Project Progress Report

Seyed Sajad Kahani
22222815

January 23, 2023

1 Literature Review

1.1 General Quantum Compilers

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

[4] 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).

[6] 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].

[9] 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).

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

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

[15] 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).

[17] 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).

[18] 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).

[19] 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

[2] 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).

[8] 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 Problems in Compilation and Resource Allocation

[1] 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

To design and implement a problem-specific container, based on 2QAN, to be

- designed to use, not just a research project.
- made upon all previous efforts rather than just sticking to an specific idea.

- implemented based on the principles of compiler design.

3 Results

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).

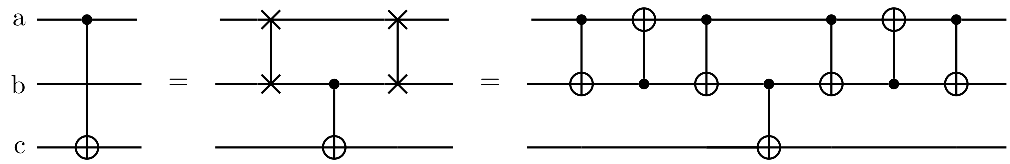


Figure 1: Applying a CNOT gate on (a, c) using a SWAP gate

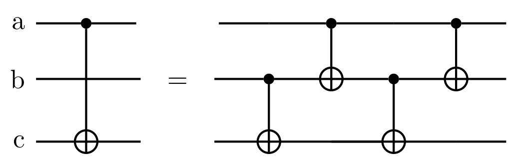


Figure 2: Applying a CNOT gate on (a, c) using a bridge gate

This example, could be generalized and the bridge gate must be generalized as well. Hereby we define the generalized version of the bridge gate and we show the optimality of the bridge gate in terms of the number of CNOT gates and the depth of the circuit.

Definition 1. *Generalized Bridge Gate For an even n*

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

but for an odd n, it is a little bit different.

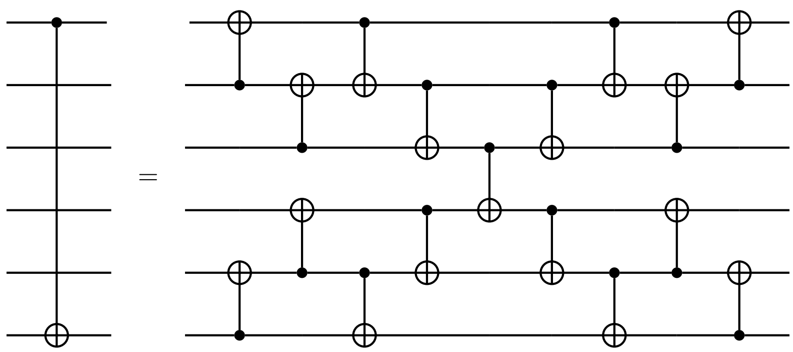


Figure 3: The bridge gate for $n = 6$

Theorem 1. *Assume that we decomposed a CNOT between the first and the last qubits in a chain, in terms of local circuit. This circuit must carry information from the first bit to the last and reverse. For the forward information flow, the information is an X gate placed on the first qubit, will travel along the circuit, so we need to have a consequential n CNOT gates to carry this information. The same argument is valid for a backward series of gates. Ignoring the first and the last qubit, for each qubit this argument is also valid that, the gate in those series must not be the first neither the last CNOT gate applied on the qubit. Therefore, at least we need to have two series, each has to have a range of CNOT gates, before and after. One could easily manage to find out that the simplest structure would be similar (in terms of complexity) to the proposed bridge gate.*

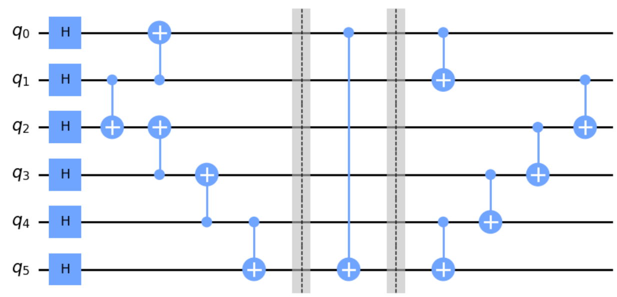
As far as we know, the family of bridge gates with more than one qubit in the chain is not studied yet, and neither implemented in any quantum compiler.

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.

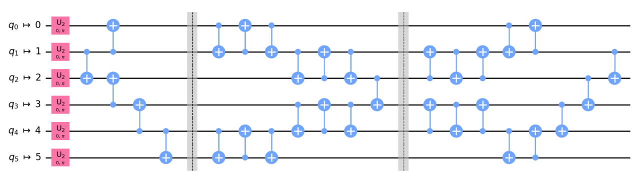
4 Challenges

- 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)

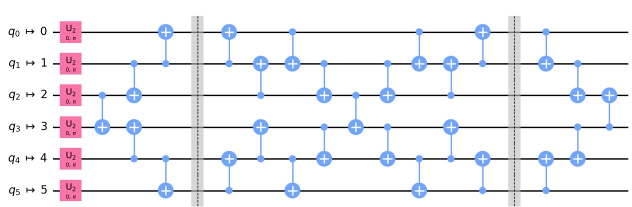


Figure 4: a) The original circuit, consisting of some local operations, then a far away CNOT and then some local operations. b) The circuit after transpiling using Qiskit. c) The circuit after transpiling using bridge gate as an intermediate gate.