Project Progress Report

Seyed Sajad Kahani
22222815

February 4, 2023

1 Literature Review

The literature review is divided into three parts: general-purpose quantum compilers, specific-purpose quantum compilers for Hamiltonians, and classical problems and algorithms related to the compilation.

1.1 General Quantum Compilers

The first part, consists of papers which are directly related to quantum compilation. While most of the papers are useful for our case, we are also looking for two specific subjects in these papers.

First, we are looking for the approaches to the gate-set specially two-qubit gates through the process of compiling. Secondly, our aim is to extract the algorithms and principles for qubit allocation, and routing gates through the device.

We have analysed almost all of the papers listed below, in terms of the gate-set, but we are still looking through the papers to organize and categorize their algorithms and their corresponding problems in qubit allocation and routing.

[5] Andrew M Childs, Eddie Schoute, and Cem M Unsal. "Circuit Transformations for Quantum Architectures". In: (), p. 29.

[6] Andrew Cross et al. "OpenQASM 3: A Broader and Deeper Quantum Assembly Language". In: *ACM Transactions on Quantum Computing* 3.3 (Sept. 30, 2022), pp. 1–50. ISSN: 2643-6809, 2643-6817. DOI: 10.1145/3505636. (Visited on 11/11/2022).

[9] Toshinari Itoko et al. "Optimization of Quantum Circuit Mapping using Gate Transformation and Commutation". In: (July 2019). DOI: 10.48550/ARXIV.1907.02686. arXiv: 1907.02686 [quant-ph].

[13] Gushu Li, Yufei Ding, and Yuan Xie. "Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices". In: *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. ASPLOS '19: Architectural Support for Programming Languages and Operating Systems. Providence RI USA: ACM, Apr. 4, 2019, pp. 1001–1014. ISBN: 978-1-4503-6240-5. DOI: 10.1145/3297858.3304023. (Visited on 11/11/2022).

[15] Alexandru Paler. *On the Influence of Initial Qubit Placement During NISQ Circuit Compilation*. Jan. 30, 2019. arXiv: 1811.08985.

[16] Alexandru Paler, Alwin Zulehner, and Robert Wille. "NISQ circuit compilers: search space structure and heuristics". In: (), p. 9.

[19] Marcos Yukio Siraichi et al. "Qubit allocation". In: *Proceedings of the 2018 International Symposium on Code Generation and Optimization*. CGO '18: 16th Annual IEEE/ACM International Symposium on Code Generation and Optimization. Vienna Austria: ACM, Feb. 24, 2018, pp. 113–125. ISBN: 978-1-4503-5617-6. DOI: 10.1145/3168822. (Visited on 11/11/2022).

[21] Chi Zhang et al. "Time-optimal Qubit mapping". In: *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. ASPLOS '21: 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. Virtual USA: ACM, Apr. 19, 2021, pp. 360–374. ISBN: 978-1-4503-8317-2. DOI: 10.1145/3445814.3446706. (Visited on 11/11/2022).

[22] Xiangzhen Zhou, Sanjiang Li, and Yuan Feng. "Quantum Circuit Transformation Based on Simulated Annealing and Heuristic Search". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.12 (Dec. 2020), pp. 4683–4694. ISSN: 0278-0070, 1937-4151. DOI: 10.1109/TCAD.2020.2969647. URL: https://ieeexplore.ieee.org/document/8970267/ (visited on 11/11/2022).

[23] Alwin Zulehner, Alexandru Paler, and Robert Wille. "Efficient mapping of quantum circuits to the IBM QX architectures". In: *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE). Dresden, Germany: IEEE, Mar. 2018, pp. 1135–1138. ISBN: 978-3-9819263-0-9. DOI: 10.23919/DATE.2018.8342181. URL: http://ieeexplore.ieee.org/document/8342181/ (visited on 11/11/2022).

1.2 Hamiltonian Quantum Compilers

Compilation of Hamiltonians has its own literature, although it is not as extensive as that of general quantum compilers. Any paper related to the compilation of Hamiltonians is considered as a part of this section and is being carefully studied.

The paper [12] is being used as a baseline for our project.

[4] Earl Campbell. "Random Compiler for Fast Hamiltonian Simulation". In: *Physical Review Letters* 123.7 (Aug. 2019), p. 070503. DOI: 10.1103/PhysRevLett.123.070503. URL: https://link.aps.org/doi/10.1103/PhysRevLett.123.070503 (visited on 01/22/2023).

[12] Lingling Lao and Dan E. Browne. *2QAN: A quantum compiler for 2-local qubit Hamiltonian simulation algorithms*. Nov. 7, 2021. arXiv: 2108.02099.

1.3 Classical Compilation and Related Problems

Many of the problems in designing and implementing quantum compilers have counterparts in the classical world that are well-studied. A deep review of classical compilation is necessary for this project, as well as a search for algorithms from related fields such as resource allocation in distributed systems and scheduling in operating systems.

We have started reviewing the two main books [1] and [3], and are still looking for additional relevant papers.

[2] Mansoor Alicherry and T.V. Lakshman. "Network aware resource allocation in distributed clouds". In: *2012 Proceedings IEEE INFOCOM*. IEEE, Mar. 2012. DOI: 10.1109/infcom.2012.6195847.

2 The Aim

The objective of this project is to survey the current literature on compiling local Hamiltonians, and use this information to enhance existing approaches, such as 2QAN. In order to effectively tackle the complex compilation process, the sub-problems must be decomposed into smaller and more manageable tasks.

The results of this project will then be tested and evaluated for performance and quality, with the goal of creating a prototype implementation that can be used and further developed by other researchers. It should be noted that the focus will be on exploring solutions rather than creating a polished and highly-engineered implementation.

Note that this objective, is carried on from the beginning of the project, and it is still valid by this time.

3 Results

Besides all the insights and the knowledge that we have gained from the literature review, we have also a few ideas that can result in improving the existing compilers.

The main idea which is developed pretty well, is related to the routing problem.

When we have a two-qubit gate, (for example, a CNOT gate), and it shall be applied between two qubits that their corresponding qubits on the device

are not connected, we will use a chain of SWAP operators to move the qubits and make them adjacent to each other, and then apply the gate. This is a very common approach [**cowtan2019**].

Here we introduce a possible simplification of this approach, using simplified bridge gates, instead of using a chain of SWAP operators.

3.1 Bridge Gate

For a simple case, that we have three qubits, called a, b, c, and we want to apply a CNOT gate on (a, c), but the connectivity only allows us to apply a CNOT gate on (a, b) and (b, c), the first solution would be to use a SWAP gate, which is shown in figure 3.1 (7 gates, depth of 7) While another approach is to use a bridge gate, which is shown in figure 3.1 (4 gates, depth of 4). [**shende2006**]

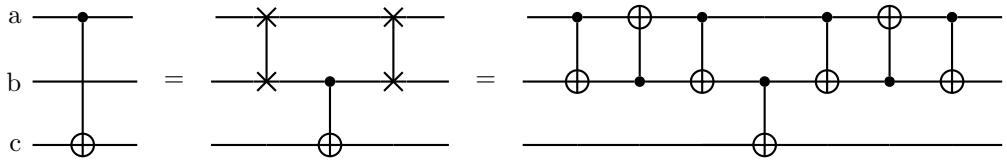


Figure 1: Applying a CNOT gate on (a, c) using a SWAP gate

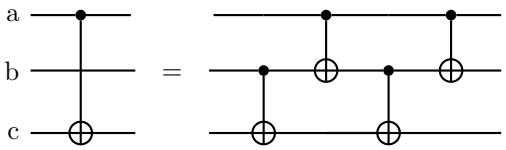


Figure 2: Applying a CNOT gate on (a, c) using a bridge gate

The bridge gate is already generalized using a recursive approach [**shende2006**], with $4n + O(1)$ depth and $4n + O(1)$ gates complexity. (where n is the number of qubits in the chain). Here we present a simplified version of the bridge gate, with $n + O(1)$ depth and $4n + O(1)$ gates complexity.

Note that a simple swapping approach will have $3n + O(1)$ depth and $6n + O(1)$ gates complexity.

Definition 1. *Simplified Bridge Gate We can define a simplified bridge gate, recursively, as follows:*

$$\begin{aligned}\text{Bridge}_n(1,n) =& \text{CNOT}(2,1)\text{CNOT}(1,2)\text{CNOT}(n,n-1)\text{CNOT}(n-1,n) \\ & \text{Bridge}_{n-1}(2,n-1) \\ & \text{CNOT}(2,1)\text{CNOT}(1,2)\text{CNOT}(n,n-1)\text{CNOT}(n-1,n)\end{aligned} \tag{1}$$

where Bridge_2 *is defined as fig. 3.1 and* Bridge_1 *is just a CNOT.*

We ignore to show that this gate is equivalent to the recursive definition of the bridge gate, since it is not the main focus of the progress report.

It is easy to show that such circuit contains $4n + O(1)$ gates, but in order to show the depth, we need to use the fact that two CNOTs applying on the same gates, commute with each other, therefore, like for the case of even ns, we can show the recursive formula is equivalent to the following:

$$\begin{aligned}\text{Bridge}_n(1,n) = &\prod_{i=1}^{n/2-1} \text{CNOT}(i+1,i)\text{CNOT}(n-i+1,n-i) \\ &\prod_{i=1}^{n/2-1} \text{CNOT}(i,i+1)\text{CNOT}(n-i,n-i+1) \\ &\text{Bridge}_1(n/2-1,n/2) \\ &\left(\prod_{i=1}^{n/2-1} \text{CNOT}(i+1,i)\text{CNOT}(n-i+1,n-i)\right)^\dagger \\ &\left(\prod_{i=1}^{n/2-1} \text{CNOT}(i,i+1)\text{CNOT}(n-i,n-i+1)\right)^\dagger\end{aligned} \tag{2}$$

where each term of each products in the RHS and the LHS can be applied simultaneously (4 gates at a time). Therefore, the depth of the circuit is $n+O(1)$. Hereby we can see an example of the bridge gate for $n = 6$ in fig. 3.

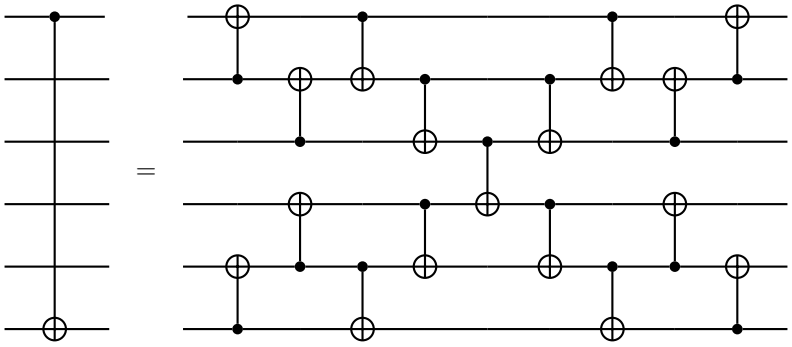


Figure 3: The bridge gate for $n = 6$

Furthermore, we can use the same idea and create a bridge gate to express $e^{iZ^1X^n}$ using CNOTs and $e^{iZ^1X^2}$ gates. Then we are working on the most general way to bridge a qubit gate between two qubits in a chain.

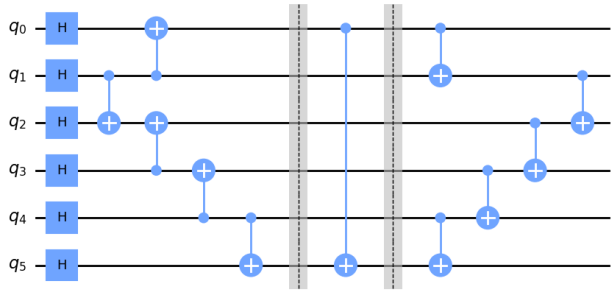
It is also proved that any effort to decompose a CNOT into a chain of CNOTs, cannot result in any complexity better than the simplified bridge gate. Therefore, the simplified bridge gate is the best possible gate to bridge a qubit gate between two qubits in a chain.

As far as we know, the family of bridge gates (the previous one or the simplified one) is not implemented (explicitly or implicitly) in any quantum compiler. It can be easily shown that for a simple case like the one in the figure **??** that we need to swap and return the qubits, using bridge gate can reduce both depth and number of gates.

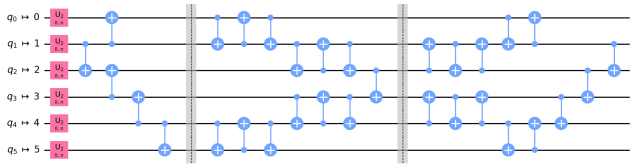
4 Challenges

- Interdisciplinary work, a wide range of keywords, a wide range of languages in the papers.
- The subtle line between theoretical works and applicability

a)



b)



c)

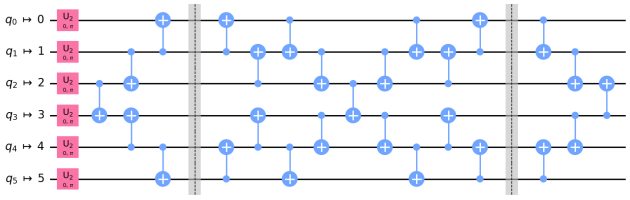


Figure 4: a) The original circuit, consisting of some local operations, then a far away CNOT and then some local operations. b) The circuit after transpiling using Qiskit. c) The circuit after transpiling using bridge gate as an intermediate gate.