FH Aachen
Fachbereich Elektrotechnik und Informationstechnik
Prof. Dr. Marko Schuba

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Bachelorarbeit

Implementierung einer Bibliothek von Quantenalgorithmen zur Kryptoanalyse

Eingereicht von:
Simon Maximilian Kalytta
Matrikelnummer: 3190683

Studienrichtung: Informatik

23. August 2023

Betreuerin: B.Sc. Janis König
Prüfer: Prof. Dr. Marko Schuba

In Kooperation mit it.sec GmbH

Kurzfassung

Schlagwörter: placeholder

Abstract

Keywords: placeholder

Inhaltsverzeichnis

1 Einleitung **3**
1.1 Ziele und Vorgehen . . . 3
1.2 Motivation . . . 4
1.3 Gliederung . . . 4

2 Schwerpunkt **4**
2.1 Entschlüsselungsfunktion für RSA . . . 4

3 Quantencomputer **5**
3.1 Entwicklung . . . 5
3.2 Vorteile gegenüber klassischen Rechnern . . . 8

4 Fundament **8**
4.1 Literatur . . . 8

5 Grundlagen **9**
5.1 Qubits . . . 9
5.2 Quantengatter . . . 10
5.3 Qiskit . . . 10
5.4 RSA . . . 10
5.5 Quanten-Fourier-Transformation . . . 10
5.6 Quanten-Phase-Estimation . . . 14

6 Shor-Algorithmus **20**
6.1 Zweck . . . 20
6.2 Funktionsweise . . . 20
6.3 Ordnungsbestimmung . . . 20
6.4 Klassische Nachberechnung . . . 22

7 Implementierung **23**
7.1 Quantenalgorithmus . . . 23
7.1.1 Addition . . . 24

8 Resultate **28**

9 Verlauf **28**
9.1 Rückblick 28
9.2 Ausblick 28

Abbildungsverzeichnis

1 IBM-Quantum-Roadmap [23b] 7
2 3-Qubit QFT ohne Swaps 14
3 3-Qubit inverse QFT ohne Swaps 14
4 3-Kontroll-Qubit QPE 18
5 3-C-Qubit QPE Messergebnis 18
6 QPE unpräzises Messergebnis 19
7 QPE für Shor 24
8 Quantum-Addition 25
9 Quantum-Addition fixierte Phasenverschiebungen 27

Abkürzungsverzeichnis

Glossar

1 Einleitung

1.1 Ziele und Vorgehen

Das Ziel dieser Arbeit ist die Entwicklung und Implementierung einer auf Quantenalgorithmen basierenden Bibliothek zur Kryptoanalyse aktueller Verschlüsselungsverfahren. Angesichts der Komplexität und des Fachwissens, das zur Entwicklung von Quantenalgorithmen erforderlich ist, konzentriert sich diese Arbeit auf die Implementierung und Anpassung prominenter Quantenalgorithmen aus der Literatur. Die verwendeten Quantenalgorithmen können bestimmte Problemstellungen, die aufgrund ihrer Komplexität eine zentrale Bedeutung in modernen Verschlüsselungsverfahren haben, deutlich schneller lösen als die effizientesten klassischen Algorithmen.

In den zugrunde liegenden wissenschaftlichen Arbeiten werden diese Quantenalgorithmen als abstrakte oder konzeptuelle Algorithmen vorgestellt, ohne dass auf konkrete Implementierungsdetails eingegangen wird. Diese Arbeit beseitigt die Diskrepanz zwischen theoretischen Konzepten und praktischen Realisierungen, indem auf der Basis der abstrakten Quantenalgorithmen eine konkrete Implementierung entwickelt wird. Anschließend werden die resultierenden Implementierungen der Quantenalgorithmen mit

klassischen Algorithmen kombiniert, um die Problemstellungen, die für die Sicherheit der Verschlüsselungsverfahren entscheidend sind, effektiv lösen zu können.

Diese Arbeit implementiert die Bibliothek auf der Abstraktionsebene von Quantenschaltkreisen. Dazu wird das Open-Source-Softwareentwicklungskit Qiskit genutzt das auf der Programmiersprache Python basiert.

1.2 Motivation

In den letzten Jahren haben Fortschritte in der Forschung und Entwicklung von Quantencomputern neue Möglichkeiten für die praktische Untersuchung von Quantenalgorithmen ermöglicht. Gegenwärtig ermöglichen sowohl Simulatoren von Quantencomputer als auch real existierende, wenn auch leistungsbegrenzte, Quantencomputer die Durchführung praktischer Tests. Zur Zeit der ursprünglichen Konzeption der in dieser Arbeit verwendeten Quantenalgorithmen war eine praktische Erprobung entweder undenkbar oder nur durch umständliche Experimente mit stark vereinfachten Versuchen möglich.

Indem diese technologischen Möglichkeiten zur Ausführung und Erprobung der implementierten Quantenalgorithmen genutzt werden, eröffnet sich ein neuer Standpunkt, der die Betrachtung aus anderen Blickwinkeln erlaubt.

1.3 Gliederung

Hier wird dann beschrieben in welchen Kapiteln der Leser welche Inhalte erwarten kann und wieso die Reihenfolge der Kapitel so gewählt wurden, inklusive der Struktur der Arbeit

2 Schwerpunkt

2.1 Entschlüsselungsfunktion für RSA

Der Fokus dieser Arbeit liegt auf der Implementierung einer Entschlüsselungsfunktion, die in der Lage ist, den privaten Schlüssel des RSA-Verfahrens aus dem zugehörigen öffentlichen Schlüssel abzuleiten.

Das RSA-Verfahren stellt ein sogenanntes asymmetrisches Kryptosystem dar. Bei asymmetrischen Kryptosystemen kommt ein mathematisch verknüpftes Schlüsselpaar zum Einsatz. Dieses Schlüsselpaar besteht aus einem öffentlichen und einem privaten Schlüssel, wobei Letzterer ausschließlich dem Eigentümer des Schlüsselpaares zugänglich sein sollte. Mit dem privaten Schlüssel ist der Eigentümer in der Lage, Nachrichten zu signieren. Ein Nutzer kann unter der Verwendung des öffentlichen Schlüssels die Authentizität der signierten Nachricht überprüfen. Hingegen erlaubt der öffentliche Schlüssel die Verschlüsselung von Nachrichten, deren anschließende Entschlüsselung ausschließlich mit

dem privaten Schlüssel durchführbar ist. Eine Bedingung der asymmetrischer Kryptosysteme ist, dass die Ableitung des privaten Schlüssels aus dem öffentlichen Schlüssel eine Herausforderung von erheblicher Komplexität darstellt [DH76]. Um dieser Anforderung gerecht zu werden, basieren asymmetrische Kryptosysteme auf mathematischen Problemstellungen, deren Lösung sogar mit dem Nutzen eines Computers von erheblicher Komplexität ist.

Die Sicherheit des kryptographischen Verfahren RSA beruht auf der Annahme, dass die Faktorisierung eines Produkts bestehend aus zwei großen Primzahlen in keiner vertretbaren Zeit berechenbar ist. Andernfalls wäre es möglich, aus dem öffentlichen Schlüssel die beiden Primfaktoren zu extrahieren, um anschließend damit den privaten Schlüssel zu bestimmen. Daraus ergibt sich die Folgerung, dass, sofern die Faktorisierung großer Zahlen mit geringen Aufwand bewältigt werden kann, das RSA-Verfahren als kompromittiert angesehen werden muss [Cor+09].

Bislang ist kein klassischer Algorithmus bekannt, der die Primfaktorzerlegung effizient berechnen kann [Hoe22]. In diesem Zusammenhang bedeutet "effizient", dass die Laufzeit des Algorithmus in einem höchstens polynomialen Verhältnis zur Größe der Eingabezahl anwächst.

Im Kontext der Entschlüsselungsfunktion für RSA wird eine Funktion implementiert, die in der Lage ist, die Primfaktorzerlegung effizient zu berechnen. Zudem wird diese Funktion spezifisch darauf ausgerichtet, aus einem öffentlichen Schlüssel den zugehörigen privaten Schlüssel des RSA-Verfahrens abzuleiten. Die Entschlüsselungsfunktion besteht aus einem Quantenalgorithmus und einem klassischer Algorithmus. Der eingesetzte Quantenalgorithmus kann effektiv die Ordnung eines Elements in einer Gruppe bestimmen. Unter Einbeziehung der zuvor bestimmten Ordnung werden anschließend die Primfaktoren des öffentlichen Schlüssels mithilfe eines klassischen Algorithmus berechnet [Sho97]. In einem abschließenden Schritt wird der private Schlüssel auf der Grundlage der berechneten Primfaktoren ermittelt.

3 Quantencomputer

3.1 Entwicklung

Die Grundidee der Quantencomputer findet ihren Ursprung in dem Jahre 1980 als Paul Benioff ein theoretisches Modell einer klassischen Turing-Maschine beschrieb, die den Gesetzen der Quantenmechanik unterlag. Benioff demonstrierte, dass die Zustände einer Turing-Maschine in einem Quantensystem darstellbar sind. Er zeigte außerdem, dass klassische Operationen, die in einer Turing-Maschine ausgeführt werden, durch entsprechende Quantengatter auf einem Quantensystem äquivalent durchgeführt werden können [Ben80].

Kurz nach der Veröffentlichung von Benioffs Arbeit beschäftigte sich Richard Feynman im Jahr 1981 mit der Frage, wie effizient ein klassischer Computer bei der Simulation

physikalischer Prozesse ist. Feynman befasste sich mit der Schwierigkeit, die Quantenmechanik mithilfe eines klassischen Computers zu simulieren. Er stellte fest, dass die Anforderungen für die Simulation der Quantenmechanik auf einem klassischen Computer mit jedem zusätzlichen Quantenteilchen exponentiell ansteigen. Die Begründung dafür liegt in der exponentiell wachsenden Menge an Zustandsinformationen, welche für die Beschreibung des Quantensystems benötigt werden. Als Lösungsansatz nannte Feynman einen Quantencomputer, der selbst auf den Prinzipien der Quantenmechanik basiert und dadurch eine effiziente Simulation von Quantensystemen ermöglicht [Fey82].

David Deutsch präsentierte im Jahr 1985 den ersten Quantenalgorithmus, der eine spezifische Fragestellung effizienter lösen konnte als jegliche bekannten klassischen Algorithmen. Allerdings bezog sich diese Fragestellung auf ein eher unrealistisches Problem. Ein klassischer Algorithmus würde zur Beantwortung dieser Frage zwei Funktionsauswertungen benötigen. Hingegen benötigt der Quantenalgorithmus, aufgrund der Nutzung von Quantenparallelismus, für die gleiche Fragestellung nur eine einzige Funktionsauswertung [Deu85].

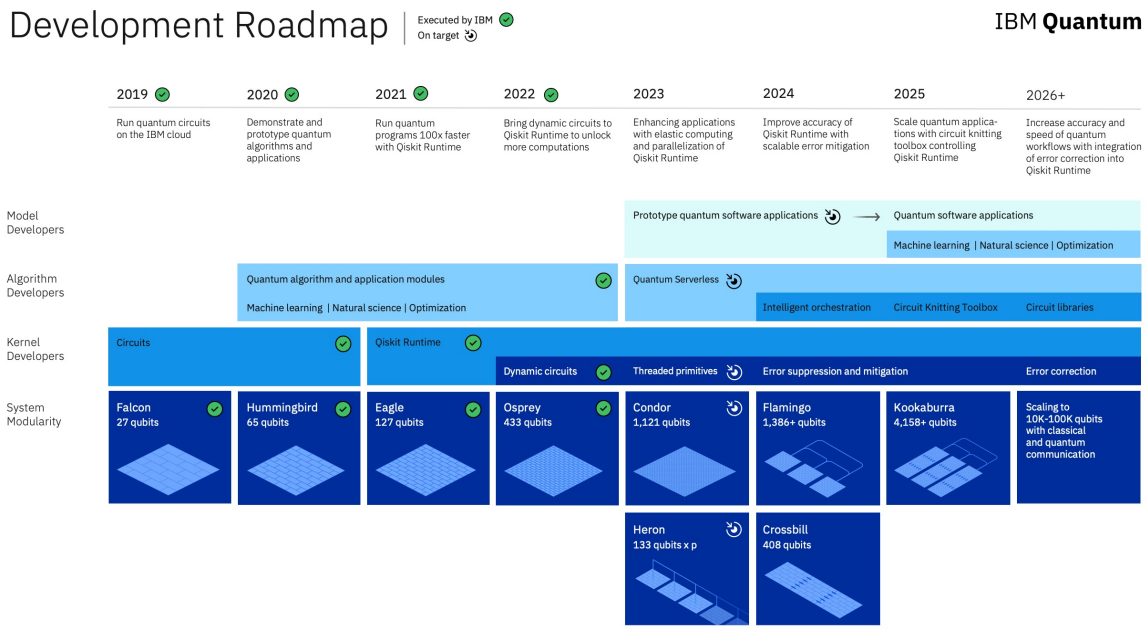
Im Jahr 1994 stellte Shor einen Quantenalgorithmus vor, der in der Lage ist, die Ordnung beziehungsweise Periode eines Elements in der multiplikativen Gruppe eines Modulus mit nur polynomialem Aufwand zu ermitteln. Da trotz erheblicher Anstrengungen lediglich klassische Algorithmen mit exponentiellem Aufwand für diese Berechnung zur Verfügung stehen, stellte Shors Entdeckung eine Errungenschaft dar, die das allgemeine Interesse für Quantencomputing enorm anregte [Sho97].

Zwei Jahre später stellte Lov K. Grover einen Quantenalgorithmus vor. Genau wie der Quantenalgorithmus von Shor, kann auch Grover's Quantenalgorithmus ein bestimmtes Problem effizienter lösen als jeglicher klassische Algorithmen. Der Quantenalgorithmus ermöglicht die Suche in einer unsortierten Datenbank mit einem Aufwand von $\mathcal{O}(\sqrt{N})$, während der beste klassische Such-Algorithmus diese Aufgabe in $\mathcal{O}(N)$ bewältigt. Darüber hinaus konnte Grover zeigen, dass es auch unter vollständiger Ausnutzung der Prinzipien der Quantenmechanik nicht möglich ist, die unstrukturierte Suche in weniger als $\mathcal{O}(\sqrt{N})$ Schritten durchzuführen [Gro96].

Die erste erfolgreiche physikalische Implementation eines Quantenalgorithmus wurde im Jahre 1998 durchgeführt. In dem Versuch wurde der Quantenalgorithmus von David Deutsch auf einem zwei Qubit Quantencomputer Implementiert. Der eingesetzte Quantencomputer nutzte die Kernspinresonanz-Technologie, wobei die zwei Qubits durch Ausnutzung der Spin-Zustände von Atomkernen in spezifischen Molekülstrukturen realisiert wurden [JM98].

In der fortlaufenden Entwicklung der Quanteninformationstechnologie wurden verschiedene Arten von Quantencomputern konzeptioniert und realisiert. Zu diesen zählen adiabatische Quantencomputer, schaltkreisbasierte und topologische Quantencomputer. Im Kontext des Quantencomputing wird der Begriff „Quantencomputer“ oft auf universell einsetzbare Quantencomputer angewendet. Der Eigenschaft „universell“ wird in dem Zusammenhang gemäß DiVincenzo's Kriterien definiert. Im Wesentlichen besagen DiVincenzo's Kriterien, dass ein universeller Quantencomputer in der Lage sein muss,

Abbildung 1: IBM-Quantum-Roadmap [23b]



jede Quantenberechnung auszuführen, vorausgesetzt, er verfügt über ausreichende Ressourcen [DiV00]. Die Schaltkreis-basierende und topologische Quantencomputer erfüllen die DiVincenzo's Kriterium und zählen somit als universelle Quantencomputer. Hingegen sind adiabatische Quantencomputer nicht universell Programmierbar und werden explizit zum Lösen von Optimierungsproblemen eingesetzt.

Das Unternehmen International Business Machines (IBM) präsentierte im Jahr 2019 den ersten kommerziell verfügbaren, schaltkreisbasierten Quantencomputer. Dieses System, bekannt als „IBM Q System One", verfügt über 20 Qubits. Seit der Vorstellung des System One hat IBM die Qubit-Kapazität kontinuierlich erhöht. Aktuelle Systeme, wie sie in der Abbildung 1 dargestellt sind, verfügen über 433 Qubits.

In Anbetracht zukünftiger Entwicklungen haben Unternehmen wie Google, IBM und Microsoft umfangreiche Pläne zur Erweiterung ihrer Quantencomputing-Kapazitäten. Google plant beispielsweise die Errichtung eines Quanten-Campus, der bis 2030 über einen Quantencomputer mit einer Million Qubits verfügen soll [23c]. Microsoft verfolgen ähnliche Ziele, wobei der konkrete Umfang ihrer Projekte bislang nicht öffentlich spezifiziert wurde. Des weiteren erwartet das Bundesamt für Sicherheit in der Informationstechnik die Existenz kryptografisch relevanter Quantencomputer zu Beginn der 2030er Jahre [23a].

3.2 Vorteile gegenüber klassischen Rechnern

Quantenparallelismus

4 Fundament

4.1 Literatur

Shor's Algorithmus

Der Algorithmus wurde erstmals in der Publikation *„Algorithms for Quantum Computation: Discrete Logarithms and Factoring“* von Peter W. Shor veröffentlicht.

Die Arbeit von Shor umfasste zwei Algorithmen. Der erste Algorithmus ermöglicht die effiziente Berechnung der Primfaktorzerlegung, während der zweite Algorithmus die effiziente Berechnung des diskreten Logarithmus ermöglicht. Aufgrund ihrer konzeptionellen Ähnlichkeiten werden beide Algorithmen häufig kollektiv als „Shor's Algorithmus“ bezeichnet.

Die Quantenberechnungen beider Algorithmen basieren auf arithmetische Operationen in Restklassen und der Quanten-Fourier-Transformation. Da Shor die Umsetzung von arithmetischen Operationen in Restklassen sowie die Quanten-Fourier-Transformation nicht explizit behandelt, stützt sich die Implementierung auf den Ergebnissen weiterer Arbeiten. Diese untersuchen insbesondere effiziente Methoden zur Durchführung von arithmetischer Operationen in Restklassen innerhalb eines Quantenschaltkreises. Einige dieser Arbeiten untersuchen die Quanten-Fourier-Transformation, da diese ein notwendiges Element für gewisse Berechnungen darstellt.

In der Publikation *„Quantum Networks for Elementary Arithmetic Operations“* erklären Vlatko Vedral, Adriano Barenco und Artur Ekert, wie die modulare Exponentiation in einem Quantenschaltkreis berechnet werden kann.

Stéphane Beauregard baut auf den Erkenntnissen von Vedral, Barenco und Ekert auf und verbessert den Quantenschaltkreis für die arithmetische Operation der modularen Exponentiation. Hierfür ersetzt Beauregard einen Teil des ursprünglichen Quantenschaltkreis, der in der modularen Exponentiation für die Berechnung der Addition genutzt wurde, durch einen effizienteren Quantenschaltkreis, wie ihn Thomas G. Draper in *„Addition on a Quantum Computer“* beschreibt. Des weiteren verwendet Beauregard diese Optimierungen in seiner Arbeit *„Circuit for Shor's algorithm using 2n+3 qubits“*, um eine Realisierung von Shor's Algorithmus zur Faktorisierung zu beschreiben.

Andere Arbeiten, die sich mit der Implementierung von Shors's Algorithmus auseinandersetzen, realisieren die modularen Exponentiation, indem klassische Schaltkreise identisch in den Quantenkontext übersetzt werden. Der Nachbau von klassischen Operationen ist aufgrund der unitären Natur von Quantenoperationen nicht die effizienteste Realisierung. Des Weiteren meiden andere Arbeiten die Realisierung der modularen Exponentiation für allgemeine Eingaben. Ohne modulare Exponentiation sind diese Varian-

ten nicht in der Lage die Berechnung variabler Werte zu verarbeiten. Stattdessen dienen diese ausschließlich der Demonstration der Funktionsweise von Shor's Algorithmus und sind nur in der Lage, ausgewählte Eingaben zu verarbeiten.

Bei Recherchen wurde keine allgemeine Variante gefunden, die weniger Qubits benötigt als die Variante aus der Arbeit *„Circuit for Shor's algorithm using 2n+3 qubits“*, deswegen bildet diese die Grundlage der Implementierung ab.

5 Grundlagen

5.1 Qubits

„Der Anfang enthält also beides, Sein und Nichts; ist die Einheit von Sein und Nichts, – oder ist Nichtsein, das zugleich Sein, und Sein, das zugleich Nichtsein ist.“ - Georg Wilhelm Friedrich Hegel

Die Repräsentation und Speicherung von Information ist ein zentraler Aspekt aller Computer, unabhängig von der verwendeten Technologie oder Architektur. Bei klassischen Computern basiert diese Repräsentation auf dem Binärsystem. In der klassischen Repräsentation ist das Bit die kleinste Informationseinheit, die eine von zwei mögliche Zustände annehmen kann: 0 oder 1. Ein Bit hat zu jedem Zeitpunkt einen klar definierten Zustand. Mehrmaliges Auslesen eines Bits führt zu keiner Zustandsänderung, sofern keine Operationen zwischen den Auslesungen durchgeführt werden.

Quantencomputer hingegen funktionieren grundlegend anders. Sie stützen sich auf die Prinzipien der Quantenmechanik und verwenden anstelle von Bits Quatenbits beziehungsweise Qubits zur Informationsrepräsentation. Ein einzelnes Qubit stellt die kleinstmögliche Informationseinheit dar über die ein Quantencomputer verfügt. Um quantenmechanische Zustände von Qubits zu beschreiben wird die Dirac-Notation als mathematische Schreibweise genutzt [Dir39].

Ein Qubit kann den Zustände 0 oder den Zustand 1 annehmen:

$$|0\rangle \text{ oder entsprechend } |1\rangle$$

Des weiteren kann ein Qubit auch beide der Basiszustände gleichzeitig einnehmen:

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

Dieses Phänomen ist eine der charakteristischen Eigenschaften von Qubits und wird als **Superposition** bezeichnet. Dabei handelt es sich bei den Vorfaktoren α und β um komplexe Zahlen für die gilt:

$$|\alpha|^2 + |\beta|^2 = 1$$

Es gibt also unendlich viele Zustände, die ein Qubit in einer Superposition der beiden Basiszustände annehmen kann.

In der klassischen Welt findet man nur schwer eine Analogie zur Superposition. Man kann sich dies jedoch in etwa an einer Münze vorstellen [Hoe23a]:

Ein klassisches Bit kann dabei entweder auf der Kopf- oder auf der Zahl-Seite liegen. Ein Qubit hingegen ist eine Münze die auf der Kante um die eigene Achse rotiert. Dabei geben die Vorfaktoren α und β an, wie die Münze zu der einen oder zu der anderen Seite tendiert. Ohne die Münze zu beeinflussen ist es nicht möglich die Tendenz, also die Vorfaktoren, zu bestimmen.

Möchte man nun aber doch ein konkretes Ergebnis haben, muss man das Qubit lesen, beziehungsweise genauer gesagt messen. Dabei wird die Superposition zerstört und das Qubit kollabiert in einen der beiden Basiszustände.

In der Analogie zu der Münze wird die Rotation der Münze gezielt gestoppt, sodass diese auf eine der beiden Seiten kippt.

Die Wahrscheinlichkeiten dafür werden durch die Vorfaktoren des Qubits bestimmt:

$$|\alpha|^2 \text{ für } |0\rangle, |\beta|^2 \text{ für } |1\rangle$$

Es ist also nicht möglich den Zustand eines Qubit während einer Berechnung nur „anzuschauen“, da ansonsten die Superposition zerstört wird und sich somit der Zustand des Qubits verändert. Aus diesem Grund erfolgt die Messung in der Regel erst am Ende eines Quantenalgorithmus, um das endgültige Ergebnis zu ermitteln.

5.2 Quantengatter

5.3 Qiskit

5.4 RSA

5.5 Quanten-Fourier-Transformation

Die Quanten-Fourier-Transformation bildet einen wesentlichen Bestandteil des implementierten Quantenalgorithmus. Im folgen Abschnitt wird die allgemeine Anwendung der Quanten-Fourier-Transformation erklärt. Darüber hinaus wird die Implementierung des Quantenschaltkreis anhand der Formel der Quanten-Fourier-Transformation hergeleitet.

Im Prinzip handelt es sich bei der Quanten-Fourier-Transformation um eine Transformation, die Qubits von der Standardbasis ($|0\rangle$, $|1\rangle$), in die entsprechende Fourierbasis ($|+\rangle$, $|-\rangle$) überführt [Hom]. Bei dem Basiswechsel werden die Informationen des vorherigen Standardbasiszustandes in die Phase des neuen Zustandes übertragen [RG17]. Anschließend können in der Fourierbasis Rechnungen durchgeführt werden die im Grunde durch Manipulationen der Phase realisiert werden. Mit diesem Ansatz ist es möglich arithmetische Operationen, wie beispielsweise die Addition, effizienter zu berechnen [Dra00][RG17].

Unter Verwendung der inversen Quanten-Fourier-Transformation, die als Rücktransformation dient, ist es möglich wieder zurück in die Standardbasis zu transformieren. Dies bewirkt, die Extraktion der Information aus der Phase in einen messbaren Zustand.

Die Quanten-Fourier-Transformation ist für ein $N = 2^n$ mit n Qubits für die Basisvektoren $|x\rangle, x = 0, ..., N-1$ wie folgt definiert [Hoe23b]:

$$QFT_N |x\rangle_n = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2\pi i x y}{N}} |y\rangle_n$$

Anhand dieser Definition kann der Quantenschaltkreis nicht direkt hergeleitet werden. Stattdessen musst die Formel umgeformt werden.

Die folgende Herleitung stammt aus dem Textbuch *Quantum Computation and Quantum Information* Seite 218 [NC10]. Die Herleitung wird der Übersicht halber um Zwischenschritte ergänzt:

Indem y in der ersten Formel in die Binärschreibweise übertragen wird, erhält man:

$$y = \sum_{k=1}^{n} 2^{n-k} y_{n-k+1}$$

$$QFT_N |x\rangle_n = \frac{1}{\sqrt{N}} \sum_{y_n=0}^{1} ... \sum_{y_1=0}^{1} e^{\frac{2\pi i x \sum_{k=1}^{n} 2^{n-k} y_{n-k+1}}{N}} |y_n ... y_2 y_1\rangle \tag{1}$$

Mit $N = 2^n$ kann der Bruch im Exponenten gekürzt werden:

$$QFT_N |x\rangle_n = \frac{1}{\sqrt{N}} \sum_{y_n=0}^{1} ... \sum_{y_1=0}^{1} e^{\frac{2\pi i x \sum_{k=1}^{n} 2^{n-k} y_{n-k+1}}{2^n}} |y_n ... y_2 y_1\rangle$$

$$QFT_N |x\rangle_n = \frac{1}{\sqrt{N}} \sum_{y_n=0}^{1} ... \sum_{y_1=0}^{1} e^{2\pi i x \sum_{k=1}^{n} 2^{-k} y_{n-k+1}} |y_n ... y_2 y_1\rangle$$

Anschließend kann der Ausdruck 2^{-k} zu $\frac{1}{2^k}$ umgeformt werden:

$$QFT_N |x\rangle_n = \frac{1}{\sqrt{N}} \sum_{y_n=0}^{1} ... \sum_{y_1=0}^{1} e^{\frac{2\pi i x \sum_{k=1}^{n} y_{n-k+1}}{2^k}} |y_n ... y_2 y_1\rangle$$

Die Summe im Exponenten der Basis e kann als Produkt umgeschrieben werden. Anstatt dem Produktzeichen $\prod$ wird das Tensorprodukt $\bigotimes$ verwendet, da es sich um Qubits handelt:

$$QFT_N |x\rangle_n = \frac{1}{\sqrt{N}} \sum_{y_n=0}^{1} ... \sum_{y_1=0}^{1} \bigotimes_{k=1}^{n} e^{\frac{2\pi i x y_k}{2^k}} |y_k\rangle$$

Der Ausdruck kann weiter vereinfacht werden indem das Tensorprodukt vorgezogen wird:

$$QFT_N \left|x\right\rangle_n = \frac{1}{\sqrt{N}} \bigotimes_{k=1}^{n} [\sum_{y_k=0}^{1} e^{\frac{2\pi i x y_k}{2^k}} \left|y_k\right\rangle]$$

$$QFT_N \left|x\right\rangle_n = \frac{1}{\sqrt{N}} \bigotimes_{k=1}^{n} [\left|0\right\rangle + e^{\frac{2\pi i x}{2^k}} \left|1\right\rangle]$$

Schreibt man das Tensorprodukt voll aus und notiert x in Binärschreibweise, erhält man:

$$\frac{1}{\sqrt{N}}(\left|0\right\rangle + e^{\frac{2\pi i(2^{n-1}x_n + \ldots + 2^1 x_2 + 2^0 x_1)}{2^1}} \left|1\right\rangle) \bigotimes \left|0\right\rangle + e^{\frac{2\pi i(2^{n-1}x_n + \ldots + 2^1 x_2 + 2^0 x_1)}{2^2}} \left|1\right\rangle \ldots \left|0\right\rangle + e^{\frac{2\pi i(2^{n-1}x_n + \ldots + 2^1 x_2 + 2^0 x_1)}{2^n}} \left|1\right\rangle$$

Die komplexe Exponentialfunktion ergibt für eine natürliche Zahl k: $e^{2\pi i k} = 1$. Mit dieser Eigenschaft kann man beispielsweise die Phasenverschiebung des ersten Tensors vereinfachen:

$$e^{\frac{2\pi i(2^{n-1}x_n + \ \ldots \ + 2^1 x_2 + 2^0 x_1)}{2^1}} \equiv e^{\frac{2\pi i(2^{n-1}x_n)}{2^1}} \ \ldots \ e^{\frac{2\pi i(2^1 x_2)}{2^1}} e^{\frac{2\pi i(2^0 x_1)}{2^1}} \equiv e^{2\pi i(2^{n-2}x_n)} \ \ldots \ e^{2\pi i(2^0 x_2)} e^{2\pi i(2^{-1}x_1)}$$

Dabei ergibt nur der Term $e^{2\pi i(2^{-1}x_1)} \neq 1$ und verursacht somit eine relevante Phasenverschiebung.

Abschließend lässt sich das gesamte Tensorprodukt vereinfachen:

$$QFT_N \left|x\right\rangle_n = \frac{1}{\sqrt{N}}(\left|0\right\rangle + e^{\frac{2\pi i(2^0 x_1)}{2^1}} \left|1\right\rangle) \bigotimes (\left|0\right\rangle + e^{\frac{2\pi i(2^1 x_2 + 2^0 x_1)}{2^2}} \left|1\right\rangle) \ldots (\left|0\right\rangle + e^{\frac{2\pi i(2^{n-1}x_n + \ldots + 2^1 x_2 + 2^0 x_1)}{2^n}} \left|1\right\rangle)$$

Ein einzelner Tensor repräsentiert die Wirkung der Schaltung auf ein einzelnes Qubit. Somit sind die Phasenverschiebungen erkenntlich die auf ein Qubit wirken. Ausserdem verdeutlicht die Binärschreibweise dass die angewendete Phasenverschiebung vom Zustand anderer Qubits abhängt.

Für die Implementierung der Quanten-Fourier-Transformation sind nur die Terme relevant welche eine Phasenverschiebung von $\neq 1$ bewirken. Dies kann man mit folgender Formel beschreiben:

$$QFT_N \left|x\right\rangle_n = \frac{1}{\sqrt{N}} \bigotimes_{k=1}^{n} [\left|0\right\rangle + e^{\frac{2\pi i \sum_{b=0}^{k-1} 2^b x_{b+1}}{2^k}} \left|1\right\rangle]$$

Im weiteren wird die Formel verwendet um ein Quantenschaltkreis der Quanten-Fourier-Transformation für drei Qubits zu implementieren:

$$QFT_8 \left|x\right\rangle_3 = \frac{1}{\sqrt{8}}[(\left|0\right\rangle + e^{\frac{2\pi i(2^0 x_1)}{2^1}} \left|1\right\rangle) \bigotimes (\left|0\right\rangle + e^{\frac{2\pi i(2^1 x_2 + 2^0 x_1)}{2^2}} \left|1\right\rangle) \bigotimes (\left|0\right\rangle + e^{\frac{2\pi i(2^2 x_3 + 2^1 x_2 + 2^0 x_1)}{2^3}} \left|1\right\rangle)]$$

Man kann die Phasenverschiebung die durch den Zustand eines einzelnen Qubits erzeugt wird verdeutlichen, indem man die Addition im Exponenten zu einer Multiplikation der gleichen Basen umformt:

$$= \frac{1}{\sqrt{8}}[(|0\rangle + e^{\frac{2\pi i(2^0 x_1)}{2^1}} |1\rangle) \bigotimes (|0\rangle + e^{\frac{2\pi i(2^1 x_2)}{2^2}} e^{\frac{2\pi i(2^0 x_1)}{2^2}} |1\rangle) \bigotimes (|0\rangle + e^{\frac{2\pi i(2^2 x_3)}{2^3}} e^{\frac{2\pi i(2^1 x_2)}{2^3}} e^{\frac{2\pi i(2^0 x_1)}{2^3}} |1\rangle)]$$

Die Phasenverschiebung wird eindeutiger indem man die Brüche kürzt:

$$= \frac{1}{\sqrt{8}}[(|0\rangle + e^{\pi i x_1} |1\rangle) \bigotimes (|0\rangle + e^{\pi i x_2} e^{\frac{\pi i x_1}{2}} |1\rangle) \bigotimes (|0\rangle + e^{\pi i x_3} e^{\frac{\pi i x_2}{2}} e^{\frac{\pi i x_1}{4}} |1\rangle)]$$

In einer abschließenden Umformung lässt sich die $\frac{1}{\sqrt{8}}$ aufteilen. Dadurch erinnern die einzelnen Tensoren, beziehungsweise Qubits an die Form die bei eine Hadamard-Transformation entsteht:

$$= \frac{1}{\sqrt{2}}(|0\rangle + e^{\pi i x_1} |1\rangle) \bigotimes \frac{1}{\sqrt{2}}(|0\rangle + e^{\pi i x_2} e^{\frac{\pi i x_1}{2}} |1\rangle) \bigotimes \frac{1}{\sqrt{2}}(|0\rangle + e^{\pi i x_3} e^{\frac{\pi i x_2}{2}} e^{\frac{\pi i x_1}{4}} |1\rangle)$$

Der Ausdruck $e^{\pi i x_k}$ ist in jedem der einzelnen Qubits vorhanden. Des weiteren ist an dem x_k erkennbar das die angewendete Phasenverschiebung von dem Zustand des Qubits abhängig ist, auf welches die Verschiebung auch angewendet wird. Konkret bedeutet dass, das auf jedes Qubit mit dem Zustand $|x_k\rangle = |1\rangle$ eine Phasenverschiebung von $e^{\pi i}$ wirkt. Anhand des ersten Qubits, beziehungsweise Tensor, würde das bedeuten, dass der Zustand bei $x_1 = 0$ zu $|0\rangle + e^{\pi i 0} |1\rangle \equiv |0\rangle + |1\rangle$ wird. Bei $x_1 = 1$ würde man $|0\rangle + e^{\pi i 1} |1\rangle$ erhalten, was $|0\rangle - |1\rangle$ entspricht. Aufgrund des Vorfaktors von $\frac{1}{\sqrt{2}}$ entsprächen beide Fälle also der Hadamard-Transformation. Da auch alle anderen Tensoren den Ausdruck $e^{\pi i x_k}$ und den Vorfaktor $\frac{1}{\sqrt{2}}$ beinhalten, wirkt auf jedes Qubit ein Hadamard-Gatter.

Die weiteren Tensoren des Tensorproduktes beinhalten zunehmend mehr Ausdrücke die für unterschiedliche Phasenverschiebungen sorgen. Die Phasenverschiebung von einem einzelnen Ausdruck kann mit einem Phasen-Gatter realisiert werden. Das Phasengatter entspricht einer Rotation mit $P(\lambda) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix}$.

Beispielsweise wirkt auf dem zweiten Qubit noch ein Phase-Gatter mit $P(\frac{\pi i}{2})$. Diese Phasenverschiebung soll aber nur angewendet werden wenn $x_1 = 1$ ist. Deswegen wird das Phase-Gatter durch den Zustand des ersten Qubits kontrolliert und mit einem kontrollierten-Phase-Gatter realisiert. Das gleiche Prinzip gilt für alle weiteren Qubits. Anschließend erhält man einen Quantenschaltkreis wie in Abbildung 2.

Wie im Quantenschaltkreis in Abbildung 2 erkennbar, spiegelt die Quanten-Fourier-Transformation die Reihenfolge der Qubits [Hoe23b]. Um die Reihenfolge wiederherzustellen werden am Ende der Quanten-Fourier-Transformation Swap-Operationen verwendet.

Anhand der Implementierung wird ersichtlich, dass die Quanten-Fourier-Transformation unitär ist. Diese Eigenschaft ergibt sich aus der Tatsache, dass der zugehörige Quantenschaltkreis ausschließlich unter Verwendung von unitären Gattern realisierbar ist.

Abbildung 2: 3-Qubit QFT ohne Swaps

Abbildung 3: 3-Qubit inverse QFT ohne Swaps

Wie bereits oben erwähnt, wird für die Rücktransformation aus der Fourierbasis in die Standardbasis die inversen Quanten-Fourier-Transformation angewendet. Um einen Quantenschaltkreis aus unitären Gattern zu invertieren, wird die inverse der verwendeten Gatter in umgekehrter Reihenfolge der originalen Schaltung angewendet. Die Swap Operationen stehen somit bei der inversen Quanten-Fourier-Transformation am Anfang. In Abbildung 3 ist beispielhaft die inverse Quanten-Fourier-Transformation für drei Qubits abgebildet.

5.6 Quanten-Phase-Estimation

Im nachfolgenden Abschnitt wird die Anwendung und Funktionsweise des Quantum-Phase-Estimation Quantenalgorithmus erläutert. Die Quantum-Phase-Estimation ist ein Bestandteil einiger fortgeschrittener Quantenalgorithmen und eine integrale Komponente des spezifischen Quantenalgorithmus, der in dieser Arbeit implementiert wird. Genauer gesagt basiert der implementierte Algorithmus auf den Prinzipien der Quantum-Phase-Estimation und verwendet die selbe Methodik für einen spezialisierten Kontext.

Schwerpunktmäßig konzentriert sich die Erklärung primär auf das Verständnis der Funktionsweise der Quantum-Phase-Estimation und nimmt an, dass einige Voraussetzungen gegeben sind. Diese Voraussetzungen hängen vom spezifischen Kontext ab, in dem die Quantum-Phase-Estimation angewendet wird. Im weiteren Verlauf der Arbeit werden diese Voraussetzungen im Hinblick auf den Anwendungsfall des implementierten Quantenalgorithmus konkretisiert.

Voraussetzungen ist, dass ein Eigenvektor $|x\rangle_n$ von einer unitäre Transformation $U^{n\times n}$ bekannt ist. Wendet man die Transformation $U^{n\times n}$ auf $|x\rangle_n$ an, so gilt: $U^{n\times n}|x\rangle_n = \lambda_x |x\rangle_n$ [NC10]. Dabei erhält man, abhängig vom gewählten Eigenvektor $|x\rangle_n$, einen der Eigenwert λ_x von $U^{n\times n}$. Ein Eigenwert λ_x besitzt die Form eines Phasenfaktors: $e^{2\pi i\varphi}$ mit $0 \leq \varphi < 1$. Im Prinzip wird durch die unitäre Transformation also eine globale Phasenverschiebung auf den Eigenvektor angewendet.

Wie bereits im Kapitel zu den Grundlagen gezeigt wurde, ist es nicht möglich eine globale Phase durch eine gewöhnliche Messung der Qubits zu bestimmen. Das liegt daran, dass eine globale Phase die Amplituden eines Qubits nicht verändert und somit die Wahrscheinlichkeiten der Messergebnisse unverändert bleiben. Stattdessen muss man die Qubits manipulieren, so dass die globale Phase doch Einfluss auf die Amplituden hat.

Der Quanten-Phase-Estimation Quantenalgorithmus ist in der Lage, den Eigenwert aus $U^{n\times n}|x\rangle_n = \lambda_x |x\rangle_n$, also die Phasenverschiebungen, repräsentiert durch $\lambda_x = e^{2\pi i\varphi}$, auf die Amplitude eines anderen Qubits zu verschieben. Um λ_x auf ein anderes Qubit zu

übertragen wird der Effekt des **Phase-Kickback** genutzt. Anschließend wird der Wert φ des Eigenwertes durch die inverse Quanten-Fourier-Transformation in einen messbaren Zustand überführt.

Der Phase-Kickback tritt auf, wenn eine unitäre Transformation $U^{n\times n}$ kontrolliert durch ein Qubit $|y\rangle_1$ in Superposition, auf einen Eigenvektor $|x\rangle_n$ von $U^{n\times n}$ anwendet wirkt. Dabei wird der Eigenwert, beziehungsweise λ_x, auf den $|1\rangle$-Anteil von $|y\rangle_1$ übertragen.

Sei: $|y\rangle_1 \equiv \alpha\,|0\rangle + \beta\,|1\rangle$ mit $\alpha, \beta \neq 0$, dann:
$CU^{(n+1)\times(n+1)}(|y\rangle_1 \bigotimes |x\rangle_n) = (\alpha\,|0\rangle + \lambda_x\beta\,|1\rangle) \bigotimes |x\rangle_n$.

Es folgt ein Beispiel welches den Effekt verdeutlicht: Beachtet werden zwei Qubits im Zustand $|+\rangle_1 \bigotimes |-\rangle_1$ auf die ein kontrolliertes X-Gatter angewendet wird. Dabei ist $|-\rangle_1$ der Eigenvektor einer X-Transformation mit zugehörigen Eigenwert $-$, also $X^{1\times1}\,|-\rangle_1 = -\,|-\rangle_1$.

Auf das zweite Qubit $|-\rangle_1$ wirkt ein $CX^{2\times2}$ Gatter welches durch das erste Qubit $|+\rangle_1$ kontrolliert wird:

$$CX^{2\times2}(|+\rangle_1 \bigotimes |-\rangle_1) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \frac{1}{2} \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \bigotimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = |-\rangle_1 \bigotimes |-\rangle_1$$

Mithilfe des Phase-Kickback kann man also den Eigenwert einer unitären Transformation in die Phase eines Kontrollqubits verschieben. Der Vorteil davon ist, dass diese Phasenverschiebung nur den Vorfaktor von $|1\rangle$ des Kontrollqubits betrifft und keine globale Phase darstellt.

Der Aufbau eines Quanten-Phase-Estimation Quantenschaltung sieht beispielsweise wie folgt aus:

Beachtet wird die unitäre Transformation eines Phase-Gatter(P) mit einer variablen Phasenverschiebung von $P(2i\pi\varphi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{2i\pi\varphi} \end{pmatrix}$.

Ein zugehöriger Eigenvektor dieser Transformation ist $|1\rangle_1$ den $P(2i\pi\varphi)\,|1\rangle_1 = e^{2i\pi\varphi}\,|1\rangle_1$.

Um den Effekt das Phase-Kickback nutzen zu können, muss sich das Kontrollqubit in einer Superposition befinden. Dafür wird das Kontrollqubit im Zustand $|0\rangle_1$ initialisiert und anschließend mit einem Hadamard-Gatter(H) in die gleichmäßige Superposition $|+\rangle_1$ versetzt.

$$|0\rangle_1 \bigotimes |1\rangle_1 \xrightarrow{H^{\otimes 1}} |+\rangle_1 \bigotimes |1\rangle_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \bigotimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Über das kontrollierte Phase-Gatter wird der Eigenwert auf das Kontrollqubit verschoben und befindet sich deswegen nicht mehr im Zustand $|+\rangle_1$:

$$\frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\0\\1\end{pmatrix} \xrightarrow{CP^{2\times 2}(2i\pi\varphi)} \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&e^{2i\pi\varphi}\end{pmatrix} \cdot \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\0\\e^{2i\pi\varphi}\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\e^{2i\pi\varphi}\end{pmatrix} \bigotimes \begin{pmatrix}0\\1\end{pmatrix}$$

Anschließend wird auf die Kontrollqubits die inverse Quanten-Fourier-Transformation angewendet. Die inverse Quanten-Fourier-Transformation sorgt dafür, dass der Eigenwert die Amplituden der Kontrollqubits beeinflusst. Die Qubits mit dem Eigenvektor sind für den weiteren Ablauf der Quanten-Phasen-Estimation nicht mehr relevant und werden nicht weiter beachtet. Im Beispiel entspricht die inverse Quanten-Fourier-Transformation einem Hadamard-Gatter da nur ein einzelnes Kontrollqubit existiert:

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\e^{2i\pi\varphi}\end{pmatrix} \xrightarrow{H^{\bigotimes 1}} \frac{1}{\sqrt{2}}\begin{pmatrix}1&1\\1&-1\end{pmatrix} \cdot \frac{1}{\sqrt{2}}\begin{pmatrix}1\\e^{2i\pi\varphi}\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1+e^{2i\pi\varphi}\\1-e^{2i\pi\varphi}\end{pmatrix}$$

Anhand des Ergebnisses des Beispiels ist zu erkennen, dass der Eigenwert praktisch auf die Amplitude vom Kontrollqubit transformiert wird.

Verwendet man ein Phase-Gatter mit $\varphi = 0.075$ sieht der Quantenschaltkreis wie in Abbildung **??** aus. Mit dieser Phasenverschiebung sollte man bei einer Messung mit einer Wahrscheinlichkeit von ungefähr 0.9455 den Zustand $|0\rangle$ erhalten und $|1\rangle$ mit der Wahrscheinlichkeit 0.0544. Die Ergebnisse von 20.000 Messungen in Abbildung **??** bestätigen die Größenordnung der Wahrscheinlichkeiten. Jedoch ergeben die Messungen aus Abbildung **??** nicht ganz genau den ausgerechneten Wahrscheinlichkeiten. Dies liegt an der probabilistischen Natur der Messung. Bei einer zunehmenden Anzahl an Messungen würden die Ergebnisse an die Wahrscheinlichkeitswerte konvergieren. Somit benötigt man sehr viele Durchläufe des Quantenalgorithmus um anhand der Messungen ein verlässliches Ergebnis zu erhalten.

Es ist möglich die Präzision der Quanten-Phase-Estimation zu verbessern, indem mehr Qubits verwendet werden. Diese Qubits werden dann als weitere Kontrollqubits verwendet. Die Anzahl der Qubits für den Eigenvektor bleibt gleich der Bitanzahl, die ausreicht, um den Wert des Eigenvektors zu definieren. Jedes einzelne Kontrollqubit kontrolliert ein U^{2^x}-Gatter. Bei n Kontrollqubits kontrolliert das least-significant-bit ein U^{2^0}-Gatter, das darauffolgende ein U^{2^1}-Gatter, während das letzte Kontrollqubit ein $U^{2^{n-1}}$-Gatter kontrolliert. Dabei kann U^{2^x} als 2^x viele U-Gatter realisiert werden oder als ein einzelnes Gatter, welches den Eigenwert λ mit 2^x multipliziert anwendet. Anschließend wirkt die inverse Quanten-Fourier-Transformation auf alle Kontrollqubits. Anhand der Messung kann dann φ bestimmt werden. Der Aufbau der Schaltung ist in Abbildung **??** abgebildet.

Wird die Quanten-Fourier-Transformation wie in Abbildung **??** realisiert, kann man den Zustand der Kontrollqubits vor der inversen Quanten-Fourier-Transformation wie

folgt beschreiben:

$$\frac{1}{\sqrt{N}}[(|0\rangle + CU^{2^0}|1\rangle) \bigotimes (|0\rangle + CU^{2^1}|1\rangle) \bigotimes ... \bigotimes (|0\rangle + CU^{2^{n-1}}|1\rangle)]$$

Mit $U^{n\times n}|x\rangle_n = e^{2\pi i\varphi}|x\rangle_n$ wird der Eigenwert $e^{2\pi i\varphi}$ wegen des Phase-Kickbacks über die CU-Gatter auf die Kontrollqubits übertragen:

$$\frac{1}{\sqrt{N}}[(|0\rangle + e^{2\pi i 2^0\varphi}|1\rangle) \bigotimes (|0\rangle + e^{2\pi i 2^1\varphi}|1\rangle) \bigotimes ... \bigotimes (|0\rangle + e^{2\pi i 2^{n-1}\varphi}|1\rangle)]$$

Schreibt man φ als Binärbruch:

$$\varphi = \frac{\varphi_n}{2^1} + \frac{\varphi_{n-1}}{2^2} + ... + \frac{\varphi_1}{2^n}$$

Kann die Formel in einer ähnlichen Form wie die Quanten-Fourier-Transformation umgeformt werden [NC10]:

$$\frac{1}{\sqrt{N}}[(|0\rangle + e^{2\pi i(\frac{\varphi_n}{2})}...e^{2\pi i(\frac{\varphi_2}{2^{n-1}})}e^{2\pi i(\frac{\varphi_1}{2^n})}|1\rangle)...(|0\rangle + e^{2\pi i(\frac{\varphi_2}{2})}e^{2\pi i(\frac{\varphi_1}{4})}|1\rangle) \bigotimes (|0\rangle + e^{2\pi i(\frac{\varphi_1}{2})}|1\rangle)]$$

Die Formel besitzt die gespiegelte Struktur wie die Quanten-Fourier-Transformation ohne Swap-Gatter:

$$QFT_N|x\rangle_n = \frac{1}{\sqrt{N}}(|0\rangle + e^{2\pi i\frac{(x_1)}{2}}|1\rangle) \bigotimes (|0\rangle + e^{2\pi i\frac{(x_2)}{2}\frac{(x_1)}{4}}|1\rangle)...(|0\rangle + e^{2\pi i(\frac{x_n}{2})}...e^{2\pi i(\frac{x_2}{2^{n-1}})}e^{2\pi i(\frac{x_1}{2^n})}|1\rangle)$$

Durch die Verwendung der Swap Gatter kann die Reihenfolge der quanten-Fourier-Transformation gespiegelt werden, anschließend sind beide Formeln strukturell identisch.

Wie im Kapitell zur Quanten-Fourier-Transformation erklärt, transformiert die Quanten-Fourier-Transformation den Zustand der Eingangsqubits $|x\rangle_n$ in die Phasen der Ausgangsqubits. Hingegen kehrt die inverse Quanten-Fourier-Transformation diesen Vorgang um, indem die Phaseninformationen der Eingangsqubits in Zustände der Standartbasis transformiert werden.

Die Anwendung der inversen Quanten-Fourier-Transformation, inklusive Swap-Gatter, bewirkt also:

$$iQFT(\frac{1}{\sqrt{N}}[(|0\rangle + e^{2\pi i(\frac{\varphi_n}{2})}...e^{2\pi i(\frac{\varphi_2}{2^{n-1}})}e^{2\pi i(\frac{\varphi_1}{2^n})}|1\rangle)...(|0\rangle + e^{2\pi i(\frac{\varphi_2}{2})}e^{2\pi i(\frac{\varphi_1}{4})}|1\rangle) \bigotimes (|0\rangle + e^{2\pi i(\frac{\varphi_1}{2})}|1\rangle)])$$

$$= |\varphi_1\varphi_2...\varphi_n\rangle_n = |2^n\varphi\rangle_n$$

Schließlich kann φ mit einer division durch 2^n bestimmt werden.

Als Beispiel wird die Quanten-Phase-Estimation für $U^{1\times 1}|1\rangle_1 = e^{2\pi i\frac{3}{8}}|1\rangle_1$ also mit $\varphi = \frac{3}{8}$ beachtet. Damit der Quantenschaltkreis in Abbildung 4 möglichst gut erkennbar

Abbildung 4: 3-Kontroll-Qubit QPE

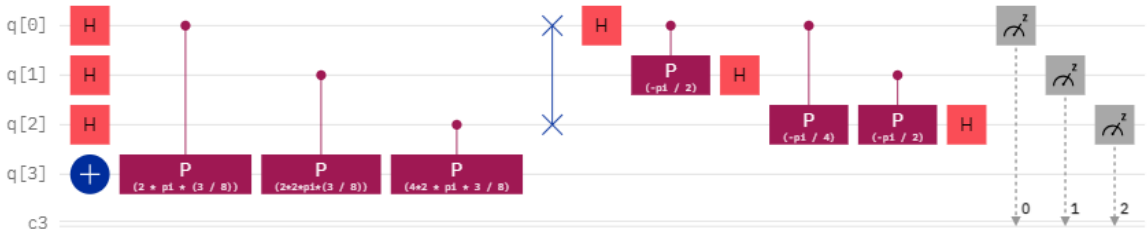


Abbildung 5: 3-C-Qubit QPE Messergebnis

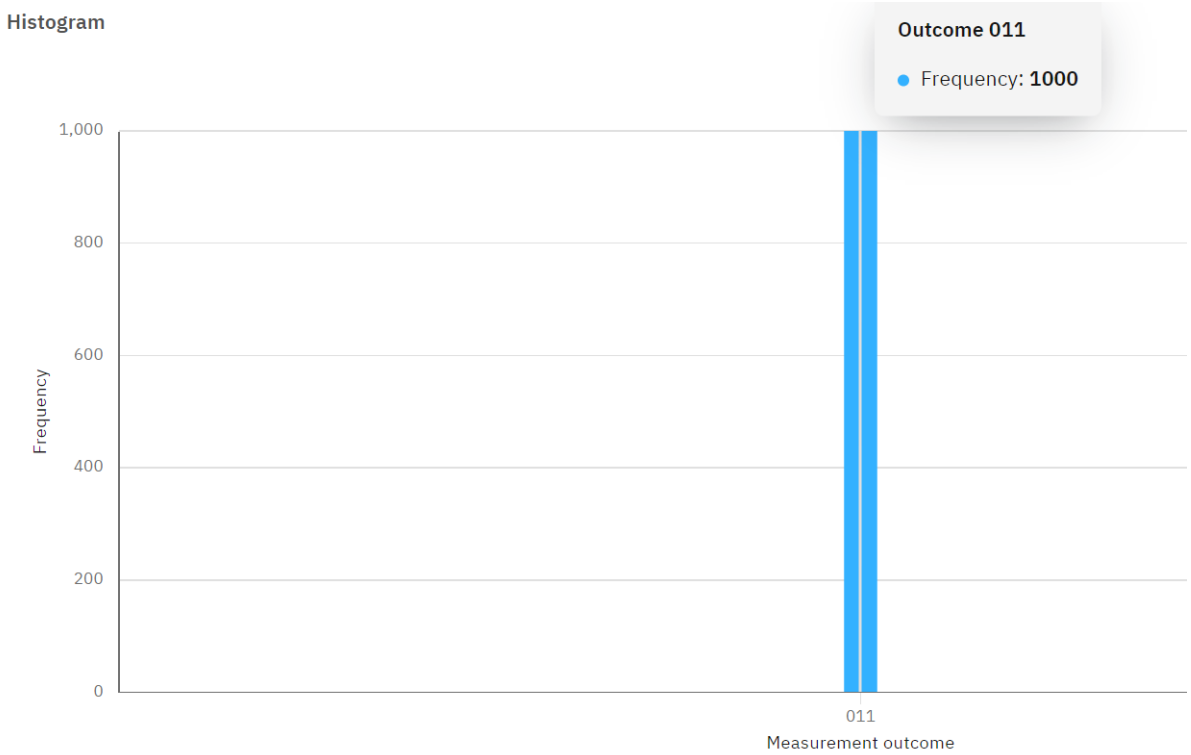
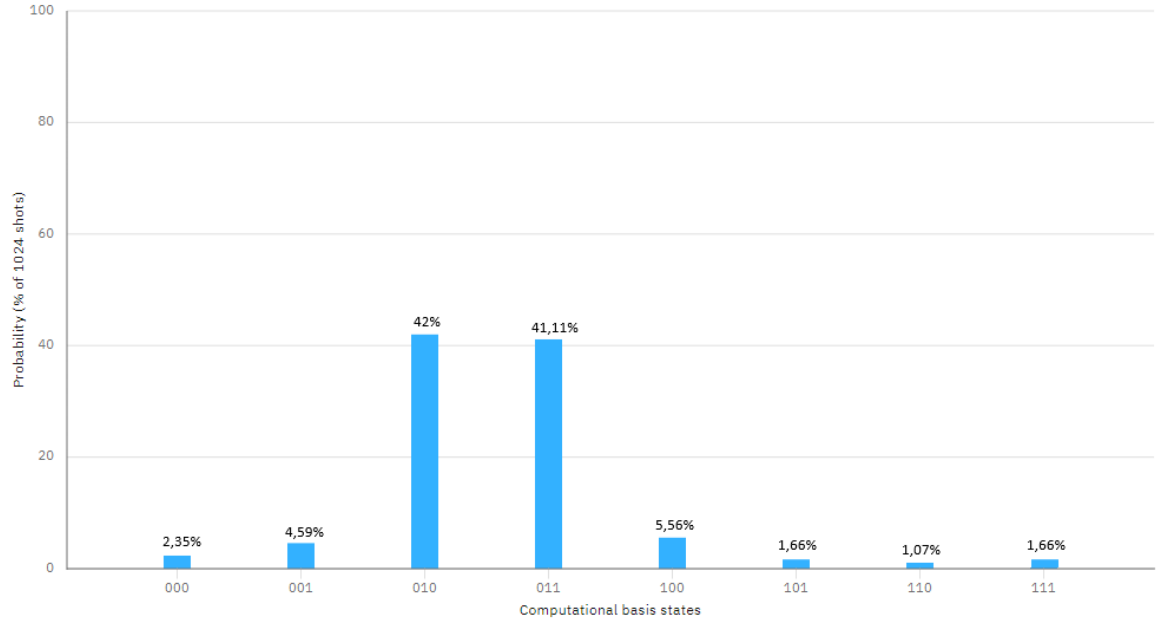


Abbildung 6: QPE unpräzises Messergebnis



ist, werden die kontrollierten U^{2^x}-Gatter werden als Phase-Gatter mit $P(e^{2^x 2\pi i \frac{3}{8}})$ realisiert. Die Messung in Abbildung 5 ergibt konsistent bei allen Durchläufen den Zustand $|3\rangle_3$. Da $|3\rangle_3 = |2^3\varphi\rangle_3$ entspricht, kann mit einer Division $\varphi = \frac{3}{8}$ bestimmt werden. Es ist zu beachten, dass die Reihenfolge der Wertigkeit der Qubits gleich bleibt. Normalerweise vertauscht die inverse wie auch die normale Quanten-Fourier-Transformation die Wertigkeiten. Jedoch wird dieser Effekt mit Swap-Gattern korrigiert.

Im vorherigen Beispiel liefern die Messungen aller Durchläufe den gleichen Zustand mit dem φ eindeutig bestimmbar ist. Diese Eindeutigkeit tritt auf wenn die verwendete Anzahl an Kontroll-Qubits ausreicht, um φ eindeutig zu repräsentieren. Im oberen Beispiel kann φ eindeutig mit den drei verwendeten Kontrollqubits dargestellt werden: $\frac{3}{8} = 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3}$. Verwendet man nicht ausreichend Kontroll-Qubits kann φ nicht eindeutig repräsentiert werden. Als Konsequenz wird die Messung ungenau. Mit hoher Wahrscheinlichkeit kollabieren die Qubits bei einer Messung in die darstellbaren Zustände, die den genauen Wert am besten approximieren. In Abbildung 6 sind die Messergebnisse von einer Quanten-Phase-Estimation abgebildet, die die Phase $\varphi = \frac{5}{16}$ bestimmen soll. Da $\frac{5}{16}$ nicht mit 3-Qubits darstellbar ist, gibt es kein eindeutiges Messergebnis. Anhand der Messergebnisse ist aber erkennbar, dass die Messungen mit hoher Wahrscheinlichkeit, zu den bestmöglichen Zuständen kollabieren. Diese entsprechen $\frac{2}{8} = \frac{4}{16}$ und $\frac{3}{8} = \frac{6}{16}$, also genau die Werte um $\frac{5}{16}$.

6 Shor-Algorithmus

Im folgenden Kapitel wird der Shor-Algorithmus zum Faktorisieren von Zahlen beschrieben. Der Inhalt bezieht sich auf den Zweck, die Funktionsweise und den Aufbau des Algorithmus. Der Aufbau beinhaltet nicht die konkrete Implementierung der Bestandteile des Algorithmus. Stattdessen werden die Details bezüglich der Implementierung im nächsten Kapitel behandelt.

6.1 Zweck

Der Shor-Algorithmus wurde mit dem spezifischen Ziel entwickelt, große zusammengesetzte Zahlen effizient auf Quantencomputern in ihre Primfaktoren zu faktorisieren. Im Gegensatz zu klassischen Faktorisierungsverfahren, die exponentielle Zeit erfordern [KL23], ermöglicht Shor's Ansatz die Faktorisierung in polynomialer Zeit, in Bezug auf die Anzahl an Bits der zu faktorisierenden Zahl [Sho97]. Dies stellt eine signifikante Beschleunigung gegenüber den besten bekannten klassischen Faktorisierungsverfahren dar. Der Shor-Algorithmus bekräftigt die These, das Quantencomputer bestimmte Probleme wesentlich schneller lösen können als ihre klassischen Gegenstücke.

6.2 Funktionsweise

Der Shor-Algorithmus verwendet zwei Teilberechnungen, die zusammen die Faktorisierung berechnen. Die erste Teilberechnung erfolgt mittels eines Quantenalgorithmus. Der zweite Teil basiert auf einem klassischen Algorithmus. Im quantenmechanischen Teil des Shor-Algorithmus geht es um die Bestimmung der Ordnung in der multiplikativen Gruppe modulo N. Hierbei ist anzumerken, dass der Quantenalgorithmus nicht direkt die Primfaktoren der zu faktorisierenden Zahl berechnet. Stattdessen wird die Eigenschaft ausgenutzt, dass das Problem der Faktorisierung äquivalent zu dem Problem der Ordnungsbestimmung ist [NC10]. Daher impliziert eine effiziente Lösung für die Berechnung der Ordnung eine ebenso effiziente Methode zur Faktorisierung. Der nachfolgende klassische Algorithmus verwendet die ermittelte Ordnung, um die Primfaktoren abzuleiten. Beide Teilberechnungen, sowohl die quantenmechanische als auch die klassische, führen ihre Berechnungen in polynomialer Zeit durch. Daher liegt die gesamte Laufzeit des Shor-Algorithmus ebenfalls in einer polynomialen Größenordnung.

6.3 Ordnungsbestimmung

Zu bestimmen sind die Primfaktoren der Zahl N. Zuerst wird ein a mit $0 < a < N$ gewählt. Falls der ungewöhnlichen Fall eintritt, dass a nicht teilerfremd zu N ist, entspricht a einem der Primfaktoren. Anschließend wird die Ordnung beziehungsweise Periode p der Funktion $f(x) = a^x \mod N$ mit einem Quantenalgorithmus bestimmt:

Die Periode p beschreibt das kleinste ganzzahlige Element mit $p > 0$, für das gilt: $f(p) = 1 \mod N$.

x		0	1	2	3	4	5
7^x	mod 15	1	7	4	13	1	7

$\longmapsto p = 4$

Im Wesentlichen handelt es sich bei der quantenmechanischen Berechnung des Shor-Algorithmus um die Quanten-Phase-Estimation. Die Architektur dieser spezifischen Quanten-Phase-Estimation korrespondiert weitgehend mit der in Abschnitt 5.6 vorgestellten Struktur. Hierbei ersetzen speziell für die gegebene Anwendung definierte U-Gatter die allgemeinen.

Für den konkreten Kontext der Periodenberechnung, realisieren die U-Gatter die Transformation:

$$U\left|y\right\rangle = \left|ay \mod N\right\rangle$$

Die Ausführung der Quanten-Phase-Estimation erfordert die Erzeugung eines Eigenvektors der Transformation U. Da die Quanten-Phase-Estimation φ aus dem Eigenwert extrahiert, darf der Eigenvektor nicht den trivialen Eigenwert 1 besitzen. Stattdessen ist es notwendig, dass die Periode der Transformation im Eigenwert enthalten ist.

Wie in [NC10] gezeigt wird, gibt es zu U Eigenvektoren $\left|u_s\right\rangle$, mit $0 \leq s \leq p-1$:

$$\left|u_s\right\rangle \equiv \frac{1}{\sqrt{p}} \sum_{k=0}^{p-1} e^{\frac{-2\pi i s k}{p}} \left|a^k \mod N\right\rangle$$

Für $a = 7$ bei $N = 15$ entspricht $r = 4$. Ein Eigenvektor zu $s = 1$ lautet dann:

$$\left|u_1\right\rangle_4 = \frac{1}{\sqrt{4}}(\left|1\right\rangle_4 + e^{-\frac{2\pi i}{4}}\left|7\right\rangle_4 + e^{-\frac{4\pi i}{4}}\left|4\right\rangle_4 + e^{-\frac{6\pi i}{4}}\left|13\right\rangle_4)$$

$$U\left|u_1\right\rangle_4 = \frac{1}{\sqrt{4}}(\left|7\right\rangle_4 + e^{-\frac{2\pi i}{4}}\left|4\right\rangle_4 + e^{-\frac{4\pi i}{4}}\left|13\right\rangle_4 + e^{-\frac{6\pi i}{4}}\left|1\right\rangle_4)$$

$$U\left|u_1\right\rangle_4 = e^{\frac{2\pi i}{4}} \cdot \frac{1}{\sqrt{4}}(e^{-\frac{2\pi i}{4}}\left|7\right\rangle_4 + e^{-\frac{4\pi i}{4}}\left|4\right\rangle_4 + e^{-\frac{6\pi i}{4}}\left|13\right\rangle_4 + e^{-\frac{8\pi i}{4}}\left|1\right\rangle_4)$$

$$U\left|u_1\right\rangle_4 = e^{\frac{2\pi i}{4}} \cdot \frac{1}{\sqrt{4}}(e^{-\frac{8\pi i}{4}}\left|1\right\rangle_4 e^{-\frac{2\pi i}{4}}\left|7\right\rangle_4 + e^{-\frac{4\pi i}{4}}\left|4\right\rangle_4 + e^{-\frac{6\pi i}{4}}\left|13\right\rangle_4) = e^{\frac{2\pi i}{4}} \cdot \left|u_1\right\rangle_4$$

Das kann verallgemeinert werden:

$$U\left|u_s\right\rangle = e^{\frac{2\pi i s}{p}}\left|u_s\right\rangle$$

Wie man in der Definition von $\left|u_s\right\rangle$ sieht, benötigt die Initialisierung eines Eigenvektors $\left|u_s\right\rangle$ mit einem konkreten s die Periode p. Man kann diese Problematik jedoch umgehen

indem man anstelle eines einzelnen Eigenvektors $|u_s\rangle$ eine Superposition verwendet, die alle $|u_s\rangle$ umfasst. Die Superposition entspricht [NC10]:

$$\frac{1}{\sqrt{p}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle$$

Also werden die Qubits, die für den Eigenvektor bestimmt sind, mit dem Zustand $|1\rangle$ initialisiert.

Die Superposition der Eigenvektoren hat zur Folge, dass nach der inversen Quanten-Fourier-Transformation, also am Ende der Quanten-Phase-Estimation, ebenfalls eine Superposition mit dem φ_s der Eigenwerte λ_{u_s} aller möglichen Eigenvektoren $|u_s\rangle$ existiert. Sei k die Anzahl an Kontroll-Qubits dann:

$$\frac{1}{\sqrt{p}} \sum_{s=0}^{p-1} \left| 2^k \cdot \frac{s}{p} \right\rangle = \frac{1}{\sqrt{p}} (|0\rangle + \left| 2^k \cdot \frac{1}{p} \right\rangle + \left| 2^k \cdot \frac{2}{p} \right\rangle ... + \left| 2^k \cdot \frac{p-1}{p} \right\rangle)$$

Bei einer Messung wird also zufällig eines der $\varphi_s \approx \frac{s}{p}$ gemessen.

Die Genauigkeit von $\frac{s}{p}$ gegenüber dem φ_s hängt davon ab, mit wie viele U^{2^x}-Gatter beziehungsweise mit wie viele Kontroll-Qubits, die Quanten-Phase-Estimation ausgeführt wurde.

Mit einer ausreichenden Anzahl an verwendeten Kontroll-Qubits, können die Zustände vollkommen beschrieben werden und somit bei einer Messung die genauen φ_s gefunden.

Falls nicht ausreichend Kontroll-Qubits vorhanden sind, wird die Messung mit hoher Wahrscheinlichkeit, in den Zustand kollabieren der dem genauen Ergebnis am nächsten ist. Die Ergebnisse sind dann jedoch probabilistischer Natur, vergleichbar mit dem Beispiel aus dem Quanten-Phase-Estimation Kapitel, in Abbildung 6.

Anhand des gemessenen $\frac{s}{p}$ erfolgt die Primfaktorzerlegung in einer klassischen Nachberechnung.

6.4 Klassische Nachberechnung

Wie der Name dieses Abschnittes vermuten lässt, wird die Nachberechnung in der Regel mit einem klassischen Algorithmus durchgeführt.

Aus dem Messergebnis des Quanten-Phase-Estimation soll die Phase extrahiert werden. Anhand der Messung sind die ersten k Bits von φ_s bekannt. Dabei steht k für die Genauigkeit der Quanten-Phase-Estimation. Die Genauigkeit k wird durch die Anzahl an Kontroll-Qubits festgelegt. Sollten k Bits nicht vollständig ausreichen um φ_s zu beschreiben, wird man bei der Messung eine Kommazahl erhalten die nah von φ_s liegt, dieser aber nicht ganz entspricht. Wendet man den Kettenbruch-Algorithmus auf das Messergebnis an, wird von der Kommazahl aus, der nächste ganzzahlige Bruch gefunden. Man kann zeigen, dass die Verwendung von $k = 2n + 1$ Kontrollqubits eine ausreichende Genauigkeit liefert, so dass das Messergebnis unter Verwendung des Kettenbruch-Algorithmus zu einem Näherungsbruch $\frac{s}{p}$ von φ_s führt [NC10].

Nichtsdestotrotz kann selbst bei der Verwendung von ausreichend Kontroll-Qubits ein $\frac{s'}{p'}$ berechnet werden, welches nicht die Periode p enthält. Dies tritt auf wenn s und p einen gemeinsamen Teiler haben der die Kürzung von $\frac{s}{p}$ auf $\frac{s'}{p'}$ ermöglicht.

Findet man in der ersten Messung $\frac{s'}{p'}$ und in einer zweiten $\frac{s''}{p''}$, wobei s' und s'' keine Faktoren teilen, so kann aus dem kleinsten gemeinsamen Vielfachen von p' und p'' p berechnen.

Man kann zeigen, dass bei $2\log(N)$ Messungen, die Wahrscheinlichkeit sehr hoch ist, mindestens einmal ein $\frac{s}{p}$ zu messen, bei dem s und p teilerfremd sind [NC10].

Ob die korrekte Periode gefunden wurde, kann mit $a^p = 1 \mod N$ geprüft werden.

Nach einige fehlgeschlagenen Versuchen(z.B. $2\log(N)$) wiederholt man die Suche mit einem anderen a.

Sobald die korrekte Periode p gefunden wurde, können die Primfaktoren von Nmit dem gemeinsamen Teiler(gcd) berechnet werden:

$$gcd(a^{\frac{p}{2}} - 1, N), gcd(a^{\frac{p}{2}} + 1, N)$$

Dies schlägt nur fehl falls r ungerade ist oder falls $a^{\frac{p}{2}} = -1 \mod N$ erfüllt [Sho97]. Die Wahrscheinlichkeit dass einer der beiden genannten Fälle eintritt beträgt $1 - \frac{1}{2^k}$, wobei k die Anzahl an unterschiedlicher Primfaktoren von N angibt [Sho97]. In einem solchen Fall, wiederholt man die Periodenberechnung mit einem anderen a.

7 Implementierung

7.1 Quantenalgorithmus

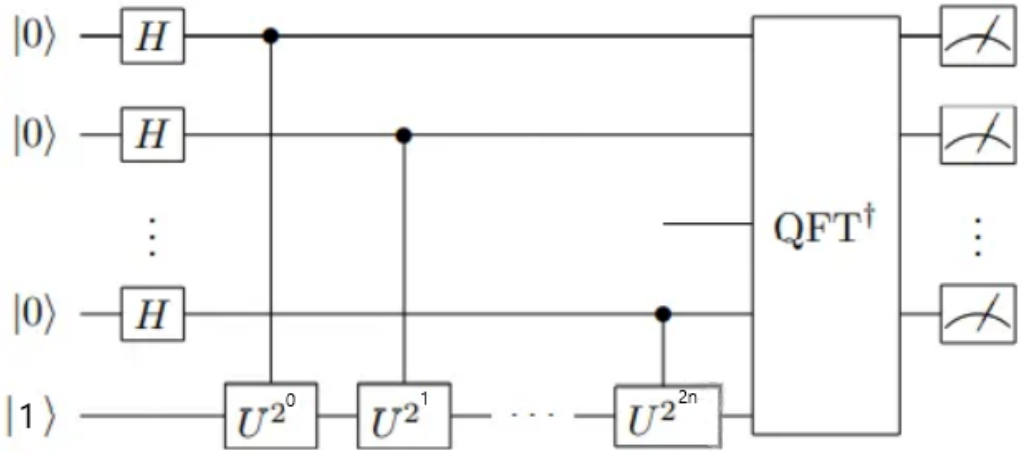
Wie im Kapitel 6.2 zur Funktionsweise erklärt, wird für die Periodenbestimmung die Quanten-Phase-Estimation genutzt.

Um den Quanten-Phase-Estimation Algorithmus für die Periodenberechnung zu nutzen, benötigt man ein Gatter U welches die modulare Multiplikation $U|y\rangle = |ay \mod N\rangle$, als eine unitär Transformation realisiert. Mit den passenden U-Gatter wird der Quantenschaltkreis wie in Abbildung 7 strukturiert.

Die Realisierung der Transformation bedingt die Implementierung einiger arithmetischer Operationen in Form eines Quantenschaltkreises. Diese fungieren als Bausteine, die zusammengesetzt zur Konstruktion des übergeordneten Quantenschaltkreises für die modulare Multiplikation beitragen. Zu den erforderlichen arithmetischen Operationen gehört die Addition, Subtraktion sowie die modulare Addition.

In den folgenden Abschnitten werden die untergeordneten arithmetischen Operationen bis hin zur modularen Multiplikation implementiert.

Abbildung 7: QPE für Shor



7.1.1 Addition

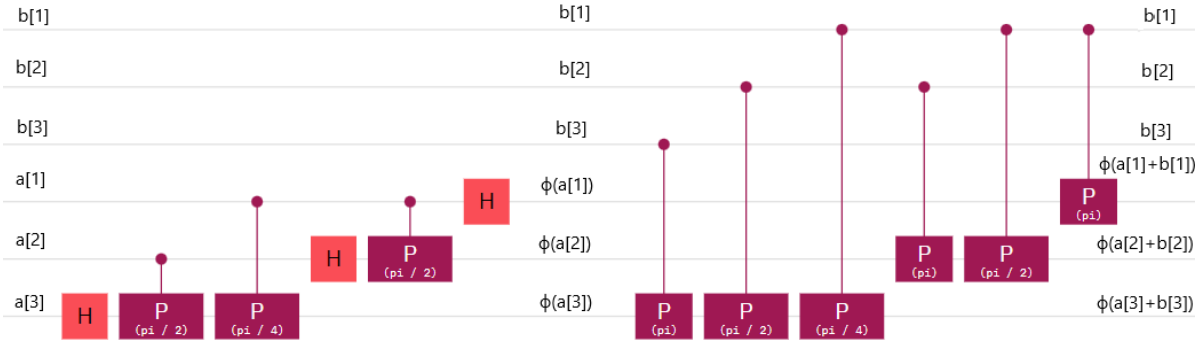
Der Quantenschaltkreis für die Addition bildet das Fundament der U-Gatter und stellt einen der am häufigsten verwendeten Bausteine dar. Deswegen hat die Implementierung der Addition einen erheblichen Einfluss auf den Ressourcenbedarf des gesamten Quantenalgorithmus und sollte daher möglichst effizient implementiert werden.

Eine Möglichkeit, die Addition als Quantenschaltkreis zu realisieren, besteht im Nachbau eines klassischen Schaltkreises aus Volladdierern. Da es nicht möglich ist, die notwendige klassischen Gatter wie AND und OR als unitäre Transformation mit nur zwei Qubits darzustellen [Hoe23b], werden zusätzliche Hilfsqubits benötigt. Die zusätzlichen Hilfsqubits bewirken, dass der Nachbau eines klassischen Schaltkreis für die Addition zweier n-Bit Zahlen, also solche der Größenordnung 2^n, mindestens $3n$ Qubits benötigt [Zal98].

Eine effizientere Methode, die ohne Hilfsqubits auskommt, ist die Quanten-Addition [Dra00]. Die Quanten-Addition führt die Berechnung auf quantenmechanische Weise durch. Im Wesentlichen wird dabei die Addition in der Fourier-Basis berechnet, wobei die Phasen der Qubits eines Summanden mit kontrollierte Phasenverschiebungen auf die Qubits des anderen Summanden wirkt.

Im Folgenden wird ein Beispiel für die Quanten-Addition zweier Qubit-Register $|a\rangle_3$ und $|b\rangle_3$, jeweils bestehend aus drei Qubits, betrachtet:

Abbildung 8: Quantum-Addition



Die Registermarkierungen in der Mitte von Abbildung 8 unterteilen die Darstellung in zwei Hälften. Die linke Hälfte repräsentiert die Quanten-Fourier-Transformation, während die rechte Hälfte die Quanten-Addition zeigt.

Wie man an der Struktur der Quanten-Addition erkennen kann, ist die Anordnung der Gatter fast identisch mit der Quanten-Fourier-Transformation. Ein Unterschied besteht darin, dass die Hadamard-Gatter durch kontrollierte $P(\pi)$ Phasen-Gatter ersetzt wurden. Sowohl das Hadamard-Gatter als auch das $P(\pi)$ Phasen-Gatter erzeugen eine relative Phase von $e^{\pi i}$. Ein weiterer Unterschied zur Quanten-Fourier-Transformation besteht darin, dass die Phasen-Gatter nicht durch das gleiche Register$|a\rangle_3$ kontrolliert werden, auf das die Gatter auch wirken. Stattdessen kontrollieren die Qubits des Registers $|b\rangle_3$ die Phasen-Gatter. Dabei wird das $P(\pi)$ Phasen-Gatter durch das Qubit des $|b\rangle_3$ kontrolliert, welches die selbe Wertigkeit hat wie das Zielqubit des $|a\rangle_3$ Registers. Jedes weitere kontrollierte Phasen-Gatter für das gleiche Zielqubit wird fortlaufend von dem nächstkleineren Qubit von $|b\rangle_3$ kontrolliert.

Im Prinzip handelt es sich bei dieser Quantenschaltung um eine Anwendung derselben Phasenverschiebungen wie bei der Quanten-Fourier-Transformation. Der grundlegende Unterschied liegt darin, dass diese Phasenverschiebungen kontrolliert auf ein anderes Quantenregister angewendet werden.

Die Wirkung der Quanten-Addition wird anhand der Abbildung 8 verdeutlicht: Am Anfang der linken Hälfte befinden sich beide Register in der Standardbasis. Auf das Zielregister $|a\rangle_3$ wirkt die Quanten-Fourier-Transformation ohne Swap Gatter. Dadurch befindet sich $|a\rangle_3$ nun in der Fourier-Basis Φ, also $|\Phi(a)\rangle_3$:

$$|\Phi(a)\rangle_3 = \frac{1}{\sqrt{8}}[(|0\rangle + e^{\frac{2\pi i(2^0 a_1)}{2^1}}|1\rangle) \bigotimes (|0\rangle + e^{\frac{2\pi i(2^1 a_2 + 2^0 a_1)}{2^2}}|1\rangle) \bigotimes (|0\rangle + e^{\frac{2\pi i(2^2 a_3 + 2^1 a_2 + 2^0 a_1)}{2^3}}|1\rangle)]$$

Anschließend wirkt auf das hinterste Tensorprodukt ein $P(\pi)$ Phasen-Gatter, welches durch $|b_3\rangle_1$ kontrolliert wird. Wenn sich $|b_3\rangle_1$ im Zustand $|0\rangle_1$ befindet, passiert nichts. Wenn es sich im Zustand $|1\rangle_1$ befindet, dass das Phasen-Gatter angewendet wird. Dieses Verhalten kann man für beide Fälle mit den entsprechenden Matrizen $\begin{pmatrix} 1 & 0 \\ 0 & e^{\pi i b_3} \end{pmatrix}$ bezie-

hungsweise $\begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i(2^2 b_3)}{2^3}} \end{pmatrix}$ beschreiben. Schreibt man das hinterste Tensorprodukt als Vektor, ergibt sich die folgende Formulierung:

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{2\pi i(2^2 a_3 + 2^1 a_2 + 2^0 a_1)}{2^3}} |1\rangle) \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ e^{\frac{2\pi i(2^2 a_3 + 2^1 a_2 + 2^0 a_1)}{2^3}} \end{pmatrix}$$

Dann wird durch das Ergebnis der Verrechnung mit dem Phasen-Gatter deutlich, dass die Addition im Wesentlichen in der Phase des Quantenzustands stattfindet:

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i(2^2 b_3)}{2^3}} \end{pmatrix} \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ e^{\frac{2\pi i(2^2 a_3 + 2^1 a_2 + 2^0 a_1)}{2^3}} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ e^{\frac{2\pi i(2^2 (a_3+b_3) + 2^1 a_2 + 2^0 a_1)}{2^3}} \end{pmatrix}$$

Wie in der Abbildung 8 erkenntlich, wirken auf das hinterste Tensorprodukt auch noch die beiden Phasen-Gatter $P(\frac{\pi}{2})$ und $P(\frac{\pi}{4})$ mit:

$$P(\frac{\pi}{2}) = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{\pi}{2} i b_2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i(2^1 b_2)}{2^3}} \end{pmatrix} \; ; \; P(\frac{\pi}{4}) = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{\pi}{4} i b_1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i(2^0 b_1)}{2^3}} \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i(2^0 b_1)}{2^3}} \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i(2^1 b_2)}{2^3}} \end{pmatrix} \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ e^{\frac{2\pi i(2^2 (a_3+b_3) + 2^1 a_2 + 2^0 a_1)}{2^3}} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ e^{\frac{2\pi i(2^2 (a_3+b_3) + 2^1 (a_2+b_2) + 2^0 (a_1+b_1))}{2^3}} \end{pmatrix}$$

Wendet man alle weiteren Phasen-Gatter auf das vollstände Tensorprodukt an, erhält man:

$$\frac{1}{\sqrt{8}}[(|0\rangle + e^{\frac{2\pi i(2^0 (a_1+b_1))}{2^1}} |1\rangle) \bigotimes (|0\rangle + e^{\frac{2\pi i(2^1 (a_2+b_2) + 2^0 (a_1+b_1))}{2^2}} |1\rangle) \bigotimes (|0\rangle + e^{\frac{2\pi i(2^2 (a_3+b_3) + 2^1 (a_2+b_2) + 2^0 (a_1+b_1))}{2^3}} |1\rangle)]$$

Setzt man in diese Formel zwei Zahlen in Binärschreibweise ein, wird man den selben Zustand erhalten, wie wenn man die Summe der beiden Zahlen in die Formel der Quanten-Fourier-Transformation einsetzt. Beispielsweise sei $a = 3$ also binär $a_3 = 0$, $a_2 = 1$,$a_1 = 1$ und $b = 1$ also $b_3 = 0$, $b_2 = 0$,$b_1 = 1$:

$$\frac{1}{\sqrt{8}}[(|0\rangle + e^{\frac{2\pi i(2^0 (1+1))}{2^1}} |1\rangle) \bigotimes (|0\rangle + e^{\frac{2\pi i(2^1 (1+0) + 2^0 (1+1))}{2^2}} |1\rangle) \bigotimes (|0\rangle + e^{\frac{2\pi i(2^2 (0+0) + 2^1 (1+0) + 2^0 (1+1))}{2^3}} |1\rangle)]$$

$$= \frac{1}{\sqrt{8}}[(|0\rangle + |1\rangle) \bigotimes (|0\rangle + |1\rangle) \bigotimes (|0\rangle + e^{\pi i} |1\rangle)]$$

Das dies tatsächlich die Summe in Fourier-Basis entspricht, wird deutlich wenn man das selbe Tensorprodukt aus der Quanten-Fourier-Transformation bildet:

$$QFT(|c\rangle_3) \frac{1}{\sqrt{8}}[(|0\rangle + e^{\frac{2\pi i(2^0 (c_1))}{2^1}} |1\rangle) \bigotimes (|0\rangle + e^{\frac{2\pi i(2^1 (c_2) + 2^0 (c_1))}{2^2}} |1\rangle) \bigotimes (|0\rangle + e^{\frac{2\pi i(2^2 (c_3) + 2^1 (c_2) + 2^0 (c_1))}{2^3}} |1\rangle)]$$

Die Summe von a und b entspricht $c = 4$ also $c_3 = 1,\ c_2 = 0,\ c_1 = 0$:

$$QFT(|4\rangle_3)\frac{1}{\sqrt{8}}[(|0\rangle+e^{\frac{2\pi i(2^0(0))}{2^1}}|1\rangle)\bigotimes(|0\rangle+e^{\frac{2\pi i(2^1(0)+2^0(0))}{2^2}}|1\rangle)\bigotimes(|0\rangle+e^{\frac{2\pi i(2^2(1)+2^1(0)+2^0(0))}{2^3}}|1\rangle)]$$

$$= \frac{1}{\sqrt{8}}[(|0\rangle + e^{\frac{2\pi i(0)}{2^1}}|1\rangle)\bigotimes(|0\rangle + e^{\frac{2\pi i(0)}{2^2}}|1\rangle)\bigotimes(|0\rangle + e^{\frac{2\pi i(2^2(1))}{2^3}}|1\rangle)]$$

$$= \frac{1}{\sqrt{8}}[(|0\rangle + |1\rangle)\bigotimes(|0\rangle + |1\rangle)\bigotimes(|0\rangle + e^{\pi i}|1\rangle)]$$

Das Ergebnis der Quanten-Addition zweier Summanden $a = 3,\ b = 1$, jeweils in einem Register mit drei Qubits, ist somit identisch mit dem Zustand, der durch die Anwendung der Quanten-Fourier-Transformation auf ein Register aus ebenfalls drei Qubits mit der Summe der beiden Zahlen entsteht.

Mit einer anschließenden inversen Quanten-Fourier-Transformation, kann die Summe in die Standardbasis und somit in einen Messbaren Zustand transformiert werden:

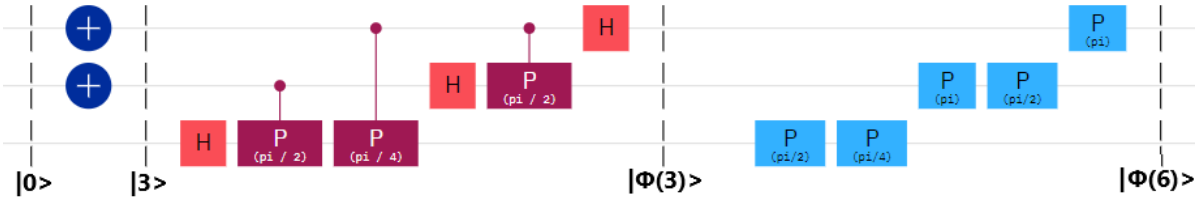
$$iQFT(|\Phi(4)_3\rangle) \equiv iQFT(\frac{1}{\sqrt{8}}[(|0\rangle + |1\rangle)\bigotimes(|0\rangle + |1\rangle)\bigotimes(|0\rangle + e^{\pi i}|1\rangle)]) = |4\rangle_3$$

Für die Realisierung der modularen Multiplikation wird zu keinem Zeitpunkt der Berechnung eine Addition zweier Zwischenergebnisse benötigt. Genauer gesagt, ist es nicht nötig, ein Quantenregister auf ein anderes zu addieren. Stattdessen wird die Quanten-Addition benutzt, um eine vorab bekannte Zahl auf ein Quantenregister zu addieren.

Bei der Quanten-Addition mit zwei Register wie in Abbildung 8 erfolgt die Phasenverschiebung kontrolliert, also in Abhängigkeit des Inhaltes von Register $|b\rangle_3$. Ist der Inhalt von Register $|b\rangle_3$ vorab bekannt, können gewöhnliche Phasen-Gatter anstelle von kontrollierten verwendet werden [Bea03].

Wenn bei der Quanten-Addition mit zwei Registern ein Phasen-Gatter aufgrund des zugehörigen Kontrollqubits im Zustand $|1\rangle$ angewendet wird, wird es in der Variante mit einem einzelnen Register als gewöhnliches Phasen-Gatter verwendet. Ist das Kontrollqubit hingegen im Zustand $|0\rangle$, wodurch das Phasen-Gatter bei der Quanten-Addition mit zwei Registern nicht zur Anwendung kommt, wird dieses Phasen-Gatter in der Variante mit nur einem Register weggelassen.

Abbildung 9: Quantum-Addition fixierte Phasenverschiebungen



In Abbildung 9 ist die Quanten-Addition für ein 3-Qubit Register abgebildet. Die blauen Phasen-Gatter sorgen für die Quanten-Addition mit einem fixierten Wert von 3. Vergleicht man die Abbildung 9 mit der Abbildung 8 fällt auf, dass das aller erste Phasen-Gatter der Quanten-Addition nicht vorkommt. Im Quantenschaltkreis der Abbildung 8 würde ein Registerinhalt von $b = 3$ das Kontrollqubit b_3 nicht setzen. Somit kommt das erste Phasen-Gatter der Quanten-Addition nicht zum Einsatz und wird deswegen in der Variante aus Abbildung 9 weggelassen.

8 Resultate

9 Verlauf

9.1 Rückblick

9.2 Ausblick

Literaturverzeichnis

[23a] *Gemeinsame Umfrage von BSI und KPMG in Deutschland zu „Kryptografie und Quantencomputing“*. 2023. URL: https://www.bsi.bund.de/DE/Service-Navi/Presse/Alle-Meldungen-News/Meldungen/BSI_KMPG_Quanten_230418.html (besucht am 02. 08. 2023).

[23b] *IBM Quantum Development Roadmap*. 2023. URL: https://www.ibm.com/quantum/roadmap (besucht am 02. 08. 2023).

[23c] *Unveiling our new Quantum AI campus*. 2023. URL: https://blog.google/technology/ai/unveiling-our-new-quantum-ai-campus/ (besucht am 02. 08. 2023).

[Bea03] Stephane Beauregard. *Circuit for Shor's algorithm using 2n+3 qubits*. 2003. arXiv: quant-ph/0205095 [quant-ph].

[Ben80] Paul Benioff. „The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines“. In: *Journal of Statistical Physics* 22.5 (1980), S. 563–591. DOI: 10.1007/BF01011339.

[Cor+09] Thomas H. Cormen u. a. *Introduction to Algorithms*. 3rd. MIT Press, 2009, S. 963.

[Deu85] David Deutsch. „Quantum theory, the Church–Turing principle and the universal quantum computer“. In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400 (1985), S. 97–117. DOI: 10.1098/rspa.1985.0070.

[DH76] W. Diffie und M. Hellman. „New directions in cryptography“. In: *IEEE Transactions on Information Theory* 22.6 (1976), S. 644–654. DOI: 10.1109/TIT.1976.1055638.

[Dir39] P. A. M. Dirac. „A new notation for quantum mechanics“. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 35.3 (1939), S. 416–418. DOI: 10.1017/S0305004100021162.

[DiV00] David P. DiVincenzo. „The Physical Implementation of Quantum Computation“. In: *Fortschritte der Physik* 48.9-11 (Sep. 2000), S. 771–783. DOI: 10.1002/1521-3978(200009)48:9/11<771::aid-prop771>3.0.co;2-e. URL: https://doi.org/10.1002%2F1521-3978%28200009%2948%3A9%2F11%3C771%3A%3Aaid-prop771%3E3.0.co%3B2-e.

[Dra00] Thomas G. Draper. *Addition on a Quantum Computer*. 2000. arXiv: quant-ph/0008033 [quant-ph].

[Fey82] Richard P. Feynman. „Simulating physics with computers“. In: *International Journal of Theoretical Physics* 21.6 (1982), S. 467–488. DOI: 10.1007/BF02650179. URL: https://doi.org/10.1007/BF02650179.

[Gro96] Lov K. Grover. *A fast quantum mechanical algorithm for database search*. 1996. arXiv: quant-ph/9605043 [quant-ph].

[Hoe22] Georg Hoever. *Kryptologie*. Skript zur Vorlesung Kryptologie, Fachhochschule Aachen. 2022, S. 28.

[Hoe23a] Georg Hoever. *Münzbeispiel*. Mündlich/Symbolisch in der Vorlesung Quantencomputing, Fachhochschule Aachen. 2023.

[Hoe23b] Georg Hoever. *Quanten Computing*. Skript zur Vorlesung Quanten Computing, Fachhochschule Aachen. 2023, S. 107.

[Hom] Matthias Homeister. *Quantum Computing verstehen*. 5. Aufl. Springer, S. 215.

[JM98] J. A. Jones und M. Mosca. „Implementation of a quantum algorithm on a nuclear magnetic resonance quantum computer“. In: *The Journal of Chemical Physics* 109.5 (Aug. 1998), S. 1648–1653. DOI: 10.1063/1.476739. URL: https://doi.org/10.1063%2F1.476739.

[KL23] Jonathan Katz und Yehuda Lindell. *Introduction to modern cryptography*. 2. Aufl. Boca Raton, FL: CRC Press, 2023. ISBN: 9781466570269.

[NC10] Michael A. Nielsen und Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511976667.

[RG17] Lidia Ruiz-Perez und Juan Carlos Garcia-Escartin. „Quantum arithmetic with the quantum Fourier transform“. In: *Quantum Information Processing* 16.6 (2017), S. 152. ISSN: 1573-1332. DOI: 10.1007/s11128-017-1603-1. URL: https://doi.org/10.1007/s11128-017-1603-1.

[Sho97] Peter W. Shor. „Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer“. In: *SIAM Journal on Computing* 26.5 (Okt. 1997), S. 1484–1509. DOI: 10.1137/s0097539795293172. URL: https://doi.org/10.1137%2Fs0097539795293172.

[Zal98] Christof Zalka. *Fast versions of Shor's quantum factoring algorithm.* 1998. arXiv: quant-ph/9806084 [quant-ph].

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe. Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht. Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Aachen, den 23. August 2023