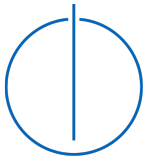
FRAUNHOFER INSTITUT (AISEC)

TECHNISCHE UNIVERSITÄT MÜNCHEN

Seminararbeit

ZX-Calculus

Author: Manuel Lerchner
Submission Date: June 30, 2023



CONTENTS

ZX-Calculus

Manuel Lerchner
Technical University of Munich
Munich, Germany

***Abstract*—ZX-Calculus is a graphical language which extends classical quantum circuits by splitting up logic gates into even smaller building blocks. Those building blocks are called spiders and are represented by colored nodes in a graph. Together with edges connecting those nodes, they form a ZX-diagram. Using a set of rewrite rules, ZX-diagrams can be transformed into each other. This allows for a more intuitive way of reasoning about the optimization of quantum circuits, as there are fewer rewrite rules to remember than in the classical logic gate model. In this paper, I will introduce the ZX-Calculus and its rewrite rules, as well as some of its applications. I will give a special focus on the optimization of quantum circuits, as this is one of the main applications of the ZX-Calculus.**

***Index Terms*—quantum computing, ZX-Calculus, quantum circuits, circuit optimization**

I. Introduction

In the last few decades, quantum computing has become a very active field of research. The main reason for this is the fact that quantum computers promise to outperform classical computers in certain tasks. The most famous example of this is Shor's algorithm [8] which can factorize large numbers in polynomial time. This is a problem that is believed to be intractable for classical computers. Another example is Grover's algorithm [3] which can search an unsorted database in $\mathcal{O}(\sqrt{N})$ time. This is a quadratic speedup compared to the classical $\mathcal{O}(N)$ time.

Usually those algorithms described in a very high-level "language", the so called (quantum-) circuit model. This allows for a very intuitive way reasoning about quantum algorithms, because every unitary operation applied to the quantum state can be compactly represented by a single gate. However, this model does not take the restrictions of real quantum computers into account. [9]

A. Quantum Circuit Compilation

Real quantum computers come with a set of restrictions. For example, the set of gates is typically limited to a small set of universal gates (e.g. the Clifford+T gate set). Furthermore, the connectivity of the qubits is limited. This means that not every operation can be applied to every pair of qubits. As a consequence, the original circuits needs to be *compiled* into a circuit that can be executed on the specific quantum computer. This process is called *quantum circuit compilation*.

In circuit 1 the Toffoli gate is shown. However it is not represented in the Clifford+T gate set. In order to execute this circuit on a real quantum computer, we need to transform it into the Clifford+T gate set first, as this allows an efficient, and fault-tolerant implementation using surface code error correction. [5]

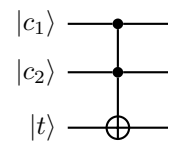


Fig. 1. The Toffoli gate.

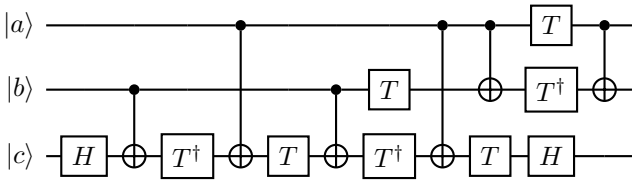


Fig. 2. Decomposition of the Toffoli gate in the Clifford+T gate set.

B. Quantum Circuit Optimization

Such a decomposition is shown in figure 2. Notice how the amount of gates increases significantly. This is a common problem in quantum circuit compilation. It means that the compiled circuit will be slower and therefore the execution time can exceed the coherence time of the qubits, and thus making the computation useless. [7]

This is where Circuit Optimizers come into play. They try to reduce the amount of gates in a circuit. This can be done by applying a set of rewrite rules to the circuit. The goal is to find a circuit that is equivalent to the original circuit, but has less gates. An important metric for simplifying quantum circuits is the so called *T-Count*. The T-Count is the amount of T gates in a circuit. Since the T-Gate is a non-Clifford gate it is very expensive to simulate and requires order of magnitudes more resources than the other clifford gates. [5]

This process is called *quantum circuit optimization*.

C. Classical Circuit Optimization

There are many different approaches to quantum circuit optimization. The most basic approach is to apply a set of rewrite rules directly to the logic-gate representation of the circuit. This approach is called *gate-level optimization*. [6]

However, this approach is typically very inefficient, as there exists a huge amount of possible rewrite rules (Some of them are shown in figure 3). Furthermore, rewrite rules are typically not independent of each other. This means that applying a rewrite rule can introduce new opportunities for other rewrite rules. This makes it very hard to find an optimal solution. [4]

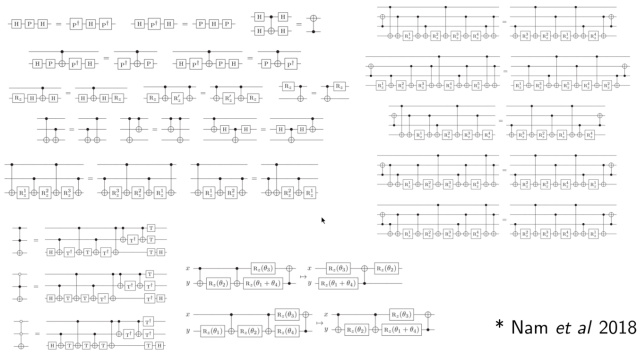


Fig. 3. Small subset of rewrite rules for classical circuit optimization [4]

D. Quantum Circuit Optimization using the ZX-Calculus

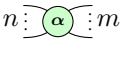
A more efficient approach for quantum circuit optimization is to use the ZX-Calculus. The ZX-Calculus is a graphical language for quantum computing, differing from the logic-gate representation by using connected nodes to represent operations on qubits. Since this new representation utilizes way fewer *gate-types* than the classical representation, there exist way fewer rewrite rules.

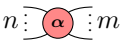
Test

II. Introduction to ZX-Calculus

ZX-Calculus has been kickstarted by Coecke and Duncan in 2008 [1]. Since then, it has been used to prove a lot of interesting results in quantum computing. It used in the field of Measurement Based Quantum Computing (MBQC) [2]. Recently [10] it has found wide application in quantum circuit optimization and verification.

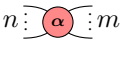
A ZX-diagram is a graph consisting of nodes and edges. The nodes are called spiders and are colored either green or red. Additionally, they may carry a phase value $\alpha \in [0, 2\pi)$. Spiders are visuallized in the following way:



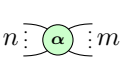


Here n and m are the number of incoming and outgoing edges, respectively. The phase value α is optional and can be omitted if it is zero.

It is important to remember that each spider represents a linear map from n to m qubits. This linear maps can be directly calculated using the following formulas:



$$\equiv |\underbrace{0\ldots0}_{m}\rangle\langle\underbrace{0\ldots0}_{n}| + e^{i\alpha}|\underbrace{1\ldots1}_{m}\rangle\langle\underbrace{1\ldots1}_{n}|$$



$$\equiv |\underbrace{+\cdots+}_{m}\rangle\langle\underbrace{+\cdots+}_{n}| - e^{i\alpha}|\underbrace{-\cdots-}_{m}\rangle\langle\underbrace{-\cdots-}_{n}|$$

By just using single spiders, we can already represent a lot of quantum gates. The most important ones are shown in figure 4. Note that in ZX-Calculus we ignore the global scalar factor of a quantum gate

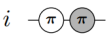
Name	Diagram	Matrix
identity		$\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$
Pauli Z		$\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$
Pauli X NOT gate		$\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$
Pauli Y		$\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}$
Hadamard gate		$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$
S gate		$\begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}$
V gate		$\frac{1}{2}\begin{pmatrix}1+i & 1-i\\ 1-i & 1+i\end{pmatrix}$
T gate		$\begin{pmatrix}1 & 0\\ 0 & e^{i\frac{\pi}{4}}\end{pmatrix}$

Fig. 4. Spiders representing quantum gates [10][P.87]

Verifying some Unitary Gates

Let's verify that the Pauli-Z gate and the Pauli-X gate can indeed be represented by the spiders shown in figure 4.

$$\equiv |\underbrace{0\ldots0}_{1}\rangle\langle\underbrace{0\ldots0}_{1}| + e^{i\pi}|\underbrace{1\ldots1}_{1}\rangle\langle\underbrace{1\ldots1}_{1}| \quad (1)$$

$$\equiv |0\rangle\langle 0| - |1\rangle\langle 1| \quad (2)$$

$$\equiv \begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}1 & 0\end{pmatrix} - \begin{pmatrix}0\\1\end{pmatrix}\begin{pmatrix}0 & 1\end{pmatrix} \quad (3)$$

$$\equiv \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix} \quad (4)$$

$$\equiv Z \quad (5)$$

III. Motivation

IV. Problem Statement

V. Solution

VI. Evaluation

VII. Results

VIII. Future Work

IX. Conclusion

References

[1] Bob Coecke and Ross Duncan. A graphical calculus for quantum observables. *Preprint*, 2007.

[2] Ross Duncan. A graphical approach to measurement-based quantum computing, 2012.

[3] Lov K. Grover. A fast quantum mechanical algorithm for database search, 1996.
[4] Aleks Kissinger. An introduction to the zx-calculus. https://www.cs.ox.ac.uk/people/aleks.kissinger/slides/fullstack-25mins.pdf, 2020. accessed: 30.06.2023.
[5] Aleks Kissinger and John van de Wetering. Reducing t-count with the zx-calculus, 2020.
[6] Ross N.J. Su Y. et al. Nam, Y. Automated optimization of large quantum circuits with continuous parameters, 2018.
[7] Michael A. Nielsen and Isaac L. Chuang. Quantum computation and quantum information, 2010.
[8] Peter W. Shor. Algorithms for quantum computation: discrete logarithms and factoring, 1994.
[9] Robert Wille Tom Peham, Lukas Burgholzer. Equivalence checking of quantum circuits with the zx-calculus. https://www.cda.cit.tum.de/files/eda/2022_jetcas_equivalence_checking_of_quantum_circuits_with_the_zx_calculus.pdf, 2022. accessed: 30.06.2023.
[10] John van de Wetering. Zx-calculus for the working quantum computer scientist, 2020.