

# Design Space Exploration in the Mapping of Reversible Circuits to IBM Quantum Computers

Philipp Niemann<sup>\*†</sup>, Alexandre A. A. de Almeida<sup>‡</sup>, Gerhard Dueck<sup>§</sup>, and Rolf Drechsler<sup>\*†</sup>

<sup>\*</sup>Department of Computer Science, University of Bremen, Bremen, Germany

<sup>†</sup>Cyber-Physical Systems, DFKI GmbH, Bremen, Germany

<sup>‡</sup>School of Engineering, Ilha Solteira, São Paulo State University, Brazil

<sup>§</sup>Faculty of Computer Science, University of New Brunswick, Canada

pniemann@uni-bremen.de, alexandre.amaral@unesp.br, gdueck@unb.ca, drechsle@informatik.uni-bremen.de

**Abstract**—With more and more powerful quantum computers becoming available, there is an increasing interest in the efficient mapping of a given quantum circuit to a particular quantum computer (so-called *technology mapping*). In most cases, the limitations of the targeted quantum hardware have not been taken into account when generating these quantum circuits in the first place. Thus, the technology mapping is likely to induce a considerable overhead for such circuits. In this paper, we consider the realization of reversible circuits consisting of multiple-controlled Toffoli gates on IBM quantum computers. We show that choosing different quantum-level decompositions can indeed have a significant impact on the mapping overhead. Based on this observation, we present an approach to perform design space exploration to obtain quantum circuits with reduced overhead by exploiting information about the targeted quantum hardware as well as the reversible circuit. An experimental evaluation shows that this approach often leads to considerable reductions of the technology mapping overhead with negligible runtime.

## I. INTRODUCTION

Quantum computing [1] is a flourishing research area. By exploiting quantum-mechanical phenomena like superposition, entanglement and phase shift, quantum computing has the prospect of solving certain practically relevant problems significantly faster than any known classical algorithm. Due to this fact, the synthesis and technology mapping of quantum circuits, i.e. the automatic generation of quantum circuits realizing a given quantum functionality on a targeted quantum computer architecture, has become an active research area. As quantum logic synthesis is a very complex and challenging problem, Boolean functions—which constitute a major component in many quantum algorithms—are usually treated separately using a two-step approach: the desired Boolean function is first realized in terms of a reversible circuit, i.e., by means of classical reversible logic gates, after which the resulting circuit is transformed to a functionally equivalent quantum circuit by mapping each reversible gate to a corresponding cascade of quantum gates [2]–[4]. The resulting quantum circuit usually does not take into account the limitations of the targeted quantum computer architecture regarding the (non-)availability of certain gates. Thus, a so-called *technology mapping* is required in order to make the circuit executable on a particular quantum computer. This

leads to an overhead, since gates that are not natively available have to be rewritten as a functionally equivalent cascade of native gates.

In recent years, there has been much progress in the development of quantum computers with increasing computational power due to improvements regarding the number and quality of the basic computational entities (qubits) [5]–[7]. In present architectures, there is usually only one multiple-qubit gate available, e.g. the two-qubit controlled-NOT or controlled-Z gates, and this gate can only be applied on a subset of qubit pairs. This also holds for the publicly available QX series by IBM that has been made available to the public via free-of-charge cloud access and is also considered in this paper.

Researchers have developed different strategies to solve such quantum architecture constraints [8], [9], most of which rely on inserting a (minimal) number of SWAP gates that move the desired qubits close to each other. To solve the problem for the IBM architectures where all CNOTs additionally can only be applied in one direction, several approaches have been proposed, see e.g. [10]–[14]. Again, most proposals focus on the insertion of SWAP gates and the work in [14] even presented a method to determine the minimal number of SWAP and  $H$  operations for small circuits by using Boolean satisfiability solvers. In contrast, there are also promising proposals to not permute the qubits using SWAP gates, but to use precomputed sequences of gates to realize arbitrary CNOT gates [15], [16].

However, the mapping of reversible circuits, i.e., circuits composed of multiple-controlled Toffoli (MCT) gates, still is an area to be explored. Some works have addressed the problem [16], [17], but are limited to MCT gates with up to two controls. The approach from [18] shows appealing results, but it requires MCT circuits that have been synthesized in a particular way with many adjacent gates sharing the same target. Recently, there have been proposals for optimized decompositions of arbitrary MCT gates [4], [19], but these focus on reducing the number of  $T$  gates rather than the technology mapping overhead that results from the restricted availability of CNOT gates.

In this paper, we consider the mapping of MCT gates with the aim to reduce the technology mapping overhead and show that there is a large potential for improvement when employing different decompositions. We propose an algorithm for exploring the design space of the possible decompositions for each MCT gate—despite its exponential size. An experimental evaluation shows this leads to considerable reductions in the technology mapping overhead with relatively small run-times.

The remainder of this paper is structured as follows. The next section introduces notations and preliminaries needed in this paper. Section III analyses the potential of different MCT decompositions, followed by Section IV presenting our method for design space exploration which determines optimized mappings for MCT gates. Experimental results are presented in Section V. Finally, the paper is concluded in Section VI.

## II. BACKGROUND AND PRELIMINARIES

To keep the paper self-contained, this section briefly introduces the basics of reversible and quantum circuits.

### A. Reversible Functions and Circuits

A multi-output Boolean function  $f: \{0, 1\}^n \rightarrow \{0, 1\}^n$  is called *reversible* if  $f$  is *bijective*, i.e., if each input pattern is mapped to a unique output pattern and vice versa.

Reversible functions on  $n$  inputs are realized by reversible circuits consisting of at least  $n$  lines (carrying binary values). These reversible circuits are cascades of reversible gates belonging to a particular gate library, with no fan-out or feedback. The most popular gate library is given by multiple-controlled Toffoli (MCT) gates which are defined as follows:

**Definition 1 (Multiple-Controlled Toffoli gate):** Given a set of circuit lines  $X = \{x_1, x_2, \dots, x_n\}$ , an  $m$ -controlled Toffoli gate  $T(C; t)$  is given by a possibly empty set of control lines  $C = \{x_{c_1}, \dots, x_{c_m}\} \subset X$  (where  $|C| = m$ ), and a target line  $t \in X \setminus C$ . On the target line, the gate performs the mapping  $t \mapsto t \oplus (x_{c_1} \wedge \dots \wedge x_{c_m})$ , i.e. the target line is inverted if, and only if, all control lines are in the 1-state. All other lines pass through unaltered.

**Example 1:** An MCT gate with no control line always inverts the target line and is thus the well-known *NOT gate*. An MCT gate with a single control line is called a *controlled-NOT (CNOT) gate*. The case of two control lines is the original *Toffoli gate*.

When drawing reversible and quantum circuits, a  $\oplus$  indicates the target line and  $\bullet$  to indicate the control connection of an MCT gate. For instance, the left-hand side of Fig. 1a represents a (2-controlled) Toffoli gate.

### B. Quantum Circuits

A quantum circuit is a model of quantum computation representing a sequence of quantum operations [1]. Each operation is represented by a quantum gate and the circuit is a cascade of quantum gates where the circuit lines represent the *qubits (quantum bits)* of a quantum system.

In this work, we consider the Clifford+ $T$  gate library [20] which is popular due to the fact that it only contains a

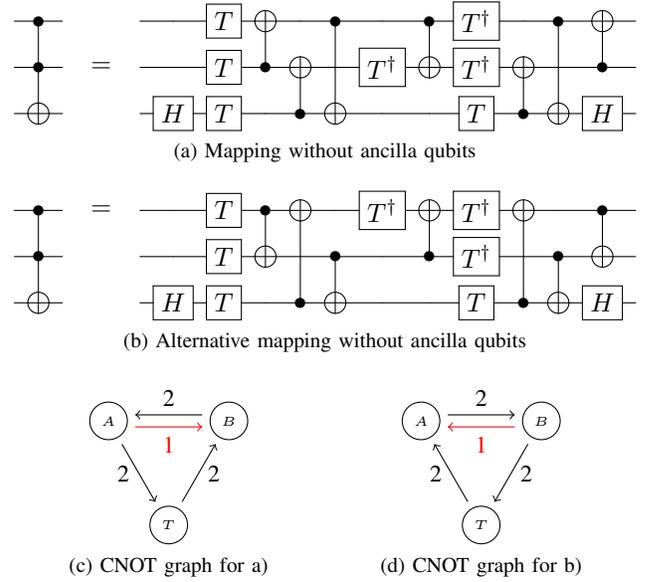


Fig. 1. Mapping of Toffoli gate into Clifford+ $T$  circuits [21] and corresponding CNOT graphs.

small, discrete number of gates and due to its compatibility to schemes for error correction which are required for large-scale quantum computing. More precisely, the library contains the *CNOT* gate, the *NOT* gate as well as some other single-qubit gates ( $H, S, S^\dagger, T, T^\dagger$ ) whose precise definitions are not relevant for this paper and who will rather be considered as blackboxes in the following.

**Example 2:** Figure 1a shows the realization of the Toffoli gate, i.e. a two-controlled MCT gate, in Clifford+ $T$  [21, Fig. 7(a)]. The gates are to be applied from left to right.

### C. Mapping MCT circuits to Clifford+ $T$

The mapping of reversible MCT circuits to Clifford+ $T$  quantum circuits is typically conducted in two steps. First, MCT gates with  $c \geq 3$  controls are decomposed into MCT gates with less than three controls, i.e. NOT, CNOT, and Toffoli gates (which constitute the so-called *NCT* library). Afterwards, each NCT gate is mapped individually to an equivalent cascade of Clifford+ $T$  gates.

Usually, the well-known decomposition of MCT into NCT gates proposed by Barenco et al. [22] is used. The basic construction requires  $c - 2$  additional circuit lines/qubits (so-called *ancilla* lines/qubits) that are used to store intermediate computations, but are finally restored to their initial value/state (cf. Fig. 2a for the case  $c = 4$ ). Note that these ancilla can have an arbitrary value/state.

This basic construction can be employed as a building block in order to perform a decomposition also when less than  $c - 2$  ancilla are available (cf. Fig. 2b for the case  $c = 5$  and a single ancilla).

To map the NCT circuit to Clifford+ $T$ , usually the mapping from Fig. 1a is employed for Toffoli gates, while NOT and CNOT gates do not have to be mapped.

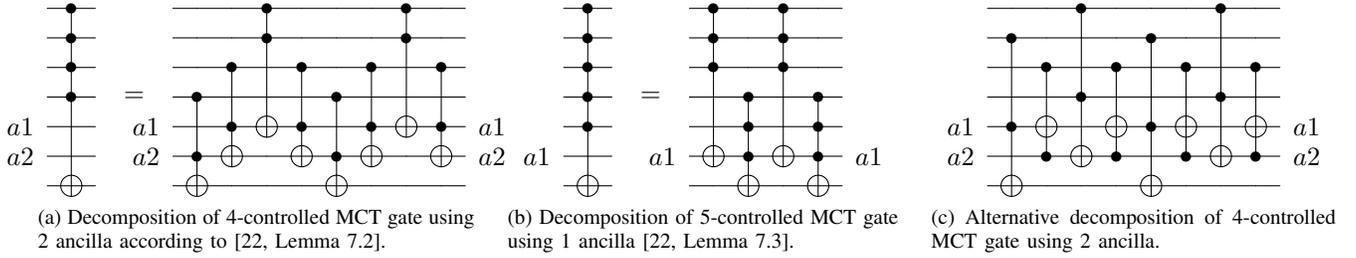


Fig. 2. Decompositions of MCT to NCT gates from Barenco et al. [22].

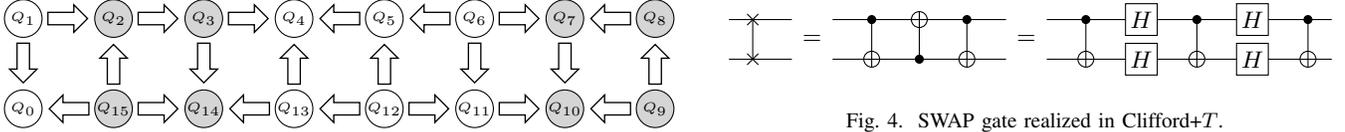


Fig. 3. IBM QX5 architecture [23].

#### D. Technology Mapping to IBM Quantum Computers

The IBM Q project provides public access to quantum computers via a cloud service. The purpose of this access is to provide the opportunity to *experiment* with real devices. Currently, there are architectures with 5, 14 and 16 qubits available (not continuously online) along with a simulator for up to 32 qubits [5]. In spite of continuous improvement, the architectures are still in development and have limitations regarding the number and the fidelity of qubits. As a consequence, the execution of quantum gates in the real devices is susceptible to errors such that the output may not be the expected one.

Another limitation of the architectures is the types of quantum gates that can be used in the circuits. All architectures essentially support arbitrary single-qubit gates (especially the ones from the Clifford+ $T$  library) as well as the CNOT gate, although, its availability is limited. More precisely, the architectures restrict the interaction of the physical qubits, i.e., it is only possible to apply a CNOT gate to a defined set of qubits.

These CNOT restrictions are expressed in a *coupling graph*. Figure 3 shows the coupling graph of the 16 qubits IBM QX5 (Rueschlikon) architecture [23]. The circles represent the physical qubits ( $Q_0, Q_1, Q_2, \dots, Q_{15}$ ) and the arrows indicate the availability of a CNOT gate between the qubits. To this end, a CNOT gate can only be applied if the qubit at the base of the arrow is the control qubit and the tip of the arrow represents the target qubit. For instance, a CNOT with control qubit  $Q_1$  and the target in qubit  $Q_0$  is natively available in the IBM QX5 architecture. Overall, for quantum circuits with 16 logical qubits, there are  $16 \cdot 15 = 240$  different CNOT gates possible, but only 22 are natively available in the IBM QX5 architecture. This shows the technology mapping challenge to be solved in order to realize a quantum circuit on these devices.

More precisely, the challenge is to find an efficient way to map the misplaced CNOT gates, i.e., to transform the non-

available CNOT gates. Clearly, it is promising to map the logical qubits to the available physical ones in such a way that the number of misplaced gates is minimized. While there might be some corner cases where it is possible to perform this mapping in such a way that all used CNOT gates are available, this is clearly not the case in general. Thus, in order to realize the remaining misplaced CNOTs, either the target and control qubits can be moved to appropriate positions (e.g., using so-called SWAP gates which correspond to a cascade of 3 CNOT and 4  $H$  gates as shown in Fig. 4) or predefined cascades of native gates can be employed as suggested in [15].

### III. MOTIVATION AND GENERAL IDEA

As discussed in Section II-C, Clifford+ $T$  quantum can be derived from MCT circuits by mapping each gate to a corresponding cascade of quantum gates. The technology mapping overhead of the decompositions is usually not taken into account in this mapping of MCT circuits to the quantum level, since this overhead can hardly be quantified in advance. However, as we will motivate in the following, the decompositions are quite sensitive to permutations of the control lines and also to the particular choice of the ancilla qubits. While all such possible configurations finally realize the desired functionality, the associated technology mapping overhead can vary significantly.

To this end, let us consider the Clifford+ $T$  realization of a Toffoli gate from Fig. 1a. Ignoring the single CNOT in the center, each qubit is used twice as a control and also twice as a target of a CNOT. This is illustrated in the CNOT graph in Fig. 1c where the vertices correspond to the qubits and a directed edge represents one or two CNOTs between the corresponding qubits. Here one can see that the CNOTs form a cyclic structure (aside from a single “reverse” CNOT which is applied in the opposite direction). Hence, in the best case, all but the reverse CNOT are native operations and can be realized without any technology mapping overhead. In contrast, in the worst case, only the reverse CNOT is a native operation and a large mapping overhead is induced.

Note that the best case can become the worst case and vice versa, by simply swapping the two controls. This swap can be realized with SWAP gates (which increase the technology mapping overhead), but also without any additional overhead by simply using a different decomposition of the Toffoli gate to Clifford+T. More precisely, the decomposition given in Fig. 1b serves this purpose (as can be seen from the CNOT graph in Fig. 1d). As it was derived from the original decomposition by swapping the two control lines, it has the same gate count and, thus, no additional overhead.

Similarly, the decomposition of MCT gates to NCT gates offers a large degree of freedom. In fact, the decomposition selects triples of control and ancilla qubits and creates corresponding Toffoli gates (cf. Fig. 2a). However, there is *a priori* no order implied neither on the controls nor on the ancilla qubits. Hence, there are many possibilities of selecting the triples—all of which lead to decompositions that are functionally equivalent but may induce significantly different overheads. Again, the optimal among those decompositions can be derived by inserting SWAP gates or by using a reordered decomposition. For instance, Fig. 2c shows an alternative decomposition of a 4-controlled MCT gate that has been gained from the decomposition in Fig. 2a by permuting control and ancilla qubits and, thus, has the same gate count and no additional overhead.

In general, for a  $c$ -controlled MCT gate on an  $m$ -qubit quantum computer and the basic decomposition using  $c - 2$  ancilla qubits, there are  $c!$  permutations of the controls and the  $c - 2$  ancillae can be chosen freely from the  $m - c - 1$  qubits that are neither target nor control. This means there are  $c! \cdot \binom{m-c-1}{c-2} \cdot (c - 2)!$  possible permutations/decompositions to be considered. Hence, for  $m = 16$  there are already 86,400 permutations for  $c = 5$  and 2,177,280 permutations for  $c = 6$ . Note that these numbers do not yet take into account the additional degree of freedom that results from the control permutations of the individual Toffoli gates.

If the 1-ancilla decomposition is used, the ancilla can be chosen freely from the  $m - c - 1$  qubits that are neither target nor control. For each of these choices, the controls plus the ancilla need to be distributed to MCT gates with  $\lfloor \frac{c+1}{2} \rfloor$  and  $\lceil \frac{c+1}{2} \rceil$  controls (cf. Fig. 2b). This can be done in  $\binom{c+1}{\lfloor \frac{c+1}{2} \rfloor}$  ways. For instance, for  $c = 10$  there are MCT gates with 5 and 6 controls and overall 2,310 possibilities to choose the ancilla and distribute the controls to these gates. By multiplying this number with the number of decompositions for the smaller MCT gates from above, we arrive at the number of 5,229,100,800 possible decompositions.

#### IV. ALGORITHM FOR DESIGN SPACE EXPLORATION

The main idea of the proposed approach is to exploit the potential of different MCT decompositions for a reduced technology mapping overhead by performing a design space exploration of the possible decompositions for each MCT gate—taking into account the topological restrictions of the targeted quantum computer. This can either be done

- *during the mapping* phase (as long as a designated placement of the involved logical qubits has already been derived), or
- as a *post-mapping optimization* (as long as the “boundaries” of the original MCT gates at the quantum level are known such that in-place replacements with optimized quantum level realizations become possible).

However, the design space for single MCT gates grows very quickly as the analysis in Section III has shown. Consequently, we do not enumerate and explicitly construct all possible decompositions for all MCT gates. We rather perform a *single* decomposition of the respectively considered MCT gate to the quantum level (which yields a quantum circuit that realizes this, and only this, single MCT gate) and then make use of a *constrained technology mapping* approach in order to compute a decomposition into NCT gates with small technology mapping overhead (as illustrated in the top of Fig. 5). Note that an off-the-shelf technology mapping of such a single-MCT-gate “circuit” yields a realization without any misplaced CNOT, but usually the logical qubits are mapped to the physical qubits in some arbitrary fashion in order to reduce the technology mapping overhead. Especially, the controls and the target of the MCT gate can be mapped anywhere, far away from their designated positions and a potentially expensive rearrangement of the qubits might be necessary in order to actually use this realization.

Thus, we enforce that the target is mapped to its designated position and that controls are mapped to any of the designated control positions (while the ancilla still can be mapped arbitrarily). As a consequence, the actual mapping determined by the accordingly constrained technology mapping approach corresponds to some particular choice of ancilla qubits and some particular permutation of the controls, i.e. some NCT decomposition of the original MCT gate with small technology mapping overhead. In a second step (see the bottom of Fig. 5), the NCT decomposition is checked for further optimizations that might be derived by swapping controls of individual Toffoli gates (which corresponds to applying the alternative Toffoli gate decomposition shown in Fig. 1b). Note that this step is comparably in-expensive as the optimal order of controls can be pre-computed (once for all possible combinations of target and control qubits) and be stored in a database. The resulting optimized Clifford+T circuit is then employed to realize the MCT gate.

This idea has been implemented on top of the technology mapping algorithm presented in [16]. More precisely, the algorithm in [16] computes a placement of a Clifford+T quantum circuit to a given quantum circuit architecture using a set of pre-computed optimized sequences of gates to realize the CNOTs rather than SWAP gates. The resulting placements are optimal w.r.t. the technology mapping overhead, under the premise that only this particular set of pre-computed optimized sequences of gates is used to realize the CNOTs—and ignoring potential optimizations that could result from employing the alternative Toffoli gate mapping from Fig. 1b.

In order to achieve this optimality, the qubit placement

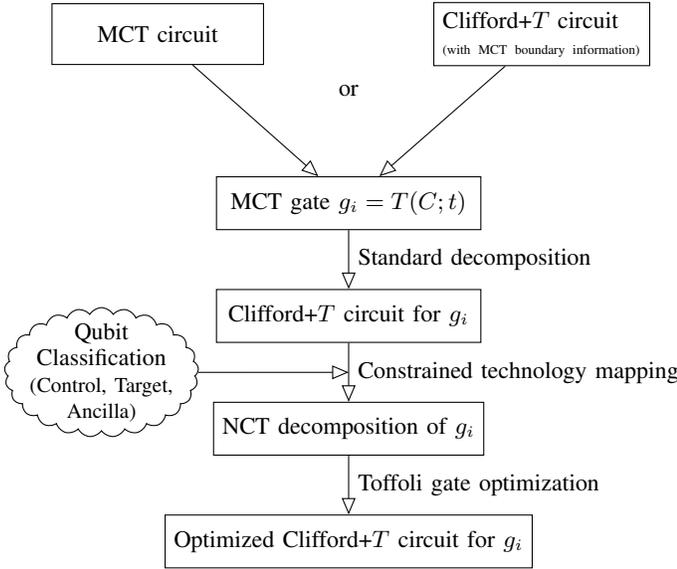


Fig. 5. Illustration of the proposed Design Space Exploration.

problem is formulated as an Integer Linear Programming (ILP) problem over the variables  $G_{ij}C_{kl} \in \{0, 1\}$  which denote whether the CNOT gate with control on logical qubit  $i$  and target on logical qubit  $j$  will be mapped to a CNOT gate with control on physical qubit  $k$  and physical target  $l$ . The ILP problem contains several constraints to ensure that only such variable assignments are valid that correspond to a well-defined qubit placement, e.g. constraints like  $\sum_{k,l=0}^m G_{ij}C_{kl} = 1$  enforce that all logical qubits that occur as target or control of a CNOT gate are indeed mapped to some physical qubit, while further constraints are required to ensure that this mapping is indeed unique. This ILP problem is passed to a solver that computes an optimal assignment of the variables that minimizes the provided objective function (which, in this case, expresses the technology mapping overhead of the associated qubit placement). This assignment can then easily be translated to a particular qubit placement. Note that in the original ILP formulation used in [16], qubits are placed arbitrarily, such that in the case of single-MCT-gate quantum circuits considered here, target and controls of the MCT gate could be placed anywhere and the resulting placement would in general not correspond to a decomposition of the considered MCT gate. In fact, it would realize an MCT gate with the same number of controls, but different target or controls.

To this end, we extended the LP formulation in order to specify a classification of the logical as well as physical qubits into target, control or ancilla qubits and enforce that each logical qubit may only be placed at an (arbitrary) physical qubit with the same classification. This is enforced by the constraint

$$\sum_{(i,j,k,l) \in S} G_{ij}C_{kl} = 0$$

where the set  $S$  describes all incompatible qubit classifications

$$S = \{(i, j, k, l) \mid \text{class}(i) \neq \text{class}(k) \vee \text{class}(j) \neq \text{class}(l)\}.$$

By adding this constraint, the LP solver determines the optimal permutation of the controls as well as the optimal choice of ancilla qubits for the given decomposition type and qubit placement, and thus a decomposition of the original MCT gate into NCT with small technology mapping overhead.

In order to speed-up computation times for circuits, computed decompositions can be stored in a lookup table in order to avoid redundant computations. Moreover, a timeout can be provided to the solver after which the solver stops and returns the best solution found so far (which, however, might then be a sub-optimal solution).

## V. EXPERIMENTAL EVALUATION

In this section, we evaluate the potential of the proposed design space exploration in comparison with previous work. To this end, the proposed scheme has been implemented in C++ on top of RevKit [24] and has been tested on single MCT gates as well as a suite of benchmarks taken from RevLib [25]. As target architecture, we employed the IBM QX5 with  $m = 16$  qubits in order to compare with the results from [16].

### A. Single MCT gates

In order to quantify the potential benefit of a topology-aware technology mapping for single MCT gates, we constructed corresponding NCT decompositions according to [22] using the `nct` command provided by RevKit and mapped the resulting NCT circuits to Clifford+T using the mapping in Fig. 1a.

For each number of controls, we then randomly chose 10 configurations of controls, targets, and ancilla then asked the solver to find the best as well as the worst permutation of controls and ancillae (given a timeout of 60 CPU seconds). Moreover, we also searched for global best cases where the target, controls and ancilla can be placed arbitrarily. For these runs, the timeout was set to 3600 CPU seconds.

The results are summarized in Table I. Here, the first column denotes the number of controls of the MCT gate ( $c$ ), the following columns show the average technology mapping overhead of the 10 configurations without any permutations (Conv.), the average overhead improvement for the corresponding best case ( $\Delta_{best}$ ) as well as the average overhead degradation for the corresponding worst case ( $\Delta_{worst}$ ). The final columns denote the global best cases observed for MCT gates with the respective number of controls (Global Best) as well as the average difference between the global optimum and the best permutation for the respective configuration ( $\Delta_{global\_best}$ ).

First of all, let us note that the case of Toffoli gates ( $c = 2$ ) behaves quite differently from the others. For each configuration, there are only two possible decompositions to be considered, and if one of those has a large technology mapping overhead, then the same also holds for the other one. To this end, the IBM QX5 architecture is “balanced”. Note that the minimal technology mapping overhead of 14 can be found for  $MCT(\{Q_{14}, Q_{15}\}, Q_2)$ ,  $MCT(\{Q_{14}, Q_{15}\}, Q_3)$  and  $MCT(\{Q_9, Q_{10}\}, Q_8)$ ,  $MCT(\{Q_9, Q_{10}\}, Q_7)$ . At these locations, the coupling graph of IBM QX5 is quite similar

TABLE I  
TECHNOLOGY MAPPING OVERHEAD FOR SINGLE MCT GATES

$c$	Conv.	$\Delta_{best}$	$\Delta_{worst}$	Global Best	$\Delta_{global\_best}$
2	154	-3%	4%	14	1,004%
3	544	-50%	68%	62	316%
4	1,120	-69%	94%	126	163%
5	1,580	-73%	105%	198	101%
6	2,108	-75%	102%	270	91%
7	2,818	-79%	67%	336	70%
8	3,091	-67%	74%	436	110%
9	6,569	-71%	88%	1,396	34%
10	7,490	-66%	94%	1,876	34%

to the CNOT graph from Fig. 1c. More precisely, as also highlighted in gray in Fig. 3, there are cycles with three of four arrows pointing in the same direction: clockwise for  $Q_{15} \Rightarrow Q_2 \Rightarrow Q_3 \Rightarrow Q_{14}$  and counter-clockwise for  $Q_9 \Rightarrow Q_8 \Rightarrow Q_7 \Rightarrow Q_{10}$ .

In contrast, a significant potential for improvement can be observed for  $c \geq 2$ . In fact, the average possible improvement compared to the randomly chosen initial control permutation is between 50% and 79% and, on the other hand, also the average deviation of this permutation from the worst case is between 67% and 105%. However, we also see that the achieved best cases still incur significantly more overhead than the determined global optimum. This means that it might be beneficial to precompute configurations with small technology mapping overhead and then apply SWAP gates in order to transform the given configuration into one of those configurations if no sufficiently good permutation can be found. Note that this precomputation only needs to be performed once for each architecture and can then be re-used for any circuit.

### B. Mapping of Reversible Circuits

We evaluated the proposed approach on a suite of benchmarks taken from [25]. In Table II, we report the results for the set of benchmarks considered in [16].

Here, the first two columns describe the benchmark in terms of its name (ID) and its number of qubits (L). The next columns denote the technology mapping overhead reported in [16] as well as obtained by the proposed approach. The remaining columns list the relative improvement of the proposed approach, the share of the improvement that originates from the subsequent Toffoli gate control permutations at NCT level (NCT impr.) as well as the overall run-time (in CPU seconds).

On the one hand, the results show that the approach is able to provide considerable improvements in many cases (up to 46%). On the other hand, the improvements are often not as significant as the above evaluations for single MCT gates might suggest. A possible explanation is that many benchmarks do not contain MCT gates with more than 2 controls and the potential improvement for Toffoli gates in the IBM QX5 architecture is rather small (around 5%) as shown above.

TABLE II  
EXPERIMENTAL RESULTS

Benchmark		Technology Mapping Overhead				
ID	L	[16]	Proposed	$\Delta_{conv}$	NCT impr.	run-time
4mod5-bdd_287	7	74	70	-5.4%	0%	0.15
0410184_169	14	284	276	-2.8%	0%	0.36
alu-bdd_288	7	117	117	0.0%	0%	0.12
sf_274	6	926	862	-6.9%	75%	0.66
sf_276	6	926	862	-6.9%	75%	0.68
cnt3-5_179	16	310	290	-6.5%	0%	0.24
4gt4-v0_80	6	214	214	0.0%	0%	0.26
rd53_133	7	934	866	-7.3%	71%	1.00
f2_232	8	1,963	1,661	-15.4%	13%	10.84
cm152a_212	12	2,110	1,980	-6.2%	12%	3.06
cm82a_208	8	1,014	966	-4.7%	33%	1.12
ex3_229	6	560	418	-25.4%	28%	0.31
ham7_104	7	539	503	-6.7%	44%	0.38
ex2_227	7	1,067	881	-17.4%	13%	0.89
majority_239	7	991	851	-14.1%	23%	1.12
4gt4-v0_78	6	388	264	-32.0%	0%	0.23
rd53_135	7	514	434	-15.6%	30%	0.43
rd53_131	7	854	678	-20.6%	14%	1.39
rd53_138	8	212	200	-5.7%	0%	0.18
mod5adder_127	6	857	755	-11.9%	16%	0.68
4gt4-v0_73	6	660	584	-11.5%	42%	0.43
rd53_251	8	2,382	1,988	-16.5%	10%	1.22
4gt4-v0_72	6	392	358	-8.7%	0%	0.27
C17_204	7	909	833	-8.4%	53%	1.91
mod8-10_177	6	778	616	-20.8%	35%	0.43
rd53_311	13	748	704	-5.9%	0%	0.40
rd73_140	10	448	432	-3.6%	0%	0.36
sys6-v0_111	10	410	390	-4.9%	0%	0.34
wim_266	11	2,214	1,868	-15.6%	23%	4.13
mini_alu_305	10	334	314	-6.0%	0%	0.26
rd53_130	7	2,237	1,643	-26.6%	19%	6.96
dc1_220	11	4,437	3,839	-13.5%	23%	2.23
hwb5_53	6	2,715	2,153	-20.7%	13%	1.53
cm42a_207	14	3,960	3,060	-22.7%	9%	7.61
sym6_145	7	9,215	6,919	-24.9%	12%	17.58
con1_216	9	2,377	1,525	-35.8%	7%	79.44
sym6_316	14	812	740	-8.9%	0%	0.53
hwb6_56	7	16,985	12,101	-28.8%	10%	23.67
sym9_146	12	1,045	1,031	-1.3%	0%	0.62
z4_268	11	10,725	5,761	-46.3%	4%	287.60

## VI. CONCLUSIONS

In this paper, we considered the mapping of reversible MCT circuits to IBM quantum computers. So far, the topological constraints of the targeted quantum hardware are hardly taken into account when performing the quantum level decomposition of the circuits. We showed that there is indeed significant potential to reduce the resulting technology mapping overhead if the characteristics of the actual quantum computer are taken into account for this decomposition. Moreover, we proposed an algorithm for determining decompositions with minimal technology mapping overhead. Due to the formulation as an ILP problem, the approach allows for a complete exploration of the exponentially large design space in acceptable run-times. For many benchmarks, a reduction of the technology mapping overhead of around 20% can be observed.

## VII. ACKNOWLEDGMENT

The authors are grateful for the support of CAPES PDSE process n° 88881.189547/2018-01. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

## REFERENCES

- [1] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge Univ. Press, 2000.
- [2] A. A. A. de Almeida, G. W. Dueck, and A. C. R. da Silva, "Efficient realization of toffoli and NCV circuits for IBM QX architectures," in *Reversible Computation - 11th International Conference, RC 2019, Lausanne, Switzerland, June 24-25, 2019, Proceedings*, 2019, pp. 131–145.
- [3] R. Wille, M. Soeken, C. Otterstedt, and R. Drechsler, "Improving the mapping of reversible circuits to quantum circuits using multiple target lines," in *ASP Design Automation Conf.*, 2013, pp. 85–92.
- [4] N. Abdessaied, M. Amy, M. Soeken, and R. Drechsler, "Technology mapping of reversible circuits to Clifford+T quantum circuits," in *Int'l Symp. on Multiple-Valued Logic*, 2016, pp. 150–155.
- [5] IBM. Project IBM Q. [Online]. Available: <https://www.ibm.com/quantum-computing/>
- [6] R. Courtland, "Google aims for quantum computing supremacy," *IEEE Spectrum*, vol. 54, no. 6, pp. 9–10, 2017.
- [7] J. Hsu, "CES 2018: Intel's 49-qubit chip shoots for quantum supremacy," *IEEE Spectrum Tech Talk*, 2018.
- [8] J. Ding and S. Yamashita, "Exact synthesis of nearest neighbor compliant quantum circuits in 2D architecture and its application to large-scale circuits," *IEEE Trans. on CAD*, pp. 1–1, 2019.
- [9] S. Hu, D. Maslov, M. Pistoia, and J. Gambetta, "Efficient circuits for quantum search over 2D square lattice architecture," in *Design Automation Conf.*, 2019, pp. 1–2.
- [10] A. Zulehner, A. Paler, and R. Wille, "An efficient methodology for mapping quantum circuits to the IBM QX architectures," *IEEE Trans. on CAD*, vol. 38, no. 7, pp. 1226–1236, July 2019.
- [11] A. Matsuo, W. Hattori, and S. Yamashita, "Reducing the overhead of mapping quantum circuits to IBM Q system," in *Int'l Symp. Circ. and Systems*, Sapporo, May 2019, pp. 1–5.
- [12] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, "A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits," *IEEE Trans. on CAD*, vol. 32, no. 6, pp. 818–830, 2013.
- [13] M. Y. Siraichi, V. F. d. Santos, S. Collange, and F. M. Q. Pereira, "Qubit allocation," in *International Symposium on Code Generation and Optimization*, ser. CGO 2018. Vienna: ACM, 2018, pp. 113–125.
- [14] A. Ash-Saki, M. Alam, and S. Ghosh, "QURE: Qubit re-allocation in noisy intermediate-scale quantum computers," in *Design Automation Conf.* New York, NY, USA: ACM, 2019, pp. 141:1–141:6.
- [15] R. Wille, L. Burgholzer, and A. Zulehner, "Mapping quantum circuits to IBM QX architectures using the minimal number of SWAP and H operations," in *Design Automation Conf.* Las Vegas: ACM, 2019, pp. 142:1–142:6.
- [16] A. A. A. de Almeida, G. W. Dueck, and A. C. R. da Silva, "CNOT gate mappings to Clifford+T circuits in IBM architectures," in *Int'l Symp. on Multiple-Valued Logic*, 2019, pp. 7–12.
- [17] A. A. A. de Almeida, G. W. Dueck, and A. C. R. da Silva, "Finding optimal qubit permutations for ibm's quantum computer architectures," in *2019 32nd Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2019, pp. 1–6.
- [18] M. B. Ali, T. Hirayama, K. Yamanaka, and Y. Nishitani, "Quantum cost reduction of reversible circuits using new Toffoli decomposition techniques," in *CSCI*, Dec 2015, pp. 59–64.
- [19] M. Soeken, F. Mozafari, B. Schmitt, and G. DeMicheli, "Compiling permutations for superconducting QPUs," in *Design, Automation and Test in Europe*, Florence, March 2019, pp. 1349–1354.
- [20] P. Niemann, A. Gupta, and R. Drechsler, "T-depth optimization for fault-tolerant quantum circuits," in *Int'l Symp. on Multiple-Valued Logic*, May 2019, pp. 108–113.
- [21] P. O. Boykin, T. Mor, M. Pulver, V. Roychowdhury, and F. Vatan, "A new universal and fault-tolerant quantum basis," *Information Processing Letters*, vol. 75, no. 3, pp. 101–107, 2000.
- [22] A. Barenco, C. H. Bennett, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, "Elementary gates for quantum computation," *Phys. Rev. A*, vol. 52, pp. 3457–3467, 1995.
- [23] IBM. IBM Q 16 Rueschlikon V1.x.x. [Online]. Available: <https://github.com/Qiskit/ibmq-device-information/tree/master/backends/rueschlikon/V1>
- [24] M. Soeken, S. Frehse, R. Wille, and R. Drechsler, "RevKit: A toolkit for reversible circuit design," in *Workshop on Reversible Computation*, 2010, pp. 69–72, RevKit is available at <http://www.revkit.org>.
- [25] 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*, 2008, pp. 220–225, RevLib is available at <http://www.revlib.org>.