

Design Automation Challenges and Benefits of Dynamic Quantum Circuit in Present NISQ Era and Beyond

(Invited Paper)

Abhoy Kole
Cyber-Physical Systems
DFKI GmbH
Bremen, Germany
Email: abhoy.kole@dfki.de

Kamalika Datta
Institute of Computer Science
University of Bremen / DFKI GmbH
Bremen, Germany
Email: kdatta@uni-bremen.de

Rolf Drechsler
Institute of Computer Science
University of Bremen / DFKI GmbH
Bremen, Germany
Email: drechsler@uni-bremen.de

Abstract—Over the last two decades there has been immense progress in the field of quantum computing. Although today we have demonstrable quantum computers with more than 1000 qubits, researchers are still trying to show how these machines can be utilized to get substantial benefit for certain applications. We are also in the *Noisy Intermediate Scale Quantum* (NISQ) era that imposes certain restrictions in utilizing the entire physical qubit space. Implementation of various large-scale quantum algorithms suffers from limited number of available qubits and also the qubit coupling restrictions of the target quantum processor. To this end an advanced class of quantum circuits called *Dynamic Quantum Circuits* (DQC) has been proposed, which can work with very few additional qubits using various non-unitary operations (viz., active reset, mid-circuit measurement and classically controlled gate operations). This paper particularly presents various design automation challenges that exists in the current NISQ era and shows how DQC can be exploited to overcome some of the challenges.

Index Terms—Quantum Computing, Dynamic Quantum Circuit, Noisy Intermediate Scale Quantum (NISQ) Platform

1. Introduction

Quantum computing aims to solve some computationally hard problems, that are beyond the capability of traditional computers. The *Noisy Intermediate-Scale Quantum* (NISQ) era [1] marks an challenging phase in quantum computing, characterized by the advent of processors with limited number of qubits, shorter coherence time, and higher gate and measurement error rates. Such processors lack the error correction capabilities necessary for fault-tolerant quantum computation which incurs large overheads in terms of additional qubits and gates, e.g. surface code [2] to effectively mitigate errors and preserve quantum information.

Designing quantum algorithms that are robust against device errors and suitable for NISQ hardware pose a significant challenge. Algorithms like *Variational Quantum Eigensolver* (VQE) [3], *Quantum Approximate Bayesian Optimization Algorithm* (QABOA) [4], *Quantum Support*

Vector Machines (QSVM) [5], etc. have been developed over the years to run on such NISQ computers. Unlike conventional algorithms, e.g. *Grover's Search Algorithm* (GSA) [6], these algorithms leverage quantum circuits with tunable parameters, which are optimized iteratively using classical optimization techniques. As qubits are susceptible to decoherence and errors induced by environmental noise and imperfections in hardware, these algorithms must be adapted to work within the device constraints while ensuring operational correctness, reliability and scalability. Although current quantum processors already consist of a few hundred of qubits and their numbers are growing, their scalability remains a significant challenge.

There exist various technologies using which a quantum computer can be built like superconducting materials [7], ion-trap [8], photonic [9], etc. In particular, qubits built using technologies like superconducting further impose connectivity restrictions for realizing 2-qubit operations. There also exist various quantum computing toolkits and software development kits to design and simulate quantum circuits, viz. Qiskit [10], t|ket> [11], and Cirq [12]. Various works have been reported for the mapping of quantum circuits efficiently onto certain quantum architectures, e.g. [13], [14]. Currently the main thrust is towards developing efficient error correction techniques tailored to NISQ devices. This is one of the major ongoing research challenges. Some other challenges that characterize this era include designing quantum algorithms of shorter depth, involving fewer qubits and minimizing the number of gate operations.

In this context, *Dynamic Quantum Circuits* (DQCs) has emerged as a promising alternative to overcome the limitations posed by current quantum hardware, e.g. [15], [16]. As DQC has the potential to realize large circuits using small number of qubits, the most important task in hand is to develop dynamic version of the quantum algorithms. DQC of Toffoli gate and *Multiple Control Toffoli* (MCT) gate has already been addressed in some recent works [17], [18], but efficient realization of existing quantum algorithms is still an open area of research. In this regard design automation plays a crucial role in realizing the potential benefits of DQC, while also addressing the unique challenges they pose. In

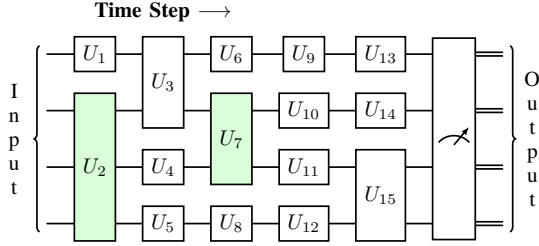


Figure 1: The circuit model representation of a quantum algorithm.

this paper we present the design automation challenges and how DQC can be utilized to mitigate some of these.

The rest of the paper is organized as follows. Section 2 presents the design automation flow and the steps required for quantum circuit compilation. Section 3 describes DQC in general and discuss about circuit clustering and iterative design in particular. Section 4 finally concludes the paper.

2. Quantum Circuit Design Automation

Typically, *quantum algorithms* are framed around a *circuit-model* (see Figure 1). In this model, data stored in quantum registers comprising of number of *qubits* undergo a series of specific and predefined quantum gate operations. The quantum measurement operations are then applied at the end to yield the final outcome, which are stored in classical registers. Considerable advancements have been made in the development of quantum compilers, e.g. Qiskit [10], t|ket [11], etc. These compilers consist of various stages (see Figure 2) to reinterpret quantum algorithms for simulating them on a *Quantum Processing Unit* (QPU).

2.1. Decomposition

Any quantum algorithm can be represented using a cascade of quantum gates. Firstly, larger abstract quantum gates are decomposed using relatively smaller gates and finally they are described using single and 2-qubit gates. Considering the case of *Multiple Control Toffoli* (MCT) gate, Figure 3 shows a 5-input version of the gate that requires two additional qubits—often termed as ancilla—for decomposing it into a network of 3-input Toffoli gates [19]. Further, a network with less Toffoli gates can be obtained when the ancilla are clean, i.e. in state $|0\rangle$. It requires more Toffoli gates when the ancilla states are arbitrary (e.g., $|\psi\rangle$) or dirty. Typically, these abstract operations are finally replaced with single- and 2-qubit gates from a specific gate library like *Clifford+T* [20]. There exist various works in the literature that introduce some improvements in decomposing larger Toffoli gates into simpler gates [21], [22], [23].

2.2. Layout Mapping

After the circuit is described using single and 2-qubit gates, the next step is to generate the correct layout map-

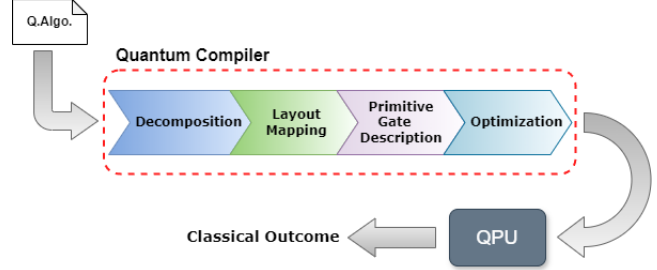


Figure 2: Typical compilation stages for rewriting quantum algorithms before executing on a QPU.

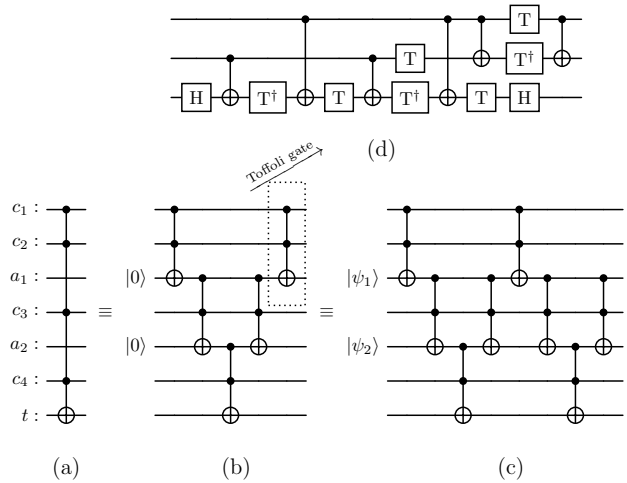


Figure 3: (a) A 5-qubit MCT gate, (b) Realization using 2 dirty ancilla, (c) Realization using 2 clean ancilla, (d) elementary gate realization of Toffoli operation.

ping with respect to some quantum architecture. Various mapping algorithms have been explored in literature mostly targeting the IBMQ machines [24], [25], [26], [27], [28], [29]. Consider the circuit shown in Figure 4(a), that we want to map into a coupling restricted qubit architecture. Figure 4(b) shows a possible mapping of the logical to physical qubits ($q_i \rightarrow Q_j$) into the 7-qubit IBM Falcon architecture. As the circuit only has 6 qubits, one of the physical qubits is remained unmapped. To simulate the circuit we require additional Swap gates. For the example circuit (see Figure 4(a)), we require two Swap gates to bring qubits q_2 and q_4 adjacent, so that the last gate can be executed. For all other gates we do not require any Swap gates as they can be directly executed using the current mapping (see Figure 4(b)). However, for mapping large quantum algorithms to some architectures we may require many Swap gates. Minimization of Swap gates is an important phase in layout mapping.

2.3. Primitive Gate Description

After the first two steps of the compilation process is over, the third step is primitive gate description. As we know

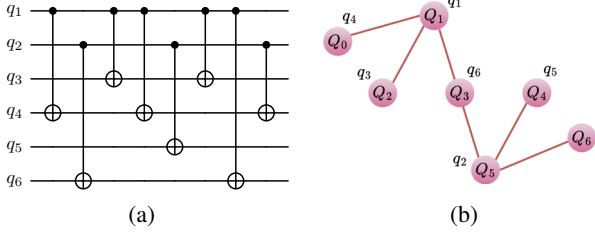


Figure 4: (a) A quantum circuit, (b) Falcon architecture.

that the development of quantum hardware platforms has witnessed notable advancements, utilizing various technologies such as superconducting materials [7], trapped-ion [8], non-linear photonic [9], etc. Specific primitive gates are supported by the QPUs built on these technologies. While the layout mapping level is independent of any underlying technology but primitive gate description is target dependent. For example, the primitive gate set supported by IBM superconducting QPUs are: rotation about Z axis (R_Z), not (X), square-root-of X (\sqrt{X}), and control- X (CX). Similarly, for simulating quantum algorithms on a IonQ QPU one has to consider the primitive gate set supported by that QPU based on ion trap technology, e.g. single-qubit rotation about X axis (R_X), rotation about Y axis (R_Y), and rotation about Z axis (R_Z), and 2-qubit rotation about XX (R_{XX}). Hence quantum circuit during this compilation stage needs to be transformed into lower level description using primitive gates supported by the targeted computing platform considered for simulation.

2.4. Optimization

The last step in the compilation process is optimization. Optimization plays a crucial role in improving all the stages, viz. (decomposition, layout mapping and primitive gate description). This step in particular tries to reduce the resources in terms of qubit count, gate count and circuit depth. For example, Qiskit [10] employs techniques like *CommutativeCancellation*, *InverseCancellation*, *Optimize1qGatesDecomposition*, *RemoveDiagonalGatesBeforeMeasure*, etc. to optimize the quantum circuit at various compilation stages. Similarly, t|ket> [11] uses approaches like *PeepholeOptimise*, *CliffordSimp*, *RemoveRedundancies*, etc. to simplify the designed circuit. Further machine learning techniques have also been employed to learn generic optimization strategies [30], [31]. Several works have targeted to optimize the decomposition of MCT gates. A recent work has also incorporated architectural level information for improving the decomposition [32]. Such optimization can be applied after each step to further reduce the circuit complexity.

2.5. Current Limitations

Over a period of time, quantum compilers have matured refining the schedules of these compilation stages [30]. But

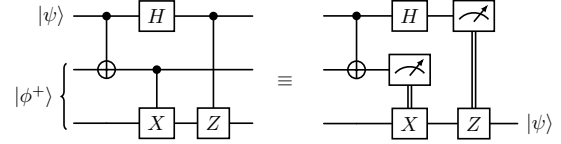


Figure 5: Realization of quantum teleportation circuit with or without the aid of mid-circuit measurement and classical controlled gate operation.

the biggest challenge is the *noise* that gets introduced in the classical outcome obtained by running a quantum algorithm on any of the available QPU (see Figure 2). This is due to the imperfection in qubits, materials used, controlling apparatus, state preparation and measurement errors, and a variety of other external factors [33]. Owing to various sources of noise, the fragility of quantum information processing leads to the exploration of different viable alternatives, e.g. enabling *Fault-Tolerant Quantum Computing* (FTQC) or designing NISQ algorithms.

The major concern is that the platform requirements for useful NISQ and FTQC are different. The qubit precision level required for implementing FTQC system seems less demanding than for executing NISQ algorithm in the quantum advantage regime [34]. But implementation of large scale FTQC system faces daunting challenges in building QPUs with myriads of qubits. While getting an answer to these problems may become feasible in the long run with the advancement of quantum fabrication technologies, design alternative that appears to be the more effective in the present-day context needs to be explored.

3. Potential of Dynamic Quantum Circuit

Dynamic Quantum Circuit (DQC) refers to the computing ability that can adapt or change during the course of a computation based on certain conditions or intermediate results. For example, the sequence of unitary operations that need to be executed for teleporting quantum information [35] may be realized either: (i) using conventional quantum gates, or (ii) using mid-circuit measurement and classical conditioned quantum operation as depicted in Figure 5. The composite state $|\phi^+\rangle$ indicates one of the Bell states or EPR pairs [36] representing the entangled quantum state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Design of later type is termed as dynamic quantum circuit that unlike the conventional or static quantum circuit may provide certain design alternatives to circumvent the major obstacles like scalability and reliability to the usage of today's limited resource QPUs.

3.1. Remote Execution

Quantum teleportation besides providing communication functionalities in the form of *Quantum Internet*, e.g. [37], enables non-local qubit interaction in a distributed computing environment as shown in the Figure 6. The qubit

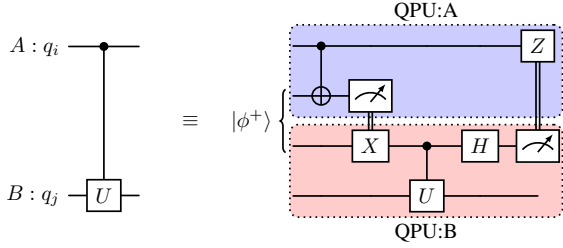


Figure 6: Realization of non-local CNOT operation in distributed computing environment using dynamic circuit.

from each QPU A and B sharing the Bell state, $|\phi^+\rangle$, which is used in non-local gate operation is often referred to as the *communication qubit* or *e-bit* [38] to distinguish it from other qubits that are specifically dedicated to quantum information processing rather than quantum communication. The execution model is called the *Local Operations and Classical Communication* (LOCC) that only requires classical communication between the two QPUs involved in remote gate operation provided both of them possess a local share of the e-bit.

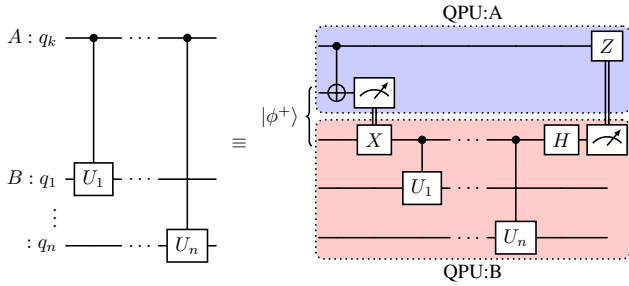


Figure 7: Realization of multiple non-local controlled gate operations using single e-bit.

The EPR-based teleportation together with the remote gate execution play a key role in distributed quantum computing. Since entanglement distribution over a quantum network is non-deterministic and incurs higher latency with respect to the coherence time of local qubits [39], efficient way of utilizing these entangled resources is essential for the success of distributed quantum computing. For example, only one e-bit is enough for implementing LOCC to realize an arbitrary sequence of non-local gate operations, provided the gates act on the same qubit from one of the QPUs as shown in Figure 7.

3.2. Circuit Clustering

Circuit clustering is one of the possible ways that allow us to simulate large quantum circuits on a QPU with fewer qubits by partitioning them into a collection of corresponding smaller subcircuits [40]. The scheme is also referred to as *circuit knitting* that with the aid of classical simulation

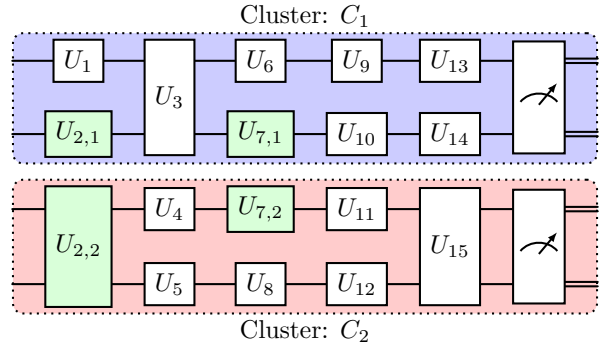


Figure 8: Clustered design of quantum circuit in corresponding centralized or distributed computing environment.

knit together the subcircuit results to obtain the desired final outcome. For an arbitrary quantum circuit, all the qubits must be grouped into clusters such that: (i) all the resulting subcircuits can fit on the targeted smaller QPUs, and (ii) only few gates may operate across the region of each subcircuit. Figure 8 shows two such clusters of the quantum circuit depicted in Figure 1 that require consideration of two non-local gate operations U_2 and U_7 .

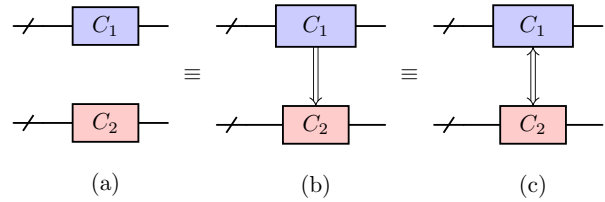


Figure 9: Communication settings for simulating non-local gates across the clusters.

For simulating non-local gates, there are three viable alternatives [41] that allow local realization of non-local gates present in subcircuits pertaining to individual clusters. Figure 9 shows an outline of the communication scenarios that can be considered for realizing non-local gates U_2 and U_7 acting on the clusters C_1 and C_2 (see Figure 8). The design alternatives are based on the type of classical communication required between the clusters for simulating non-local gates: (a) without any classical communication, (b) using one-way classical communication, and (c) using both-way classical communication between the clusters. An added advantage of the subcircuits based on the first two design alternatives is that both of them can be simulated on a QPU in some arbitrary sequence (i.e., $C_1 \Rightarrow C_2$ or $C_2 \Rightarrow C_1$ for scheme (a)) or in a specific order (i.e., $C_1 \Rightarrow C_2$ for scheme (b)). The scheme (c) is well suited for distributed computing environment comprising of at least two QPUs with LOCC enabled.

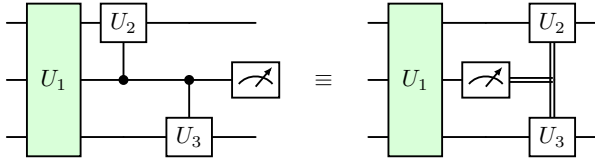


Figure 10: Commutation relationship between measurement based classical conditional and controlled quantum gates.

3.3. Iterative Design

Another major benefits of adopting this dynamic computing technique is that the number of qubits required for conventional circuit design can be minimized by *qubit recycling* [9]. According to the deferred measurement principle [42] classical conditional gate operations based on intermediate quantum measurement outcome are equivalent to controlled quantum operations as shown in the Figure 10. Once the state of a qubit is measured, it can be recycled for conducting gate operations on other qubits.

This enables simulation of large gate operation as well as quantum algorithms on a QPUs with small number of qubits. Figure 11 shows the inverse realization of 3-qubit quantum Fourier transform (QFT) by recycling a single qubit 3 times. The single qubit QFT realization allows us to implement $(n + \frac{n}{2})$ -qubit Shor’s factoring algorithm for finding factors of semiprime N where $n = \lceil 2 \log_2 N \rceil$ using $(\frac{n}{2} + 1)$ qubits [9]. Similarly, *Quantum Phase Estimation* (QPE) algorithm for finding m -bit approximation of the phase of an n -qubit operator is demonstrated using $n + 1$ qubit only [43]. Besides this, qubit recycling also make it feasible to realize an n -qubit multiple control Toffoli gate operation using only 2 or 3 qubits [18].

Of course qubit recycling minimizes the scope of parallel execution while increasing the depth of the realized circuit. But, the iterative design approach provides a way to improve the device scalability by reducing the number of required qubits for simulating quantum algorithms that demands large number of qubits for conventional design.

3.4. Design Challenges

Considering the noise characteristics of quantum computing platforms and scalability of quantum devices, dynamic circuits lay out certain design constructs like *circuit clustering* and *qubit recycling* to address these issues. It provides opportunities to re-adjust circuit depth and width depending on the computing platform opted for the simulation. Small-scale practical demonstration of dynamic circuits are also conducted for evaluating performance against their conventional realization. The benefits of dynamic circuits come with the challenges they pose for design automation. It is thus crucial to find and adapt new scalable design techniques in the existing tools considering broadly the following factors for compiling quantum circuits in order to simulate them on the targeted computing platform:

- (i) What will be the design trade-off for computing platforms ranging from single to multiple QPUs with fixed number of qubits on each QPU, restriction on 2-qubit gate operations, available quantum network and noise information?
- (ii) How efficiently do we obtain the modular realization of arbitrarily large quantum operations using small number of qubits while retaining their operational correctness?

4. Conclusion

This paper presents the design automation challenges that exist in the current NISQ era. As NISQ era put forward a crucial challenge, existing algorithms must be redesigned to be executed within the device restriction thereby ensuring reliability, correctness and scalability. In this regard DQC plays an important role. Although DQC can be exploited to minimize the qubits as well as circuit depth, it also poses an unique challenge for its design automation. In this paper we particularly emphasize on the remote execution, circuit clustering and iterative design aspects of DQC. The most important question that needs to be addressed is how arbitrary size quantum operations can be effectively realized using smaller number of qubits, thereby preserving the operational correctness. Further research in this regard will help in addressing the various design automation challenges in the current NISQ era.

References

- [1] J. Preskill, “Quantum Computing in the NISQ era and beyond,” *Quantum*, vol. 2, p. 79, Aug. 2018.
- [2] M. Takita, A. D. Córcoles, E. Magesan, B. Abdo *et al.*, “Demonstration of weight-four parity measurements in the surface code architecture,” *Phys. Rev. Lett.*, vol. 117, p. 210505, Nov 2016.
- [3] S. Liu, S.-X. Zhang, C.-Y. Hsieh, S. Zhang, and H. Yao, “Probing many-body localization by excited-state variational quantum eigensolver,” *Phys. Rev. B*, vol. 107, p. 024204, Jan 2023.
- [4] J. E. Kim and Y. Wang, “Quantum approximate bayesian optimization algorithms with two mixers and uncertainty quantification,” *IEEE Trans. Quant. Eng.*, vol. 4, pp. 1–17, 2023.
- [5] Z. Li, X. Liu, N. Xu, and J. Du, “Experimental realization of a quantum support vector machine,” *Phys. Rev. Lett.*, vol. 114, p. 140504, Apr 2015.
- [6] L. Grover, “A fast quantum mechanical algorithm for database search,” in *ACM Symp. Theory Comput.*, Jul 1996, pp. 212–219.
- [7] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, “A quantum engineer’s guide to superconducting qubits,” *Appl. Phys. Rev.*, vol. 6, no. 2, p. 021318, 06 2019.
- [8] H. Häffner, C. Roos, and R. Blatt, “Quantum computing with trapped ions,” *Physics Reports*, vol. 469, no. 4, pp. 155–203, 2008.
- [9] E. Martín-López, A. Laing, T. Lawson, R. Alvarez *et al.*, “Experimental realization of shor’s quantum factoring algorithm using qubit recycling,” *Nature Photonics*, vol. 6, no. 11, pp. 773–776, Nov 2012.
- [10] Qiskit Contributors, “Qiskit: An open-source framework for quantum computing,” 2023.
- [11] S. Sivarajah, S. Dilkes, A. Cowtan, W. Simmons, A. Edgington, and R. Duncan, “|ket>: A retargetable compiler for NISQ devices,” *Quantum Sci. Technol.*, vol. 6, no. 1, p. 014003, 2020.

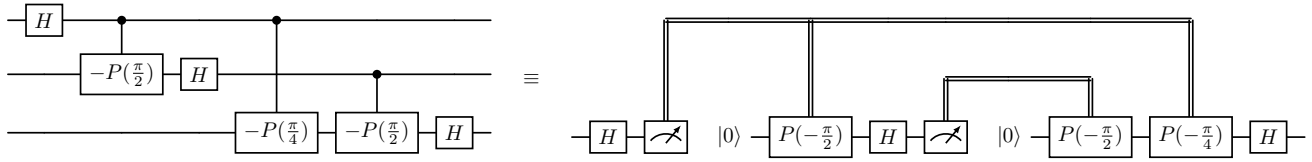


Figure 11: Dynamic circuit realization of a 3-qubit inverse QFT operator using a single qubit that is recycled 3 times.

- [12] Cirq Developers, “Cirq: A python library for writing, manipulating, optimizing and running quantum circuits,” Jul. 2023.
- [13] A. Kole, S. Hillmich, K. Datta, R. Wille, and I. Sengupta, “Improved mapping of quantum circuits to IBM QX architectures,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2375–2383, 2020.
- [14] K. Datta, A. Kole, I. Sengupta, and R. Drechsler, “Improved cost-metric for nearest neighbor mapping of quantum circuits to 2-dimensional hexagonal architecture,” in *Reversible Computation (RC)*. Springer, 2023, pp. 218–231.
- [15] P. Deliyannis, J. Sud, D. Chamaki, Z. Webb-Mack, C. W. Bauer, and B. Nachman, “Improving quantum simulation efficiency of final state radiation with dynamic quantum circuits,” *Phys. Rev. D*, vol. 106, p. 036007, Aug 2022.
- [16] E. Bäumer, V. Tripathi, D. S. Wang, P. Rall *et al.*, “Efficient long-range entanglement using dynamic circuits,” *arXiv: 2308.13065 [quant-ph]*, 2023.
- [17] A. Kole, A. Deb, K. Datta, and R. Drechsler, “Extending the design space of dynamic quantum circuits for Toffoli based network,” in *Des. Autom. Test Eur. Conf. & Exh. (DATE)*, 2023, pp. 1–6.
- [18] —, “Dynamic realization of multiple control Toffoli gate,” in *Des. Autom. Test Eur. Conf. & Exh. (DATE)*, 2024, pp. 1–6.
- [19] A. Barenco *et al.*, “Elementary gates for quantum computation,” *Phys. Rev. A*, vol. 52, no. 5, pp. 3457–3467, Nov 1995.
- [20] H. N. N.H. Nickerson, Y. Li, and S. Benjamin, “Topological quantum computing with a very noisy network and local error rates approaching one percent,” *Nature Communications*, vol. 4, p. 1756, 2013.
- [21] D. Miller, R. Wille, and Z. Sasanian, “Elementary quantum gate realizations for multiple-control Toffoli gates,” in *Int. Symp. Multiple-Valued Logic (ISMVL)*, May 2011, pp. 288–293.
- [22] 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. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 6, pp. 818–830, 2013.
- [23] A. Kole and K. Datta, “Improved NCV gate realization of arbitrary size Toffoli gates,” in *Int. Conf. VLSI Des. and Embedded Sys. (VLSID)*, 2017, pp. 289–294.
- [24] A. Zulehner and R. Wille, “Compiling SU (4) quantum circuits to IBM QX architectures,” in *Asia and South Pacific Des. Autom. Conf.*, 2019, pp. 185–190.
- [25] A. Zulehner, A. Paller, and R. Wille, “An efficient methodology for mapping quantum circuits to the IBM QX architectures,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 7, pp. 1226–1236, 2019.
- [26] G. Li, Y. Ding, and Y. Xie, “Tackling the qubit mapping problem for NISQ-era quantum devices,” in *Int. Conf. Arch. Sup. Prog. Lan. and Operating Sys. (ASPLOS)*, 2019, p. 1001–1014.
- [27] X. Zhou, S. Li, and Y. Feng, “Quantum circuit transformation based on simulated annealing and heuristic search,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 12, pp. 4683–4694, 2020.
- [28] C. Zhang, A. B. Hayes, L. Qiu, Y. Jin, Y. Chen, and E. Z. Zhang, “Time-optimal qubit mapping,” in *Int. Conf. Arch. Sup. Prog. Lan. and Operating Sys. (ASPLOS)*, 2021, p. 360–374.
- [29] A. Kole, K. Datta, P. Niemann, I. Sengupta, and R. Drechsler, “Exploiting the benefits of clean ancilla based Toffoli gate decomposition across architectures,” in *Reversible Computation (RC)*. Springer, 2023, pp. 232–244.
- [30] N. Quetschlich, L. Burgholzer, and R. Wille, “Compiler optimization for quantum computing using reinforcement learning,” in *Des. Autom. Conf.*, 2023, pp. 1–6.
- [31] T. Fösel, M. Niu, F. Marquardt, and L. Li, “Quantum circuit optimization with deep reinforcement learning,” *arXiv: 2103.07585 [quant-ph]*, 2021.
- [32] S. Sengupta, A. Kole, K. Datta, I. Sengupta, and R. Drechsler, “AQuCiDe: Architecture aware decomposition of quantum circuits,” in *Quantum Computing: Circuits, Systems, Automation and Applications*, H. Thapliyal and T. Humble, Eds. Springer, 2024, pp. 69–87.
- [33] J. M. Gambetta, J. M. Chow, and M. Steffen, “Building logical qubits in a superconducting quantum computing system,” *npj Quantum Inf.*, vol. 3, no. 1, p. 2, Jan 2017.
- [34] O. Ezratty, “Where are we heading with NISQ?” *arXiv: 2305.09518 [quant-ph]*, 2023.
- [35] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, “Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels,” *Phys. Rev. Lett.*, vol. 70, pp. 1895–1899, Mar 1993.
- [36] A. Einstein, B. Podolsky, and N. Rosen, “Can quantum-mechanical description of physical reality be considered complete?” *Phys. Rev.*, vol. 47, pp. 777–780, May 1935.
- [37] A. S. Cacciapuoti, M. Caleffi, F. Tafuri, F. S. Cataliotti *et al.*, “Quantum internet: Networking challenges in distributed quantum computing,” *IEEE Network*, vol. 34, no. 1, pp. 137–143, 2020.
- [38] J. Eisert, K. Jacobs, P. Papadopoulos, and M. B. Plenio, “Optimal local implementation of nonlocal quantum gates,” *Phys. Rev. A*, vol. 62, p. 052317, Oct 2000.
- [39] J.-Y. Wu, K. Matsui, T. Forrer, A. Soeda, P. Andrés-Martínez, D. Mills, L. Henaut, and M. Muraio, “Entanglement-efficient bipartite-distributed quantum computing,” *Quantum*, vol. 7, p. 1196, Dec. 2023.
- [40] T. Peng, A. W. Harrow, M. Ozols, and X. Wu, “Simulating large quantum circuits on a small quantum computer,” *Phys. Rev. Lett.*, vol. 125, p. 150504, Oct 2020.
- [41] C. Piveteau and D. Sutter, “Circuit knitting with classical communication,” *IEEE Trans. Inf. Theory*, vol. 70, no. 4, pp. 2734–2745, 2024.
- [42] M. Nielsen and I. Chuang, *Quantum computation and quantum information*. Cambridge Univ. Press, Oct 2000.
- [43] A. D. Córcoles, M. Takita, K. Inoue, S. Lekuch, and others., “Exploiting dynamic quantum circuits in a quantum algorithm with superconducting qubits,” *Phys. Rev. Lett.*, vol. 127, p. 100501, Aug 2021.