

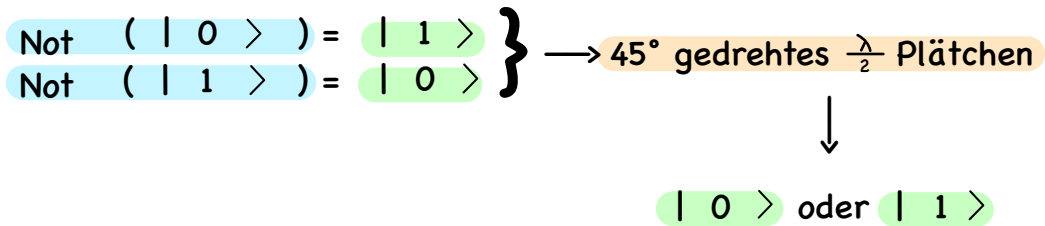
Quantencomputer Gatter

Made by

Jannes Ruder



Not

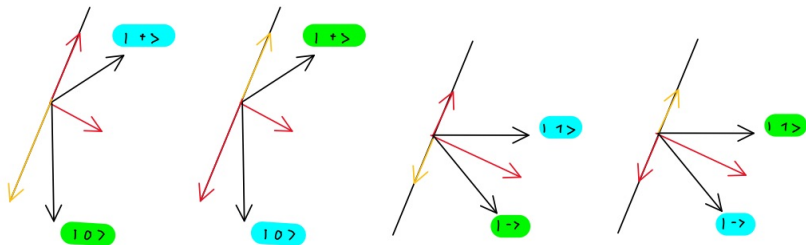


Hadarmad

$$\begin{aligned}
 \text{Hadarmad } (|0\rangle) &= |+\rangle \\
 \text{Hadarmad } (|1\rangle) &= |-\rangle \\
 \text{Hadarmad } (|+\rangle) &= |0\rangle \\
 \text{Hadarmad } (|-\rangle) &= |1\rangle
 \end{aligned}
 \left. \vphantom{\begin{aligned} \text{Hadarmad } (|0\rangle) &= |+\rangle \\ \text{Hadarmad } (|1\rangle) &= |-\rangle \\ \text{Hadarmad } (|+\rangle) &= |0\rangle \\ \text{Hadarmad } (|-\rangle) &= |1\rangle \end{aligned}} \right\} \rightarrow 22,5^\circ \text{ gedrehtes } \frac{\lambda}{2} \text{ Plättchen}$$

↓

$|0\rangle$, $|1\rangle$, $|+\rangle$ oder $|-\rangle$



C-Not

$$\text{C-Not} (| 0 \rangle | 0 \rangle) = | 0 \rangle | 0 \rangle$$

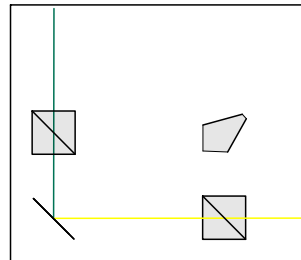
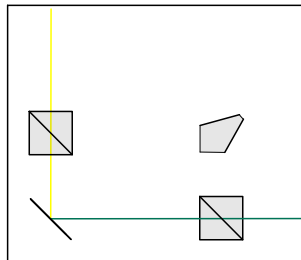
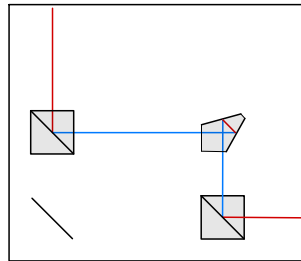
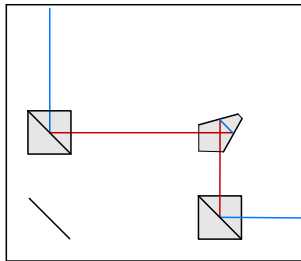
$$\text{C-Not} (| 0 \rangle | 1 \rangle) = | 0 \rangle | 1 \rangle$$

$$\text{C-Not} (| 1 \rangle | 0 \rangle) = | 1 \rangle | 1 \rangle$$

$$\text{C-Not} (| 1 \rangle | 1 \rangle) = | 1 \rangle | 0 \rangle$$



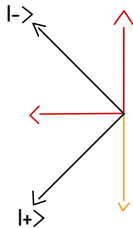
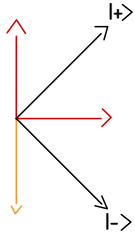
$| 0 \rangle | 0 \rangle$ — blue
 $| 0 \rangle | 1 \rangle$ — red
 $| 1 \rangle | 0 \rangle$ — yellow
 $| 1 \rangle | 1 \rangle$ — green
 Pentaprisma
 Beamsplitter
 Spiegel



$| 0 \rangle$ oder $| 1 \rangle$

Z

$$\begin{aligned} Z (| + \rangle) &= | - \rangle \\ Z (| - \rangle) &= | + \rangle \end{aligned} \quad \left. \vphantom{\begin{aligned} Z (| + \rangle) &= | - \rangle \\ Z (| - \rangle) &= | + \rangle \end{aligned}} \right\} \longrightarrow 0^\circ \text{ gedrehtes } \frac{\lambda}{2} \text{ Plättchen}$$



$| + \rangle$ oder $| - \rangle$

T

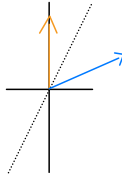
$$T(|0\rangle) = |0\rangle$$

$$T(|1\rangle) = |1\rangle$$

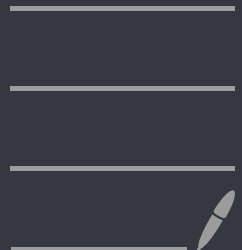
$$T(|+\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$T(|-\rangle) = |0\rangle - |1\rangle$$

}



Quantencomputer Berechnungen und Formeln



Vektor Schreibweise für Qubits

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

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

Kontroll Qubit Ziel Qubit

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

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

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

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

Matrix Schreibweise für Gatter

$$\text{C-Not} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\text{Hadamard} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$\text{Not} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\text{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\text{T} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Beispiel Berechnungen des C-Not

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

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = |1\rangle \otimes |0\rangle$$

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

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |1\rangle \otimes |1\rangle$$

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

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = |0\rangle \otimes |0\rangle$$

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

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |0\rangle \otimes |1\rangle$$

Beispiel Berechnungen des Hadarmad Gatters

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

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

Beispiel Berechnungen des NOT Gatters

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{bmatrix} 0 \cdot 0 + 1 \cdot 1 \\ 1 \cdot 0 + 0 \cdot 1 \end{bmatrix} = | 0 \rangle$$

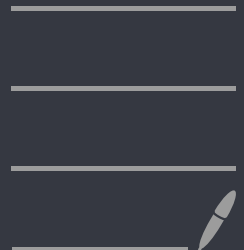
$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{bmatrix} 0 \cdot 1 + 1 \cdot 0 \\ 1 \cdot 1 + 0 \cdot 0 \end{bmatrix} = | 1 \rangle$$

Beispiel Berechnungen des Z Gatters






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

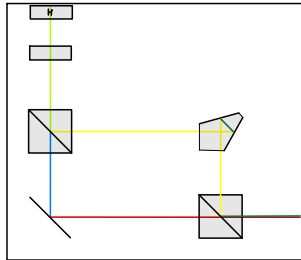
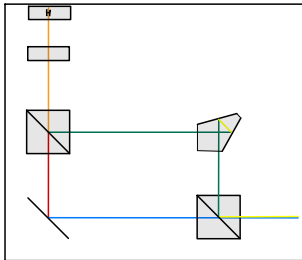
$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{bmatrix} 1 \cdot 1 + 0 \cdot 1 \\ 0 \cdot -1 + -1 \cdot -1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = | + \rangle$$

Quantencomputer Algorithmen



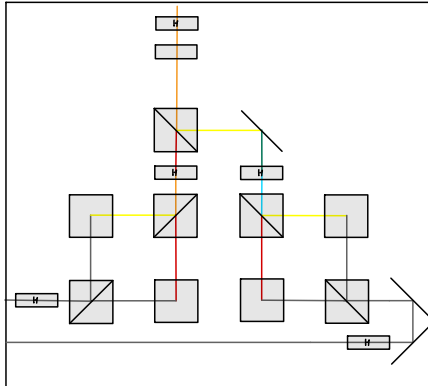
Deutsch - Algorithmus

$|0\rangle|0\rangle$ — blue
 $|0\rangle|1\rangle$ — red
 $|1\rangle|0\rangle$ — yellow
 $|1\rangle|1\rangle$ — green
 $|+\rangle|0\rangle$ — light green
 $|+\rangle|1\rangle$ — orange
 Pentaprisma 
 Beamsplitter 
 Spiegel 
 Hologramm 
 Hadarmad 

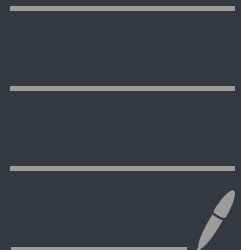


Deutsch-Jozsa- Algorithmus

$ 0\rangle 0\rangle$	Blue
$ 0\rangle 1\rangle$	Red
$ 1\rangle 0\rangle$	Yellow
$ 1\rangle 1\rangle$	Green
$ +\rangle 0\rangle$	Light Green
$ +\rangle 1\rangle$	Orange
$ -\rangle 0\rangle$	Purple
$ -\rangle 1\rangle$	Cyan
Beamsplitter	
Spiegel	
Hologramm	
Pentaprisma	
Hadarmad	



Quantencomputer, Algorithmen und das OAM



Jugend forscht Wettbewerb 2025

Lüneburg



(Bild 1) Die Abbildung zeigt das Prinzip eines C-NOT-Gatters im Quantencomputer. Auch findet man auf der Abbildung zwei 3D gedruckte Bauteile: in dem Gatter die erste Bodenhalterung einschließlich Stab für das Dove Prisma nach dem Laser, und die hellgrüne Halterung auf der rechten Seite der Abbildung

Teilnehmer:	Jannes Ruder (15) Johann Wischner (17)
Erarbeitungsort:	Himmelpforten, Stade
Projektbetreuer:	Dr. Hans Otto-Carmesin
Fachgebiet:	Technik
Wettbewerbssparte:	Jugend forscht
Bundesland:	Niedersachsen
Wettbewerbsjahr:	2025

Kurzfassung

Quantencomputer, Algorithmen und das OAM

Unsere Arbeit ist die Weiterführung eines ein Quantencomputer-Demonstrator-Projekts aus dem Vorjahr von Philipp Schöneberg, Phil Gustke und Jannes Ruder. Quantencomputer nutzen Superposition und Verschränkung, um bestimmte Probleme schneller zu lösen als klassische Computer. Wir haben gezeigt, wie Hologramme genutzt werden, um Licht mit Orbital Angular Momentum zu erzeugen. Dieses eignet sich neben der Polarisation als zweites Qubit. Andererseits haben wir theoretisch wie experimentell nachgewiesen, dass auch optische Quantencomputer, die mit Laserlicht arbeiten, das Qubit-Prinzip nutzen können. Diese verzichten durch Nutzung der Lasertechnologie auf Einzel-Photonen und bieten so einen besonders einfachen und eleganten sowie günstigen Aufbau ohne Verlust der Funktion der Qubits.

Inhaltsverzeichnis

Kurzfassung.....	2
1. Einleitung.....	4
2. Gatter und Methode.....	5
2.1 Quantengatter.....	5
2.1.1 CNOT-Gatter.....	5
2.1.2 Hadamard-Gatter.....	6
2.1.3 NOT-Gatter.....	6
2.1.5 Z-Gatter.....	7
2.2 Das OAM.....	7
2.2.1 Einführung in den Bahndrehimpuls (OAM).....	7
2.2.1.1 Definition und Eigenschaften des Bahndrehimpulses.....	7
2.2.2 Erzeugung von OAM-Zuständen.....	8
2.2.2.1 Theoretische Grundlagen der OAM-Erzeugung.....	8
2.2.2.2 Zielzustände: $\ell = +1$ und $\ell = -1$	9
2.2.3 Hologramme als Werkzeuge zur OAM-Erzeugung.....	9
2.2.3.1 Rolle der Hologramme in der OAM-Manipulation.....	9
2.2.3.2 Probleme bei der Herstellung von Hologrammen.....	9
2.2.3.3 Lösung: Microsheet Hologramme.....	10
2.2.3.5. Integration in den Quantencomputer-Demonstrator.....	10
2.2.3.5.1 Kombination von Polarisation und OAM als Qubits.....	10
3 Algorithmen.....	10
3.1 Deutsch-Algorithmus.....	10
3.1.1 Theorie.....	10
3.1.2 Aufbau.....	11
3.1.3 Durchführung.....	11
3.1.4 Ergebnisse.....	12
3.2 Erklärung und Verallgemeinerung.....	12
3.2.1. Beweismethode.....	12
3.2.2. Step 1.....	12
3.2.3. Step 2.....	13
3.2.4. Step 3.....	13
3.2.5. Step 4.....	13
3.2.6.1. Step 5a.....	13
3.2.6.2. Step 5b.....	14
3.2.6.3. Step 5c.....	14
3.2.7. Step $U_f(b\rangle)$	14
3.2.8. Fazit.....	14
3.3 Deutsch-Jozsa.....	15
3.3.1 Theorie.....	15
3.3.2 Aufbau.....	16
3.3.3. Durchführung.....	16
4. Ergebnisse und Ziele.....	17
4.1 Fazit.....	17
4.2 Ziele.....	17
5. Quellen- und Literaturverzeichnis.....	18
6. Danksagung.....	19

1. Einleitung

Quantencomputer sind eine revolutionäre Technologie, die die Grenzen der klassischen Informatik überschreiten. Basierend auf den Prinzipien der Quantenmechanik versprechen sie, komplexe Probleme zu lösen, die für herkömmliche Computer nahezu unlösbar sind. Diese Arbeit untersucht die Funktionsweise von Quantencomputern, Algorithmen und Quantengattern.

Unser Ziel für dieses Jahr war es das OAM als unabhängiges Qubit ausgearbeitet und einsatzfähig zu machen, wir hatten uns auch vorgenommen das Projekt umweltfreundlicher zu gestalten. Außerdem haben wir uns vorgenommen Algorithmen auf unseren Quanten Computer rechnen zu lassen, wobei unser Fokus auf dem Deutsch-Jozsa-Algorithmus lag.

2. Gatter und Methode

In diesem Abschnitt stellen wir an Beispielen wesentliche Methoden und Gatter dar.

2.1 Quantengatter

Quantengatter sind fundamentale Bausteine im Quantencomputing, die es ermöglichen, die Zustände von Qubits gezielt zu verändern und zu verschränken. In unserer Arbeit am Quantencomputer haben wir mit den Gattern {CNOT, H, T, Not, Z} geforscht, um einen Algorithmus zu entwerfen.

2.1.1 CNOT-Gatter

Die Controlled-NOT-Abbildung oder kurz C-NOT- beziehungsweise CX-Abbildung führt eine NOT-beziehungsweise X-Operation am Zielqubit durch, welche durch den Zustand des Kontrollqubits kontrolliert wird. Dementsprechend werden zwei Qubits benötigt, welche durch die Abbildung miteinander verknüpft werden.

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Die mathematische Umsetzung als Matrix ist entsprechend vierdimensional und hängt davon ab, ob Qubit $|q0\rangle$ oder Qubit $|q1\rangle$ als kontrollierendes Qubit verwendet wird.

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

Will man dies nun ausrechnen, erkennt man das sich die Polarisation des Zielqubits ändert, wenn Qubit $|q1\rangle$ als kontrollierendes Qubit verwendet wird. (Development Team)

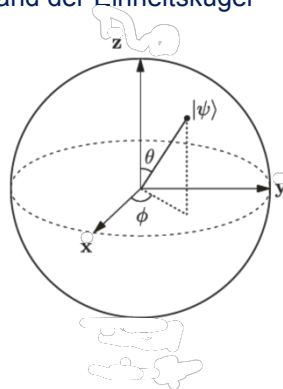
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

2.1.2 Hadamard-Gatter

Die Hadamard-Abbildung ist eine Einzel-Qubit-Abbildung, welche es ermöglicht, ein Qubit von einem Basiszustand in eine uniforme Superposition und wieder zurückzusetzen.

Prinzipiell lässt sich die Abbildung anhand der Einheitskugel



(Bild 2)

Die Abbildung zeigt eine Einheitskugel, die den Hilbert-Raum darstellt.

durch eine Rotation um die X + Z-Achse (Roy) erklären und resultiert deshalb beispielsweise ausgehend vom Startzustand $|0\rangle$ ($|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$) im Zustand $|+\rangle$ ($|+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$). Da die Hadamard-Abbildung in den späteren Berechnungen mit zwei Qubits, nicht nur mit sich selbst verknüpft wird, sondern teilweise auch auf einzelne Qubits angewendet wird, zeigen wir hier unsere drei verschiedenen Verknüpfungen. Die erste ist die Verknüpfung zweier Hadamard-Abbildungen, welche eine Anwendung der Abbildung auf beide Qubits ermöglicht. Die zweite ist die Verknüpfung einer Hadamard-Abbildung und einer Einheitsmatrix. Diese ermöglicht eine Anwendung der Hadamard-Abbildung nur auf das erste Qubit, hier $|q_0\rangle$, wobei das zweite Qubit $|q_1\rangle$ keine Rolle spielt. Die dritte Möglichkeit ist die ausschließliche Anwendung der Hadamard-Abbildung auf das zweite Qubit des Systems, wofür die Einheitsmatrix und die Hadamard-Abbildung in einer anderen Reihenfolge verknüpft werden.

Für diese Abbildung nutzen wir das um $22,5^\circ$ gedrehte $\frac{\lambda}{2}$ Plättchen für das Qubit mit Polarisation.

(Universum Physik Sekundarstufe 2) Rechnerisch sieht die Abbildung wie folgt aus:

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

2.1.3 NOT-Gatter

Das NOT-Gatter ermöglicht, dass ein Qubit von einem Basiszustand in einen anderen Basiszustand versetzt wird.

Anhand der Einheitskugel kann man sich das NOT-Gatter erklären.

Die NOT-Abbildung sorgt dafür, dass sich der Basiszustand zum entgegengesetzten spiegelt. Die sorgt zum Beispiel, dass sich der Basiszustand 0, nachdem dieser auf ein NOT-Gatter angewendet wird,

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \times \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{bmatrix} 0 \times 1 + 1 \times 0 \\ 1 \times 1 + 0 \times 0 \end{bmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

zu dem Basiszustand 1 ($|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$) wechselt.

2.1.5 Z-Gatter

Das Z-Gatter, auch als Pauli-Z-Gatter bekannt, ist ein fundamentales Quantengatter, das auf ein Qubit angewendet wird. Es wird durch die Matrix Z

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

Dargestellt und rotiert den Qubit-Zustand um π (180°) um die z-Achse der Einheitskugel. In der Dirac-Notation transformiert das Z-Gatter die Basiszustände wie folgt:

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \times \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{bmatrix} 1 \times 1 + 0 \times 0 \\ 0 \times 1 + -1 \times 0 \end{bmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \times \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{bmatrix} 1 \times 0 + 0 \times 1 \\ 0 \times 0 + -1 \times 1 \end{bmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

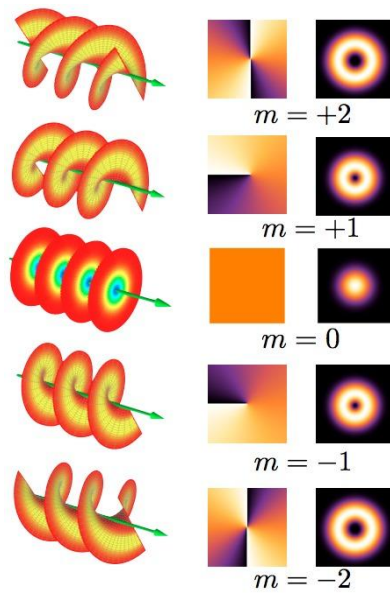
Dies bedeutet, dass das Z-Gatter die Phase des $|1\rangle$ -Zustands invertiert, während der $|0\rangle$ -Zustand unverändert bleibt.

2.2 Das OAM

2.2.1 Einführung in den Bahndrehimpuls (OAM)

2.2.1.1 Definition und Eigenschaften des Bahndrehimpulses

Der Bahndrehimpuls ist ähnlich wie der Spin-Drehimpuls eine Form des elektromagnetischen Drehimpulses. Er beschreibt die Verteilung der Wellenfront entlang der Azimutalachse - bei einem normalen Laser also eine plane Ebene. Im Gegensatz zum Spin welcher nur + oder - sein kann wird der Bahndrehimpuls in topologischer Ladung (ℓ) angegeben, welche Zustände von $+n$ bis $-n$ annehmen kann und so eine höhere Zustandsvielfalt bietet. Licht mit einer topologischen Ladung größer 0 ist aufgrund der auffälligen, ringförmigen Struktur gut erkennbar.

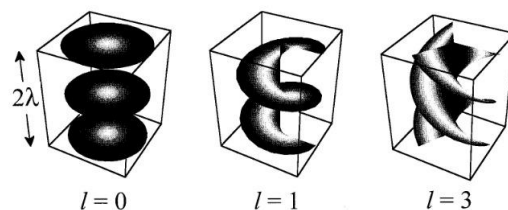


(Bild 3) Die Abbildung zeigt Lichtwellenfronten mit verschiedenen topologischen Ladungen m von -2 bis +2. Auf der linken Seite eine dreidimensionale Darstellung der helikalen Struktur, in der Mitte die Phasenfront und rechts die Intensitätsverteilung am Schirm.

2.2.2 Erzeugung von OAM-Zuständen

2.2.2.1 Theoretische Grundlagen der OAM-Erzeugung

Die topologische Ladung beschreibt die Anzahl an Drehungen pro Wellenlänge.



(Bild 4) Die Abbildung zeigt Wellenfronten für verschiedene topologische Ladungen $l(0, 1, 3)$ in einer Strecke von $\frac{\lambda}{2}$.

Das Licht wird also in eine helikale Struktur gebracht, die durch den Phasenterm $\exp(il\varphi)$ beschrieben wird. Dementsprechend sind auch Zustände wie $l = 10$ möglich, praktisch aber nicht mehr gut umsetzbar da bei größerer topologischer Ladung ein Intensitätsverlust zu vermerken ist. Die Ringstruktur ist eine Folge der Phasenverschiebungen, welche zur Auslöschung der Lichtintensität im Kern führen.

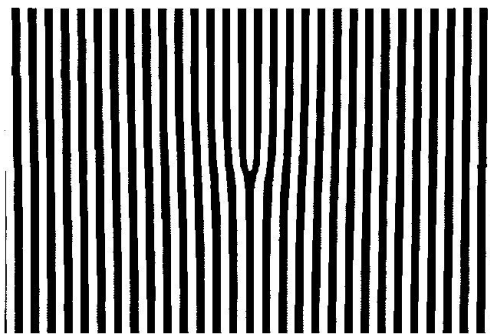
2.2.2.2 Zielzustände: $\ell = +1$ und $\ell = -1$

Positive und negative topologische Ladungen unterscheiden sich in ihrer Drehrichtung: linksherum oder rechtsherum. Trotz dessen, dass der Bahndrehimpuls mehr Möglichkeiten bietet, bleiben wir vorerst beim Bit-System und haben so als Zielzustände nur $\ell = +1$ und $\ell = -1$. Die Drehung (rechts- / linksherum) lässt sich mithilfe von Spiegeln beeinflussen, so wird z.B. aus $\ell = +1$, wenn es gespiegelt wird, $\ell = -1$. Spiegel funktionieren wie Gatter.

2.2.3 Hologramme als Werkzeuge zur OAM-Erzeugung

2.2.3.1 Rolle der Hologramme in der OAM-Manipulation

Um Licht mit topologischer Ladung größer 0 zu erzeugen, eignen sich computergenerierte Hologramme mit einem Gabelmuster.

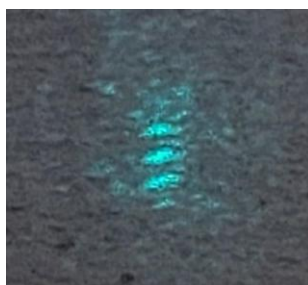


(Bild 5) Die Abbildung zeigt ein Computer generiertes Hologramm. Wird es mit einer ebenen Wellenfront beleuchtet, gibt es eine Welle mit helikalen Wellenfronten aus.

Beim Auftreffen eines Lichtstrahls ist der Beugungswinkel auf der einen Seite größer. Das hat eine Phasenverschiebung zur Folge, welche sich spiralförmig um das Zentrum des Strahls ändert.

2.2.3.2 Probleme bei der Herstellung von Hologrammen

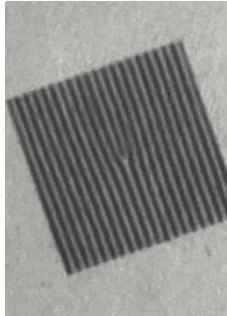
Die Beschaffung der Hologramme hat sich vorerst als schwierig erwiesen. Unsere ersten Exemplare haben wir bei örtlichen Druckereien besorgt. Diese Exemplare haben allerdings nicht die gewünschte Qualität gehabt, was zur Folge hatte, dass die Ringstruktur nicht zu erkennen war



(Bild 6/7) Die Abbildung 7 zeigt ein Foto vom Beugungsmuster, gegeben durch das Hologramm von nicht ausreichender Qualität. Man erkennt keine Ringstruktur. Die Abbildung 6 zeigt ein Foto von einem Hologramm mit ungenügender Qualität. Zu erkennen sind die Unregelmäßigkeiten in den schwarzen Linien.

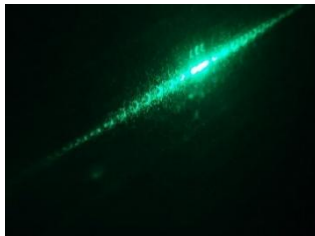
2.2.3.3 Lösung: Microsheet Hologramme

Letztendlich haben wir bei einem Mikrofilm-Unternehmen (Wandel: <https://www.dd-wandel.de>) angefragt, was eine teurere Herstellung bedeutet, aber sich auch lohnt, da die Exemplare eine gute Qualität haben und auch die Ringstruktur klar zu erkennen ist.



(Bild 8) Die Abbildung zeigt ein Foto von einem Hologramm hergestellt mit Mikrofilmtechnologie. Man erkennt klare durchgängige Linien von hoher Qualität.

Selbst die Unterscheidung zwischen Links- und Rechtsdrehung ist mit dem bloßen Auge möglich



(Bild 9) Die Abbildung zeigt ein Foto vom Beugungsmuster, gegeben durch das Hologramm von hoher Qualität. Die Ringstruktur ist bei dem äußeren Maximum klar zu erkennen. Außerdem erkennt man, dass es links vom nullten Maximum eine Rechtsdrehung und rechts von dem nullten Maximum eine Linksdrehung in der Wellenfront gibt.

2.2.3.5. Integration in den Quantencomputer-Demonstrator

2.2.3.5.1 Kombination von Polarisation und OAM als Qubits

Da Licht einen Bahndrehimpuls hat, und man diesen mit einfachen Bauteilen wie Hologrammen gezielt beeinflussen kann, eignet sich dieser neben der Polarisation gut als zweites Qubit.

3 Algorithmen

3.1 Deutsch-Algorithmus

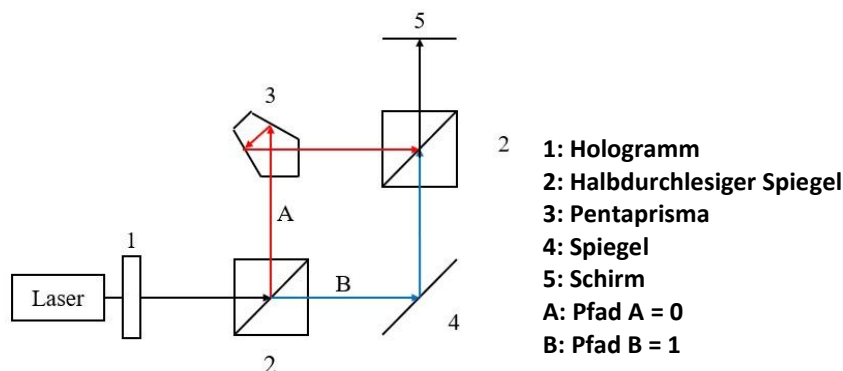
3.1.1 Theorie

Der Deutsch-Algorithmus ist ein Spezialfall des Deutsch-Jozsa-Algorithmus, für welchen die Eingabe ein Bit groß ist. Es gibt also nur die Eingabemöglichkeiten 0 oder 1. Der Algorithmus soll überprüfen, ob eine binäre Funktion konstant oder balanciert ist. Konstant bedeutet für jedes x ist $f(x)$ gleich: entweder $f(x) = 0$ oder $f(x) = 1$. Balanciert bedeutet $f(x) = x$ oder $f(x) = NOT(x)$. Daraus ergeben sich zwei

konstante Funktionen f_a , f_b und zwei balancierte f_c , f_d . Das interessante am Algorithmus ist, dass man vorab kein Wissen über f hat, sondern nur eine Eingabe $x \in \{0, 1\}$ machen kann und die Ausgabe $f(x)$ überprüfen kann. Ein klassischer Computer müsste also zwei Eingaben $f(0)$ und $f(1)$ machen und die Ausgaben vergleichen, um zu überprüfen, ob die Funktion konstant oder balanciert ist. Der Quantencomputer kann das Problem mit nur einer Eingabe lösen, indem er die beiden Zustände $|0\rangle$ und $|1\rangle$ in eine Superposition versetzt. Dadurch wird mit nur einer Eingabe genug Information gesammelt.

3.1.2 Aufbau

Der Aufbau eines Deutsch-Algorithmus, also eines Deutsch-Jozsa mit einem Qubit ähnelt in der praktischen Umsetzung dem CNOT-Gatter. Wir behalten den Grundaufbau bei



(Bild 10) Die Abbildung zeigt den Aufbau des Deutsch-Algorithmus für die balancierte Funktion f_d . 1 ist das Hologramm, 2 sind Polwürfel, 3 ein Pentaprisma und 4 ein Spiegel. 5 ist der Schirm, an dem das Ergebnis ausgelesen wird.

und ändern lediglich je nach Funktion (f_a bis f_d) die Operation auf den Pfaden, sprich wann wo entweder ein Spiegel oder ein Pentaprisma steht. Das resultiert in folgenden Aufbauten:

- f_a : Spiegel auf beiden Pfaden.
- f_b : Pentaprisma auf beiden Pfaden.
- f_c : Spiegel auf Pfad A und Pentaprisma auf Pfad B.
- f_d : Pentaprisma auf Pfad A und Spiegel auf Pfad B.

Damit der Algorithmus auch wie gewünscht funktioniert wird als erstes ein Hadamard-Gatter auf die Polarisation angewendet, damit das Licht auch beide Pfade durchläuft.

3.1.3 Durchführung

Um den Versuch gemäß der Theorie durchzuführen, gibt man mit einem horizontalen Polarisationsfilter $|x\rangle = |0\rangle$ ein und wendet dann ein Hadamard-Gatter ($\frac{1}{2}$ -Plättchen) auf den Zustand an. Dadurch liegt eine Überlagerung von Horizontaler und vertikaler Polarisation vor. Hierdurch werden beide Wege

gleichzeitig abgetastet. Außerdem erzeugt man mit dem Hologramm einen Bahndrehimpuls $\ell = -1$. Nun kann man am Schirm ablesen, ob es sich um eine konstante $f(0) = f(1)$ oder balancierte $f(0) \neq f(1)$ Funktion handelt: fa erzeugt auf beiden Wegen $\ell = +1$. fb erzeugt auf beiden Wegen $\ell = -1$. fc sowie fd erzeugt eine Überlagerung von $\ell = +1$ und $\ell = -1$

3.1.4 Ergebnisse

Wir konnten zeigen, dass eine Umsetzung des Deutsch-Algorithmus auch mit Laserlicht möglich ist.

3.2 Erklärung und Verallgemeinerung

3.2.1. Beweismethode

Die Tabellen zeigen die Rechenschritte des Deutsch($A(1)$) und Deutsch-Jozsa-Algorithmus für $n = 2$ Eingaben ($A(2)$). Schritt für Schritt analysieren wir die Tabellen und mithilfe der vollständigen Induktion beweisen wir die Funktionalität des Algorithmus. Die vollständige Induktion beschreibt das Verfahren, bei dem man sich anguckt, was bei $A(1)$ passiert und was sich zwischen $A(1)$ und $A(2)$ ändert. Diesen Zwischenschritt verallgemeinert man zu $A(n + 1)$, wobei man die Annahme tätigt, dass n durch jeden ganzzahligen Wert k ersetzt werden kann. Schlussendlich beweist man also auf der Grundlage von $A(1)$ und $A(1 + 1)$ was für $A(n + 1)$ passieren würde. Das wäre die vollständige Induktion. Wir zeigen nur eine beinahe vollständige Induktion. Das heißt wir analysieren wie bei einer vollständigen Induktion $A(1)$ und gucken uns an was zu $A(2)$ passiert, wenden die Erkenntnisse dann aber lediglich auf $A(3)$ an und nicht allgemein auf $A(n + 1)$. Soweit es uns möglich ist, probieren wir dennoch die Schritte zu verallgemeinern.

3.2.2. Step 1

Im ersten Schritt legen wir einen Testzustand fest bei dem alle $x_n \rightarrow |0\rangle$ und $y \rightarrow |1\rangle$. Für $n = 1$ also: $|x_1\rangle, |y\rangle = |0\rangle, |1\rangle$ und für $n = 2$: $|x_1\rangle, |x_2\rangle, |y\rangle = |0\rangle, |0\rangle, |1\rangle$. Für $n = 3$ müsste man nur $x_3 = |0\rangle$ hinzufügen. Verallgemeinert lautet der Schritt: $|x_1\rangle \dots |x_n\rangle, |y\rangle = |0\rangle \dots |0\rangle, |1\rangle$.

3.2.3. Step 2

In Schritt zwei werden die Superpositionen nach der Anwendung eines Hadamard-Gatters wie folgt definiert: $H(|0\rangle) = |0\rangle + |1\rangle \div \sqrt{2} = |+\rangle$ und $H(|1\rangle) = |0\rangle - |1\rangle \div \sqrt{2} = |-\rangle$.

3.2.4. Step 3

In Schritt drei wird das Tensorprodukt der Qubits in Superposition ausgerechnet. Für $n = 1$ also $H(x1) \otimes H(y)$ und für $n = 2$ $H(x1) \otimes H(x2) \otimes H(y)$. Für $n = 3$, müsste man dem Term noch $(\otimes)H(x3)$ hinzufügen. Eine Verallgemeinerte Darstellung gelingt durch: $(\bigotimes_{i=1}^n H(x_i)) \otimes y$. Der Term rechnet mithilfe der Zählvariable i von 1 bis n das Tensorprodukt der n Qubits aus. Das Ergebnis daraus bildet alle möglichen Kombinationen der Qubits ab. Für ein Qubit also 0 und 1. Dieses Ergebnis wird dann noch $\otimes y$ gerechnet, um alle Kombinationen von x und y darzustellen. Für $n = 1$ also 00, 01, 10, 11. Da $\sqrt{2}$ kein Tensor ist wird es einfach $n + 1$ -mal mit sich selbst multipliziert, woraus sich unter dem Bruch für $n = 1$ $\sqrt{2^{1+1}}$ und für $n = 2$: $\sqrt{2^{2+1}}$ ergibt. Für $n = 3$ also $\sqrt{2^{3+1}}$. Allgemein gilt dementsprechend $\sqrt{2^{n+1}}$. Der Bruch aus 3b) wird als Zustand $|b\rangle$ definiert.

3.2.5. Step 4

In Schritt vier definiert man die Abbildung Uf , in der das Qubit y mit den Qubits x_n durch den Term $f(x_1, \dots, x_n) \oplus y$ verbunden wird.

3.2.6.1. Step 5a

In Schritt fünf wendet man die Abbildung Uf auf den Zustand $|b\rangle$ aus Schritt 3 an. Das bedeutet, wenn man in 3 den Zustand $|x = 0\rangle|y = 1\rangle$ hat, wird aus $y \rightarrow f(x) \oplus y$. Aus $|0\rangle|0\rangle$ wird also $|0\rangle|f(0) \oplus 0\rangle$. Das wird für jede Kombination $|x\rangle|y\rangle$ gemacht. Daraus ergibt sich dann der Term in 5a). Für $n = 1$ hat man wie bereits gesagt die Kombinationen 00, 01, 10, 11, bei $n = 2$ von 000 bis 111. In diesem Schritt wird immer aus dem letzten Qubit y der Kombination: $f(x) \oplus y$. Für $n = 3$ erweitert man die Kombinationen lediglich auf vier Stellen also von 0000 bis 1111. Allgemein ergeben sich also immer die Binärzahlen von 0 bis $(2n + 1 - 1)$.

3.2.6.2. Step 5b

In diesem Teilschritt vereinfachen wir den Term aus 5a indem wir x ausklammern. Im Term aus 5a sehen wir, dass wir in $|x\rangle|f(x) \oplus 0\rangle$ und in $|x\rangle|f(x) \oplus 1\rangle$ haben. Dieses $|x\rangle$ kann man ausklammern. Daraus ergibt sich: $|x\rangle[|f(x) \oplus 0\rangle - |f(x) \oplus 1\rangle]$. Es gilt: $f(x) \oplus 0 = f(x)$. Daraus ergibt sich die gekürzte Version aus 5b.

3.2.6.3. Step 5c

Hier wird weiter gekürzt und eine Schlüsseleigenschaft des Deutsch-Jozsa-Algorithmus macht sich erkenntlich. Aus $[|f(x)\rangle - |f(x) \oplus y\rangle]$ wird $(-1)^{f(x)[|0\rangle - |1\rangle]}$. Diese Umstellung des Terms ergibt sich durch eine Verschiebung der Amplitudeninformation in die Phaseninformation. Der Term $|0\rangle - |1\rangle$ bleibt unverändert, da er die Superposition in Abhängigkeit von der Phase $(-1)^{f(x)}$ abbildet. Das $\oplus 1$ invertiert. Wenn $f(x) = 0 : [|0\rangle - |1\rangle]$, und wenn $f(x) = 1 : -|0\rangle - |1\rangle]$. Die Schlüsseleigenschaft besteht eben darin, dass alle möglichen Funktionen für $n = 1: 0, 1$, für $n = 2: 00, 01, 10, 11$ und für $n = 3$ wäre es: $000, 0001...111$ - vorhanden sind. Das bedeutet, dass obwohl man nur $f(1)$ eingibt, man eine Ausgabe für alle möglichen Funktionen erhält. Das zeigt sich auch beim Experiment ganz gut. So tätigt man zwar eine Eingabe ($x_1 = 0, x_2 = 0$), aber dennoch kann man am Schirm alle vier Funktionen $(00, 01, 10, 11)$ ablesen.

3.2.7. Step $U_f(|b\rangle)$

Dieser Schritt ist ausschließlich eine Verkürzung des Terms. So erkennt man das in Schritt 5c) mehrfach $|0\rangle - |1\rangle$ vorkommt, was durch eine Multiplikation mit $1/\sqrt{2}$ zur Superposition $|-\rangle$ wird (siehe Schritt 2).

3.2.8. Fazit

Es zeigt sich, dass die Zwischenschritte, also von $A(1)$ zu $A(2)$ eigentlich immer demselben Schema folgen und der Term bloß für mehr Qubits länger wird. Im Endeffekt hat man immer dasselbe Ergebnis, nämlich einen Term, der alle möglichen x für $f(x)$ enthält und somit die Überlegenheit der Quantencomputer zeigt. Es ist eben so, dass der Quantencomputer alle Eigenschaften simultan überprüfen kann, anstatt jede Möglichkeit einzeln zu überprüfen.

Step	A(1)
1	$ x_1\rangle, y\rangle = 0\rangle, 1\rangle$
2	$H(0\rangle) = \frac{1}{\sqrt{2}} \cdot 0\rangle + 1\rangle / H(1\rangle) = \frac{1}{\sqrt{2}} \cdot 0\rangle - 1\rangle$
3a	$H(x_1\rangle) \otimes H(y\rangle) = H(0\rangle) \otimes H(1\rangle)$
3b	$\frac{ 0\rangle 0\rangle - 0\rangle 1\rangle + 1\rangle 0\rangle - 1\rangle 1\rangle}{\sqrt{2^{1+1}}} := b\rangle$
4	$U_f : (x_1\rangle, y\rangle) := (x_1\rangle, f(x_1) \oplus y\rangle)$
5a	$U_f(b\rangle) = \frac{1}{\sqrt{2^{1+1}}} \cdot 0\rangle f(0) \oplus 0\rangle - 0\rangle f(0) \oplus 1\rangle + 1\rangle f(1) \oplus 0\rangle - 1\rangle f(1) \oplus 1\rangle$
5b	$\frac{1}{2} \cdot 0\rangle \cdot [f(0)\rangle - f(0) \oplus 1\rangle] + 1\rangle \cdot [f(1)\rangle - f(1) \oplus 1\rangle]$
5c	$\frac{1}{2} \cdot 0\rangle \cdot (-1)^{f(0)}[0\rangle - 1\rangle] + 1\rangle \cdot (-1)^{f(1)}[0\rangle - 1\rangle]$
$U_f(b\rangle)$	$\frac{1}{\sqrt{2}} \cdot (-1)^{f(0)} 0\rangle -\rangle + (-1)^{f(1)} 1\rangle -\rangle$
$U_f(b\rangle)$	$\frac{1}{\sqrt{2}} \cdot [-\rangle \cdot [(-1)^{f(0)} 0\rangle + (-1)^{f(1)} 1\rangle]]$

Step	A(2)
1	$ x_1\rangle, x_2\rangle, y\rangle = 0\rangle, 0\rangle, 1\rangle$
2	$H(0\rangle) = \frac{1}{\sqrt{2}} \cdot 0\rangle + 1\rangle / H(1\rangle) = \frac{1}{\sqrt{2}} \cdot 0\rangle - 1\rangle$
3a	$H(x_1\rangle) \otimes H(x_2\rangle) \otimes H(y\rangle) = H(0\rangle) \otimes H(0\rangle) \otimes H(1\rangle)$
3b	$\frac{ 000\rangle - 001\rangle + 010\rangle - 011\rangle + 100\rangle - 101\rangle + 110\rangle - 111\rangle}{\sqrt{2^{2+1}}} := b\rangle$
4	$U_f := (x_1\rangle, y\rangle) := (x_1\rangle, x_2\rangle, f(x_1, x_2) \oplus y\rangle)$
5a	$U_f(b\rangle) = \frac{1}{\sqrt{2^{2+1}}} \cdot 0\rangle 0\rangle f(0 \cdot 0) \oplus 0\rangle - 0\rangle 0\rangle f(0 \cdot 0) \oplus 1\rangle + 0\rangle 1\rangle f(0 \cdot 1) \oplus 0\rangle - 0\rangle 1\rangle f(0 \cdot 1) \oplus 1\rangle \dots$ $\dots + 1\rangle 0\rangle f(1 \cdot 0) \oplus 0\rangle - 1\rangle 0\rangle f(1 \cdot 0) \oplus 1\rangle + 1\rangle 1\rangle f(1 \cdot 1) \oplus 0\rangle - 1\rangle 1\rangle f(1 \cdot 1) \oplus 1\rangle$
5b	$U_f(b\rangle) = \frac{1}{\sqrt{2^{2+1}}} \cdot 0\rangle 0\rangle \cdot [f(0,0)\rangle - f(0,0) \oplus 1\rangle] + 0\rangle 1\rangle \cdot [f(0,1)\rangle - f(0,1) \oplus 1\rangle] \dots$ $\dots + 1\rangle 0\rangle \cdot [f(1,0)\rangle - f(1,0) \oplus 1\rangle] + 1\rangle 1\rangle \cdot [f(1,1)\rangle - f(1,1) \oplus 1\rangle]$
5c	$\frac{1}{2\sqrt{2}} \cdot 0\rangle 0\rangle \cdot (-1)^{f(0,0)}[0\rangle - 1\rangle] + 0\rangle 1\rangle \cdot (-1)^{f(0,1)}[0\rangle - 1\rangle] \dots$ $\dots + 1\rangle 0\rangle \cdot (-1)^{f(1,0)}[0\rangle - 1\rangle] + 1\rangle 1\rangle \cdot (-1)^{f(1,1)}[0\rangle - 1\rangle]$
$U_f(b\rangle)$	$\frac{1}{2} \cdot (-1)^{f(0,0)} 0\rangle 0\rangle -\rangle + (-1)^{f(0,1)} 0\rangle 1\rangle -\rangle + (-1)^{f(1,0)} 1\rangle 0\rangle -\rangle + (-1)^{f(1,1)} 1\rangle 1\rangle -\rangle$
$U_f(b\rangle)$	$\frac{1}{2} \cdot [-\rangle \cdot [(-1)^{f(0,0)} 0\rangle 0\rangle + (-1)^{f(0,1)} 0\rangle 1\rangle + (-1)^{f(1,0)} 1\rangle 0\rangle + (-1)^{f(1,1)} 1\rangle 1\rangle]]$

(Tabelle 1) Die Tabellen zeigen die jeweiligen Algorithmen: Deutsch Algorithmus (oben) und Deutsch-Jozsa-Algorithmus mit zwei Eingabequbits (unten). Es handelt sich um selbstgeschriebene Mathematik bzw. Quantenalgorithmen in der Dirac-Notation. Sie werden in 3.2 Erklärung und Verallgemeinerung Schritt für Schritt erklärt.

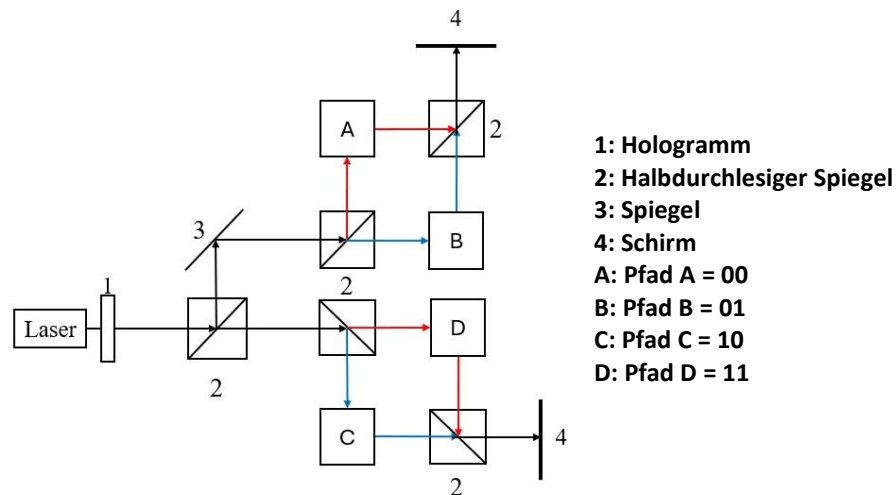
3.3 Deutsch-Jozsa

3.3.1 Theorie

Wie bereits erklärt ist der Deutsch-Jozsa-Algorithmus nur eine Erweiterung des Deutsch-Algorithmus auf n Qubits. Er hat dasselbe Ziel eine Funktion auf konstant oder balanciert zu überprüfen. Es handelt sich also um eine Verallgemeinerung des Problems. Ein klassischer Computer bräuchte im schlimmsten Fall $(n \div 2) + 1$ Eingaben: also die Hälfte der möglichen Eingaben plus eins, um sicherzustellen, ob die Funktion konstant oder balanciert ist. Ein Quantencomputer benötigt wieder nur eine Eingabe.

3.3.2 Aufbau

Um zu zeigen, wie man theoretisch einen Aufbau für n Qubits umsetzen könnte haben wir erstmal den Aufbau für 2 Qubits als $x \in \{00,01,10,11\}$ zu umgesetzt. Dafür erweitert man einfach den Aufbau für den Deutsch-Algorithmus um eine Verzweigungstiefe



(Bild 11) Die Abbildung zeigt den allgemeinen Versuchsaufbau des Deutsch-Jozsa-Algorithmus mit 2 Eingabequbits. 1: Hologramm, 2: Polwürfel, 3: Spiegel, 4: Schirm. A, B, C und D stellen die Pfade bzw. Operationsfelder dar, welche abhängig von der Funktion (f_a - f_h) durch Spiegel oder Pentaprismen ersetzt werden. Die erste Eingabe durch Polfilter durchläuft ein Hadamard-Gatter, danach ein Hologramm. Dann geht es zum ersten Polwürfel. Wie beim Deutsch-Algorithmus führt eine ausbalancierte Funktion zu einer Überlagerung von $\ell = +1$ und $\ell = -1$ an wenigstens einen der beiden Schirme.

Wieder werden je nach Funktion die Positionen der Spiegel und Pentaprisma abgewandelt. Da es nun vier Pfade A bis D gibt ist dieser benötigt dieser Aufbau deutlich mehr Materialien. Er zeigt aber gut, wie die Umsetzung für n Qubits funktionieren könnte, so fügt man einfach mehr Verzweigungen hinzu.

3.3.3. Durchführung

Die Versuchs Durchführung ist in der Bildunterschrift von Bild 11 Beschrieben

4. Ergebnisse und Ziele

4.1 Fazit

In diesem Jahr haben wir die Idee des OAMs als unabhängiges Qubit ausgearbeitet und einsatzfähig gemacht. Wir hatten auch daran gearbeitet das Projekt umweltfreundlicher zu gestalten, indem wir damit begannen benötigte Teile selbst aus PLA (Polylactide: $C_3H_4O_2$) zu drucken, anstatt diese zu bestellen (siehe Bild 19).

Außerdem haben wir es geschafft Algorithmen auf unseren Quantencomputer rechnen zu lassen. Wir setzen den Deutsch-Algorithmus, sowie den Deutsch-Jozsa-Algorithmus mit zwei Eingabequbits experimentell um. Theoretisch ist es uns gelungen, den Deutsch-Jozsa-Algorithmus in einer effizienten Schreibweise für einen, zwei und auch arithmetisiert für n Eingabequbits darzustellen.

Zudem konnten wir praktisch beweisen, dass unsere universellen Quantengatter auch auf optischen Quantencomputern mit Laserlicht laufen. Dazu haben wir den Beweis von De Ro 2021 auf unseren Quantencomputer übertragen

Als Fazit lässt sich sagen, dass wir in diesem Jahr, alle im Vorjahr gesetzten Ziele gemeistert haben und einen guten Schritt in Richtung eines einsatzfähigen Quantencomputer gegangen sind.

4.2 Ziele

Unser Ziel für das kommende Jahr ist es den Grover-Algorithmus erstmalig auf unserem Quantencomputer laufen zu lassen. Auch arbeiten wir daran den Quantencomputer zu miniaturisieren, kostengünstiger zu gestalten und eine Schnittstelle zu herkömmlichen Computern zu schaffen.

5. Quellen- und Literaturverzeichnis

Orbital angular momentum of light. Wikipedia, 2024

Padgett, Miles John / Allen, Les: The angular momentum of light: Optical spanners and the rotational frequency shift. Research Gate, 1999

IBM Development Team, Q: Circuit Library. IBM, 2021

Roy, Pradosh: Quantum Logic Gates. Research Gate, 2020

Dilba, Dennis: schaeffler-tomorrow.com, 2019

Ruder, Jannes / Schöneberg, Philipp / Gustke, Phil. Experimentelle und theoretische Analyse von Quantencomputern. Jugend-Forscht, 2024

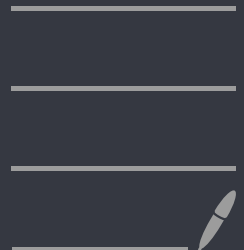
Carmesin, Hans-Otto et al.: Universum Physik Sekundarstufe 2. Cornelsen, 2024

De Ro, Nicolas: Universal Sets of Gates in Quantum Computing. Research Gate, 2021

6. Danksagung

Besonders danken möchten wir Herrn Dr. Hans-Otto Carmesin, Fachleiter AG Jugend forscht am Gymnasium Athenaeum in Stade für die ausgezeichnete Betreuung und Unterstützung bei der Durchführung und Umsetzung der gesamten Arbeit. Außerdem möchten wir uns bei unseren Eltern für ihre moralische Unterstützung bedanken, die uns auf unserem Weg stets motiviert haben.

Experimental and Theoretical Analysis of Quantum Computing



Experimental and Theoretical Analysis of Quantum Computing

Philipp Schöneberg¹, Hans-Otto Carmesin^{1, 2, 3}, Phil Gustke¹ and Jannes Ruder¹

¹Gymnasium Athenaeum, Harsefelder Straße 40, 21620 Stade

²Studienseminar Stade, Bahnhofstr. 5, 21682 Stade

³Universität 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 Gordon Moore 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).

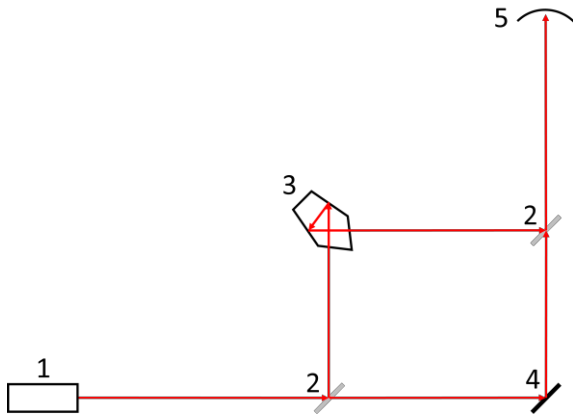


Fig. 1: Schematic experimental setup of the C-NOT gate demonstration: 1. laser, 2. polarizing beamsplitter, 3. pentaprism, 4. mirror, 5. detector

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.

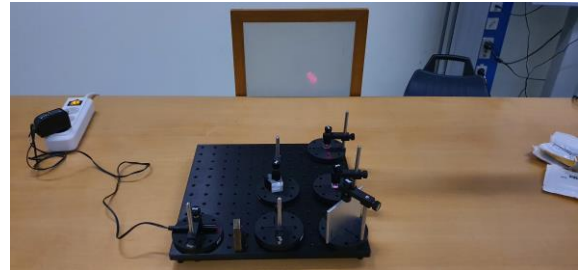


Fig. 5: C-NOT experiment realizing the input qubit states through the usage of a horizontal polarizer and the rotation of the laser, creating an antidiagonal oval.



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).

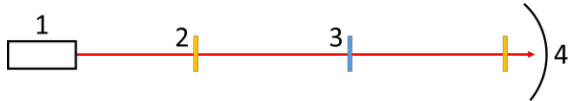


Fig. 7: Schematic experimental setup for the NOT gate demonstration: 1. laser, 2. horizontal polarizer, 3. X gate, 4. detector

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).

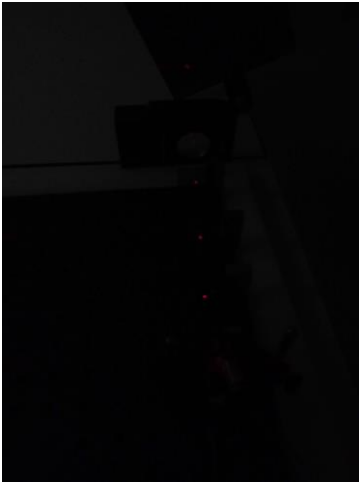


Fig. 8: Realization of the experimental setup in fig. 7 with a diverging lens added before the detector. The point on the detector is barely visible to the naked eye.

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).

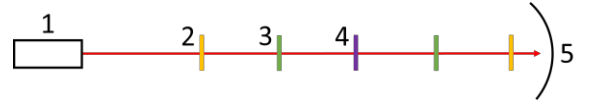


Fig. 9: Schematic experimental setup for the Hadamard and Z gate demonstration: 1. laser, 2. horizontal polarizer, 3. Hadamard gate, 4. Z gate, 5. detector

Therefore, by placing the half wave plates in the given order we transform the polarization qubit from $|0\rangle$ 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).

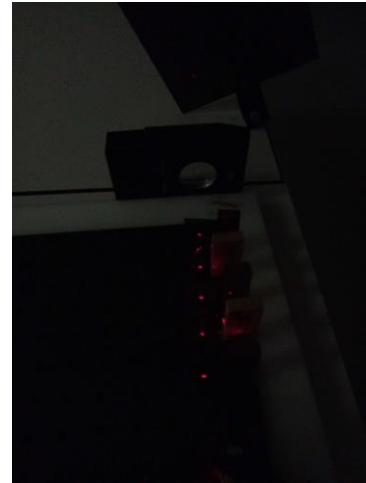


Fig. 10: Realization of the experimental setup in fig. 9 with a scattering lens added before the detector. The point on the detector is barely visible to the naked eye.

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).

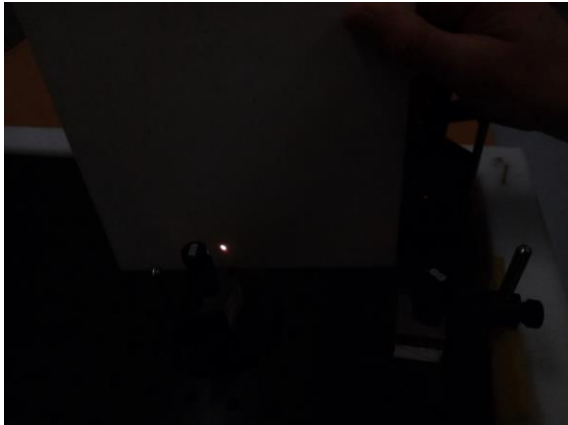


Fig. 11: Intersection of the two laser beams between the pentaprism and the second beamsplitter.

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

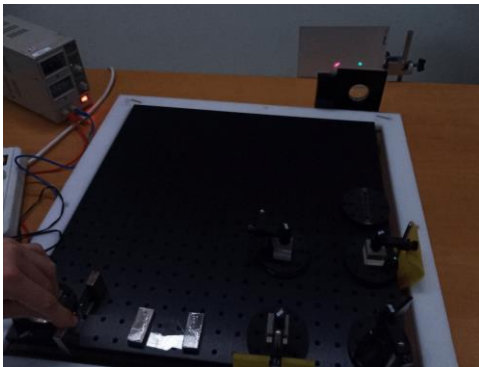


Fig 12: C-NOT experiment realizing the input qubit states $|0\rangle$ and $|0\rangle$ in the red laser as well as $|1\rangle$ and $|0\rangle$ in the green laser. The qubits implemented in the red laser are realized by the usage of a vertical polarizer and the rotation of the laser, creating an antidiagonal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

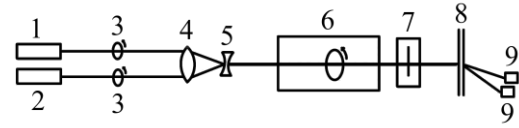


Fig. 13: Exemplary optical setup to use wavelength-based multiplexing for optical computation: 1. laser of wavelength a, 2. laser of wavelength b, 3. wave plate to initialize input state of the given qubit, 4. collecting lens, 5. scattering lens, 6. optical operation, 7. polarizer for measurement, 8. 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 $|1\rangle$ and a balanced function by returning it in the state $|0\rangle$.

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):

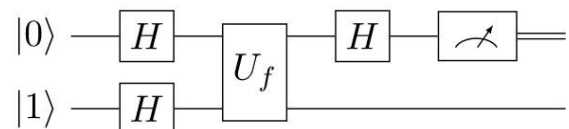


Fig. 14: A diagram of the circuit representing the Deutsch algorithm.

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).

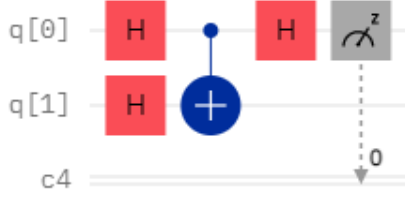


Fig. 15: A diagram of the circuit representing the modified Deutsch algorithm.

To show that the Deutsch 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}).

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

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

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

$$\begin{aligned} |q_0q_1\rangle &= |q_1\rangle \otimes |q_0\rangle = |1\rangle \otimes |0\rangle \\ &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \{3\} \end{aligned}$$

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

$$\begin{aligned} &H_{q_0,q_1} \cdot |q_0q_1\rangle \\ &= \frac{1}{2} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \\ &= \frac{1}{2} \cdot \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \quad \{4\} \end{aligned}$$

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

$$\begin{aligned} &CX_{q_0,q_1} \cdot |q_0q_1\rangle \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \quad \{5\} \end{aligned}$$

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

$$\begin{aligned} &H_{q_0} \cdot |q_0q_1\rangle \\ &= \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \cdot \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \\ &= \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \\ &= |1\rangle \quad \{6\} \end{aligned}$$

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 0x13f3>

#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).

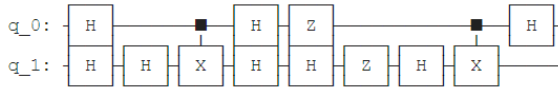


Fig. 18: Schematic quantum circuit for our exemplary version of the Grover algorithm for two qubits.

In this instance, the oracle is selecting the $|11\rangle$ state and is derived from a simple realization by the qiskit 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).

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

$$= \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \{7\}$$

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

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

$$= \frac{1}{2} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \{8\}$$

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).

$$H_{q_1} CX_{q_0, q_1} H_{q_1}$$

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

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

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} = CZ_{q_0 q_1} \{9\}$$

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

$$= \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{pmatrix} \{10\}$$

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

$$\begin{aligned}
 &= \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} \quad \{11\} \\
 H_{q_1} \cdot |q_0 q_1\rangle &= \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} \\
 &= \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \quad \{12\}
 \end{aligned}$$

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).

$$\begin{aligned}
 &H_{q_0, q_01} \cdot |q_0 q_1\rangle \\
 &= \frac{1}{2} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \cdot \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \\
 &= \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \quad \{13\}
 \end{aligned}$$

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).

$$\begin{aligned}
 &\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \\
 &= Z_{q_0, q_01} C Z_{q_0, q_01} = Z_{q_0, q_01} H_{q_1} C X_{q_0, q_01} H_{q_1} \\
 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \\
 &\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \quad \{14\}
 \end{aligned}$$

$$\begin{aligned}
 Z_{q_0, q_01} \cdot |q_0 q_1\rangle &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \\
 &= \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \quad \{15\}
 \end{aligned}$$

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

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

$$\begin{aligned}
 &C X_{q_0, q_01} \cdot |q_0 q_1\rangle \\
 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \quad \{17\}
 \end{aligned}$$

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).

$$\begin{aligned}
 &H_{q_1} \cdot |q_0 q_1\rangle \\
 &= \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \\
 &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |11\rangle \quad \{18\}
 \end{aligned}$$

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 Siemens 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.