Towards Exploring the Potential of Alternative Quantum Computing Architectures

Arighna Deb*, Gerhard W. Dueck†, Robert Wille‡
*School of Electronics Engineering, KIIT University, India
†Faculty of Computer Science, University of New Brunswick, Canada
‡Institute for Integrated Circuits, Johannes Kepler University Linz, Austria
airghna.debfet@kiit.ac.in gdueck@unb.ca robert.wille@jku.at

***Abstract*—The recent advances in the physical realization of *Noisy Intermediate Scale Quantum* (NISQ) computers have motivated research on design automation that allows users to execute quantum algorithms on them. Certain physical constraints in the architectures restrict how logical qubits used to describe the algorithm can be mapped to physical qubits used to realize the corresponding functionality. Thus far, this has been addressed by inserting additional operations in order to overcome the physical constrains. However, all these approaches have taken the existing architectures as invariant and did not explore the potential of changing the quantum architecture itself—a valid option as long as the underlying physical constrains remain satisfied. In this work, we propose initial ideas to explore this potential. More precisely, we introduce several schemes for the generation of alternative coupling graphs (and, by this, quantum computing architectures) that still might be able to satisfy physical constraints but, at the same time, allow for a more efficient realization of the desired quantum functionality.**

I. Introduction

Quantum computing [1] received significant interests because of its ability to provide efficient solutions for certain complex tasks such as quantum chemistry, optimization, machine learning, cryptography, etc. Physicists experimented with various technologies such as ion-traps, superconductors, semiconductor quantum dots, or photonic systems in order to physically realize quantum computers. Among these, the superconducting technology is considered very promising since it provides better physical realizations over other candidate technologies [2]. This motivated researchers as well as companies to focus on the development of actual quantum computers [3], [4].

Herein, the approach from IBM stands out—it provided the first publicly available quantum processors. These processors can be accessed by anyone through cloud access [5]. This allows designers to run their own quantum algorithms (usually represented in terms of circuits) on the IBM quantum computers, known as IBM QX architectures. In order to execute quantum circuits on those architectures, the initial circuits have to be decomposed into elementary quantum operations that are supported by the given architecture. To this end, several solutions exists that decompose arbitrary quantum circuits into a sequence of elementary quantum gates [6]–[8].

Once the circuits are represented in a sequence of elementary quantum gates supported by the architecture, further design steps need to be conducted. This includes the mapping of logical qubits used in the originally given quantum circuit to the physical qubits used in the architecture. This, however, cannot be done in a one-to-one fashion, because IBM QX architectures have certain physical constraints described by so-called *coupling graphs*. Current state-of-the-art methods [9]–[14] insert additional gates in order to re-arrange the qubits and/or to change the control/target connections so that the constraints imposed by the coupling graphs are satisfied. Obviously, the insertion of additional gates increases the size of the quantum circuit and, thus, reduces the fidelity of the circuit. As a result, researchers and engineers focused on developing solutions that aim to derive a proper mapping of logical qubits to physical qubits while, at the same time, keeping the number of additional gates as small as possible.

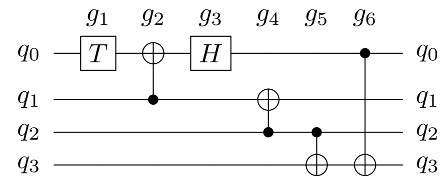

Fig. 1: Quantum circuit

However, all these approaches have taken the existing architectures as invariant and did not question the correspondingly resulting constraints. In this work, we show that there exists further potential. In fact, changing the constraints imposed by the existing quantum computing architectures is a valid option (of course, as long as the underlying physical constrains remain satisfied). In the following, we motivate that in more detail and propose initial ideas to explore the resulting potential. More precisely, we introduce several schemes for the generation of alternative coupling graphs (and, by this, quantum computing architectures) that still might be able to satisfy physical constraints but, at the same time, allow for a more efficient realization of the desired quantum functionality.

II. Quantum Circuits and Architectures

Before the general idea and the proposed schemes are introduced, we first provide a brief review on quantum circuits as well as the quantum architectures commonly used in today's NISQ devices.

Quantum bits (*qubits*) are the basic information units in quantum computation [1]. A qubit can have two basis states, $|1\rangle$ or $|0\rangle$ and can also have a superposition of both states. A quantum circuit is composed of quantum gates, where each gate represents a quantum operation. A gate can either involve one or two qubits. In the case of two-qubit quantum gates, one qubit is the target qubit and other is the control qubit. The *Clifford+T* gate library [6], [15] composed of the 1-qubit *Hadamard (H)* gate, *T* (phase shift by $\frac{\pi}{4}$) gate, and 2-qubit *controlled NOT (CNOT)* gate represents a universal gate library, i.e., all quantum operations can be performed by circuits composed of gates from this library. In order to realize an efficient quantum circuit, the total number of quantum gates in a circuit, should be kept as low as possible.

Example 1. *Fig. 1 shows an example of a quantum circuit composed of four qubits and six gates. The boxes labeled with H and T represent the single qubit gates* H *and* T*, respectively. The control and target qubits of the CNOT gates are denoted by* $\bullet$ *and* $\oplus$*, respectively.*

In order to execute a quantum circuit, they have to be mapped onto a real quantum computer. In the following, we focus on the quantum computers provided by IBM's Project Q [5]. Here, quantum algorithms to be executed (usually provided in terms of a quantum circuit) have to be composed of elementary quantum gates only. To this end, several methods decomposing the desired quantum functionality to an elementary gate library exist in literature [6]–[8]. Besides that, there are also some constraints which need to be

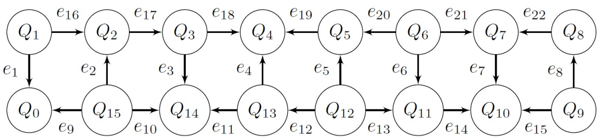

Fig. 2: Quantum architecture Rueschlikon

satisfied. In fact, 2-qubit quantum gates such as CNOT can only be applied between specific pairs of qubits. Furthermore, for each pair of qubits, which qubit will work as the control and which one will work as the target are firmly specified. This restriction is known as *CNOT-constraints*, and are usually described in terms of a *coupling graph* which depicts the layout of the quantum architecture. More formally, a coupling graph $A = (Q, E)$ over physical qubits $Q = \{Q_0, Q_1, \cdots, Q_{n-1}\}$ is a directed graph consisting of a set of vertices Q and a set of edges $E = \{(Q_u, Q_v), Q_u, Q_v \in Q, Q_u \neq Q_v\}$ representing a 2-qubit operation with the qubits Q_u and Q_v being the control and target, respectively.

Example 2. *Fig. 2 shows a coupling graph representing the restrictions of the Rueschlikon (also known as IBM QX5) architecture. As can be seen, the architecture has 16 physical qubits represented by vertices with labels Q_0 to Q_{15}. The edges e_1 to e_{22} in the graph represent the connections between the qubits. For example, edge e_1 pointing from physical qubit Q_1 to qubit Q_0 indicates that a CNOT with control qubit Q_1 and a target qubit Q_0 can be applied here. Similarly, all other edges define the other allowed qubit interactions. All remaining interactions are prohibited.*

III. Motivation and General Idea

In this section, we first briefly review the state-of-the-art process of realizing quantum functionality on real quantum computers. Afterwards, we discuss a potential that has not been utilized thus far. This provides the basis for investigations towards the generation of alternative coupling graphs that satisfy physical constraints but also allow for more efficient realizations of the desired quantum functionality.

A. Current Design Process

Thus far, the realization of quantum functionality onto real quantum computers has been conducted by simply taken the existing architectures as invariant and not questioning the correspondingly resulting constraints. This does not only yield a significantly more complex design process (in fact, realizing a given quantum functionality to a given architecture has been proven to be $\mathcal{NP}$-hard [16]), but also substantially increases the costs of the resulting realizations. This is because the given architectures substantially restrict the allowed interactions between quibts. Current state-of-the-art methods address this problem by adding additional gates which re-arrange qubits and/or change control/target connections so that they are eventually in line with the constraints imposed by the quantum architecture/coupling graph. An example illustrates the idea.

Example 3. *Consider the circuit from Fig. 1 that is to be realized on the Rueschlikon quantum computer. The constraints as defined by the coupling graph shown in Fig. 2 have to be satisfied. By directly mapping each logical qubit q_i to a physical qubit Q_i, the first three gates are supported. However, gate g_4 and gate g_6 cannot be realized under the given constraints, because an interaction between Q_2 and Q_1 is only possible if Q_2 is target and Q_1 is control (which is the opposite in g_4) and because no interaction is allowed between Q_0 and Q_3 at all (which is required in g_6), respectively.*

These issues can be addressed as follows. First, add four H gates which flip the respective control/target connections of a gate and, by this, satisfy the constraints for gate g_4 as shown in Fig. 3. Second, SWAP gates are applied which exchange two qubit values and effectively "moving" qubit values from one physical position to another. This is applied to satisfy the constraints for gate g_6 as also shown in Fig. 3. Since all these adjustments require 18 additional elementary gates (four H gates and two SWAP gates which need to be realized with seven elementary gates each), realizing this circuit onto the Rueschlikon architecture increases the gate count by a factor of 3.

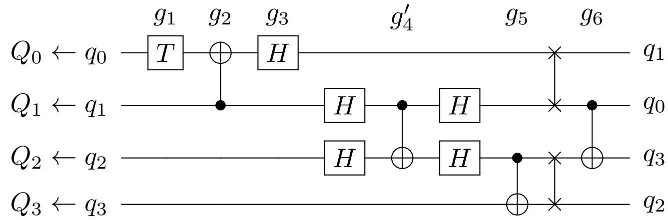

Fig. 3: Mapped circuit (assuming coupling graph from Fig. 2)

In the recent past, several methods for realizing quantum functionality under these constraints have been proposed (see e.g. [9]–[13]). They employ various heuristics, clever reordering schemes, templates, etc. Even exact solutions which guarantee a minimal overhead with respect to H/SWAP gates have been proposed (see [14]). However, all these solutions frequently yield substantial overheads in terms of a large number of additionally gates—a significant drawback since the total number of gates significantly affects the fidelity of the result. In fact, studies by IBM have shown that, if the gate overhead gets too large, the intended result cannot be determined anymore because of the noise levels are too high [17].

B. Potential Impact of Architectural Modifications

Reducing the gate overhead caused by the need to satisfy the constraints from physical realizations obviously is the main objective of solutions introduced thus far for quantum circuit realization. However, even if minimal overheads can be determined, their impact on the reliability of the resulting computations remains substantial. Hence, to further improve realizations, more avenues need to be explored. Changing the constraints imposed by the existing quantum computer architectures (and described by the coupling graphs) seems to be a promising further direction. Since those constraints resulted from physical requirements, they have been taken as invariant and were not questioned thus far. In this section, we show that, even if we recognize that physical constraints have to be satisfied, some degree of freedom exists. This allows for the design of valid alternative quantum architectures onto which certain quantum circuits can be realized with much less gate overhead than before.

The physical constraints of quantum computing technology have to be considered in more detail. We focus on the constraints of quantum superconducting (cf. [2], [18]) as a representative technology[1]. Here, each qubit is realized as an artificial atom using a non-linear inductor-capacitor circuit. The non-linear elements lead to anharmonicity which results in unequally spaced energy levels [19]. These differences in energy levels allow to address individual qubits with a different anharmonic oscillator. As a result, in a multi-qubit quantum computer, each qubit has a unique frequency [20]. In case of 2-qubit gates, the qubit with high frequency is usually used as control qubit and the qubit with low frequency is usually used as target qubit. Exceptions to this high frequency control and low frequency target arise when the qubits are degenerated

[1]Quantum superconducting is used in many quantum computers accessible today. Furthermore, the corresponding constraints are similar in other technologies. Hence, choosing it as representative allows for valid conclusions for several quantum computing technologies.

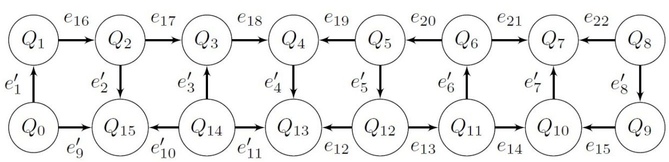

Fig. 4: Coupling graph of an alternative architecture

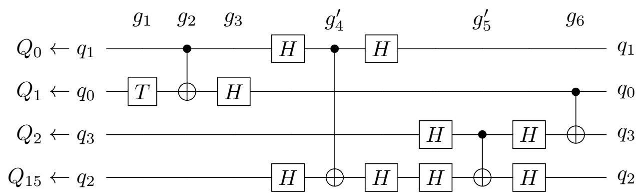

Fig. 5: Mapped circuit (assuming coupling graph from Fig. 4)

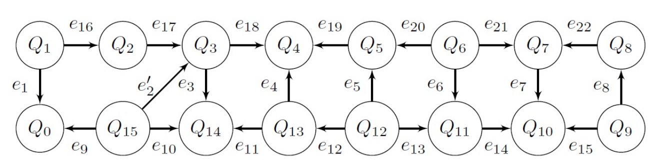

Fig. 6: Coupling graph determined by random modifications

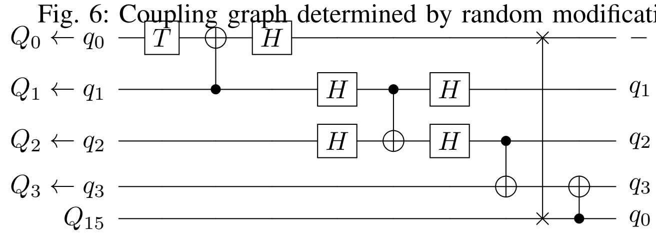

Fig. 7: Mapped circuit (assuming coupling graph from Fig. 6)

or there is an interference between coupling qubits and other qubits with low frequency.

This establishes couplings between two qubits and thus allows to perform operations on the target qubit based on the state of the control qubit [21]—eventually, realizing 2-qubit operations such as CNOT. However, such a strong coupling is only possible between two qubits and can only be established if the qubits are next to each other (otherwise, the qubits may degenerate which results in a gate operation with very low fidelity). Eventually, this led to quantum computer architectures with constraints defined by coupling graphs.

However, it is obvious that these characteristics not necessarily have to lead to quantum architectures as available thus far. In fact, a coupling between qubits that follow these characteristics can be established in numerous fashions. This allows to determine architectures with coupling graphs that are much more suited for quantum circuits to be executed on them. Again, an example illustrates the idea.

Example 4. *Fig. 4 shows the coupling graph for an alternative architecture that also satisfies the physical constraints discussed above. In fact, this coupling graph is almost identical to the coupling graph for the Rueschlikon architecture shown before in Fig. 2, but differs in the directions of the edges. Despite these minimal differences (which should not pose any obstacles with respect to a physical realization), this allows to map the quantum circuit from Fig. 1 with significatly less overhead as shown in Fig. 5. In fact, rather than 18 additionally gates, only eight additional gates are needed—an overhead reduction of 55%.*

This example sketches the possible potential in the design of quantum architectures: Rather than only satisfying physical constraints (which, of course, always remains a primary objective), it should also be considered how good/bad a derived architecture is able to realize the desired quantum functionality.

IV. Towards Exploring the Potential

In this work, we propose initial ideas towards exploring the potential sketched above. In fact, exploiting the shown potential in a naive fashion is easy. One just needs to generate alternative coupling graphs and map the respective quantum circuits to it in order to see whether this yields more efficient results as if, e.g., IBM's Rueschlikon is considered as coupling graph. However, exploring the potential using "arbitrary" coupling graphs is meaningless (in this case, a complete graph where all qubits may arbitrarily interact with each other will be the best but also physically most unrealistic solution). Hence, we consider alternative schemes for coupling graph generation that, on the one hand, allow to explore the possible potential while, on the other hand, remain as close to the characteristics of existing quantum computing architectures (and, by this, most likely will also be physically possible).

A. Flipping Edges of Existing Coupling Graphs

The first approach to generate alternative coupling graphs involves a minor modification of the existing coupling graph in order to still satisfy the physical constraints/restrictions discussed in Section III-B. We consider an existing coupling graph (such as for Rueschlikon) as a basis for generating a modified one. The modification is done by randomly reversing the directions of the edges that exist in the given coupling graphs. More precisely, given an existing coupling graph $A = (Q, E)$, we randomly choose an edge $e_{i,j} \in E$ pointing from qubit Q_i to qubit Q_j ($Q_i, Q_j \in Q$) and flip its direction which results in an edge $e_{j,i}$, now pointing from qubit Q_j to Q_i. In a similar fashion, the directions of the other edges in the graph can also be reversed. The choice of the edges to be flipped is done in a purely random fashion.

Example 5. *Consider the coupling graph for the Rueschlikon architecture as shown in Fig. 2. Applying the scheme described above may lead to an alternative coupling graph as shown in Fig. 4. As already discussed above in Example 4, this reduces the overhead by 55% from 18 to eight additional gates in case of the circuit from Fig. 1.*

B. Random Modifications

While the coupling graphs generated by the above scheme differ from the existing graphs with respect to the directions of the edges, i.e., only minor modifications are made, more substantial modifications can be made by a random approach which is proposed as second scheme to generate alternative coupling graphs. We again consider an existing coupling graph (such as for Rueschlikon) as a basis for generating an alternative one. The modification is done by randomly adding and removing edges that exist in the considered coupling graph. More precisely, given an existing coupling graph $A = (Q, E)$, we randomly select a qubit Q_i, its adjacent qubit Q_j followed by a qubit Q_k which is adjacent to Q_j ($\{Q_i, Q_j, Q_k\} \in Q$). Based on the edges with outward direction, we order the nodes as control qubit to target qubit. Without loss of generality, assume that an edge points from Q_i to Q_j, while another edge points from Q_k to Q_j resulting the order $Q_i > Q_j < Q_k$. Next we remove an existing edge between Q_i and Q_j and add an edge either pointing from Q_i to Q_k or vice-verse.

Example 6. *Consider again the coupling graph from Fig. 2. Applying the scheme sketched above, we choose qubits Q_{15}, Q_2, and Q_3 which are adjacent to each other (see Fig. 2). Now, we remove the edge pointing from Q_{15} to Q_2 and add an edge pointing from Q_{15} to Q_3. This leads to an alternative coupling graph as shown in Fig. 6. Using this coupling graph, the circuit from Fig. 1 can be realized as shown in Fig. 7. Rather than 18 additionally gates (needed in case of the Rueschlikon architecture), this requires only eleven additional gates—a reduction of the overhead by 39%.*

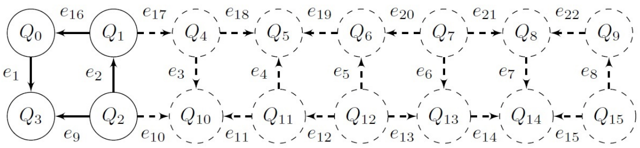

Fig. 8: Dedicated coupling graph for circuit from Fig. 1

C. Function-specific Generation with Restrictions

The schemes proposed above generate various coupling graphs and, by this, allow to evaluate the effects the coupling graphs have, in general, on the final quantum circuits. Nevertheless, in order to work towards the development of coupling graphs/architectures, schemes considering the quantum functionality to be executed in the resulting architecture can be of interest. To this end, we first determine how often which qubit pairs interact in the quantum functionality to be executed on the architecture (this can be obtained from a representative quantum circuit or defined by the designer). Based on the number of interactions between the pairs of qubits, the edges are added between the corresponding pairs, which ultimately generate a coupling graph. More precisely, for n logical qubits $\{q_0, q_1, \cdots, q_{n-1}\}$ of a given quantum circuit G, we add n physical qubits Q_i (where, $i = \{0, 1, \cdots, n-1\}$) in the coupling graph $A = (Q, E)$. Then, an edge $e_{i,j}$ is added to E with a point of direction from physical qubit Q_i to Q_j depending on the 2-qubit gate $g_l(q_i, q_j) \in G$ where, q_i and q_j denote control and target qubits respectively. In a similar manner, the other edges are applied between physical qubits based on the rest of the 2-qubit gates $g_m(c, t) \in G$ resulting in a coupling graph $A = (Q, E)$.

However, adding an edge between qubits based on every 2-qubit gates in a given circuit leads to a coupling graph where all qubits may interact with each other. According to the current physical constraints, this is unrealistic. To avoid such unrealistic coupling graph, we enforce two restrictions to the graph: (1) only one edge exists between two qubits Q_i and Q_j and (2) each qubit Q_i has an out-degree of 2. The choice of the 2 target qubits for any qubit Q_i is made depending on how frequently 2-qubit operations occur between Q_i and the corresponding target qubits. Without loss of generality, assume that the quantum gates $\{g_{l_1}(q_i, q_j), g_{l_2}(q_i, q_k), \cdots, g_{l_k}(q_i, q_m)\} \in G$ occur T_1, T_2, T_3 times ($T_1 \geq T_2 \geq T_3$) respectively, then we only add edges between qubits Q_i to Q_j and Q_i to Q_k as T_1 and T_2 are higher than that of T_3 resulting in no edge pointing from Q_i to Q_m.

Example 7. *Consider the circuit from Fig. 1 for which a coupling graph is to be generated. Applying the scheme described above, for each logical qubit q_i, we consider a physical qubit Q_i. Now, we add an edge pointing from Q_1 to Q_0 by considering the gate g_2 with q_1 as control and q_0 as target. In a similar manner, edges from Q_2 to Q_1 and to Q_3, and Q_0 to Q_3 are added according to the gates g_4, g_5 and g_6 respectively. This leads to a coupling graph depicted in Fig. 8[2]. Using this coupling graph, the circuit from Fig. 1 can be realized with no additional gates.*

Architectures described by the coupling graphs obtained from schemes proposed above most likely can be physically realized. In fact, those schemes result in coupling graphs which are heavily restricted based on the arguments discussed in Section III-B—making these architectures likely physically realizable. Since, as shown in the examples, they allow for much cheaper realizations of quantum functionality, they might be a promising alternative to existing architectures.

[2]To stay in line with the goal of generating a 16 qubit architecture, we have randomly generated the rest of the qubits Q_4 to Q_{15} which are shown with dashed circles/lines.

V. Conclusion and Further Work

In this work, we proposed initial ideas for the generation of alternative coupling graphs (i.e. quantum computing architectures) that might be able to realize quantum functionality in a more efficient fashion. The considerations may motivate physicists to develop quantum computers while not only considering physical constraints, but also taking the effect of the corresponding architectures on the quantum functionality to be executed into account. In order to provide further motivation along those lines, a more thorough evaluation of the outlined potential is left for future work.

Acknowledgments

This work has partially been supported by the LIT Secure and Correct Systems Lab funded by the State of Upper Austria.

References

[1] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge Univ. Press, 2000.

[2] M. H. Devoret and R. J. Schoelkopf. Superconducting Circuits for Quantum Information: An Outlook. *Science*, 339(6124):1169–1174, 2013.

[3] E. Pednault, J. A. Gunnels, G. Nannicini, L. Horesh, and R. Wisnieff. Leveraging Secondary Storage to Simulate Deep 54-qubit Sycamore Circuits, 2019.

[4] F. Arute et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[5] IBM Q. https://www.research.ibm.com/ibm-q.

[6] 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 of Integrated Circuits and Systems*, 32(6):818–830, 2013.

[7] D. M. Miller, R. Wille, and Z. Sasanian. Elementary quantum gate realizations for multiple-control Toffolli gates. In *Intl. Symp. on Multi-Valued Logic*, pages 288–293, 2011.

[8] R. Wille, M. Soeken, C. Otterstedt, and R. Drechsler. Improving the mapping of reversible circuits to quantum circuits using multiple target lines. In *Asia and South Pacific Design Automation Conference*, pages 145–150, 2013.

[9] IBM Quantum Information Software Kit (QISKit). https://qiskit.org/.

[10] M. Y. Siraichi, V. F. D. Santos, S. Collange, and F. M. Q. Pereira. Qubit Allocation. In *Intl. Symp. on Code Generation and Optimization*, pages 113–125, 2018.

[11] A. Matsuo, W. Hattori, and S. Yamashita. Reducing the Overhead of Mapping Quantum Circuits to IBM Q System. In *Intl. Symp. on Circuits and Systems*, pages 1–5, 2019.

[12] A. Zulehner, A. Paler, and R. Wille. An Efficient Methodology for Mapping Quantum Circuits to the IBM QX Architectures. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 38(7):1226–1236, 2019.

[13] A. Zulehner and R. Wille. Compiling SU(4) quantum circuits to IBM QX architectures. In *Asia and South Pacific Design Automation Conference*, pages 185–190, 2019.

[14] 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 Conference*, 2019.

[15] P. Selinger. Quantum circuits of T-depth one. *Physical Review A*, 87(4):042302, 2013.

[16] A. Botea, A. Kishimoto, and R. Marinescu. On the complexity of quantum circuit compilation. In *Intl Symp. on Combinatorial Search*, pages 138–142, 2018.

[17] M. Sisodia, A. Shukla, A. A. A. de Almeida, G. W. Dueck, and A. Pathak. Circuit optimization for IBM processors: A way to get higher fidelity and higher values of nonclassicality witnesses. Technical Report arXiv:1812.11602 [quant-ph], ArXiV, 2019.

[18] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I. Wang, S. Gustavsson, and W. D. Oliver. Superconducting Qubits: Current State of Play. Technical Report arXiv:1608.02792 [cs.IT], ArXiV, 2019.

[19] M. Steffen, D. P. DiVincenzo, J. M. Chow, T. N. Theis, and M. B. Ketchen. Quantum computing: An IBM perspective. *IBM Journal of Research and Development*, 55(5):13, 2011.

[20] S. S. Tannu and M. K. Qureshi. Not All Qubits Are Created Equal: A Case for Variability-Aware Policies for NISQ-Era Quantum Computers. In *Intl. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 987–999, 2019.

[21] A. D. Patterson et al. Calibration of the cross-resonance two-qubit gate between directly-coupled transmons. Technical Report arXiv:1905.05670 [quant-ph], ArXiV, 2019.