Optimization and Synthesis of Quantum Circuits with Global Gates

Alejandro Villoria Henning Basold Alfons Laarman
Leiden Institute of Advanced Computer Science, The Netherlands
{a.d.villoria.gonzalez, h.basold, a.w.laarman}@liacs.leidenuniv.nl

Extended abstract

Physical realizations of quantum computers in the current era greatly vary in multiple, non-trivial aspects. A fundamental difference between two hardware realizations of a quantum computer can be the choice of implementation of their qubits. Current candidates for qubit implementations are superconducting circuits [10], trapped ions [7], nitrogen-vacancy centers [8], and neutral atoms [9], to name a few. Each choice of qubit implementation comes with distinct features that need to be considered when compiling and synthesizing circuits for a quantum computer based on it. For example, the connectivity between qubits in a superconducting quantum computer is limited to a connectivity graph that depends on a specific topology, while on a trapped ion quantum computer the connectivity is all-to-all. This all-to-all connectivity naturally gives us a special set of quantum gates to work with, in which we have access to powerful multi-qubit entangling gates that we refer to as *global* gates. If we have a quantum circuit described on an arbitrary gate set that we want to run on a trapped ion quantum computer, we need a way to synthesize it such that we take advantage of the global gates we have available. Recent advances [13] have shown that we can do quantum circuit synthesis with the ZX-calculus by restricting ourselves to a target gate set when doing a process called *circuit extraction*. In this work we aim to combine both ideas.

Contributions. We propose a method for converting any input quantum circuit into one using the gate set native to ion trap quantum platforms. Our synthesis algorithm is based on the ZX-calculus and consists of a circuit extraction procedure that outputs a circuit with global entangling gates and arbitrary single-qubit gates. In the process, we aim to minimize the number of global gates we generate, which are powerful but more costly than single-qubit operations. We develop a Python implementation of our algorithm and run benchmarks over different classes of quantum circuits.

We assume the native gate set of an ion trap quantum computer to be arbitrary single-qubit gates and global (sometimes referred to as multi-qubit) entangling gates [14, 2]. These global entangling gates consist of a set of 2-qubit Mølmer-Sørensen (MS) XX interactions between any subset of qubits in the system. We define the MS gate on qubits i and j as the matrix exponential $XX(\alpha)_{i,j} = e^{-i\frac{\alpha}{2}X_iX_j}$, for $\alpha \in [0, 2\pi)$ and X_iX_j the Pauli X on qubits i and j. Global MS (GMS) interactions are defined as

$$\mathrm{GMS}(\alpha)_S = \prod_{i<j\in S} XX(\alpha)_{i,j} \tag{1}$$

for $S \subseteq Q$ a subset of the available qubits. Note that we can (mostly) use global XX and ZZ gates interchangeably, since we can define $ZZ(\alpha)_{i,j} := H_iH_j\, XX(\alpha)_{i,j}\, H_iH_j$ and $\mathrm{GZZ}(\alpha)_S := H_S\ \mathrm{GMS}(\alpha)_S\, H_S$ where H_S denotes applying a Hadamard gate on all qubits in S.

Global entangling gates such as the GMS have been used to more efficiently synthesize layers of CNOT gates when the CNOTs are arranged in so-called *fanout* gates [12]. A fanout gate consists of a

Submitted to:
IWQC 2024

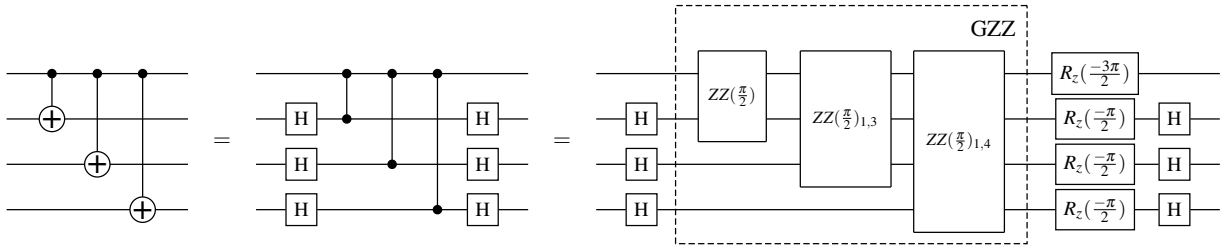
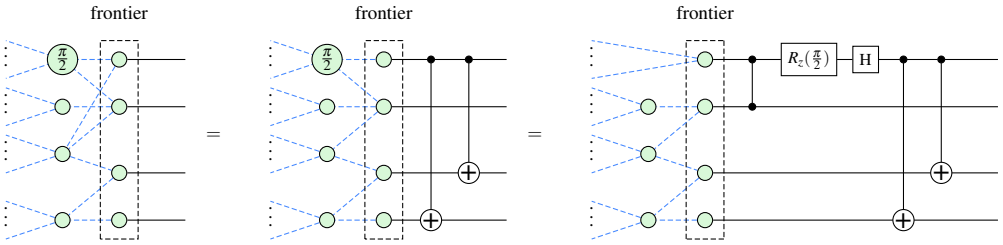


Figure 1: Implementing a fanout gate with one GZZ and single-qubit gates.

layer of CNOT gates that have a common control qubit, and can be implemented with just a single GZZ gate (see Figure 1) and some additional single-qubit gates. As we will see later, we can also potentially gather multiple fanout gates into the same GZZ gate. By extracting layers of CNOTs in the shape of fanout gates during the ZX-calculus circuit extraction process, we aim to gather as many CNOTs as we can into the least amount of global gates.

We are now going to briefly comment on how circuit optimization and extraction works in the ZX-calculus [1, 6], and how we use it to synthesize and optimize circuits with global gates. We omit here the basic definitions and equational rules of the ZX-calculus, which can be found in Appendix A, and explained thoroughly in Refs. [3, 16]. The ZX-calculus is a formal diagrammatic framework useful for reasoning about quantum mechanics [4] but also as a tool for circuit optimization [11, 1] and synthesis [5, 13]. The process of optimizing a circuit using the ZX-calculus usually involves transforming a quantum circuit as a ZX-diagram, rewriting the diagram in graph-like form and then applying a series of rewrite rules that decrease the size (number of vertices) of the diagram until they cannot be applied anymore [6]. Once this process is finished, we have a reduced ZX-diagram that implements the same unitary that we were given as a quantum circuit. To get a quantum circuit back, we are required to perform circuit extraction on our reduced ZX-diagram. Circuit extraction usually involves constructing a quantum circuit from right to left by iteratively extracting quantum gates from the "frontier" of the ZX-diagram. When we extract gates from the frontier, we progressively advance the frontier until reaching the end, which means that we have finished the extraction. For instance, two rounds of extraction in which we extract CNOT gates and then single-qubit gates and a CZ would look as follows (in which dashed blue lines represent edges with a Hadamard on them):



If we calculate the adjacency matrix M between the frontier vertices and their neighbours to the left at any given time, extracting a CNOT gate with control qubit i and target qubit j equates to performing the row operation $r_i \leftarrow r_i \oplus r_j$, for r_i, r_j rows in M and $\oplus$ addition modulo 2. These CNOTs can be added to the quantum circuit we are building as a Gaussian Elimination process until we make one or more frontier vertices have only one neighbour (equivalent to a row in M with all zero entries except for a

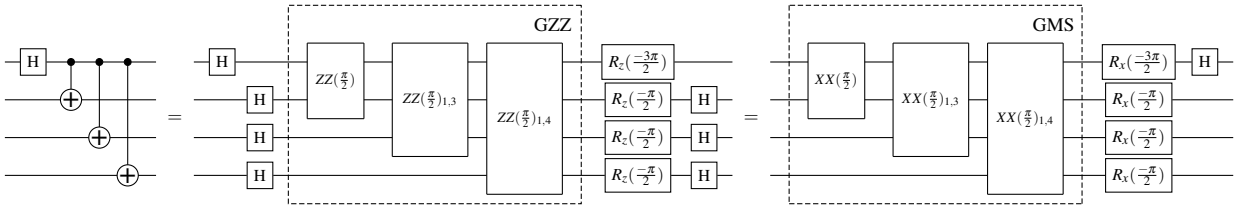


Figure 2: A better extraction of a fanout gate followed by a Hadamard.

single 1), in which case we can move the frontier vertex to the right, extracting a Hadamard gate with it unless it is the last vertex.

The main observation for our work is that the shape of the layer of CNOT gates we are extracting can be made into a sequence of fanout gates and that, under certain conditions that we will see later, some of these fanout gates can be put together into the same GMS gate. We make use of the fact that, once we extract a vertex after applying a fanout gate to the circuit, we will have a Hadamard in front of the fanout gate that we can commute through, resulting in the equations of Figure 2. If the extracted vertex is the last one on that row meaning that no Hadamard will be present, we can always add a redundancy of two Hadamards and commute one of them.

This gives us a series of $XX(\frac{\pi}{2})$ and $R_x(-\frac{\pi}{2})$ rotations and a Hadamard gate for each extracted vertex. The objective is to combine as many $XX(\frac{\pi}{2})$ rotations as possible that originate from different fanout gates into the least amount of GMS gates. We develop two algorithms to achieve this, which mostly differ in how vertices are chosen for extraction. Both algorithms are in the process of being implemented and tested against each other and other synthesis tools at the time of writing.

Algorithm 1: Max-vertices. Here we aim to maximize the number of vertices that we extract at once for each time we get a new frontier, while trying to fit their corresponding fanout gates in the least amount of GMS gates. The intuition behind it is that extracted single-qubit or CZ gates get in the way when creating GMS gates, so we try to minimize the number of times we extract anything that is not a GMS. Steps are as follows.

1. Calculate the biadjacency matrix between our current frontier and its neighbours, together with the fanout gates necessary to extract every vertex that can be extracted.

2. Construct the GMS by taking the vertices with the smallest (in the number of CNOTs) fanout gates that can fit into the same GMS. A fanout gate fits into the GMS we are currently constructing if the control qubit involved in the fanout has not been used yet, and the qubits used as targets have not been used yet or used only as targets in the previously added fanouts. If no fanouts fit in the GMS we are currently building, start building a new GMS.

3. Once decided which fanouts will get added to the GMS, we extract them together with the Hadamard gates that we will then commute as in Figure 2. Single-qubit R_x gates are also merged together whenever possible.

4. Extract single-qubit gates and CZ gates from the frontier, if any. Any qubit that had an R_z, Hadamard, or CZ gate extracted cannot be used anymore in the GMS that is currently being built, while qubits with extracted R_x gates can still be taken into account since we can commute the R_x through the XX rotations of the current GMS. If there are still extractable vertices go back to **Step 1**, otherwise we are done.

Algorithm 2: Greedy selection. The second algorithm works similarly to **Max-vertices**, with the difference that in **Step 2** we only extract the vertex with the smallest fanout gate that fits into the current GMS or create a new one if none of them fit. The motivation behind this is that early experiments show that sometimes we only need a single CNOT to extract a vertex, so there can be some advantage in moving the frontier as frequently as possible in search of vertices that only require one CNOT to be extracted.

With this work, we expand on the methods for quantum circuit synthesis with the ZX-calculus. The target hardware in mind is that of ion trap quantum computers, which are a promising technology due to their all-to-all connectivity between qubits, a property that not all quantum platforms based on other qubit implementations enjoy. We turn an input quantum circuit into a ZX-diagram and perform circuit optimization [6] on it. Then, we apply our synthesis algorithm directly on the reduced ZX-diagram. This algorithm also aims to minimize the number of global gates necessary to implement the original circuit. The entire process converts a circuit with an arbitrary gate set into one that uses GMS gates and arbitrary single-qubit unitaries.

References

[1] Miriam Backens, Hector Miller-Bakewell, Giovanni de Felice, Leo Lobski & John van de Wetering (2021): *There and back again: A circuit extraction tale*. *Quantum* 5, p. 421, doi:10.22331/q-2021-03-25-421. Available at `http://arxiv.org/abs/2003.01664`. ArXiv:2003.01664 [quant-ph].

[2] Pascal Baßler, Matthias Zipper, Christopher Cedzich, Markus Heinrich, Patrick H. Huber, Michael Johanning & Martin Kliesch (2023): *Synthesis of and compilation with time-optimal multi-qubit gates*. *Quantum* 7, p. 984, doi:10.22331/q-2023-04-20-984. Available at `http://arxiv.org/abs/2206.06387`. ArXiv:2206.06387 [quant-ph].

[3] Bob Coecke & Ross Duncan (2011): *Interacting Quantum Observables: Categorical Algebra and Diagrammatics*. *New Journal of Physics* 13(4), p. 043016, doi:10.1088/1367-2630/13/4/043016. Available at `http://arxiv.org/abs/0906.4725`. ArXiv:0906.4725 [quant-ph].

[4] Bob Coecke & Aleks Kissinger (2017): *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press.

[5] Alexander Cowtan, Silas Dilkes, Ross Duncan, Will Simmons & Seyon Sivarajah (2020): *Phase Gadget Synthesis for Shallow Circuits*. *Electronic Proceedings in Theoretical Computer Science* 318, pp. 213–228, doi:10.4204/EPTCS.318.13. Available at `http://arxiv.org/abs/1906.01734v2`.

[6] Ross Duncan, Aleks Kissinger, Simon Perdrix & John van de Wetering (2020): *Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus*. *Quantum* 4, p. 279, doi:10.22331/q-2020-06-04-279. Available at `http://arxiv.org/abs/1902.03178`. ArXiv:1902.03178 [quant-ph].

[7] C. Figgatt, A. Ostrander, N. M. Linke, K. A. Landsman, D. Zhu, D. Maslov & C. Monroe (2019): *Parallel entangling operations on a universal ion-trap quantum computer*. *Nature* 572(7769), pp. 368–372, doi:10.1038/s41586-019-1427-5. Available at `https://www.nature.com/articles/s41586-019-1427-5`. Publisher: Nature Publishing Group.

[8] Michal Gulka, Daniel Wirtitsch, Viktor Ivády, Jelle Vodnik, Jaroslav Hruby, Goele Magchiels, Emilie Bourgeois, Adam Gali, Michael Trupke & Milos Nesladek (2021): *Room-temperature control and electrical readout of individual nitrogen-vacancy nuclear spins*. *Nature Communications* 12(1), p. 4421, doi:10.1038/s41467-021-24494-x. Available at `https://www.nature.com/articles/s41467-021-24494-x`. Publisher: Nature Publishing Group.

[9] D. Jaksch J. I. Cirac H.-J. Briegel, T. Calarco & P. Zoller (2000): *Quantum computing with neutral atoms*. *Journal of Modern Optics* 47(2-3), pp. 415–451, doi:10.1080/09500340008244052. arXiv:https://www.tandfonline.com/doi/pdf/10.1080/09500340008244052.

[10] Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout van den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme & Abhinav Kandala (2023): *Evidence for the utility of quantum computing before fault tolerance*. Nature 618(7965), pp. 500–505, doi:10.1038/s41586-023-06096-3. Available at https://www.nature.com/articles/s41586-023-06096-3. Publisher: Nature Publishing Group.

[11] Aleks Kissinger & John van de Wetering (2020): *Reducing the number of non-Clifford gates in quantum circuits*. Physical Review A 102(2), p. 022406, doi:10.1103/PhysRevA.102.022406. Available at https://link.aps.org/doi/10.1103/PhysRevA.102.022406. Publisher: American Physical Society.

[12] Dmitri Maslov & Yunseong Nam (2018): *Use of global interactions in efficient quantum circuit constructions*. New Journal of Physics 20(3), p. 033018, doi:10.1088/1367-2630/aaa398. Available at http://arxiv.org/abs/1707.06356. ArXiv:1707.06356 [quant-ph].

[13] Korbinian Staudacher, Ludwig Schmid, Johannes Zeiher, Robert Wille & Dieter Kranzlmüller (2024): *Multi-controlled Phase Gate Synthesis with ZX-calculus applied to Neutral Atom Hardware*. Available at http://arxiv.org/abs/2403.10864. ArXiv:2403.10864 [quant-ph].

[14] John Van De Wetering (2021): *Constructing quantum circuits with global gates*. New Journal of Physics 23(4), p. 043015, doi:10.1088/1367-2630/abf1b3. Available at https://iopscience.iop.org/article/10.1088/1367-2630/abf1b3.

[15] Renaud Vilmart (2018): *A Near-Optimal Axiomatisation of ZX-Calculus for Pure Qubit Quantum Mechanics*. Available at http://arxiv.org/abs/1812.09114. ArXiv:1812.09114 [quant-ph].

[16] John van de Wetering (2020): *ZX-calculus for the working quantum computer scientist*. Available at http://arxiv.org/abs/2012.13966. ArXiv:2012.13966 [quant-ph].

A ZX-calculus

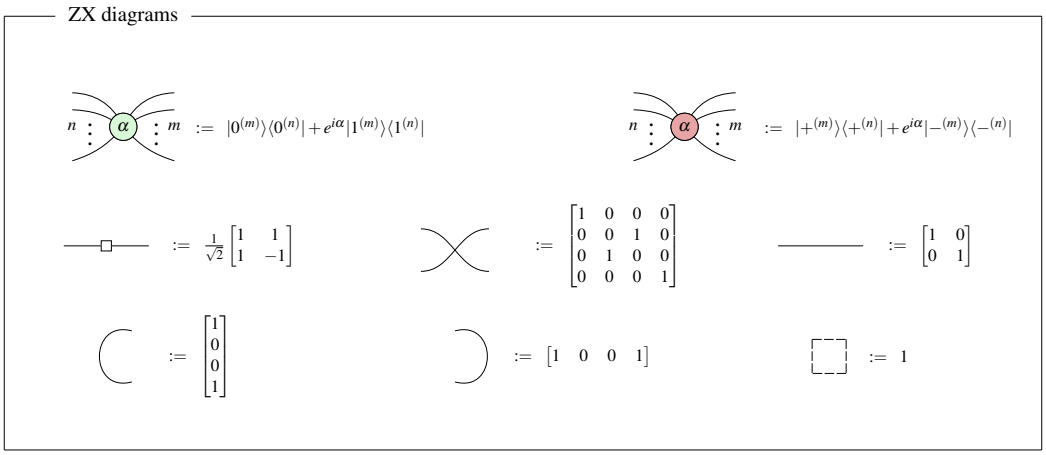


Figure 3: The generators of the ZX-calculus and their definitions, with $\alpha \in \mathbb{R}$ and $|x^{(n)}\rangle$ being shorthand for $|x\rangle \otimes \cdots \otimes |x\rangle$ done n times.

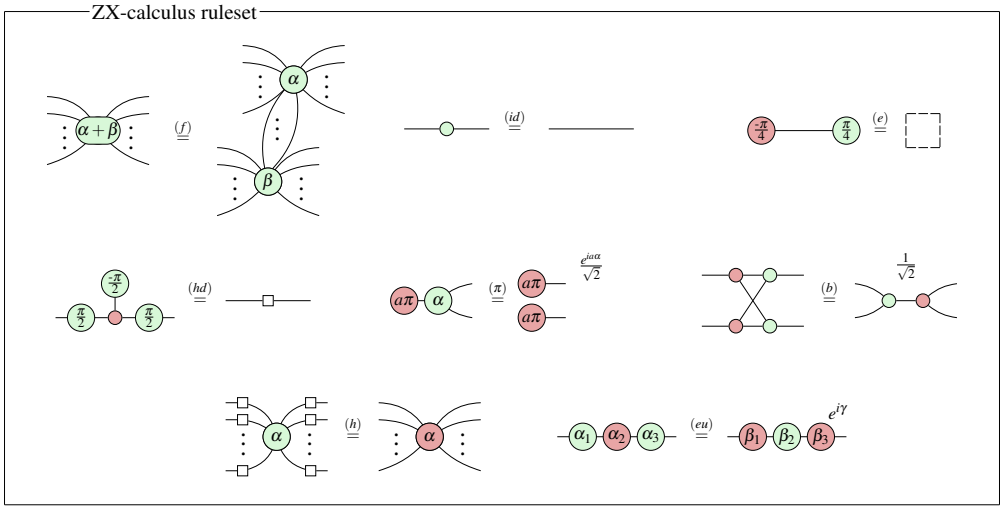


Figure 4: ZX-calculus ruleset. We omit the relationship between the phases in the (eu) rule. For more information we refer to [15].