Experimental and Theoretical Analysis of Quantum Computing

Philipp Schöneberg[1], Hans-Otto Carmesin[1, 2, 3], Phil Gustke[1] and Jannes Ruder[1]

[1]Gymnasium Athenaeum, Harsefelder Straße 40, 21620 Stade
[2]Studienseminar Stade, Bahnhofstr. 5, 21682 Stade
[3]Universität Bremen, Fachbereich 1, Postfach 330440, 28334 Bremen
hans-otto.carmesin@t-online.de

Kurzfassung

Innerhalb der Jugend forscht AG unserer Schule sowie einem Kurs über Quantengravitation lernten wir die fundamentalen Konzepte von Quantencomputern theoretisch kennen und entwickelten darauf aufbauend demonstrative Experimente. In diesem Artikel erklären und demonstrieren wir ein universelles Set bestehend aus essenzielle Quantengattern. Des Weiteren zeigen wir darauf aufbauend zwei bekannte Quantenalgorithmen und präsentieren eine Möglichkeit zum Vervielfachen der allgemeinen Rechengeschwindigkeit von Quantencomputern.

Abstract

In a student research club, we explain elemental concepts of quantum computing with theory and experiments. For it, we demonstrate multiple fundamental quantum gates. These quantum gates result in a valuable universal set. By using this set, we derive two known algorithms showing quantum supremacy. Moreover, we present and discuss an opportunity to multiply the calculation speed of quantum computing.

1. Introduction

Since the production of the first computer humans have tried to achieve continuous growth in terms of efficiency. Hereby, transistors are often used as an indicator due to their functionality. In 1965 Gordan Moor noticed a doubling of the number of transistors inside newly published computer chips in a time interval of one to two years (Moore, 1965) which is often realized by downsizing. However, an end of this trend is suspected due to physical limitations like the size of an atom. Therefore, the focus started to shift towards new technologies like quantum computers. In a student research club, we try to understand the complexity of this topic by developing functional demonstration experiments of quantum computing. In particular, we decided to focus on the technology of quantum computers based on light because of their leading efficiency (Madsen, et al., 2022). Furthermore, we even discovered a way to multiply the speed of processing inside a quantum computer by transferring the idea of multiplexing from the data transfer into the processing of optical computers including quantum computers based on light. Moreover, we learn about algorithms which have been used to demonstrate quantum supremacy in the past (Deutsch and Jozsa, 1992). To illustrate these, we employ exemplary calculations and verify these with adequate simulations.

1.1. Universal Set

To understand quantum computing completely we wanted our demonstration experiments to explain the most fundamental way of processing inside a quantum computer. This is the processing of qubits and mathematically known as unitary operations. They are realized by quantum gates. To be able to achieve every possible unitary operation we use a universal set of quantum gates. Mathematically this could be achieved by the set {C-NOT, single-qubit gate} (De_Ro, 2021). Yet it is physically impossible to realize a universal and precise single-qubit gate which would be able to convert a qubit in every possible way (Circuit Library, 2023). Therefore, we use the Set {C-NOT, H, T} approximating a universal Set (De_Ro, 2021).

1.2. Materials

For our experiments we are mainly using a laser with a wavelength of 650nm. Yet in some experiments requiring a second laser we additionally use a laser with a wavelength of 520nm.

1.3. Qubits

Based on the usage of the C-NOT gate as the only multiple qubit gate of our universal set, we need to realize a minimum of two qubits. We decided to use the linear polarization of the light as the first qubit defining it being vertically polarized as $|0\rangle$ and it being horizontally polarized as $|1\rangle$. This qubit will also be used as the controlling qubit inside the C-NOT gate.

Therefore, the second qubit will only have to show an inversion in some of the cases in the C-NOT gate. For this reason, we decided to simply use an asymmetry of our laser creating a diagonal oval as a representation of the second Qubit. Thereby, it being antidiagonally oriented from the bottom left to the top right will be defined as $|0\rangle$ whereas it being diagonally oriented from the bottom right to the top left will be defined as $|1\rangle$.

2. **Experiments**

2.1. **C-NOT Gate**

The C-NOT gate (CX gate) in optical quantum computers is characterized by using the correlation between a control and a target qubit to control the inversion of the target qubit. The crucial point here is that the C-NOT gate entangles the states of the qubits, thus, enabling complex quantum operations that are not realizable in classical systems. We have achieved this by guiding a laser beam, realizing the two previously described qubits, through an optical circuit. Inspired by an existing C-NOT realization (Lopez, et al., 2018), we designed the following setup (see fig. 1).

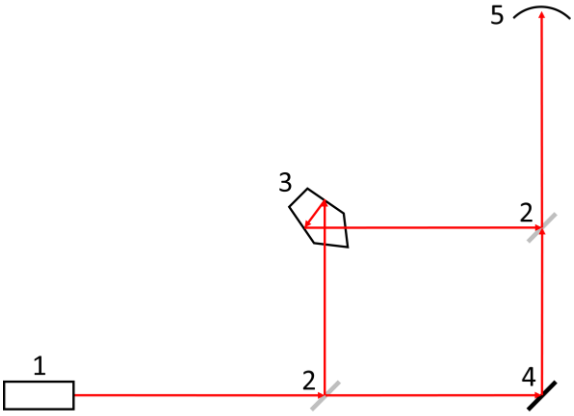

Fig. 1: Schematic experimental setup of the C-NOT gate demonstration: 1. laser, 2. polarizing beamsplitter, 3. pentaprism, 4. mirror, 5. detector

The first polarizing beamsplitter reflects or transmits the beam depending on the state of the control qubit, being the polarization. In our case, the beamsplitter transmits horizontally polarized light, while vertically polarized light is reflected. The pentaprism reflects the asymmetry of our laser, used as the second qubit, twice and thus ensures a double inversion of the target qubit. Therefore, the target qubit does not change. The mirror reflects the beam, causing the state of the target qubit to be inverted only once. The second polarizing beamsplitter combines the two previously separated light waves.

To verify the general functionality of our demonstration experiment we implemented every combination of basis states and checked if their outcome corresponds to the expected theoretical results (see fig. 2).

Input control qubit	Input target qubit	Resulting control qubit	Resulting target qubit	Corresponding experiment
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	Fig 3
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	Fig 4
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	Fig 5
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	Fig 6

Fig. 2: A table showing the implemented basis states of each qubit as well as their theoretical outcome and corresponding performed experiments.

However, although this realization works fine for most demonstrative purposes, it cannot demonstrate more complicated functionality like phase kickback yet. This occurs mostly, due to the solely demonstrative nature of the target qubit, briefly realized by the asymmetric orientation of the laser.

Fig. 3: C-NOT experiment realizing the input qubit states through the usage of a vertical polarizer and the rotation of the laser, creating an antidiagonal oval.

Fig. 4: C-NOT experiment realizing the input qubit states through the usage of a vertical polarizer and the rotation of the laser, creating a diagonal oval.

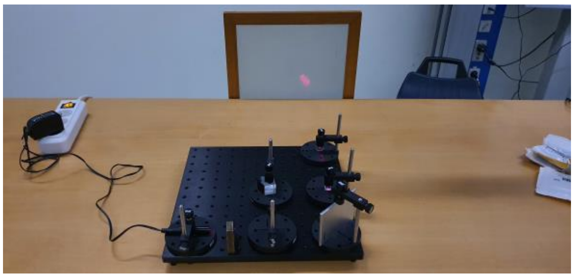

Fig. 5: C-NOT experiment realizing the input qubit states through the usage of a horizontal polarizer and the rotation of the laser, creating an antidiagonal oval.

Fig. 6: C-NOT experiment realizing the input qubit states through the usage of a horizontal polarizer and the rotation of the laser, creating a diagonal oval.

2.2. **Single Qubit Gates**

To realize single qubit gates for the qubit of polarization, we employ a variety of wave plates. Specifically, we use a half wave plate to realize the Hadamard gate specified in our chosen universal set as well as an NOT gate (X gate) and a Z gate. Well, known by most physicists is the realization of the quantum-NOT gate through rotating a half wave plate by 45° along the optical axis. This function can be experimentally demonstrated by using two linear (vertical) polarizers and placing the rotated half wave plate in between (see fig. 7).

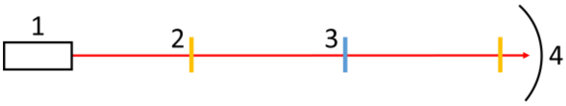

Fig. 7: Schematic experimental setup for the NOT gate demonstration: 1. laser, 2. horizontal polarizer, 3. X gate, 4. detector

Thus, we initialize the polarization qubit of the input beam with a value of $|0\rangle$ and invert its value to $|1\rangle$. Then, by employing the last polarizer as a method to read out the qubit's value, we show that the intensity of the light beam hitting the detector is approximately 0lx and the linear polarization has been inverted (see fig. 8).

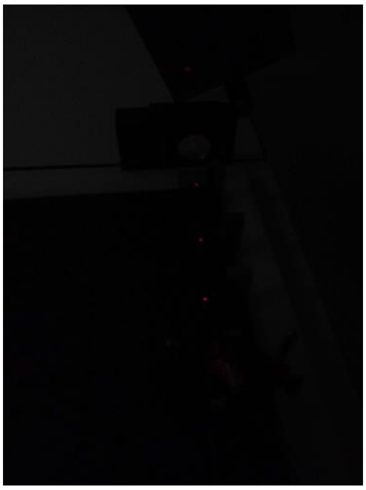

Fig. 8: Realization of the experimental setup in fig. 7 with a diverging lens added before the detector. The point on the detector is barely visible to the naked eye.

On the other hand, the T gate has not been realized due to the lack of accessible material. Nevertheless, we propose a method for realizing a T gate by employing a $\frac{\lambda}{8}$ wave plate that has been rotated by 0° degrees. Luckily, the T gate is quite similar to the Z gate regarding their same rotation angle around the optical axis. Due to this similarity, one can imagine the Z gate as a kind of replacement for the missing T gate for demonstrative purposes. This works especially well, because the Z gate can be used to replace the T gates in our exemplary calculations regarding the quantum algorithms.

Realizing a Z gate can be done by employing a half wave plate rotated by 0° around the optical axis, while the Hadamard gate is realized by using a half wave plate that is rotated by 22.5°. To demonstrate these gates experimentally, we employ the same method as we have with the X gate, although in this instance we realize the NOT operation through the sequence of gates: Hadamard gate, Z gate, Hadamard gate (Qiskit Textbook, 2024) (see fig. 9).

Fig. 9: Schematic experimental setup for the Hadamard and Z gate demonstration: 1. laser, 2. horizontal polarizer, 3. Hadamard gate, 4. Z gate, 5. detector

Therefore, by placing the half wave plates in the given order we transform the polarization qubit form $|0\rangle$ to $|+\rangle$ to $|-\rangle$ to $|1\rangle$ and thus, demonstrate the Hadamard and Z gates in the same fashion as the X gate (see fig. 10).

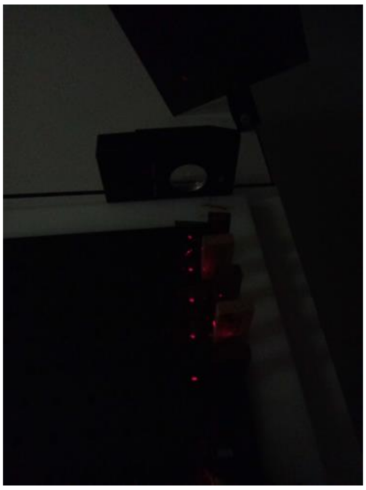

Fig. 10: Realization of the experimental setup in fig. 9 with a scattering lens added before the detector. The point on the detector is barely visible to the naked eye.

2.3. **Multiplexing**

During our research on the C-NOT gate, we had the idea to transfer the technology of multiplexing, known from data transfer to data processing in optical computers. Specifically, we focused on wavelength-based multiplexing. This is characterized by increasing the parallelism of a light-based systems by superimposing photons of different wavelengths. The crucial point here is that these superimposed photons do not interfere with each other and can be processed simultaneously in optical circuits. Thus, the bandwidth of data that such a computer can process simultaneously can be drastically multiplied.

To prove that the light beams do not influence each other during data processing through superposition, we utilized our existing setup of the C-NOT gate. We intersected the different laser beams multiple times within the gate (see fig. 11).

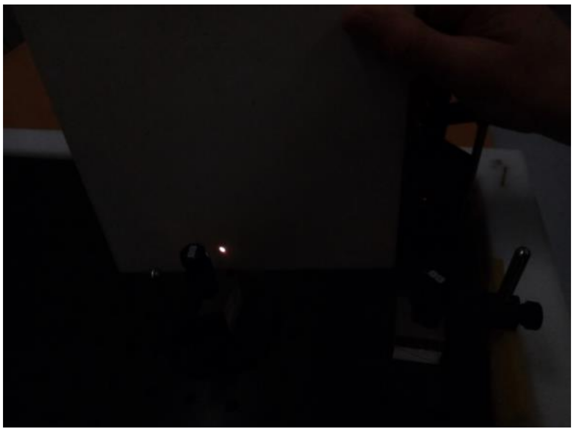

Fig. 11: Intersection of the two laser beams between the pentaprism and the second beamsplitter.

During this process, we observed continued complete and accurate data processing of the individual qubits, implemented through the different laser beams (see fig. 12).

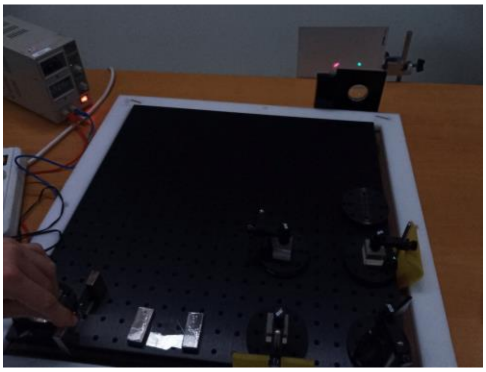

Fig 12: C-NOT experiment realizing the input qubit states |0⟩ and |0⟩ in the red laser as well as |1⟩ and |0⟩ in the green laser. The qubits implemented in the red laser are realized by the usage of a vertical polarizer and the rotation of the laser, creating an antidiagonal oval. The ones in the green laser are realized by the usage of a horizonal polarizer and a rotation of the laser creating an antidiagonal oval as well.

Another possible implementation of wavelength-based multiplexing can be achieved by cleverly superimposing multiple lasers. The lasers are superimposed using a combination of collecting and scattering lenses. Subsequently, the combined beam is sent through an optical circuit (see fig. 13).

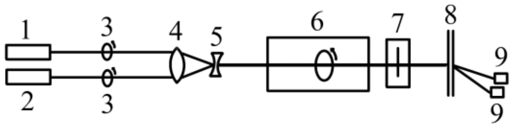

Fig. 13: Exemplary optical setup to use wavelength-based multiplexing for optical computation: 1. laser of wavelength a, 2. laser of wavelength b, 3. wave plate to initialize input state of the given qubit, 4. collecting lens, 5. scattering lens, 6. optical operation, 7. polarizer for measurement, 8. optical lattice, 9. intensity measurement device

Afterwards, a measurement is performed using a polarizer, which changes the intensity depending on the polarization. Finally, the combined beam is split again according to the wavelength of the individual components, and the individual states are determined using intensity measurement devices.

3. Calculations and Simulations

To emphasize the applicability of our universal set, we decided to illustratively calculate two exemplary quantum algorithms. Therefore, we solely use the gates contained in our chosen universal set, except for the Z gate. We have decided to use the Z gate in the following calculations because it is a gate we have experimentally realized, and it can be easily transferred to our universal set by replacing it with a sequence of four T gates. Thus, the connection of experiments and theoretic calculations becomes much clearer.

3.1. Deutsch-Jozsa Algorithm

The Deutsch-Jozsa algorithm is an algorithm to categorize a binary function into either constant or balanced. Herby the algorithm only needs to run once and implies a constant function through returning the measured qubit in the state |1⟩ and a balanced function by returning it in the state |0⟩.

To understand the Deutsch-Jozsa algorithm we focus on its simplest form using only two qubits. Its then called the Deutsch algorithm and consists of three Hadamard gates and one oracle arranged as follows (see fig. 14):

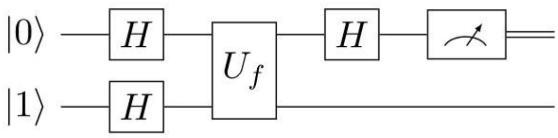

Fig. 14: A diagram of the circuit representing the Deutsch algorithm.

Thereby, the oracle represents the function. In our case we just use a C-NOT gate as an oracle because it is equivalent to a balanced function (see fig 15).

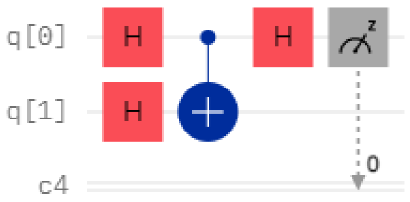

Fig. 15: A diagram of the circuit representing the modified Deutsch algorithm.

To show that the Deusch algorithm works with our universal set we start calculating this version of the algorithm by initializing the two qubits: Qubit $|q_0\rangle$ as $|0\rangle$ and qubit $|q_1\rangle$ as $|1\rangle$ (see equation {1} and {2}).

$$|q_0\rangle = |0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \qquad \{1\}$$

$$|q_1\rangle = |1\rangle = \begin{pmatrix}0\\1\end{pmatrix} \qquad \{2\}$$

Now we connect the two qubits with each other through the usage of the Kronecker product (see equation {3}).

$$|q_0 q_1\rangle = |q_1\rangle \otimes |q_0\rangle = |1\rangle \otimes |0\rangle$$

$$= \begin{pmatrix} 0 \cdot & \begin{matrix}1\\0\end{matrix} \\ 1 \cdot & \begin{matrix}1\\0\end{matrix} \end{pmatrix} = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \qquad \{3\}$$

Subsequently, we apply a Hadamard gate to both qubits (see equation {4}).

$$H_{q_0,q_{01}} \cdot |q_0 q_1\rangle$$

$$= \frac{1}{2} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \cdot \begin{pmatrix}0\\0\\1\\0\end{pmatrix}$$

$$= \frac{1}{2} \cdot \begin{pmatrix}1\\1\\-1\\-1\end{pmatrix} = \begin{pmatrix}\frac{1}{2}\\\frac{1}{2}\\-\frac{1}{2}\\-\frac{1}{2}\end{pmatrix} \qquad \{4\}$$

Next, we use the C-NOT gate to modify the qubits through the function (see equation {5}).

$$CX_{q_0,q_{01}} \cdot |q_0 q_1\rangle$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix}\frac{1}{2}\\\frac{1}{2}\\-\frac{1}{2}\\-\frac{1}{2}\end{pmatrix}$$

$$= \begin{pmatrix}\frac{1}{2}\\-\frac{1}{2}\\-\frac{1}{2}\\\frac{1}{2}\end{pmatrix} \qquad \{5\}$$

Afterwards we apply another Hadamard gate to the first qubit so that it can be measured in the next step (see equation {6}).

$$H_{q_0} \cdot |q_0 q_1\rangle$$

$$= \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \cdot \begin{pmatrix}\frac{1}{2}\\-\frac{1}{2}\\-\frac{1}{2}\\\frac{1}{2}\end{pmatrix}$$

$$= \frac{1}{\sqrt{2}} \cdot \begin{pmatrix}0\\1\\0\\-1\end{pmatrix}$$

$$= \begin{pmatrix}0\\\frac{1}{\sqrt{2}}\\0\\-\frac{1}{\sqrt{2}}\end{pmatrix}$$

$$= |1\,-\rangle \qquad \{6\}$$

Finally, we notice that the measured first qubit changed. This is due to the phase kickback inside the C-NOT gate (Lee and Selby, 2016). Thereby, it being $|1\rangle$ verifies the used C-NOT gate as a balanced function. A simulation we programmed using qiskit libraries in python verifies this result (see fig. 16 and 17).

```
#Initialisierung importieren
import matplotlib.pyplot as plt
import numpy as np

#Qiskit importieren
from qiskit import *
from qiskit.tools import job_monitor
from qiskit.quantum_info import Statevector
from qiskit_ibm_runtime import QiskitRuntimeService
from qiskit_ibm_runtime import Sampler

#Visualisierungshilfen importieren
from qiskit.visualization import plot_histogram
from qiskit.visualization import plot_bloch_vector
from qiskit.visualization import plot_bloch_multivector
from qiskit.visualization import plot_histogram
from qiskit_textbook.tools import vector2latex
from qiskit.visualization import plot_distribution
```

```
#Hadamard-Gatter auf alle Qubits
def initialize(qc, qubits):
    for q in qubits:
        qc.h(q)
    return qc
```

```
#Simulation
def simulate_state_vector(qc):
    state = Statevector(qc)
    return state
```

Fig. 16: Verification of our calculation of the Deutsch algorithm using qiskit libraries for python with jupyter notebook.

```
#Erstellen von Quantenschaltung mit 2 Qubits
n = 1
deutsch_circuit = QuantumCircuit(n+1, n)

#Umkehrung des Kontrollqubits
#von Zustand |0> auf |1>
deutsch_circuit.h(1)
deutsch_circuit.z(1)
deutsch_circuit.h(1)

<qiskit.circuit.instructionset.InstructionSet at 0x13f3:

#Initialisierung
deutsch_circuit = initialize(deutsch_circuit, [0,1])

# Oracle
deutsch_circuit.cx(0,1)

<qiskit.circuit.instructionset.InstructionSet at 0x13f7·

#Hadamard-Gatter auf Zielqubit
deutsch_circuit.h(0)
#doppelte Anwendung des Hadamards
#ist theoretisch obsolet,
#aber zu Demonstrationszwecken
#erhalten geblieben

<qiskit.circuit.instructionset.InstructionSet at 0x13f3:

#Visualisierung Zustandsvektor
vector2latex(simulate_state_vector(deutsch_circuit),
             pretext="|\\psi\\rangle =")
```

$$|\psi\rangle = \begin{bmatrix} 0 \\ 0.70711 \\ 0 \\ -0.70711 \end{bmatrix}$$

Fig. 17: Verification of our calculation of the Deutsch algorithm using qiskit libraries for python with jupyter notebook.

3.2. **Grover's algorithm**

The Grover algorithm is a quantum search algorithm that can be interpreted as searching for specific items from a given list. It consists of an initialization in a uniform superposition, an oracle representing the function selecting the search result and the diffusion operator amplifying the search result for effective measurement.

In its simplest form with two qubits, while still featuring our chosen universal set, the algorithm can be written as follows (see fig. 18).

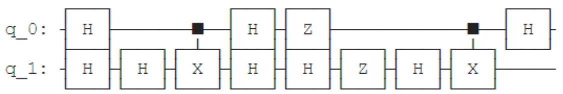

Fig. 18: Schematic quantum circuit for our exemplary version of the Grover algorithm for two qubits.

In this instance, the oracle is selecting the $|11\rangle$ state and is derived from a simple realization by the quiskit textbook (Qiskit Textbook, 2024). Gates like the CZ gate have been replaced with the sequence: Hadamard gate, cx gate, Hadamard gate to feature our universal set. Moreover, we algebraically calculate the algorithm step by step and verify the outcome with a qiskit simulation.

Firstly, we initialize both qubits in the $|0\rangle$ state and connect them via Kronecker product (see equation 7).

$$|q_0 q_1\rangle = |q_1\rangle \otimes |q_0\rangle = |0\rangle \otimes |0\rangle$$
$$= \begin{pmatrix} 1 \cdot \begin{matrix} 1 \\ 0 \end{matrix} \\ 0 \cdot \begin{matrix} 1 \\ 0 \end{matrix} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \{7\}$$

Then, we apply the Hadamard operation to both qubits to achieve a uniform superposition of $|++\rangle$ (see equation 8).

$$H_{q0,q01} \cdot |q_0 q_1\rangle = \frac{1}{2} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$
$$= \frac{1}{2} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \quad \{8\}$$

Thirdly, we require the oracle consisting of a CZ gate. Therefore, we replace it with a sequence of our chosen gates and apply those to calculate the operation (see equation 9-12).

$$H_{q_1} CX_{q0,q01} H_{q_1}$$
$$= \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$
$$= \frac{1}{2} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} = CZ_{q_0 q_{01}} \quad \{9\}$$

$$H_{q_1} \cdot |q_0 q_1\rangle = \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \cdot \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$$
$$= \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{pmatrix} \quad \{10\}$$

$$CX_{q0,q01} \cdot |q_0 q_1\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} \quad \{11\}$$

$$H_{q_1} \cdot |q_0 q_1\rangle = \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$= \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \quad \{12\}$$

Next, we want to apply the general diffusion operator for the two qubit Grover's algorithm. To achieve this, we start by applying two Hadamard operations (see equation 13).

$$H_{q_0,q_{01}} \cdot |q_0 q_1\rangle$$

$$= \frac{1}{2} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \cdot \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \quad \{13\}$$

Now, we replace the following necessary matrix with a sequence of our chosen gates and apply those to the exemplary calculation (see equation 14-17).

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$= Z_{q_0,q_{01}} CZ_{q_0,q_{01}} = Z_{q_0,q_{01}} H_{q_1} CX_{q_0,q_{01}} H_{q_1}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \cdot$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \quad \{14\}$$

$$Z_{q_0,q_{01}} \cdot |q_0 q_1\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \quad \{15\}$$

$$H_{q_1} \cdot |q_0 q_1\rangle = \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \cdot \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}$$

$$= \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 0 \\ -1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} \quad \{16\}$$

$$CX_{q_0,q_{01}} \cdot |q_0 q_1\rangle$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \quad \{17\}$$

Lastly, we need to again apply two Hadamard operations. However, since we would now apply two Hadamard operations to the first qubit, we can ignore those, because the Hadamard operation is a self-inverse matrix. Thus, we only apply a Hadamard operation to the second qubit to finish our calculations (see equation 17).

$$H_{q_1} \cdot |q_0 q_1\rangle$$

$$= \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |11\rangle \quad \{18\}$$

Finally, we verify our solution with the following qiskit simulation (see fig. 19 and 20).

```
#Initialisierung importieren
import matplotlib.pyplot as plt
import numpy as np

#Qiskit importieren
from qiskit import *
from qiskit.tools import job_monitor
from qiskit.quantum_info import Statevector
from qiskit_ibm_runtime import QiskitRuntimeService
from qiskit_ibm_runtime import Sampler

#Visualisierungshilfen importieren
from qiskit.visualization import plot_histogram
from qiskit.visualization import plot_bloch_vector
from qiskit.visualization import plot_bloch_multivector
from qiskit.visualization import plot_histogram
from qiskit_textbook.tools import vector2latex
from qiskit.visualization import plot_distribution
```

```
#Erstellen von Quantenschaltung mit 2 Qubits
n = 2
grover_circuit = QuantumCircuit(n)
```

```
#Hadamard-Gatter auf alle Qubits
def initialize(qc, qubits):
    for q in qubits:
        qc.h(q)
    return qc
```

Fig. 19: Verification of our calculation of the Grover algorithm using qiskit libraries for python with jupyter notebook.

```
#Simulation
def simulate_state_vector(qc):
    state = Statevector(qc)
    return state
```

```
#Initialisierung in uniformer Superposition
grover_circuit = initialize(grover_circuit, [0,1])
```

```
# Oracle für |11>
grover_circuit.h(1)
grover_circuit.cx(0,1)
grover_circuit.h(1)
```

```
<qiskit.circuit.instructionset.InstructionSet at 0x15cf
```

```
# Diffusion Operator
grover_circuit.h([0,1])
grover_circuit.z([0,1])
grover_circuit.h(1)
grover_circuit.cx(0,1)
grover_circuit.h(0)
```

```
<qiskit.circuit.instructionset.InstructionSet at 0x15cf
```

```
#Visualisierung Zustandsvektor
vector2latex(simulate_state_vector(grover_circuit),
             pretext="|\\psi\\rangle =")
```

$$|\psi\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Fig. 20: Verification of our calculation of the Grover algorithm using qiskit libraries for python with jupyter notebook.

4. **Results**

In our experiments we have realized the C-NOT and Hadamard gate of our universal set. Additionally, we were able to demonstrate the X- and Z gate. Thereby, the Z gate could also be used within the calculation of the algorithms as a replacement of the T gate. Furthermore, we discovered a great possibility to increase the processing speed of light-based computers through the usage of multiplexing. Moreover, we were able to show the functionality of the Deutsch-Jozsa as well as Grover algorithm mathematically and through simulations, while working in the confines of our universal set. Hence, we gained a detailed understanding of the functionality of quantum computers and their hardware as well as software.

5. **Discussion**

Obviously, our experimental demonstrations do not quite work as a complete and functioning implementation of a quantum computer yet, especially due to the solely demonstrative nature of our CX gate and the missing implementation of a T gate.

Nevertheless, using the orbital angular momentum of light as a second qubit, we are currently trying to realize the complete universal set of gates. Firstly, we are employing the C-NOT-realization from Lopez (Lopez, et al., 2018) in combination with an oam-Hadamard gate (Xinbing Song, et al., 2020) to eventually implement a swap gate (Qiskit Textbook, 2024). Combining this with our proposed solution for a T gate and the given Hadamard gate-realization, the entire universal set can be physically realized.

Moreover, we are currently working on a physical realization of the Deutsch algorithm solely using this universal set and the materials mentioned above.

In a nutshell, our experiments and calculations have helped us to greatly improve our understanding of quantum computers in all areas and even let us apply the gained knowledge in a practical way. It especially helps us to connect abstract mathematics and theory with their physical implementation and thus provides a more thorough picture of the field. The experiments are simple and visually appealing, while still conveying the essential concepts of quantum computing.

Additionally, while the costs for the C-NOT gate realization might be a lot for most schools and some universities, the demonstration experiments for the single qubit gates solely require polarizer and half wave plate foil. Thus, the costs come out to be in the low double-digit area making the experiments easily and cheaply replicable.

6. **Literature**

Qiskit Textbook: https://github.com/Qiskit/textbook (Stand: 8/2024)

Lee, Ciarán M; Selby, John H (2016): Generalised phase kick-back: the structure of computational algorithms from physical principles. In: New J. Phys. 18 033023, Url: https://iop-science.iop.org/article/10.1088/1367-2630/18/3/033023/meta (Stand: 8/2024)

Circuit Library: https://qiskit.org/documentation/stable/0.19/apidoc/circuit_library.html
(Stand: 3/2023)

De_Ro, Nicolas (2021): Universal Sets of Gates in Quantum Computing. In: ResearchGate, Url: https://www.researchgate.net/publication/357768824_Universal_Sets_of_Gates_in_Quantum_Computing
(Stand: 8/2024)

Deutsch, David; Jozsa, Richard (1992): Rapid solution of problems by quantum computation. In: Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, p. 553-558, Url: https://royalsocietypublishing.org/doi/epdf/10.1098/rspa.1992.0167
(Stand: 8/2024)

Lopez, J. H.; et al. (2018): Experimental realization of quantum CNOT gate for orbital angular momentum and polarization with linear optical spanners and the rotational frequency shift. In: arXiv:1807.06065, Url: https://arxiv.org/abs/1807.06065
(Stand: 8/2024)

Madsen, Lars S; et al. (2022): Quantum computational advantage with a programmable photonic processor. In: Nature, p. 210-214, Url: https://www.nature.com/articles/s41586-022-04725-x
(Stand: 8/2024)

Moore, Gordon E. (1965): Cramming more components onto integrated circuits. In: Electronics, Volume 38, Number 8, Url: https://ieeexplore.ieee.org/abstract/document/4785860
(Stand: 8/2024)

Xinbing, Song; Yifan, Sun; et al. (2015): Bell's measure and implementing quantum Fourier transform with orbital angular momentum of classical light. In: Scientific Reports, Url: https://www.nature.com/articles/srep14113
(Stand: 8/2024)

Acknowledgements

We would like to take this opportunity to thank the Lower Saxony sponsor pool and the Anna and Claus Heinrich Siemsen Foundation, whose generous financial support made this project possible in the first place. We would also like to thank Sparkasse for its mediation between us and the foundation.