0 and 1, but also the superposition of them. More precisely, a qubit is a linear combination of the conventional Boolean states in the two dimensional complex Hilbert space. The two orthonormal quantum states $|0\rangle = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$ and $|1\rangle = \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$ are used to represent the Boolean values 0 and 1. Thus, any state of a qubit may be written as $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $\alpha$ and $\beta$ are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$. All quantum bits reside on the Bloch sphere, a unit 2-sphere.

Each operation on these qubits can be defined by a unitary matrix [1] which is represented by means of quantum gates. However, in this work we use a definition for quantum gates that is closer to the definition of reversible gates.

**Definition 2** *A* quantum gate $\mathrm{q}(C, t)$ *over the inputs* $X = \{x_1, \ldots, x_n\}$ *consists of a single target line* $t \in X$ *and, in some cases, of a single control line* $c \in X$ *with* $t \neq c$*. That is,* $C$ *is either empty or equals* $\{c\}$*. When further assuming that the inputs to the circuit as well as the inputs to the control lines of the gates are restricted to the conventional Boolean values* 0 *and* 1[1]*, the following four gates define the commonly used quantum gate library.*

- NOT gate*: The qubit on the target line* $t$ *is inverted.*
- Controlled NOT gate *(CNOT): The target qubit* $t$ *is inverted if the control qubit* $c$ *is 1.*
- Controlled $V$ gate*: The* $V$*-operation is performed on the target qubit* $t$ *if the control qubit* $c$ *is 1. The* $V$*-operation is also known as the square root of NOT, since two consecutive* $V$*-operations are equivalent to an inversion.*
- Controlled $V^\dagger$ gate*: The* $V^\dagger$ *gate performs the inverse operation of the* $V$ *gate, i.e.* $V^\dagger = V^{-1}$*.*

Due to the assumption for the circuit inputs and control lines, the set of possible values $\{|0\rangle, |1\rangle, |v_0\rangle, |v_1\rangle\}$ is closed under the above mentioned operations. That is, we are dealing with 4-valued logic, where $|v_0\rangle = \frac{1+i}{2}\left[\begin{smallmatrix}1\\-i\end{smallmatrix}\right]$ and $|v_1\rangle = \frac{1+i}{2}\left[\begin{smallmatrix}-i\\1\end{smallmatrix}\right]$.

**Example 2** *Fig. 2 shows a quantum circuit composed of several of the gates introduced above realizing a reversible function. Control lines are again denoted by* $\bullet$ *while target lines are either denoted by* $\oplus$ *in case of NOT and CNOT or by* $\boxed{V}$ *and* $\boxed{V^\dagger}$ *in case of* $V$ *gates and* $V^\dagger$ *gates, respectively. This circuit maps e.g. the input* 101 *to the output* 111 *and vice versa, whereby the intermediate value* $|v_0\rangle$ *occurs.*

All elementary gates are assumed to have unit cost [12].

[1]Note that, as also shown in the next section, this restriction is common when considering the design of Boolean components of quantum circuits.

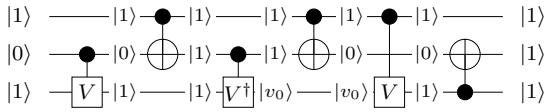
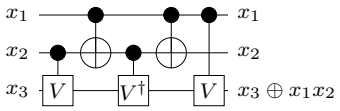
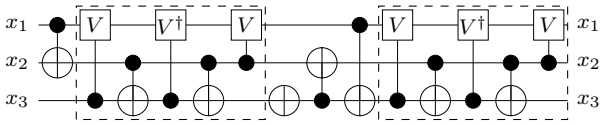



Fig. 2. Quantum circuit




(a) For the single Toffoli gate shown in Fig. 1(a)




(b) For the reversible circuit shown in Fig. 1(b)

Fig. 3. Mapping reversible circuits to quantum circuits

### C. Mapping Reversible Circuits to Quantum Circuits

Since any quantum operation can be represented by a unitary matrix [1], each quantum circuit is inherently reversible. Consequently, every reversible circuit can be transformed into a quantum circuit. To this end, each gate of the reversible circuit is *mapped* into a cascade of functionally equivalent quantum gates.

**Example 3** *Consider a Toffoli gate with two control lines as shown in Fig. 1(a). A functionally equivalent realization in terms of quantum gates is depicted in Fig. 3(a). This cascade can be applied to fully map the reversible circuit shown in Fig. 1(b) into an equivalent quantum circuit. For this purpose, all corresponding Toffoli gates are respectively substituted with a corresponding quantum gate cascade. The* 1st*,* 3rd*,* 4th*, and* 5th *gate remain unchanged as they already represent quantum gates. The resulting fully equivalent quantum circuit is shown in Fig. 3(b).*

Similar mappings exist for Toffoli gates with more than two control lines. But with increasing number of control lines, the resulting quantum circuits become more expensive, i.e. require more quantum gates. Furthermore, also the number of the *ancillary lines*, i.e. the number of circuit lines which neither are a control line nor a target line, affect the size of the resulting quantum circuit. To provide some examples, Table I(a) lists the respective costs for different Toffoli gate configurations according to the mapping scheme introduced in [12, 13] and often used in the literature. However, how to efficiently determine mappings from a single reversible gate to a cascade of quantum gates is an active research area which led to better results in the meantime. The costs obtained by the current state-of-the-art [14] are provided in Table I(b). Note that, in this work, we apply this metric since it represents the best results available so far. But besides that, the approach proposed in this work is also applicable to other mapping schemes (e.g. the one recently presented in [16]).

In the design of quantum circuits, Boolean components (e.g. the oracle transformation in the Deutsch-Jozsa algorithm, the database in Grover's search algorithm, and the modulo exponentiation in Shor's algorithm) play a significant role. Motivated by the mapping capability sketched above, an established design of such components for quantum circuits is conducted in two steps:

- First, the desired function is realized in terms of a reversible circuit making use of existing synthesis approaches (e.g. [7, 8, 9, 10, 11]) and,
- afterwards, the resulting circuit is mapped into a quantum circuit using techniques shown above.

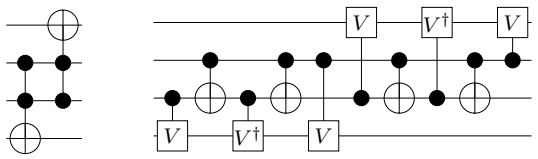
TABLE I
QUANTUM COSTS FOR DIFFERENT TOFFOLI GATE CONFIGURATIONS

(a) According to [12, 13]

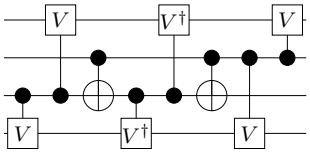
| Control lines | Ancillary lines 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 1 | | | |
| 1 | 1 | | | |
| 2 | 5 | | | |
| 3 | 14 | | | |
| 4 | | 29 | 26 | |
| 5 | | 61 | 52 | 38 |
| 6 | | 125 | 80 | 50 |
| 7 | | 253 | 100 | 62 |

(b) Current state-of-the-art [14]

| Control lines | Ancillary lines 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | | |
| 1 | 1 | | |
| 2 | 5 | | |
| 3 | | 14 | |
| 4 | | 20 | |
| 5 | | 32 | |
| 6 | | 44 | |
| 7 | | 64 | 56 |




(a) Circuit (b) Established mapping




(c) Improved mapping

Fig. 4. Exploiting multiple targets

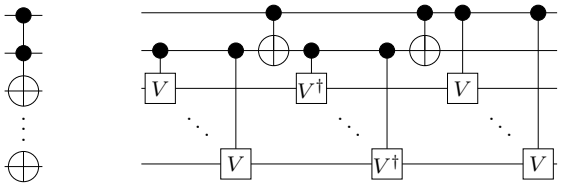
## III. General Idea

Mapping reversible circuits to quantum circuits has mainly been considered in a local manner, i.e. single Toffoli gates only have been mapped to corresponding quantum gate cascades as reviewed in Section II.C. Exceptions are e.g. the work in [17] where the mapping of pairs of Toffoli gates are considered. However, considering arbitrary sequences of Toffoli gates in order to determine more efficient mappings to quantum gate cascades is a non-trivial task and computationally expensive due to the high number of combinations.

On the other side, synthesis methods for reversible circuits often lead to realizations with a certain structure which can be exploited. In fact, it can be observed that many reversible circuits are composed of cascades of Toffoli gates which differ only in the position of their respective target lines, but have an equal set of control lines. This can be exploited as the following example illustrates.

**Example 4** *Consider the reversible circuit composed of two Toffoli gates as shown in Fig. 4(a). Mapping this circuit to a quantum circuit according to the procedure outlined in Section II.C leads to a cascade of ten quantum gates as shown in Fig. 4(b). However, due to the fact that both Toffoli gates have the same set of control lines, the resulting quantum gate cascade includes redundancies. In fact, gates checking for the values of the control lines are added twice. In contrast, Fig. 4(c) shows a more efficient realization which checks for the values of the control lines only once. This enables a realization with only eight gates.*

In this work, we present an approach which exploits those structures. We are introducing an extension of the commonly applied Toffoli gate which allows multiple target lines in a single gate. As the experiments in Section VI confirm, this leads to significantly cheaper quantum cascades and, hence, significantly reduces the cost of the resulting quantum circuits.




(a) Toffoli gate (b) Quantum gate cascade

Fig. 5. Mapping Toffoli gate with multiple targets

## IV. Reversible Gates with Multiple Target Lines

Motivated by the general idea outlined above, this section introduces Toffoli gates with multiple target lines and shows how such gates can be mapped to smaller quantum gate cascades. In order to exploit the outlined idea, the definition of Toffoli gates as provided in Section II is extended as follows:

**Definition 3** *A* Toffoli gate with multiple target lines $g(C, T)$ *over the inputs* $X = \{x_1, \ldots, x_n\}$ *consists of a (possibly empty) set of* control lines $C \subset X$ *and a non-empty set of* target lines $T \subseteq X \setminus C$. *The Toffoli gate inverts the value on all target lines if all values on the control lines are set to* 1 *or if* $C = \emptyset$. *All remaining values are passed through unaltered.*

Toffoli gates with multiple target lines enable an easier and more efficient mapping to quantum gate cascades. For this purpose, the existing mapping methods for Toffoli gates with single target lines can be exploited. More precisely, a Toffoli gate $g(C, T)$ with multiple target lines $T = \{t_1, t_2, \ldots, t_k\}$ can be mapped to a quantum gate cascade as follows:

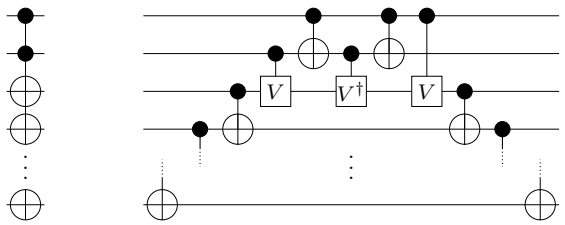
1. Consider a Toffoli gate $g(C, t)$ with a single target line $t \in T$.
2. Map $g(C, t)$ to a functionally equivalent quantum gate cascade using any of the existing methods introduced in the past.
3. Traverse the resulting quantum circuit. Substitute any quantum gate $q(C, t)$ with target line $t$ with a cascade of quantum gates $q(C, t_1), q(C, t_2), \ldots, q(C, t_k)$.

**Example 5** *Consider the Toffoli gate with multiple target lines as shown in Fig. 5(a). Applying the proposed scheme, a functionally equivalent quantum gate cascade as shown in Fig. 5(b) results.*

Exploiting Toffoli gates with multiple target lines can significantly reduce the resulting quantum circuit costs for Toffoli gate cascades that differ only in their target lines, but not in their control lines. As an example, Toffoli gates with two control lines and $k$ target lines can be mapped to a quantum circuit composed of $2 + 3k$ gates only – using the established mapping from Fig. 3(a) considering single target lines, the quantum circuit would be composed of $5k$ gates. Analogously, the size of the quantum gate cascade scale with increasing number of control lines.

While this already represents a substantial decrease in the size of the resulting quantum circuits, further alternatives are possible. As another example, a Toffoli gate with multiple target lines $g(C, T)$ with $T = \{t_1, t_2, \ldots, t_k\}$ can be mapped to a quantum gate cascade as follows:

1. Consider a Toffoli gate $g(C, t)$ with a single target line $t \in T$.
2. Map $g(C, t)$ to a functionally equivalent quantum gate cascade using any of the existing methods introduced in the past.




(a) Toffoli Gate (b) Quantum gate cascade

Fig. 6. Improved mapping of Toffoli gate with multiple targets

TABLE II
QUANTUM COSTS FOR TOFFOLI GATES WITH MULTIPLE TARGET LINES

(a) Current state-of-the-art [14]

| Control lines \ Target lines | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |
| 1 | 1 | 2 | 3 | 4 |
| 2 | 5 | 10 | 15 | 20 |
| 3 | 14 | 28 | 42 | 56 |
| 4 | 20 | 40 | 60 | 80 |
| 5 | 32 | 64 | 96 | 128 |
| 6 | 44 | 88 | 132 | 176 |
| 7 | 64 | 128 | 192 | 256 |

(b) Proposed solution

| Control lines \ Target lines | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |
| 1 | 1 | 2 | 3 | 4 |
| 2 | 5 | 7 | 9 | 11 |
| 3 | 14 | 16 | 18 | 20 |
| 4 | 20 | 22 | 24 | 26 |
| 5 | 32 | 34 | 36 | 38 |
| 6 | 44 | 46 | 48 | 50 |
| 7 | 64 | 66 | 68 | 70 |

3. For each remaining target line $t' \in T \setminus \{t\}$, add a CNOT gate $\mathrm{q}(\{t\}, t')$ before and after the quantum gate cascade generated in Step 2.

**Example 6** *Consider the Toffoli gate with multiple target lines as shown in Fig. 6(a). Applying the proposed scheme, a functionally equivalent quantum gate cascade as shown in Fig. 6(b) results.*

Note that the proposed mapping scheme is not recommended for Toffoli gates $\mathrm{g}(C, T)$ with $|C| \leq 1$, i.e. gates with no or only a single control line. As reviewed in Section B, gates $\mathrm{g}(C, t)$ with $C$ being empty or being equal to $\{c\}$ with $c \in X$ are already treated as quantum gates. Hence, multiple target versions of such gates can simply be realized by a cascade of NOT/CNOT gates $\mathrm{q}(C, t_1), \ldots, \mathrm{q}(C, t_k)$ with $\{t_1, \ldots, t_k\} = T$.

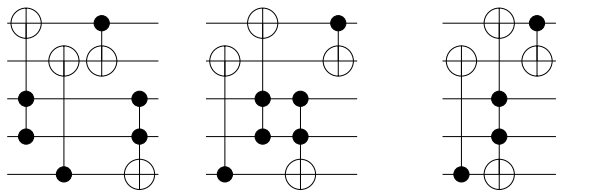
Using this improved mapping, the size of the resulting quantum gate cascades can further be reduced from $2+3k$ to $3 + 2k$ for a Toffoli gate with two control lines and $k$ target lines (again, the current state-of-the-art leads to quantum gate cascades of size $5k$). In the following, only this improved mapping scheme is considered.

The decrease in the costs compared to the current state-of-the-art mapping scheme becomes more substantial when the number of control lines increases. To illustrate this, Table II shows the resulting costs for a selection of Toffoli gates with multiple target lines[2]. Table II(a) provides the costs that result when applying the current state-of-the-art, while Table II(b) provides the costs that result when the proposed method is used. How these promising observations can be exploited in actual circuit realizations is presented in the next section.

## V. Application

Toffoli gates with multiple target lines enable more efficient mappings to quantum circuits. However, so far only Toffoli gates with single target lines are supported by existing circuits and synthesis methods. This section first shows how existing circuits can be re-synthesized to exploit the proposed concept. Afterwards, a possible exploitation of multiple target lines in an existing synthesis method, namely the ESOP-based synthesis approach presented in [9], is illustrated.

[2]For reasons of clarity, the best number of ancillary lines is assumed here.



(a) Reversible circuit (b) Step (c) Optimized circuit

Fig. 7. Exploitation in existing circuits

### A. Exploitation in Existing Circuits

One obvious application of the proposed concept in existing circuits is to simply merge Toffoli gates that share the same set of control lines. For example, Toffoli gates with single target lines $\mathrm{g}(C, t_1)$ and $\mathrm{g}(C, t_2)$ can be substituted by one Toffoli gate with multiple target lines $\mathrm{g}(C, \{t_1, t_2\})$. Afterwards, redundancies can be removed. For example, a cascade of two Toffoli gates $\mathrm{g}(C, \{t_1, t_2\})$ and $\mathrm{g}(C, \{t_1\})$ can be simplified to a single Toffoli gate $\mathrm{g}(C, \{t_2\})$. This can be generalized as follows:

**Definition 4 (Merging Rule)** *A cascade of two Toffoli gates with the same set of control lines* $\mathrm{g}(C, T_1)$ *and* $\mathrm{g}(C, T_2)$ *can be merged to a single Toffoli gate* $\mathrm{g}(C, T_1 \triangle T_2)$ *with* $T_1 \triangle T_2 := (T_1 \cup T_2) \setminus (T_1 \cap T_2)$ *being the symmetric difference. Note that, if* $T_1 = T_2$*, both gates are simply removed as they represent the identity.*

Besides that, the *moving rule* of reversible circuits can be applied after adjusting it to the new definition. That is:

**Definition 5 (Moving Rule)** *Two adjacent gates* $\mathrm{g}(C_1, T_1)$ *and* $\mathrm{g}(C_2, T_2)$ *can be interchanged if* $C_1 \cap T_2 = \emptyset$ *and* $C_2 \cap T_1 = \emptyset$*, i.e. of none of the target lines of one gate is a control line of the other gate.*

Moving gates through the circuits enables further possibilities for the application of the merging rule and, thus, leads to further reductions. The following example illustrates the benefits.

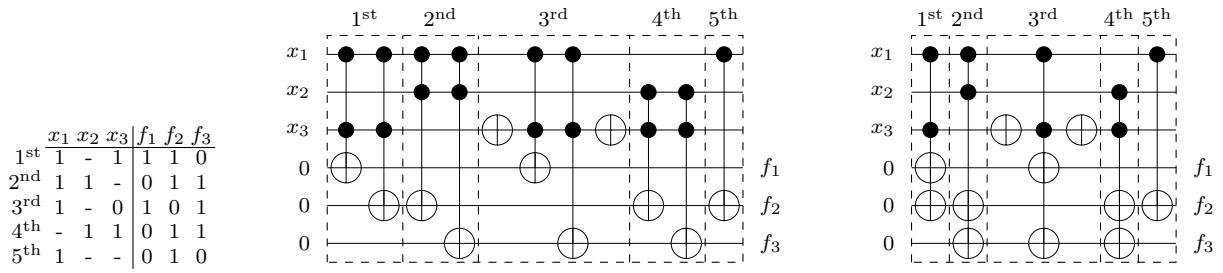
**Example 7** *Consider the circuit shown in Fig. 7(a). Moving the inner gates to the outside using the moving rule leads to a circuit as shown in Fig. 7(b). This enables to merge both Toffoli gates using the merging rule eventually leading to a circuit as shown in Fig. 7(c). Given that, the corresponding quantum circuit costs are reduced from 12 to 9.*

### B. Exploitation in ESOP-based Synthesis

The proposed concept of multiple targets can also be applied in existing synthesis methods for reversible circuits. This is demonstrated in the following by means of the ESOP-based synthesis approach presented in [10][3]. For this purpose, the general idea of this approach is briefly reviewed first before the application of multiple target Toffoli gates is discussed in more detail.

ESOP-based synthesis generates a reversible circuit from a Boolean function provided as *Exclusive Sum of Products* (ESOPs). ESOPs are two-level descriptions of Boolean functions that are represented as the exclusive disjunction (EXOR) of conjunctions of literals (called *products*). A *literal* either is a propositional variable or its negation. That is, an ESOP is the most general form of two-level AND-EXOR expressions.

[3]Note that similar applications are possible in other approaches e.g. the transformation-based approach presented in [18] or the exact synthesis approach presented in [19].




| | $x_1$ | $x_2$ | $x_3$ | $f_1$ | $f_2$ | $f_3$ |
|---|---|---|---|---|---|---|
| 1st | 1 | - | 1 | 1 | 1 | 0 |
| 2nd | 1 | 1 | - | 0 | 1 | 1 |
| 3rd | 1 | - | 0 | 1 | 0 | 1 |
| 4th | - | 1 | 1 | 0 | 1 | 1 |
| 5th | 1 | - | - | 0 | 1 | 0 |

(a) ESOP

(b) Previously obtained circuit

(c) Optimized circuit

Fig. 8. Exploitation in ESOP-based synthesis

Having an ESOP representing a function $f : \mathbb{B}^n \to \mathbb{B}^m$, the ESOP-based synthesis approach generates a circuit with $n + m$ lines, whereby the first $n$ lines also work as primary inputs. The last $m$ circuit lines are initialized to constant 0 and work as primary outputs. Having that, Toffoli gates are selected such that the desired function is realized. This selection exploits the fact that a single product $x_{i_1} \dots x_{i_k}$ of an ESOP description directly corresponds to a Toffoli gate with control lines $C = \{x_{i_1}, \dots, x_{i_k}\}$. In case of negative literals, NOT gates (i.e. Toffoli gates with $C = \emptyset$) are applied accordingly. Based on these ideas, a circuit realizing a function given as ESOP can be derived as illustrated in the following example.

**Example 8** *Consider the function $f$ to be synthesized as depicted in Fig. 8(a)*[4]*. The first product $x_1x_3$ affects $f_1$ and $f_2$. Hence, two Toffoli gates which have target lines $f_1$ and $f_2$ and control lines $C = \{x_1, x_3\}$ are added (see Fig. 8(b)). The third product $x_1\overline{x}_3$ includes a negative literal. Thus, a NOT gate is needed at line $x_3$ to generate the appropriate value for the next mappings. Again, two Toffoli gates are added and afterwards, a further NOT gates is applied to restore the value of $x_3$ (needed again by the fourth product). This procedure is continued until all products have been considered. The resulting circuit is shown in Fig. 8(b).*

Applying ESOP-based synthesis often leads to cascades of Toffoli gates with the same set of control lines. In fact, each time a product affects more than one output, Toffoli gates with the same control lines are added. In the example of Fig. 8, this is the case in all products except the last one. In all these cases, a single Toffoli gate with multiple target lines is sufficient.

**Example 9** *Consider again the function $f$ to be synthesized as depicted in Fig. 8(a). Using Toffoli gates with multiple target lines, only a single Toffoli gate needs to be added for each product. This leads to an optimized realization as shown in Fig. 8(c). In this case, the costs of the respective quantum gate cascades are reduced from 43 (cost of the previously obtained realization from Fig. 8(b)) to 31.*

Note that the function to be synthesized in Fig. 8(a) is rather small. With increasing number of inputs and, therefore, an increasing size of the products, also the number of Toffoli gates suitable for multiple target lines increases. As the experiments in the next section show, this leads to substantial improvements in the resulting costs.

[4]The column on the left-hand side gives the respective products, where a "1" on the $i^{\text{th}}$ position denotes a positive literal (i.e. $x_i$) and a "0" denotes a negative literal (i.e. $\overline{x}_i$), respectively. A "–" denotes that the respective variable is not included in the product. The right-hand side gives the respective primary output patterns.

TABLE III
EVALUATIONS WITH EXISTING CIRCUITS

| Circuit | $n$ | Single Target Lines RevG | Single Target Lines QuaG | Mutliple Target Lines RevG | Mutliple Target Lines QuaG | Impr (%) |
|---|---|---|---|---|---|---|
| apex4_202 | 28 | 5,376 | 209,448 | 1,695 | 69,846 | -66.65 |
| decod_217 | 21 | 80 | 1,458 | 26 | 510 | -65.02 |
| table3_264 | 28 | 1,012 | 75,605 | 369 | 26,903 | -64.42 |
| ex1010_230 | 20 | 2,611 | 143,396 | 1,184 | 61,646 | -57.01 |
| in2_236 | 29 | 405 | 22,248 | 193 | 10,794 | -51.48 |
| dc1_220 | 11 | 39 | 371 | 19 | 182 | -50.94 |
| inc_237 | 16 | 93 | 1,815 | 45 | 900 | -50.41 |
| in0_235 | 26 | 338 | 18,668 | 175 | 9,410 | -49.59 |
| misex3_242 | 28 | 1,752 | 111,827 | 880 | 56,627 | -49.36 |
| misex1_241 | 15 | 55 | 859 | 27 | 436 | -49.24 |
| cordic_218 | 25 | 2,533 | 249,116 | 1,130 | 126,816 | -49.09 |
| apla_203 | 22 | 80 | 3,096 | 39 | 1,620 | -47.67 |
| misex3c_243 | 28 | 1,721 | 107,888 | 877 | 56,456 | -47.67 |
| sao2_257 | 14 | 88 | 4,154 | 49 | 2,274 | -46.80 |
| cm42a_207 | 14 | 35 | 324 | 17 | 177 | -45.37 |
| dist_223 | 13 | 185 | 5,378 | 107 | 3,051 | -43.57 |
| f2_232 | 8 | 19 | 209 | 12 | 125 | -40.19 |

## VI. Experimental Evaluation

In order to confirm the benefits of multiple target lines, the approaches and optimization techniques presented in Section V have been implemented in C++ on top of *RevKit* [20] and evaluated using circuits from the *RevLib* [21] benchmark library as well as the ESOP-based synthesis approach [10]. As initial mapping method, the state-of-the-art solution proposed in [14] has been applied. All experiments have been conducted on an Intel Core i5-2500 with 8GB of memory.

### A. Evaluation with Existing Circuits

Existing circuits from the *RevLib* benchmark library do not include Toffoli gates with multiple target lines. As a result, the merging rule and the moving rule introduced in Section V.A have been applied to generate reversible circuits with multiple target lines so that their benefits can be exploited. Afterwards, both the original circuits with single target lines only and the newly generated ones have been mapped to quantum circuits as proposed in Section IV.

Overall, this leads to a reduction in the size of the resulting quantum gate cascades by approx. 8% if *all* circuits available in *RevLib* are considered. Note that this includes many small circuits for which no improvement can be obtained at all. In contrast, particularly for larger circuits improvements of up to 66% are possible. Table III shows the best improvements which have been observed during our evaluation. The first columns give the name and the number $n$ of lines of the respective circuits. Afterwards, the number of reversible gates (*RevG*) and the number of the resulting quantum gates (*QuaG*) are reported for the original *RevLib* circuits (considering single target lines only) and the improved realizations (considering multiple target lines). The total improvement is provided by the last column.

### B. Evaluation with ESOP-based Synthesis

In a second evaluation, we observed how existing synthesis approaches can profit from multiple target lines. For this purpose, we extended the ESOP-based synthesis approach from [9] as described in Section V.B. Afterwards, circuits have been generated using both approaches as well as functions from *RevLib*.

TABLE IV
EVALUATION WITH ESOP-BASED SYNTHESIS

| Circuit | Single Target RevG | Single Target QuaG | Multiple Target RevG | Multiple Target QuaG | Impr (%) QuaG |
|---|---|---|---|---|---|
| seq_201 | 2243 | 245625 | 575 | 34827 | -85.82 |
| ex5p_154 | 748 | 25843 | 140 | 3817 | -85.23 |
| urf6_77 | 2911 | 274091 | 1347 | 41091 | -85.01 |
| bw_116 | 262 | 3852 | 33 | 795 | -79.36 |
| urf5_76 | 390 | 16133 | 210 | 5287 | -67.23 |
| spla_202 | 1163 | 96782 | 600 | 31781 | -67.16 |
| urf2_73 | 914 | 24695 | 401 | 8576 | -65.27 |
| urf1_72 | 1742 | 60139 | 869 | 22777 | -62.13 |
| hwb8_64 | 885 | 20973 | 410 | 7965 | -62.02 |
| urf3_75 | 3089 | 134853 | 1632 | 51273 | -61.98 |
| hwb7_15 | 382 | 7409 | 174 | 2939 | -60.33 |
| hwb9_65 | 1799 | 56621 | 868 | 22472 | -60.31 |
| hwb6_14 | 152 | 2000 | 74 | 875 | -56.25 |
| aj-e11_81 | 32 | 220 | 18 | 100 | -54.55 |
| decod24-enable_32 | 11 | 79 | 6 | 37 | -53.16 |
| 4_49_7 | 34 | 234 | 20 | 129 | -44.87 |
| frg2_161 | 2322 | 190708 | 1496 | 107444 | -43.66 |
| 4mod7_26 | 23 | 143 | 17 | 83 | -41.96 |
| 5xp1_90 | 95 | 1180 | 65 | 748 | -36.61 |
| hwb4_12 | 24 | 144 | 16 | 93 | -35.42 |
| decod24_10 | 11 | 27 | 6 | 18 | -33.33 |
| apex5_104 | 782 | 46858 | 639 | 32075 | -31.55 |
| mod5adder_66 | 48 | 482 | 35 | 335 | -30.50 |
| cycle10_2_61 | 77 | 1662 | 64 | 1203 | -27.62 |
| 3_17_6 | 13 | 45 | 8 | 33 | -26.67 |
| ham3_28 | 11 | 23 | 6 | 17 | -26.09 |
| fredkin_3 | 7 | 23 | 5 | 17 | -26.09 |
| plus127mod8192_78 | 53 | 985 | 44 | 745 | -24.37 |
| ex-1_82 | 10 | 26 | 6 | 20 | -23.08 |
| one-two-three_27 | 13 | 52 | 9 | 40 | -23.08 |
| plus63mod8192_80 | 52 | 1008 | 45 | 786 | -22.02 |
| 0410184_85 | 396 | 8740 | 307 | 6910 | -20.94 |
| alu3_97 | 112 | 2195 | 90 | 1754 | -20.09 |
| mini-alu_84 | 12 | 95 | 11 | 77 | -18.95 |
| add6_92 | 264 | 5517 | 211 | 4665 | -15.44 |
| alu4_98 | 1227 | 42204 | 1139 | 35961 | -14.79 |
| rd53_68 | 27 | 225 | 23 | 195 | -13.33 |
| adr4_93 | 67 | 663 | 53 | 588 | -11.31 |
| alu2_96 | 190 | 3760 | 176 | 3367 | -10.45 |
| rd32_19 | 9 | 29 | 8 | 26 | -10.34 |
| rd73_69 | 84 | 869 | 70 | 806 | -7.25 |
| rd84_70 | 147 | 1964 | 135 | 1847 | -5.96 |
| plus63mod4096_79 | 50 | 654 | 43 | 624 | -4.59 |
| apex2_101 | 3618 | 567406 | 3499 | 552800 | -2.57 |
| 4mod5_8 | 6 | 22 | 6 | 22 | 0.00 |
| 9symml_91 | 157 | 3442 | 157 | 3442 | 0.00 |
| alu1_94 | 36 | 231 | 33 | 231 | 0.00 |
| alu_9 | 9 | 43 | 9 | 43 | 0.00 |
| mod5d1_16 | 15 | 31 | 13 | 31 | 0.00 |
| mod5d2_17 | 20 | 36 | 14 | 36 | 0.00 |
| sf_232 | 9 | 39 | 9 | 39 | 0.00 |

The results are summarized in Table IV. The first column gives the name of the function. Afterwards, the number of reversible gates (*RevG*) and the number of the resulting quantum gates (*QuaG*) are reported for (1) the circuits obtained by the original approach considering single target lines only and (2) the circuits obtained by the proposed approach additionally exploiting multiple target lines. The last column lists the total improvement.

Also in this evaluation substantial improvements have been observed. On average, the size of the resulting quantum gate cascades can be decreased by 28% (even though also here some circuits are included where no improvement at all can be reported). In the best cases, reductions of up to 85% are possible.

## VII. Conclusions

In this work, we proposed the consideration of multiple target lines in the synthesis of reversible circuits. We observed that synthesis approaches often lead to reversible circuits which are composed of cascades of Toffoli gates with the same set of control lines. Motivated by this, we introduced an improved mapping scheme which exploits this structure when generating the corresponding quantum gate cascades. Exploiting multiple target lines helps to improve both, existing reversible circuits and existing synthesis approaches. Improvements of up to 66% in the first case and up to 85% in the latter case have been observed.

As a result, this work builds a promising basis for a further investigation in this direction. In particular, an application to other gate types (e.g. Fredkin gates), to gates including negative control lines, or nearly introduced mapping schemes (e.g. the one introduced in [16]) is of interest. Besides that, the consideration of multiple target lines in other synthesis methods (e.g. the transformation-based approach introduced in [18] or the exact synthesis approach introduced in [19]) is left for future work.

## References


[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. New York, NY, USA: Cambridge University Press, Oct. 2000.

[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Theory of computing*, 1996, pp. 212–219.

[3] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," *Foundations of Computer Science*, pp. 124–134, 1994.

[4] A. K. Lenstra and E. R. Verheul, "Selecting cryptographic key sizes," in *Int'l Workshop on Practice and Theory in Public Key Chryptography*, Jan. 2000, pp. 446–465.

[5] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," *Nature*, vol. 414, p. 883, 2001.

[6] M. Steffen, "Towards a scalable superconducting qubit architecture," in *APS March Meeting of the American Physical Society*, 2012.

[7] V. Shende, A. Prasad, I. Markov, and J. Hayes, "Synthesis of reversible logic circuits," *IEEE Trans. on CAD*, vol. 22, no. 6, pp. 710–722, June 2003.

[8] R. Wille and R. Drechsler, "BDD-based synthesis of reversible logic for large functions," in *Design Automation Conference*. ACM, July 2009, pp. 270–275.

[9] K. Fazel, M. Thornton, and J. Rice, "ESOP-based Toffoli gate cascade generation," in *IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing*. IEEE, Aug. 2007, pp. 206–209.

[10] R. Drechsler, A. Finder, and R. Wille, "Improving ESOP-based synthesis of reversible logic using evolutionary algorithms," in *Applications of Evolutionary Computation*, Apr. 2011, pp. 151–161.

[11] M. Soeken, R. Wille, C. Hilken, N. Przigoda, and R. Drechsler, "Synthesis of reversible circuits with minimal lines for large functions," in *Asia and South Pacific Design Automation Conference*, Jan. 2012, pp. 85–92.

[12] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, "Elementary gates for quantum computation," *Phys. Rev. A*, vol. 52, no. 5, pp. 3457–3467, Nov. 1995.

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

[14] D. M. Miller, R. Wille, and Z. Sasanian, "Elementary quantum gate realizations for multiple-control Toffolli gates," in *Int'l Symp. on Multiple-Valued Logic*, May 2011, pp. 217–222.

[15] T. Toffoli, "Reversible computing," in *Automata, Languages and Programming*, ser. Lecture Notes in Computer Science, J. W. de Bakker and J. van Leeuwen, Eds., vol. 85. Springer, July 1980, pp. 632–644.

[16] Z. Sasanian, R. Wille, and D. M. Miller, "Realizing reversible circuits using a new class of quantum gates," in *Design Automation Conference*, 2012, pp. 36–41.

[17] N. O. Scott and G. W. Dueck, "Pairwise decomposition of Toffoli gates in a quantum circuit," in *Great Lakes Symp. on VLSI*, 2008, pp. 231–236.

[18] D. M. Miller, D. Maslov, and G. W. Dueck, "A transformation based algorithm for reversible logic synthesis," in *Design Automation Conference*. ACM, June 2003, pp. 318–323.

[19] D. Große, R. Wille, G. W. Dueck, and R. Drechsler, "Exact multiple-control Toffoli network synthesis with sat techniques," *IEEE Trans. on CAD*, vol. 28, no. 5, pp. 703–715, May 2009.

[20] M. Soeken, S. Frehse, R. Wille, and R. Drechsler, "RevKit: An open source toolkit for the design of reversible circuits," in *Reversible Computation 2011*, ser. Lecture Notes in Computer Science, vol. 7165, 2012, pp. 64–76, RevKit is available at www.revkit.org.

[21] R. Wille, D. Große, L. Teuber, G. W. Dueck, and R. Drechsler, "RevLib: An online resource for reversible functions and reversible circuits," in *Int'l Symp. on Multiple-Valued Logic*, May 2008, pp. 220–225.