Combining SWAPs and Remote CNOT Gates for Quantum Circuit Transformation

Philipp Niemann*†, Luca Mueller*, and Rolf Drechsler*†

*Department of Computer Science, University of Bremen, Bremen, Germany
†Cyber-Physical Systems, DFKI GmbH, Bremen, Germany
{pniemann,lucam,drechsler}@uni-bremen.de

Abstract—**Quantum computers offer enormous speed advantages over their classical counterparts. Still, optimization on quantum circuits is necessary to further increase their potential. Additionally, physical realizations of quantum computers place restrictions on quantum circuits, regarding the available quantum gates. In order to satisfy these restrictions, non-native gates need to be expressed as an equivalent cascade of natively available quantum gates which induces a mapping overhead. Two complementary approaches to this problem are to move around the qubits (using SWAP gates) or to apply so-called remote gates, i.e. pre-computed cascades of native gates which keep the qubit placement.**

In this paper, we explore how combinations of movements and remote gates can be employed to reduce the required overhead regarding the number of native gates as well as the circuit depth. We also discuss ways to find out which qubits to address with the movements in order to optimize these metrics. Our general evaluation is supplemented by evaluations on two IBM quantum computer architectures to show how quantum circuits can be optimized by the presented patterns.

I. Introduction

Quantum computers [1] promise to have enormous computational power and, thus, to solve relevant problems significantly faster than their classical counterparts. In recent years, large efforts have been put on their development. While the mathematical foundations have been widely explored in the last decades and meaningful quantum algorithms have been proposed, the physical realization currently provides the biggest obstacle preventing the widespread use of quantum computers.

Driven by big players like IBM, Google, and Intel, more and more powerful quantum computer architectures have been presented in recent years with increasing quantity and quality of the so-called qubits—the basic computational entities in quantum computing. While there are several different approaches regarding the employed technology to realize qubits, one of the physical constraints that all proposed architectures have in common is the limited availability of quantum gates. Typically, multi-qubit gates are much harder to realize than single-qubit gates and in many cases there is only one multi-qubit gate natively available, namely the two-qubit controlled-NOT (CNOT) gate. This does not restrict the computational power of the architectures in general, since there are universal gate libraries consisting of the CNOT gate and single-qubit gates only, e.g. the Clifford+T library [2] which allow to realize arbitrary quantum computations. However, the CNOT is typically only available on a small subset of physically adjacent qubit pairs. Consequently, computations that require CNOT operations on distant qubits can become quite complex. Fortunately, there are ways to simulate these logical CNOTs at the physical level and transform a quantum circuit that contains non-native CNOTs to a quantum circuit containing only native gates and, thus, being ready for the execution on the targeted quantum architecture.

Many approaches to find efficient CNOT implementations have been suggested, e.g. in [3]–[10]. The underlying idea of most of these solutions is to use so-called SWAP gates in order move distant qubits to adjacent positions where a native CNOT gate can be applied. Alternatively, there have been proposals to use remote CNOTs, i.e. realize non-native CNOTs using pre-computed, optimized sequences of native gates (sometimes also referred to as *templates*) [5]. Recently, there has been a proposal to combine SWAPs with remote gates in the mapping of high-level reversible circuits [11] which demonstrated that this approach can outperform state-of-the-art mapping approaches regarding the required gate overhead. In fact, most approaches focus on the gate overhead that is introduced, but also the resulting circuit depth shall be considered, since a smaller depth leads to a shorter execution time. Due to the short decoherence time of the qubits, this is also a rather limited resource.

In this paper, we explore how a combination of SWAPs and remote gates can be adapted to the realization of non-native CNOT gates. More precisely, we explore combinations of qubit movements and remote CNOTs with a focus on reducing circuit depth without (substantially) increasing the gate overhead. We come up with patterns which combine the advantages of SWAPs (i.e., a smaller depth) and remote gates (i.e., less gate overhead). We determine cost- and depth-optimal patterns for two IBM quantum computer architectures and evaluate their impact on the transformation of entire quantum circuits. The main differences to [11] are:

- This work does not treat SWAPs as atomic operations, but explicitly makes use of optimizations at the quantum circuit level that only become possible when SWAPs are considered as cascades of CNOT (and Hadamard) gates.
- We restrict to the realization of non-native CNOT gates, while [11] is concerned with the realization of high-level gates (so-called Multiple-Controlled Toffoli gates) which are a generalization of CNOT gates. The patterns used in this paper can hardly be generalized to these gates.

The remainder of this paper is structured as follows. The next section introduces notations and preliminaries needed in the paper. Section III outlines the motivation for and the general idea of the proposed approach. In Section IV, we discuss several possibilities to combine SWAPs with remote gates (denoted as patterns) and outline possible optimizations of the patterns due to the particular structure of the SWAP gates in Section V. In Section VI, we determine depth- as well as cost-optimal instantiations of these patterns, before we evaluate their impact on the transformation of entire quantum circuits in Section VII. Finally, the paper is concluded in Section VIII.

II. Background and Preliminaries

To keep the paper self-contained, this section briefly introduces the basics of quantum computation and the technology mapping of quantum circuits to (IBM) quantum computers.

A. Quantum States and Circuits

In contrast to classical bits which can only assume two discrete states, *qubits* can represent any combination of the classical Boolean values 0 and 1. More precisely, the state space of a qubit is a 2-dimensional Hilbert space such that all possible states can be written as $|\psi\rangle = a|0\rangle + b|1\rangle = \binom{a}{b}$ where $|0\rangle, |1\rangle$ denote the computational basis states (associated with the classical Boolean values) and $a, b \in \mathbb{C}$ are complex-valued numbers such that $|a|^2 + |b|^2 = 1$. Analogously, the state space of an n-qubit quantum system has 2^n basis states ($|0\dots00\rangle, |0\dots01\rangle, \dots, |1\dots11\rangle$) and the state of such system can be described by a 2^n-dimensional complex-valued vector.

A quantum circuit is a model of quantum computation representing a sequence of quantum operations [1]. Each operation is a unitary transformation and is represented by a quantum gate. The operation of a quantum gate acting on n qubits is uniquely determined by a $2^n \times 2^n$ unitary matrix.

A popular gate library for universal quantum computation is the Clifford+T library [2] which contains controlled-NOT ($CNOT$), Hadamard (H) and T (T) gates, represented by the following matrices (where $\omega = e^{i\frac{\pi}{4}}$):

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, T = \begin{pmatrix} 1 & 0 \\ 0 & \omega \end{pmatrix}.$$

A CNOT on two qubits α and β, denoted as CNOT(α, β), performs a NOT operation on the target qubit β if, and only if, the control qubit α is in the $|1\rangle$-state.

Example 1. *The left-hand side of Fig. 1 shows the circuit notation of a CNOT. Horizontal lines denote the qubits, the control qubit connection is indicated by a small, filled circle and the target qubit is illustrated by $\oplus$. As shown on the right-hand side, control and target of a CNOT can be swapped by applying Hadamard gates before and after the CNOT gate.*

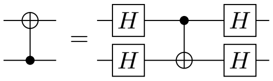

Fig. 1. Swapping control and target of a CNOT using Hadamard gates.

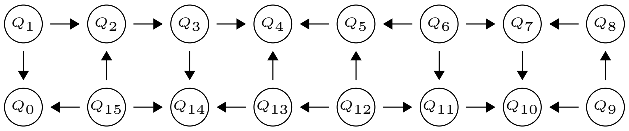

Fig. 2. IBM QX5 architecture.

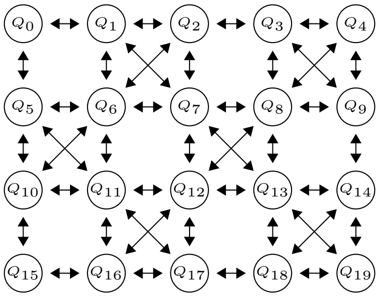

Fig. 3. IBM Q20 architecture.

B. Technology Mapping to (IBM) Quantum Computers

All IBM 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 qubit pairs.

These CNOT restrictions are expressed in a *coupling graph*. Figure 2 shows the coupling graph of the 16 qubits IBM QX5 (Rueschlikon) architecture [12]. 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, CNOT($Q1$, $Q2$), i.e. a CNOT with control on $Q1$ and target on $Q2$ is available on QX5, but not vice versa. 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. Analogously, the 20-qubit Q20 architecture shown in Fig. 3 natively supports only 74 out of $20 \cdot 19 = 380$ possible CNOTs. Thus, the challenge is to find an efficient way to realize the CNOT gates that are not natively available in order to implement arbitrary quantum algorithms.

III. General Idea

Existing approaches for the efficient realization of non-native CNOTs have mainly focused on inserting SWAP gates, which are compositions consisting of several CNOT and Hadamard gates as shown in Fig. 4. To illustrate this idea, consider the realization of a CNOT($Q3, Q0$) in IBM QX5. In order to implement this non-native CNOT, one could simply swap $Q3$ with $Q2$ and $Q2$ with $Q1$ using SWAP

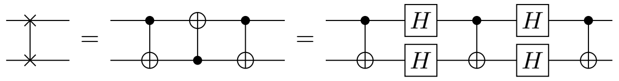

Fig. 4. SWAP gate realized in Clifford+T.

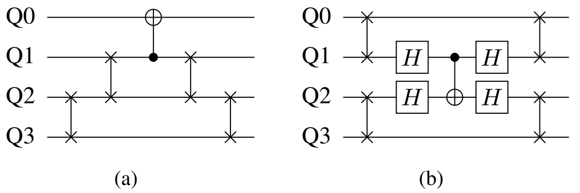

Fig. 5. Realization of CNOT($Q3, Q0$) in IBM QX5 using SWAP gates

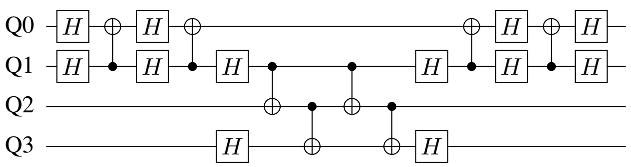

Fig. 6. Realization of CNOT($Q3, Q0$) in IBM QX5 according to [5].

gates, transferring the state of $Q3$ to that of $Q1$, and then apply the native CNOT($Q1, Q0$). This is illustrated in Fig. 5a. Alternatively, as shown in Fig. 5b, one could swap $Q3$ with $Q2$ and $Q0$ with $Q1$—this can be done in parallel and, thus, reduces the circuit depth—and then use the construction from Fig. 1 to apply the (non-native) CNOT($Q2, Q1$). In both cases, the qubit placement, i.e. the mapping of the logical qubits to the physical qubits of the quantum computer, is changed and the same set of SWAPs has to be applied in reverse order to restore the original positions of $Q1, Q2$, and $Q3$, if required. However, each SWAP introduces an additional cost of 7 gates (c.f. Fig. 4) resulting in a total cost of $4 \cdot 7 + 1 = 29$ gates, but complete SWAPs may not be required. Almeida et al. [5] identified several movements of control and target qubits which can be realized with reduced costs—resulting in a realization of CNOT($Q3, Q0$) using only 20 gates (shown in Fig. 6). If there is a (directed!) path from control to target in the coupling graph, also the multi-hop construction from [13] can be employed. It requires $4(d-1)$ CNOTs (where $d > 1$ is length of the path between control and target) and, thus, reduces the number of gates (as well as the depth) to 8 for the considered CNOT (as shown in Fig. 8).

In summary, SWAPs have the advantage of being suitable for a parallelization which might reduce the circuit depth, while it requires a higher number of native gates and the state of the target qubit is established quite late. In contrast, remote CNOTs typically require a smaller number of gates, are often able to establish the desired state of the target qubit relatively early, but usually do not offer room for parallelization have a large circuit depth.

The general idea of this paper is to combine SWAPs with remote CNOTs in order to combine the strengths of both for an efficient realization of non-native CNOTs. While most approaches to quantum circuit transformation do not require the original qubit placement to be restored after the execution of a non-native CNOT, but rather carry on with modified qubit placements, in the following we will assume that all realizations do not have any (side) effects on any, but the target qubit, i.e. the remaining qubits will be restored to the state that would have occurred if only the non-native would have been applied.

IV. Metrics and Patterns

In this section, we will discuss several possibilities to combine SWAPs with remote CNOTs (denoted as patterns) and analyze there behavior w.r.t. the following metrics:

- cos: The cost of the pattern, measured as the total number of (native) gates which have to be applied.
- dep: The total depth of the pattern, being the number of stages needed.
- dep_t: The target depth of the pattern, i.e. the number of stages until the target qubit has its desired state and is not changed afterwards.

While cost and depth have an immediate impact on the reliability of the resulting circuits, since all gates are prone to errors and the execution/decoherence time of the qubits is limited, the target depth can have a positive impact on the overall execution time of the circuits. If the target reaches its desired state earlier, the execution of the following gates which rely on this state might also be started earlier and so on.

In all patterns, the control or target qubit of the non-native CNOT (denoted as c,t) will first be moved or copied to positions c', t'. Then a remote CNOT(c', t') will be performed. Finally c', t' will be moved back to their original positions.

Note that in some patterns, we will have $c = c'$ or $t = t'$, i.e. sometimes control or target will not be moved at all. The metrics will be given w.r.t. to

- the distance d_c between c and c',
- the distance d_t between t and t', and
- the distance d between c and t.

A. X Pattern

The X pattern is the intuitive pattern in which the control and target movements are just sequentially carried out one after the other. The control and the target qubit move towards one another until a native $CNOT(c', t')$ can be carried out. Afterwards the qubits are moved back to their starting position.

Without any further optimization, the metrics for the X pattern are:

- cos: $2 * cos(SWAP) * (d_c + d_t) + 1$
- dep: $2 * dep(SWAP) * \max\{d_c, d_t\} + 1$
- dep_t: $2 * dep(SWAP) * \max\{d_c, d_t\} + 1$

where $cos(SWAP)$ and $dep(SWAP)$ denote the cost and depth of a single SWAP gate in the considered architecture. E.g., for QX5, we have $cos_{QX5}(SWAP) = 7$ and $dep_{QX5}(SWAP) = 5$, while for architectures with bidirectional CNOTs, like Q20, we have $cos(SWAP) = dep(SWAP) = 3$.

Apparently, it has no impact to cos how d_c and d_t are chosen, while the depth is minimal, if d_c and d_t are balanced, i.e. as close to each other as possible.

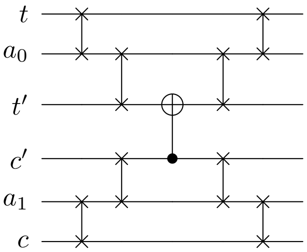

Fig. 7. The X movement pattern

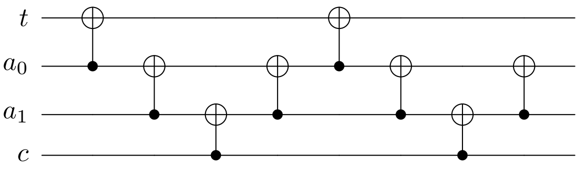

Fig. 8. The W movement pattern

B. W Pattern

The W pattern offers the potential of lowering cos as well as dep_t as compared to the X pattern. Neither control nor target are moved, i.e. $t = t'$ and $c = c'$, but the W-shape multi-hop pattern from [13] is applied directly between c and t using a path through the coupling graph. Note that by construction no parallelization is possible here. Given that all required CNOT gates are native to the architecture, the metrics for the W pattern are

- cos: $4 * (d - 1)$
- dep: $4 * (d - 1)$
- dep_t: $2 * (d - 1) + 1$

For each of the CNOT which is non-native, 4 Hadamard gates need to be added (according to Fig. 1), such that in the worst case cos grows by a factor of 5 and dep as well as dep_t grow by a factor of 3.

C. WX Pattern

The WX pattern is a combination between the X and the W pattern. While the target qubit is not moved at all (i.e., $t = t'$), the control qubit is moved to another qubit c' (like in the X pattern), and the W pattern is then applied between c' and t (whose distance is denoted as d_w). This has the advantage that the first gates of the W pattern can be applied in parallel to the control qubit movements.

The metrics for the WX pattern, given that all CNOTs of the W pattern are native gates, are

- cos: $2 * cos(SWAP) * d_c + 4 * (d_w - 1)$
- dep_t: $\max(d_w, d_c * dep(SWAP)) + d_w + 1$
- dep: $dep_t + (d_w - 1) + \max(d_w - 2, d_c * dep(SWAP))$

Again, the same worst case behavior as for the W pattern can occur.

Given the SWAP gate realization using only 2 CNOTs as discussed for the X pattern, the gate count becomes $4 * (d_c + d_w - 1) = 4 * (d - 1)$ which is the same as for the W pattern.

Before we determine optimal patterns w.r.t. to the choice of c', t' and the involved intermediate qubits, in the next section we will discuss quantum-level optimizations of the SWAP gates that lead to reduced costs.

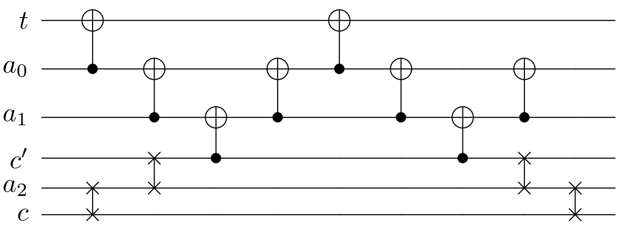

Fig. 9. The WX movement pattern

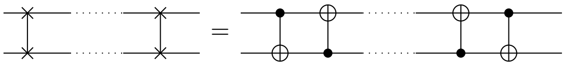

Fig. 10. SWAP gate optimization

V. SWAP Gate Optimizations

While investigating the above-mentioned patterns, we observed that SWAP gates as used in the X and WX pattern can be optimized. More precisely, complete SWAPs are not required here, but two of the three CNOT gates are sufficient as shown in Fig. 10. By this, the involved qubits remain in a "mixed" state until the SWAP is undone later on. For instance, in the X pattern and WX pattern, the state of qubit c is moved to qubit c', but the qubit c and the ancilla qubits in-between carry some mixed state when the native $CNOT(c', t')$ or the W pattern, respectively, is applied. In the X pattern, the qubit t' does not carry exactly the value of t when the native $CNOT(c', t')$ is applied, but also a mixed state. E.g., for the X pattern shown in Fig. 7, the states read (from top to bottom): a_0; t'; $t \oplus a_0 \oplus t'$; c; $c \oplus c'$; $c \oplus a_1$.

Nonetheless, all qubits are finally in their desired state.

With this optimization, we obtain $cos(SWAP) = dep(SWAP) = 2$ for architectures with bidirectional CNOTs such as, e.g., IBM Q20.

Moreover, in order to also reduce the depth, it is beneficial to interleave the CNOTs of the SWAP gates as shown in Fig. 11. By this, if there are more than two SWAP gates in a row, only the first two require a depth of 2, but for the remaining SWAPs the depth only increases by 1 each as illustrated in Fig. 12.

Thus, the depth of the first as well as the last part of the X pattern goes down to $\max\{d_c, d_t\} + 2$. In addition, also the target depth is reduced further, since the final state is established already after $\max\{d_c, d_t\} + 3 + d_t$ CNOTs.

By this, the optimized X pattern reaches a cost of $4(d - 1) + 1$, i.e. it only requires one gate more than the W pattern, but has a target depth of $d + 2$ and an overall depth of $d + 4$ in the best case (i.e. $d_c = d_t$). This is especially remarkable, since there is no way to "transfer" the value of c to t using less than d CNOTs in a row, even if would allow t to not only depend on the value of c, but also any other qubit in-between.

VI. Determining Optimal Patterns

In order to determine the optimal choices of c' and t' for the X and WX pattern and also the optimal choice of the intermediate qubits for all patterns, we performed an exhaustive search on all shortest paths between c and t.

Note that for architectures, where all CNOTs are bidirectional, one needs to consider a single CNOT(c, t) only for each

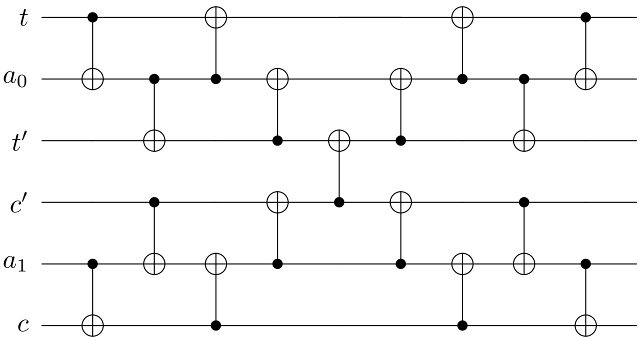

Fig. 11. Optimized X movement pattern

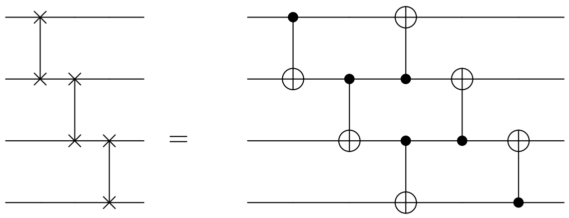

Fig. 12. Additional savings for more than two consecutive SWAP gates

possible distance d and one can restrict to one shortest path each, since the patterns will be isomorphic for all shortest paths and all CNOTs with the same distance. Since the diameter of Q20 is rather small (the maximum distance is 5) we will instead study an artificial 1-dimensional architecture LNN_{10} with 10 qubits $Q_0, \dots, Q_9$, where CNOTs exist in both directions between qubits Q_i and Q_{i+1} (for $i = 0, \dots, 8$), i.e. there is no CNOT between Q_9 and Q_0.

After constructing the corresponding patterns, we performed a simple gate reduction in which pairs of adjacent Hadamard or CNOT gates, which cancel out immediately, are removed.

A. Architectures with bidirectional CNOTs (Q20)

We performed single-objective optimizations w.r.t. the three considered cost metrics. This means for each CNOT we determined three patterns (X, W, and WX pattern) with are optimal w.r.t. either cos, dep or dep_t. As a consequence, it might be the case that the cos-optimal pattern is not dep-optimal and so on. However, it turned out that the dep_t-optimal patterns were always optimal w.r.t. to the other two metrics as well such that we will restrict to these patterns in the following.

The complete metrics for the determined X, W, and WX patterns are shown in Table I. Here the first column denotes the distance d between control and target of the CNOT. Recall that only this distance is important here, but not the actual choice of control and target qubit. Afterwards the metrics are provided for X, W, and WX patterns and also the respective value for d_c is given.

One can observe that the X patterns always require exactly one CNOT gate more than the W and WX pattern. The WX pattern outperforms the W pattern for $d \geq 3$, but itself is outperformed by the X pattern regarding circuit depth and target depth—except for $d = 2$ where the circuit depth is off by 1 and $d = 4$ where the target depth is off by 1. These exceptions are highlighted in bold font in Table I.

TABLE I
EXPERIMENTAL RESULTS FOR LNN_{10}

	X pattern				W pattern				WX pattern			
d	cos	dep	dep_t	d_c	cos	dep	dep_t	d_c	cos	dep	dep_t	d_c
1	1	1	1	0	1	1	1	–	1	1	1	0
2	**5**	**5**	3	1	4	4	3	–	4	4	3	0
3	9	5	4	1	8	8	5	–	8	7	4	1
4	13	9	**6**	2	12	12	7	–	12	9	5	1
5	17	9	7	2	16	16	9	–	16	12	7	1
6	21	11	8	3	20	20	11	–	20	15	8	2
7	25	11	9	3	24	24	13	–	24	17	9	2
8	29	13	10	4	28	28	15	–	28	19	10	3
9	33	13	11	4	32	32	17	–	32	21	11	3

For the X pattern, d_c is chosen such that d_c and d_t are balanced, i.e. $|d_t - d_c| = |d - 2d_c - 1| \leq 1$. For the WX pattern, the minimum value for dep_t—according to the formula from Section IV-C that does not take into account the SWAP gate optimizations discussed in Section V—is achieved if $|d_w - 2d_c| = |d - 3d_c| \leq 1$. It turned out that there are in general multiple possibilities to choose d_c such that the resulting dep_t is minimal. For $d = 2$ and $d = 5$, the provided values of d_c do not satisfy the above inequation, but the resulting patterns have the same optimal value for dep_t and a slightly better overall circuit depth dep.

B. Architecture with uni-directional CNOTs (QX5)

Here, we performed multi-objective optimizations w.r.t. to the three cost metrics. In the first run, we used the gate count (cos) as the dominating metric, followed by the circuit depth (dep) and target depth (dep_t). In the second run, we used circuit depth (dep) as the dominating metric, followed by the gate count (cos) and the target depth. In the third run we only used target depth (dep_t) as dominating metric.

The results show that cos and dep are conflicting metrics. More precisely, it is often not possible to determine a pattern which is optimal w.r.t. both metrics.

Depth-optimal X patterns are typically not cost-optimal, and also have a higher target depth than the cost-optimal patterns which, in most cases, also have an optimal target depth.

For W patterns, the metrics are not conflicting, while for WX patterns the depth-optimal patterns also have a minimal target depth, but lowering the depth always comes at a higher gate count.

In order to give an impression of these trends, the results for all CNOTs with a control on Q_0 and target on $t = Q_1, \dots, Q_9$ are summarized in Table II. Here, the first column denotes the target qubit t, and the remaining columns provide the metrics for cos- and dep-optimal X, W, and WX patterns. Note that for the W pattern there is only one series, since all patterns are both cos- and dep-optimal.

Comparing X against W and WX patterns, we observe that the X patterns have a higher gate count, i.e. a larger cos-value, than W and WX (for both, cos- and dep-optimal patterns), while W and WX almost have the same gate count. Regarding circuit depth, W has the largest circuit depths by far. X performs slightly better than WX for cos-optimal patterns, and significantly better than WX for dep-optimal patterns.

TABLE II
EXPERIMENTAL RESULTS FOR QX5 (CONTROL ON Q_0)

	X pattern						W pattern			WX pattern					
	cos-optimal			dep-optimal						cos-optimal			dep-optimal		
t	cos	dep	dep_t	cos	dep	dep_t	cos	dep	dep_t	cos	dep	dep_t	cos	dep	dep_t
1	5	3	3	5	3	3	5	3	3	5	3	3	5	3	3
2	11	9	5	11	9	5	10	8	5	10	8	5	10	8	5
3	19	13	7	21	9	8	14	12	7	14	11	6	14	11	6
4	27	15	8	31	13	11	18	16	9	18	13	7	18	13	7
5	35	21	11	37	13	11	28	24	15	28	18	11	28	18	11
6	43	15	12	43	15	12	36	30	17	36	22	13	40	21	12
7	51	25	13	55	17	15	42	36	19	42	28	15	46	25	13
8	59	21	16	61	19	17	52	44	25	52	36	21	56	29	16
9	53	21	15	55	17	15	48	40	23	48	27	15	48	27	15

Overall, the WX pattern seems to be a good compromise. It outperforms the W pattern in all regards, while X patterns yield the smallest circuit depths at the price of higher gate count. We will explore by means of larger benchmark circuits, how these different characteristics of the patterns show themselves in practice.

VII. EXPERIMENTAL EVALUATION

In order to evaluate the impact of the patterns on entire Clifford+T circuits, we considered a suite of benchmarks taken from RevLib [14] and the naive qubit mapping which maps i-th qubit of the circuit to qubit Qi. All non-native CNOTs are replaced with the corresponding X, W, or WX pattern and finally the same simple gate reduction is performed in which pairs of adjacent Hadamard or CNOT gates are cancelled out.

The results for Q20 are shown in Table III. The first column denotes the benchmark name and the remaining column the overall cost and depth for the resulting circuit after performing the pattern replacement and gate reduction.

As suggested by the findings from Section VI-A, the X pattern leads to a depth reduction and cost increase as compared to the W pattern. However, in contrast to what was suggested by the results from Section VI-A, the depth when using the WX pattern is often even smaller than for the X patterns. So, overall the WX pattern outperforms the X and W pattern.

For QX5, the qualitative findings from Section VI-B were confirmed also at the larger scale.

The runtime is negligible in all cases, since the replacement of the patterns as well as the gate reduction require only two to three passes of the circuit.

VIII. CONCLUSIONS

In this paper, we explored how combinations of SWAP-based qubit movements and remote gates can be employed to reduce the transformation overhead required to transform non-native gates in a quantum circuit regarding both the number of native gates as well as the circuit depth. We discussed several possibilities to combine SWAPs and remote gates (X, W, and WX pattern) and came up with SWAP gate optimizations that become possible by the particular use of SWAP gates in the patterns and yield cost as well as depth reductions. We developed a methodology to determine optimal instantiations of the patterns for quantum computer architectures with uni-directional as well as bi-directional CNOTs. Our experimental evaluations indicate that the WX pattern, i.e. a balanced combination of SWAPs and remote gates, gives the best compromise between circuit cost and depth.

TABLE III
CIRCUIT TRANSFORMATION FOR IBM Q20

	X pattern		W pattern		WX pattern	
Benchmark	cos	dep	cos	dep	cos	dep
0410184_169	405	226	370	277	370	224
4gt4-v0_72	632	390	602	499	588	418
4gt4-v0_73	804	530	724	553	716	497
4gt4-v0_78	458	281	416	300	406	256
4gt4-v0_80	423	262	393	321	391	287
4mod5-bdd_287	176	108	159	127	163	108
alu-bdd_288	197	119	185	145	181	118
C17_204	1089	694	961	741	965	666
cm152a_212	2606	1620	2174	1445	2192	1375
cm42a_207	3478	1953	3087	2068	3123	1966
cm82a_208	1262	803	1171	915	1139	788
cnt3-5_179	382	215	332	227	338	213
con1_216	1989	1250	1747	1320	1773	1215
dc1_220	4581	2546	4273	3294	4253	2808
decod24-bdd_294	118	75	109	83	109	72
ex2_227	1243	781	1087	795	1107	735
ex3_229	969	608	896	682	890	576
f2_232	2005	1200	1799	1267	1787	1154
ham7_104	889	576	813	688	805	579
hwb5_53	2934	1821	2664	2002	2638	1758
hwb6_56	12583	7745	11517	8578	11391	7527
majority_239	1296	801	1149	840	1173	780
mini_alu_305	242	120	230	163	224	130
mod5adder_127	1378	891	1259	1010	1251	868
mod8-10_177	1179	727	1081	845	1073	728
rd53_130	2312	1423	2086	1631	2098	1428
rd53_131	920	594	868	659	854	558
rd53_133	1330	826	1278	1036	1238	832
rd53_135	546	358	516	409	500	353
rd53_138	234	153	210	155	204	144
rd53_251	2610	1615	2208	1578	2232	1474
rd53_311	542	329	458	306	460	291
rd73_140	330	188	289	190	291	175
sf_274	1498	1007	1372	1001	1314	916
sf_276	1498	1007	1372	1001	1314	916
square_root_7	13450	7715	11378	7070	11760	7086
sym6_145	7659	4951	6929	5381	6901	4760
sym6_316	566	319	482	286	486	284
sym9_146	778	479	698	540	706	470
sys6-v0_111	351	191	312	208	312	187
wim_266	2339	1290	2121	1579	2139	1355
z4_268	6241	3686	5640	4104	5600	3618

References

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[2] P. Boykin, T. Mor, M. Pulver, V. Roychowdhury, and F. Vatan, "A new universal and fault-tolerant quantum basis," *Information Processing Letters*, vol. 75, pp. 101–107, 08 2000.

[3] 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.

[4] 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.* ACM, 2019, pp. 142:1–142:6.

[5] 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*. IEEE, 2019, pp. 7–12.

[6] X. Zhou, S. Li, and Y. Feng, "Quantum circuit transformation based on simulated annealing and heuristic search," *IEEE Trans. on CAD*, pp. 1–1, 2020.

[7] 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.

[8] 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.

[9] S. Li, X. Zhou, and Y. Feng, "Qubit mapping based on subgraph isomorphism and filtered depth-limited search," *IEEE Transactions on Computers*, pp. 1–1, 2020.

[10] A. Farghadan and N. Mohammadzadeh, "Quantum circuit physical design flow for 2d nearest-neighbor architectures," *Int. J. Circuit Theory Appl.*, vol. 45, no. 7, p. 989–1000, Jul. 2017. [Online]. Available: https://doi.org/10.1002/cta.2335

[11] P. Niemann, C. Batthyopadyay, and R. Drechsler, "Combining SWAPs and remote toffoli gates in the mapping to IBM QX architectures," in *Design, Automation and Test in Europe*, 2021.

[12] IBM. IBM Q 16 Rueschlikon V1.x.x. [Online]. Available: https://github.com/Qiskit/ibmq-device-information/tree/master/backends/rueschlikon/V1

[13] B. Nash, V. Gheorghiu, and M. Mosca, "Quantum circuit optimizations for nisq architectures," *Quantum Science and Technology*, vol. 5, no. 2, p. 025010, Mar 2020.

[14] 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.