

# Quantencomputer – Ein Spiel mit Qubits

EIN LEITFADEN ZUM VERSTÄNDNIS DES  
KOMPLEXESTEN COMPUTERS UNSERER ZEIT

MAURICE WEINGÄRTNER, G4A  
MATURAAARBEIT, KANTONSSCHULE BADEN

ERSTBETREUER: DR. CHRISTOPH BERSIER,  
ZWEITBEWERTER: DR. STEFAN GUGGENBÜHL

23.11.2021

## Inhaltsverzeichnis

1	Einleitung .....	5
2	Was ist ein Quantencomputer? .....	9
2.1	Die physikalischen Grundlagen .....	9
2.1.1	Statistische Theorie .....	10
2.1.2	Unbestimmtheit .....	11
2.1.3	Messung .....	11
2.1.4	Verschränkung .....	12
2.1.5	Weitere physikalische Grundlagen .....	12
2.2	Die mathematischen Grundlagen .....	13
2.2.1	Komplexe Zahlen .....	13
2.2.2	Vektorräume .....	13
2.2.3	Matrizen .....	14
2.3	Das Qubit .....	14
2.3.1	Realisierung von Qubits .....	18
2.3.2	Quantenregister .....	20
2.4	Verschiedene Ansätze zur Implementierung von Quantencomputern .....	20
2.4.1	Quantengattercomputer .....	20
2.4.2	Quanten-Annealing-Maschinen .....	22
3	Programmierung von Quantencomputern .....	26
3.1	Klassische Logikgatter .....	26
3.1.1	NOT / NICHT – Gatter .....	26
3.1.2	AND / UND-Gatter .....	26
3.1.3	OR / ODER-Gatter .....	26
3.1.4	XOR-Gatter .....	27
3.1.5	Wahrheitstabellen .....	27
3.2	Quantengatter .....	28
3.2.1	Pauli-Gatter .....	28
3.2.2	Hadamard-Gatter .....	30
3.2.3	Phasenverschiebungs-Gatter .....	31
3.2.4	Z-, S- und T-Gatter .....	31
3.2.5	CNOT Gatter .....	32
3.3	Quantenschaltungen .....	33
4	Spieltheorie .....	34
4.1	Klassische Spieltheorie .....	34
4.1.1	Nash-Gleichgewicht .....	34
4.2	Quanten-Spieltheorie .....	37

5	Anwendung der Quanten Spieltheorie .....	38
5.1	Coin Flip .....	38
5.1.1	Anwendung von Quanten Strategien .....	39
5.1.2	Prototyp Coin Flip.....	42
5.1.3	Resultat des Coin Flip .....	46
5.2	Quanten Tic-Tac-Toe.....	46
5.2.2	Anwendung von Quanten Strategien .....	48
5.2.3	Prototyp Quanten Tic-Tac-Toe .....	51
5.2.4	Resultate der Experimente zum Quanten Tic-Tac-Toe.....	58
6	Zusammenfassung und Ausblick .....	66
7	Anhang.....	68
7.1	Source Code des Coin Flip.....	68
7.1.1	HMTL .....	68
7.1.2	JavaScript.....	68
7.2	Source Code des Prototypen Version 1 – Tic-Tac-Toe .....	70
7.2.1	HMTL .....	70
7.2.2	JavaScript – main.js .....	70
7.3	Source Code des Prototypen Version 2 – Tic-Tac-Toe .....	73
7.3.1	HMTL .....	73
7.3.2	JavaScript.....	74
8	Literaturverzeichnis .....	78
8.1	Literaturverzeichnis mit Angabe des Autors.....	78
8.2	Literatur ohne Angabe des Autors .....	81
9	Abbildungsverzeichnis.....	85

## Abstract

Quantencomputer haben in den letzten Jahrzehnten eine gewaltige Entwicklung durchlaufen. Sie basieren anders als klassische Computer auf der Quantenphysik, der physikalischen Theorie von Gesetzmässigkeiten auf Ebene von Atomen. Die Quantenphysik ist "der Bereich der Physik, der sich mit dem Verhalten und der Wechselwirkung kleinster Teilchen befasst"<sup>1</sup> und beschreibt Phänomene, welche mit der klassischen Physik nicht erklärt werden können. Die Quantenmechanik, ein Teilgebiet der Quantenphysik, beschreibt die Zustände und Bewegungen von Elementarteilchen und deren gegenseitige Reaktionen. Die Quantenmechanik unterscheidet sich sowohl in ihren physikalischen Gesetzmässigkeiten als auch in den mathematischen Grundlagen komplett von ihrem klassischen Pendant. Daraus folgt auch, dass bei der Quantenmechanik eine komplett andere Logik als bei der klassischen Physik und damit auch bei klassischen Computern gilt. Dieser Zusammenhang motivierte mich dazu, Quantencomputer genau zu verstehen und diese auf eine verständliche und spannende Art darzustellen. Meine Maturaarbeit beschreibt, wie Quantencomputer funktionieren, wie man auf Quantencomputer programmieren kann und zeigt anhand von zwei Spielen, welche Vorteile durch die Anwendung von Quantenlogik erreicht werden können. Daneben werden auch die physikalischen und mathematischen Grundlagen der Quantencomputer beschrieben.

Die fundamentalen Unterschiede in der Logik zwischen der Quantenphysik und der klassischen Physik werden in meiner Arbeit mithilfe der Spieltheorie erklärt. Die Spieltheorie befasst sich mit der Modellierung von Entscheidungsprozessen im Rahmen von Gesellschaftsspielen sowie der Logik zur Definition möglicher Strategien. Anhand zweier Spiele, dem Münzwurf und dem Spiel Tic-Tac-Toe, wird dies verständlich aufgezeigt. Es werden zuerst die klassischen Spielvarianten mit der zugehörigen Spieltheorie anschaulich dargelegt. Anschliessend werden die Unterschiede bei Anwendung der Quanten Spieltheorie gegenüber der klassischen Spieltheorie beschrieben und mit Beispielen unterlegt. Die Quanten-Spiele wurden in mehreren Versionen als Webapplikationen mithilfe von HTML und JavaScript programmiert. Damit können unterschiedliche Quanten Strategien anschaulich nachvollzogen werden. So ist es auch für Laien einfach möglich die Phänomene der Quantenphysik zu verstehen.

## Zielpublikum, Voraussetzungen und Form

Die Arbeit ist interessant für alle, die sich gerne intensiver mit dem Thema Quantenphysik und Quantencomputer auseinandersetzen wollen und daran interessiert sind, wie dieses Thema die Informatik in den nächsten Jahren voraussichtlich stark prägen wird. Sie ermöglicht es zu verstehen, in welchen Bereichen Quantencomputer klassischen Computern überlegen sind und wie die Programmierung dieser neuen Klasse von Computern durchgeführt wird. Die physikalischen und mathematischen Grundlagen der Quantenmechanik machen diese Abschnitte auch als Zusammenfassung und Überblick interessant. Leser sollten Neugier und Unvoreingenommenheit mitbringen. Alle Grundlagen werden eingeführt, so dass keine speziellen Vorkenntnisse vorausgesetzt werden.

Betreuer

Dr. Christoph Bersier (Erstbetreuer) und Dr. Stefan Guggenbühl (Zweitbetreuer)

---

<sup>1</sup> Quantenphysik, in: chemie.de, 05.11.2021.

# 1 Einleitung

Unsere heutigen Computer basieren auf den gleichen Prinzipien wie die frühen Rechenmaschinen von 1940.<sup>2</sup> Ihre Leistungsfähigkeit wurde im Laufe der Zeit kontinuierlich und beeindruckend gesteigert. Historisch gesehen wurde die Rechenleistung in der PC-Ära alle 1,5 Jahre verdoppelt.<sup>3</sup> Diese Leistungssteigerung erreichte man durch Skalierung der Transistoren und viele andere Techniken wie System-, Architektur- und Software-Innovationen. Die schnellsten Hochleistungssysteme bezeichnet man dabei als "Supercomputer".

Es gibt hier jedoch ein grosses "Aber". Wie IBM es ausdrückt:<sup>4</sup>

**"For some problems, supercomputers aren't *that* super"**

Ein Beispiel für Probleme, die unsere heutigen Computer nicht effizient lösen können, sind die sogenannten "kombinatorischen Optimierungsprobleme". Hier geht es darum, aus einer grossen Menge von Elementen eine Teilmenge zu konstruieren, die gewissen Nebenbedingungen entspricht und bezüglich einer Kostenfunktion optimal ist.<sup>5</sup> Schon für eine überschaubare Menge von Elementen gilt es hier sehr viele mögliche Kombinationen auszuwerten, beispielsweise gibt es über 3 Millionen Möglichkeiten, 10 Leute, um einen Tisch zu platzieren (berechnet als  $10!$ ). IBM identifiziert zwei Gründe, warum Supercomputer nicht in der Lage sind, diese Probleme effizient zu lösen<sup>6</sup>

1. Supercomputer haben nicht den Arbeitsspeicher, um die unzähligen Kombinationen von Problemen der realen Welt zu speichern.
2. Supercomputer müssen jede Kombination- eine nach der anderen- analysieren, was sehr lange dauern kann.

Der Physiker Richard Feynman stellte darüber hinaus fest, dass klassische Computer schlecht darin sind, quantenmechanische Problemstellungen zu lösen. Er schlug daher als einer der ersten vor, Computer, welche Gesetze der Quantenmechanik ausnutzen, für Aufgaben zu verwenden, die für herkömmliche Computer unmöglich sind.<sup>7</sup> Ein weiterer Auslöser für die Entwicklung von Quantencomputern ist die Tatsache, dass die Bauteile von Computern immer kleiner werden. Es ist absehbar, dass es Bauteile von der Grösse einzelner Atome geben wird. Hier gelten die Gesetze der klassischen Physik nicht mehr und man muss sich auf die Quantenmechanik einlassen.<sup>8</sup>

Die Grundlagen von Quantencomputern wurden genau genommen bereits 1905 gelegt, als Albert Einstein in Zusammenhang mit der Erklärung des photoelektrischen Effekts den Begriff des Lichtquants einführte.<sup>9</sup> Seit den 80er Jahren, d.h. seit über 30 Jahren, war die Idee eines Quantencomputers in den Köpfen von Grundlagenforschern, basierend auf den Ideen von Paul Benioff, Yuri Manin und Richard Feynman. In den 1990 Jahren wurden von Peter Shor und Lov Grover zwei Quantenalgorithmen erarbeitet, einerseits für das Faktorisierungsproblem und andererseits die sogenannte unstrukturierte Datenbanksuche, die einer Lösung mit klassischen Computern überlegen sind. Ausserdem wurde klar, dass durch Quantenalgorithmen die Sicherheit der Public-Key-Infrastruktur, einer weit verbreiteten Verschlüsselungstechnologie gefährdet sein könnte, was sogar Geheimdienste aktivierte.

---

<sup>2</sup> Homeister, Quantencomputing, S.1.

<sup>3</sup> Stroh: Multi-Chip Technologien, 05.05.2021.

<sup>4</sup> What is quantum computing?, 05.05.2021.

<sup>5</sup> Kombinatorische Optimierung, in: Wikipedia, 05.05.2021.

<sup>6</sup> What is quantum computing?, 05.05.2021.

<sup>7</sup> Spentzouris : Quantum Computing, 06.05.2021.

<sup>8</sup> Homeister, Quantencomputing, S. 9.

<sup>9</sup> Press: 27 Milestones in the History of Quantum Computing, 06.05.2021.

Wo stehen wir heute? Zusammengefasst kann man sagen, dass die heute realisierten Quantencomputer sich auf Qubits in der Grössenordnung bis ca. 60 beschränken. Anwendungen benötigen aber Tausende oder gar Millionen stabile Qubits.

2019 realisierte Google einen Quantencomputer basierend auf dem Sycamore-Chip mit 53 Qubits und konnte damit eine praktisch wenig relevante Aufgabe lösen, die ein konventioneller Computer nicht in einer akzeptablen Zeitspanne bewältigen könnte. Diese Tatsache bezeichnet man als Quantenüberlegenheit (Quantum Supremacy). Der Sycamore-Chip benötigte zur Berechnung 200 s anstelle von 10.000 Jahren (Aussage Google) oder von 2,5 Tagen (Aussage IBM) – bei Ausführung auf dem zu dieser Zeit leistungsfähigsten klassischen Supercomputer Summit von IBM. Dies markiert also immerhin einen Speed-up um den Faktor 1080 bis 1,578 Mrd..<sup>10</sup>

Der grösste Quantencomputer von IBM hat zur Zeit 65 Qubits. IBM plant aber, 2023 einen Chip mit 1000 Qubits zu realisieren. Forschern und Entwicklern stellt IBM heute kleinere Quantencomputer mit 5-16 Qubits zur Verfügung.

Ende 2020 demonstrierte die Firma Xanadu aus Toronto mit ihrem Quantencomputer Quantenüberlegenheit. Allerdings benötigte sie zur Realisierung enorm viel Platz: eine komplette Tischplatte. Das Unternehmen PsiQuantum behauptet jedoch, sie hätten das Platzproblem im Griff und wären in der Lage, bis zu einer Million Qubits zu verknüpfen. Forscher in China berichten von realisierten Quantencomputern mit 66 Qubits<sup>11</sup>.

Die Firma D-Wave realisiert Quantenrechner mit über 5000 Qubits. Allerdings lassen sich diese nicht einzeln manipulieren und sind damit nur für wenige eingeschränkte Anwendungsbereiche einzusetzen. Mit den universellen, also beliebig einsetzbaren Quantencomputern von IBM und Google, sind sie daher nicht zu vergleichen.<sup>12</sup>

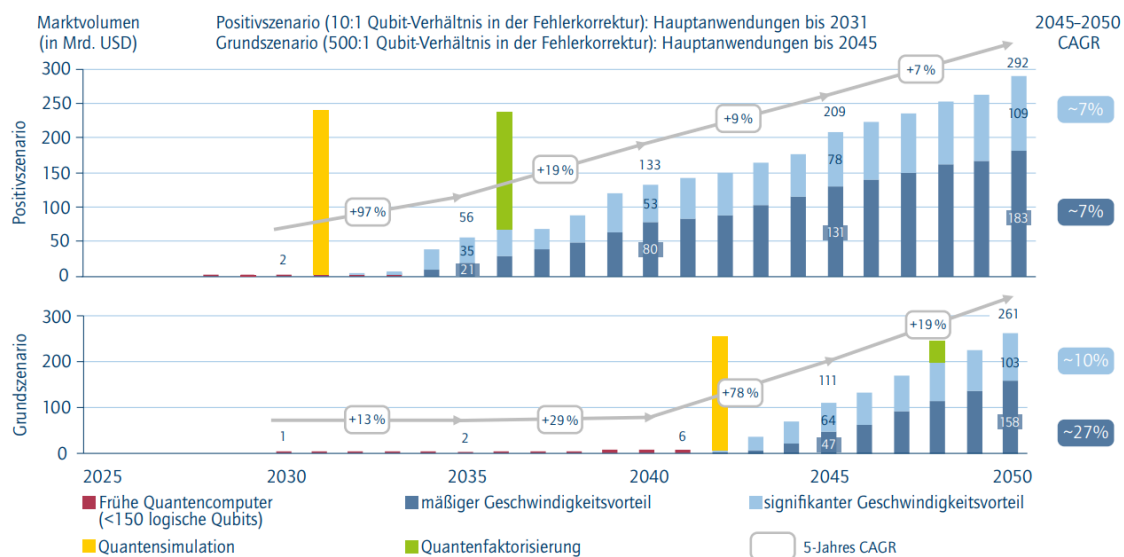


Abbildung 1: Projizierte Entwicklung des Marktpotenzials für Quantencomputer (2025–2050) mit Grund- und Positivszenarios, Actech.de: Innovationspotenziale der Quantentechnologien der zweiten Generation, 09.06.2021

<sup>10</sup> Lässig: Angewandtes Quantencomputing, 09.06.2021.

<sup>11</sup> China vermeldet Rekord-Leistung, in: dw.com, 09.06.2021

<sup>12</sup> Hossenfelder, Rechnen mit Qubits, S.124-126

Abbildung 1 zeigt eine Prognose für die Entwicklung des Marktszenarios von Quantencomputern. Der Unterschied zwischen Grundszenario und Positivszenario ist gewaltig und zeigt, wie schwer es ist, genaue Prognosen zu erstellen.

Durch den Sprung in die Quantenmechanik kann nicht mehr die gleiche Logik bei Quantencomputern genutzt werden wie bei den herkömmlichen Computern, die mit der Logik der klassischen Physik funktionieren. Bei klassischen Computern konnte man Strategien von unterschiedlichen rationalen Entscheidungsträgern in sozialen Konfliktsituationen durch die *Spieltheorie* untersuchen und definieren. Doch durch die neue Logik und die neuen Spielzüge, welche die Spieler anwenden können, verändert sich die Situation in den Spielen, wie aber auch die Theorie dahinter drastisch. So wird bei Spielen wie zum Beispiel dem Münzwurf (Coin Flip) durch die Anwendung der Logik der Quantenmechanik die herkömmliche Spieltheorie verändert und eine abgewandelte Quanten Spieltheorie definiert.

Die Quanten Spieltheorie ist eine Erweiterung der klassischen Spieltheorie, bei der die Gesetze der Quantenphysik angewendet werden. Folgende Prinzipien der Quantenmechanik sind auch in den Quanten Spieltheorie von überragender Bedeutung:<sup>13</sup>

*Überlagerung (Superposition)*: Die Fähigkeit von Quantenobjekten, zwei Zustände gleichzeitig anzunehmen.

*Verschränkung (Entanglement)*: Das Phänomen, dass entfernte Komponenten eines Quantensystems Korrelationen aufweisen, die nicht erklärt werden können. Dabei wird durch das Beeinflussen einer Komponente der Zustand der anderen Komponente auch beeinflusst.

*Kollabieren (Collapse)*: Das Phänomen, dass die Quantenzustände eines Systems auf klassische Zustände reduziert werden. Zusammenbrüche treten auf, wenn wir den Quantenkreislauf messen und dadurch das System reduzieren.

Die Quanten Spieltheorie ermöglicht es, dass sich bei Zwei-Personen-Nullsummenspielen wie dem Coin Flip ein Spieler durch eine Quanten-Strategie einen dominanten Vorteil verschaffen kann und durch Verwenden der Strategie nie verliert.

Das klassische Tic-Tac-Toe ist ein altes, einfaches Spiel, dessen Geschichte sich bis ins 12. Jahrhundert v. Chr. zurückverfolgen lässt. Ohne Rotationen und Spiegelungen gibt es lediglich 5478 verschiedene Spielsituationen, was im Vergleich beispielsweise zum Schachspiel eine sehr kleine Komplexität ist. Man kann leicht zeigen, dass beide Spieler ein Unentschieden erzwingen können- was das Spiel eigentlich "langweilig" macht.<sup>14</sup> Durch die Einbeziehung der Quanten Spieltheorie in das traditionelle Spiel Tic-Tac-Toe können sich die Gewinnchancen für einzelne Spieler bei gewissen Quanten Strategien vergrößern. Somit wird aus dem langweiligen Spiel durch die Einführung von Quanten Spieltheorie ein strategisch interessantes Spiel.

Das Ziel der vorliegenden Arbeit ist es, die Unterschiede der Logik der Quantenmechanik und der klassischen Physik zu zeigen und diese anhand von Beispielen zu verdeutlichen. Zudem werden die Grundlagen der verschiedenen Arten von Quantencomputern offengelegt und dem Leser gezeigt, wie diese funktionieren und wie es möglich ist, auf einer bestimmten Art von Quantencomputern zu programmieren. Zur Veranschaulichung der Unterschiede der Logik in der Quantenmechanik und der klassischen Physik werden Spiele anhand beider dieser Logiken vorgeführt und programmiert.

Kapitel 2 beschreibt die relevanten Grundlagen für Quantenphysik und Quantencomputer. Es erklärt die physikalischen und mathematischen Grundlagen sowie die Funktionsweise eines Qubits. Gleichzeitig beschreibt es die heute verfügbaren Quantencomputer und Quanten-Algorithmen, jeweils

---

<sup>13</sup> Sagole; Anurit; Bikash; Prasanta: Quantum Tic-Tac-Toe, 06.11.2021.

<sup>14</sup> Tic-Tac-Toe, in: Wikipedia, 06.11.2021

mit den Unterschieden zu den klassischen Ansätzen. Aufbauend auf diesen Grundlagen werden in Kapitel 3 die verschiedenen Quantengatter, im Vergleich zu den klassischen Logikgattern, eingeführt und deren Kombination zu Quantenschaltungen erklärt. Kapitel 4 rundet die Grundlagen ab, indem die Einführung in die Spieltheorie und Quanten Spieltheorie gegeben wird. Aufbauend auf diesen Grundlagen und dem damit erworbenen Verständnis über Quantenmechanik wird die praktische Anwendung der Quanten Spieltheorie an den Beispielen Coin Flip und Quanten Tic-Tac-Toe in Kapitel 5 entwickelt und analysiert. Dies stellt den Hauptteil der Arbeit dar. Die Zusammenfassung in Kapitel 6 zieht ein Fazit aus den gewonnenen Erkenntnissen und fasst die wichtigsten Ergebnisse zusammen.



## 2 Was ist ein Quantencomputer?

Wie eingangs beschrieben kann ein Quantencomputer Probleme lösen, die ein klassischer Computer nur mit viel Zeit oder Speicherplatz lösen könnte. Quantencomputer sind aber nicht generell besser als klassische Computer. Es gibt zum einen Probleme, die weder herkömmliche Computer noch Quantencomputer praktisch lösen können. Zum anderen können klassische Computer bereits viele Anwendungen umsetzen, für die man keinen Quantencomputer braucht: beispielsweise benötigen wir keine Quanten-Tabellenkalkulation.

Es gibt aber viele mögliche Anwendungsfelder und die Erwartungen sind gross: So könnten zeitkritische Optimierungsprobleme gelöst werden, beispielsweise Probleme der künstlichen Intelligenz oder Verkehrsleitsysteme.

Der Unterschied zwischen Quantencomputer und klassischem Computer entsteht einerseits dadurch, dass sie unterschiedlich aufgebaut sind. Andererseits sind auch die Bestandteile, mit denen der jeweilige Computer rechnet, unterschiedlich. Ein klassischer Computer rechnet mit Bits, einem logischen, binären Zustand, also einem Zustand, welcher nur zwei mögliche Werte annehmen kann. Diese Werte können mit wahr/falsch, ein/aus bezeichnet werden. Jedoch werden sie am häufigsten als "1" oder "0" dargestellt. Hingegen rechnet ein Quantencomputer mit sogenannten Qubits ("Quantenbits"). Ein Qubit nimmt nicht wie ein klassisches Bit einen Zustand von entweder 1 oder 0 an, sondern eine Mischung aus diesen Zuständen. Diese Mischung der Zustände wird auch „Überlagerung“ oder „Superposition“ genannt.

Der Unterschied wird noch deutlicher, wenn man mehrere Bits betrachtet: ein klassischer Computer kann mit  $n$  Bits  $2^n$  verschiedene Zahlen darstellen, zu jedem Zeitpunkt aber nur eine davon speichern. Ein Quantencomputer kann mit  $n$  Qubits  $2^n$  Zahlen gleichzeitig darstellen und in einer Recheneinheit ("Quantengatter") auf allen diesen Eingaben gleichzeitig rechnen. Dies bezeichnet man als Quantenparallelismus. Es ist ein völlig anderes Konzept als bei den klassischen Parallelrechnern, bei denen ein Computer auf mehreren Prozessoren unabhängige Berechnungen gleichzeitig ausführt und die Ergebnisse anschliessend in irgendeiner Form verknüpft.

Die parallele Rechenleistung eines Quantencomputers funktioniert aber nur, solange der Quantencomputer perfekt von seiner Umgebung getrennt ist und die empfindlichen Qubits komplett isoliert sind. Um das Ergebnis eines Quantenprogramms aus den Qubits auslesen zu können, müssen Messungen durchgeführt werden. Dadurch verlieren die Qubits ihre Superposition und man erhält eines der Rechenergebnisse, das darüber hinaus auch noch zufällig gewählt wird. Man weiss also nicht, zu welcher Eingabekombination das gemessene Ergebnis gehört. Dies muss man beim Entwurf von Algorithmen entsprechend berücksichtigen. Eine weitere wichtige Eigenschaft von Qubits ist die Verschränkung oder das Entanglement. Verschränkte Qubits können sich gegenseitig beeinflussen, auch wenn sie sich an verschiedenen Orten befinden. Manipulationen eines Qubits können den Ausgang der Messung eines anderen Bits beeinflussen. Darauf basiert die Quantenteleportation, die Übertragung eines Quantenzustands mit Hilfe von nichtklassischen Einstein-Podolsky-Rosen-(EPR-) Korrelationen. Die Quantenteleportation eignet sich unter anderem für das Übertragen von Qubits in der Quanteninformatik.<sup>15</sup>

### 2.1 Die physikalischen Grundlagen

Die Atom- und Quantenphysik ist im Wesentlichen im ersten Drittel des letzten Jahrhunderts entstanden.<sup>16</sup> Sie betrifft physikalische Objekte und Größen der Mikrophysik, also die kleinsten Teilchen und ihre Merkmale. Quantenphysikalische Teilchen sind so klein (bzw. quantenphysikalische Prozesse spielen sich auf so kleinen Dimensionen ab), dass sie von uns Menschen nicht sinnlich

---

<sup>15</sup> Quantenteleportation, in: Lexikon der Physik, 06.11.2021.

<sup>16</sup> Haken, Wolf: Atom- und Quantenphysik, S.1.

erfahrbar sind.<sup>17</sup> Quantensysteme kann man als solche nicht sehen, anfassen oder schmecken. Die Quantenphysik widerspricht insbesondere den klassischen Vorstellungen der Physik, dass die Natur stets kontinuierlich aufgebaut und immer messbar ist.

Folgende Prinzipien sind hilfreich, um über Quantenphysik nachzudenken:

"Prinzip 1: Die Quantenphysik ist eine statistische Theorie.

Die Quantenphysik macht Aussagen über die Statistik, die sich ergibt, wenn man ein Experiment oft durchführt. Über den Ausgang von Einzelmessungen gibt sie im Allgemeinen keine Auskunft.

Prinzip 2: Es gibt Unbestimmtheit in der Quantenphysik.

Es ist in der Quantenphysik nicht möglich, ein System so zu präparieren, dass für jede Messgröße bei jeder Instanz der Messung immer derselbe Messwert auftritt.

Prinzip 3: In der Quantenphysik ist keine Messung ohne Störung möglich.

Es ist in der Quantenphysik nicht möglich, durch Messung Informationen über ein System zu sammeln, ohne seinen Zustand zu verändern.

Prinzip 4: In der Quantenphysik tritt Verschränkung auf.

Es gibt in der Quantenphysik bestimmte Zustände von Systemen, die so in der klassischen Physik nicht vorkommen und somit mit unseren Alltagserfahrungen nicht erfassbar sind."<sup>18</sup>

Wie wir sehen werden, sind alle diese Prinzipien von Relevanz für das Verstehen der Funktionsweise von Quantencomputern. Daher werden die Prinzipien nun vertieft eingeführt.

### 2.1.1 Statistische Theorie

In der klassischen Physik sind Experimente fehlerbehaftet. Dies sind aber Fehler, die auf Grund nicht idealer Messgeräte oder durch nicht ideale Rahmenbedingungen entstehen. Theoretisch ist in der klassischen Physik der Ausgang eines Experiments deterministisch, d.h. bei gleichen Ausgangsbedingungen ("gleicher Präparation") sollte die Messung immer dasselbe Ergebnis liefern. In Wahrheit ist das Ergebnis eines Experiments aber eine Wahrscheinlichkeitsverteilung, siehe *Abbildung 2*.

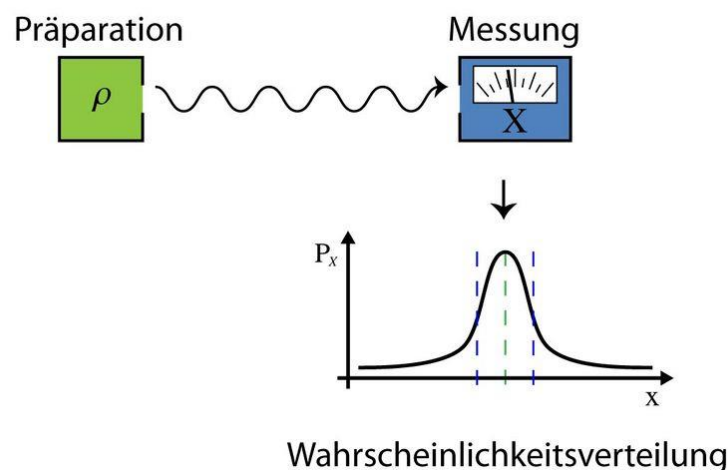


Abbildung 2: Schema eines Experiments, Uni Hannover, Prinzipien der Quantenphysik, 21.11.2021

<sup>17</sup> Prinzipien der Quantenphysik, in: Uni Hannover, 09.06.2021.

<sup>18</sup> Prinzipien der Quantenphysik, in: Uni Hannover, 09.06.2021.

In der Quantenphysik ist dies fundamental anders. Man spricht hier vom "objektiven Zufall". Dies bedeutet, dass das Ergebnis eines Experiments auch dann nicht exakt vorhergesagt werden kann, wenn man alle Gegebenheiten kennt. Es können aus identischen Ausgangssituationen verschiedene Konsequenzen folgen.<sup>19</sup> Dieser objektive Zufall ist wahrscheinlich eine der tiefendsten Entdeckungen der Naturwissenschaften des letzten Jahrhunderts.

### 2.1.2 Unbestimmtheit

Man kann Unbestimmtheit oder auch Unschärfe physikalisch klar definieren: Bringt man viele Quantenobjekte in einen Zustand, so ist die Grösse  $G$  in diesem Zustand bestimmt, wenn man für  $G$  stets dasselbe Messergebnis erhält. Der Zustand bezüglich  $G$  ist umso unbestimmter, je größer die Streuung  $\Delta G$  der Messergebnisse ist.<sup>20</sup>

1927 formulierte Heisenberg die Unschärferelation oder auch Unbestimmtheitsrelation. Sie besagt, dass zwei Messgrößen eines Teilchens (zum Beispiel sein Ort und Impuls) nicht gleichzeitig beliebig genau bestimmt sind.<sup>21</sup> Dies gilt prinzipiell und folgt nicht wegen Unzulänglichkeiten eines entsprechenden Messvorgangs. Der Begriff der Unschärfe meint dabei die Standardabweichung. Es geht dabei aber nicht darum, wie genau eine Messung ist, sondern es ist prinzipiell unmöglich, dies zu messen. Daher ist der Ausdruck „Unbestimmtheitsrelation“ eigentlich besser.

### 2.1.3 Messung

Beim Beobachten eines Teilchens beginnt das sogenannte Messproblem, bei dem man sich bis heute noch nicht einig ist, wie man es theoretisch beschreiben soll.<sup>22</sup> Der Zustand eines Teilchens wird in der Quantenphysik durch die Messung beeinflusst. Eine grundlegende Eigenschaft eines Teilchens in der Quantenphysik ist nämlich die Existenz von Überlagerungszuständen, in denen es sich in zwei Zuständen gleichzeitig befindet. Dies konnte experimentell bewiesen werden, in dem man Teilchen wie zum Beispiel ein Elektron auf einen Doppelspalt schießt und ein Interferenzmuster beobachten kann. Dies ist nur dadurch erklärbar, dass das Teilchen beide Spalten gleichzeitig passiert,<sup>23</sup> In dem man den Zustand des Teilchens misst und auf einen Wert festlegt, verliert es seinen Superpositionszustand. Möchte man im obigen Beispiel beobachten, wie das Elektron die beiden Spalte durchläuft, wird durch das Messen der Weg des Elektrons festgelegt: das Ergebnis des Experiments ist anders.

Ein interessantes Gedankenexperiment, das die Zerstörung des Superpositionszustands durch Messung veranschaulicht, wurde bereits 1935 durch Erwin Schrödinger beschrieben: es heisst "Schrödingers Katze".

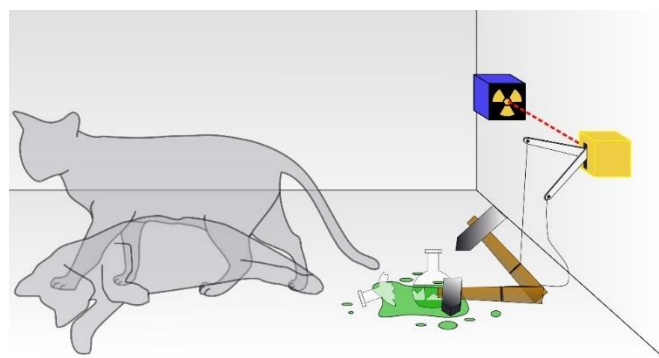


Abbildung 3: Katze in überlagertem Zustand, Wikipedia: Schrödingers Katze, 16.06.2021

<sup>19</sup> Was ist Zufall, in: physikalische Soiree, 09.06.2021.

<sup>20</sup> Heisenbergsche Unbestimmtheitsrelation, in: Schülerlexikon, 09.06.2021.

<sup>21</sup> Heisenbergsche Unschärferelation, in: chemie.de, 09.06.2021.

<sup>22</sup> Kreye, Es können nicht beide recht haben, 16.06.2021.

<sup>23</sup> Romero, Pflanzler, Quantenmechanik am Limit, 16.09.2021.

In einer verschlossenen Kiste befindet sich eine Katze. Vom Zerfall eines Atoms abhängig wird ein Mechanismus ausgelöst, so dass ein Hammer ein Gefäß mit einer giftigen Substanz zerstört. Die Katze stirbt daraufhin. Gemäss unserer Alltagserfahrung ist die Katze entweder tot oder lebendig. In der Quantenwelt ist es aber möglich, dass die Katze sowohl tot und lebendig ist: sie ist bezüglich dieses Zustands unbestimmt. Solange die Kiste unverschlossen ist, bleibt ihr Zustand uneindeutig. Der Zustand wird erst dann eindeutig, wenn wir ihn messen, in dem wir die Kiste öffnen, wir den Gesundheitszustand der Katze ablesen und das System somit mit der Umwelt interagiert. Bei einer quantenmechanischen Superposition führt dies dazu, dass das Teilchen einen eindeutigen Zustand annimmt, und zwar zufällig einen der in der Superposition enthaltenen Zustände.<sup>24</sup>

#### 2.1.4 Verschränkung

Eine der merkwürdigsten Phänomene der Quantenphysik ist, dass zwei räumlich getrennte Teilchen Informationen über ihre Eigenschaften austauschen können.

Der Spin eines Elektrons ist eine quantenmechanische Eigenschaft des Elektrons. Man nennt ihn auch Eigendrehimpuls, da er im Gegensatz zum klassischen mechanischen Drehimpuls nicht durch die Drehbewegung einer Masse hervorgerufen wird. Der Spin des Elektrons kann zwei Werte annehmen, die normalerweise als  $+1/2$  und  $-1/2$  bezeichnet werden.<sup>25</sup>

Vor der Messung ist der Spin eines Elektrons nicht festgelegt (Stern-Gerlach-Versuch). Erst durch Messung nimmt er einen der beiden Werte  $+1/2$  oder  $-1/2$  an-die Superposition wird aufgelöst. Die Verschränkung kann beobachtet werden, wenn man zwei Elektronen gleichzeitig erzeugt und ihren Spin in zwei verschiedenen Apparaturen misst. Es zeigt sich, dass die Elektronen stets einen gegenläufigen Spin besitzen. Vor der Messung ist der Spin eines Elektrons jeweils mit einer Wahrscheinlichkeit von 50% im Zustand  $+1/2$  oder  $-1/2$ . Die Wahrscheinlichkeit, dass zwei Elektronen einen gegenläufigen Spin haben, ist daher nur 50%. Dass sie aber nach der Messung stets einen gegenläufigen Spin haben, ist nur dadurch erklärbar, dass trotz räumlicher Entfernung zueinander die Elektronen als System zu verstehen sind, dass sich vor der Messung in Superposition befindet. Dies bezeichnet man als Verschränkung.<sup>26</sup> "Einstein, Podolsky und Rosen formulierten 1935 den EPR-Effekt, nach dem die Quantenverschränkung zur Verletzung des klassischen Prinzips des lokalen Realismus führen würde, was von Einstein in einem berühmten Zitat als „spukhafte Fernwirkung“ („spooky action at a distance“) bezeichnet wurde."<sup>27</sup>

#### 2.1.5 Weitere physikalische Grundlagen

##### 2.1.5.1 Supraleitung

Unter Supraleitung versteht man die praktisch widerstandsfreie Leitung von Strom. Der elektrische Widerstand von Supraleitern strebt unterhalb der sogenannten Sprungtemperatur gegen Null. Die Supraleitung bezeichnet man als "makroskopischen Quantenzustand, also als Zustand eines Systems mit vielen Teilchen, der sich nicht klassisch, sondern nur quantenphysikalisch richtig beschreiben lässt".<sup>28</sup>

Viele Metalle sind Supraleiter, aber auch andere Materialien. Die Sprungtemperaturen sind enorm tief, so hat Magnesiumdiborid die höchste Sprungtemperatur mit nur 39 K (-234 °C).

---

<sup>24</sup> Schrödingers Katze, in: LEIFIphysik, 16.06.2021.

<sup>25</sup> Elektronenspin, in: Wikipedia, 16.06.2021.

<sup>26</sup> Quantenverschränkung, in: chip.de, 16.06.2021.

<sup>27</sup> Quantenverschränkung, in: Wikipedia, 16.06.2021.

<sup>28</sup> Supraleiter, in: Wikipedia, 16.06.2021.

### 2.1.5.2 Adiabatische Zustandsänderungen

"Eine adiabatische Zustandsänderung ist ein thermodynamischer Vorgang, bei dem ein System von einem Zustand in einen anderen überführt wird, ohne Wärmeenergie mit seiner Umgebung auszutauschen."<sup>29</sup>

### 2.1.5.3 Quasiteilchen

Quasiteilchen sind mathematische Konstrukte, die zur Beschreibung bestimmter Phänomene benutzt werden.<sup>30</sup> Quasiteilchen verhalten sich wie freie Teilchen, können aber nicht als freie Teilchen existieren, da sie Elementaranregungen von Vielteilchensystemen sind. Sie werden durch die Wechselwirkungen des sie erzeugenden Systems erzeugt.<sup>31</sup>

## 2.2 Die mathematischen Grundlagen

Für Berechnungen in der Quantenmechanik werden primär komplexe Zahlen, Vektorräume sowie Matrizenoperationen, besonders unitäre Matrizen, benötigt. Diese werden im Folgenden kurz eingeführt. Für weitere Details kann die entsprechende Fachliteratur herangezogen werden.

### 2.2.1 Komplexe Zahlen

Komplexe Zahlen  $z$  können in der Form  $z = a + bi$  geschrieben werden, wobei  $a$  den Realteil und  $b$  den Imaginärteil bezeichnet. Für die imaginäre Einheit  $i$  gilt:  $i^2 = -1$ .

Komplexe Zahlen kann man als Punkte in der komplexen Zahlenebene darstellen. Die Teilmenge der reellen Zahlen befindet sich dabei auf der  $x$ -Achse, die der rein imaginären befindet, sich auf der  $y$ -Achse.

$z^* := a - bi$  heisst die zu  $z = a + bi$  konjugiert komplexe Zahl.

Der Betrag einer komplexen Zahl ist als  $|z| = \sqrt{a^2 + b^2} = \sqrt{z * z^*}$  definiert.

Ausserdem kann man jeder komplexen Zahl in der Form  $z = r e^{i\Phi}$  darstellen. Dabei ist  $e^{i\Phi}$  ein Vektor in der komplexen Zahlenebene der Länge 1.  $\Phi$  gibt den Winkel zur  $x$ -Achse an und heisst Phase,  $r$  ist die Länge des Vektors.

### 2.2.2 Vektorräume

Ein Vektorraum  $V$  ist über einem Körper  $K$  definiert. Eine Menge  $V$  zusammen mit einer inneren Verknüpfung

$$+: V \times V \rightarrow V$$

und einer äusseren Verknüpfung

$$*: K \times V \rightarrow V$$

heisst ein Vektorraum über  $K$ , wenn 8 Vektorraumaxiome erfüllt sind.<sup>32</sup>

Wir arbeiten in dieser Arbeit mit dem Vektorraum  $C^n$ , der Menge aller  $n$ -Tupel von komplexen Zahlen  $\alpha_0, \alpha_1, \dots, \alpha_{n-1}$ .

Sei  $M$  eine Menge:  $M = \{|0\rangle, |1\rangle, \dots, |n-1\rangle\}$ .

Eine Summe der Form  $\alpha_0 * |0\rangle + \alpha_1 * |1\rangle + \dots + \alpha_{n-1} * |n-1\rangle$  heisst Linearkombination über  $C^n$ .<sup>33</sup>

---

<sup>29</sup> Adiabatische Zustandsänderung, in: chemie.de, 16.06.2021.

<sup>30</sup> Fufaev, Unterschied zwischen Teilchen und Quasiteilchen, 16.06.2021.

<sup>31</sup> Quasiteilchen, in: Lexikon der Physik, 16.06.2021.

<sup>32</sup> Akfeld, Vektorraumaxiome, 16.06.2021.

<sup>33</sup> Homeister, Quantencomputing, S. 292.

### 2.2.3 Matrizen

Von besonderer Bedeutung für die Operationen auf einem Qubits sind *unitäre* Matrizen. Damit kann man beschreiben, wie sich ein abgeschottetes Quantensystem ändert.

Gegeben sei eine quadratische Matrix  $A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$ .

Die komplex konjugiert Matrix  $A^*$  ist definiert als  $A^* = \begin{pmatrix} a_{11}^* & \cdots & a_{1n}^* \\ \vdots & \ddots & \vdots \\ a_{n1}^* & \cdots & a_{nn}^* \end{pmatrix}$ .

Die zu  $A$  adjungierte Matrix ist die komplex konjugierte und transponierte Matrix  $\text{adj } A = A^{*T}$ , d.h.  $A^{*T}$  entsteht aus  $A$ , indem  $a_{ij}$  durch  $a_{ji}^*$  ersetzt wird.

Eine quadratische Matrix  $A$  heisst *unitär*, falls ihre Inverse gleich der adjungierten Matrix ist:

$$\text{adj } A = A^{-1}$$

Eine unitäre Matrix beschreibt eine unitäre Transformation.

Im speziellen ist die Hadamard-Matrix  $H$  unitär, die wie folgt definiert ist:

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

Die Matrix  $H$  ist zu sich selbst invers. Dies kann man leicht zeigen, da  $HH = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ .

Das zweimalige Anwenden von  $H$  auf einen Vektor ergibt also den ursprünglichen Vektor.<sup>34</sup>

## 2.3 Das Qubit

Aber was ist nun ein Qubit genau? Ein Qubit, auch Quantenbit genannt, ist die kleinste Informationseinheit von Quantencomputern. Ein Qubit ist ein quantenmechanisches System. Ein Qubit nimmt mit einer gewissen Wahrscheinlichkeit den Zustand "0" und mit einer gewissen Wahrscheinlichkeit den Zustand "1" an. Dabei bezeichnet man "0" und "1" als Basiszustände und schreibt sie als  $|0\rangle$  und  $|1\rangle$ , gelesen wird dies als "Ket 0" und "Ket 1". Eine Operation auf einem Qubit in Superposition wird sowohl auf dem Zustand  $|0\rangle$  als auch auf dem Zustand  $|1\rangle$  ausgeführt.

Ein Qubit ist also immer mit einer gewissen Wahrscheinlichkeit  $|0\rangle$  und mit einer gewissen Wahrscheinlichkeit  $|1\rangle$ . Die Wahrscheinlichkeiten, mit dem ein Qubit  $|0\rangle$  bzw.  $|1\rangle$  ist, kann man vor der Messung des Qubits durch zwei Parameter charakterisieren: nämlich mit  $\alpha$  und  $\beta$ . Dabei sind  $\alpha$  und  $\beta$  genau genommen Wahrscheinlichkeitsamplituden, auch Amplituden genannt. Somit nimmt ein der interne Zustand eines Qubits vor der Messung:

$$\alpha \cdot |0\rangle + \beta \cdot |1\rangle$$

$$\alpha, \beta \in \mathbb{C}$$

an, wobei  $\alpha$  und  $\beta$  komplexe Zahlen sind. Eine andere Darstellung des oberen Terms basiert darauf, dass  $\alpha$  und  $\beta$  reelle Zahlen sind. Durch Hinzufügen eines Terms, der die relative Phase zwischen  $\alpha, \beta$  zeigt, ergibt sich folgender Ausdruck:

$$|\psi\rangle = \alpha|0\rangle + e^{i\varphi}\beta|0\rangle$$

---

<sup>34</sup> Homeister, Quantencomputing, S.25

$$\alpha, \beta, \varphi \in \mathbb{R}$$

Die komplexen Zahlen haben weiterhin die folgende Eigenschaft:

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

Dabei fällt auf, dass diese Gleichung wie die Umfangsgleichung aussieht:

$$\cos^2 \alpha + \sin^2 \alpha = 1$$

Aus diesem Grund kann man ein Qubit wie folgt schreiben:

$$|\psi\rangle = \cos \alpha |0\rangle + \sin \alpha |1\rangle$$

Es gilt für  $\alpha, \beta$ :

$$\alpha = \cos \frac{\theta}{2}, \quad \beta = \sin \frac{\theta}{2}$$

Aus dieser Schlussfolgerung ist es möglich, ein beliebiges einzelnes Qubit wie folgt zu beschreiben:

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle$$

wobei  $\theta$  und  $\varphi$  reelle Zahlen sind.<sup>35</sup>

Man kann aber den Zustand eines Qubits auch als Vektor beschreiben. Dabei wird der Zustand als Vektor in einem zweidimensionalen Vektorraum über zwei komplexen Zahlen betrachtet:  $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ . Die beiden Basiszustände,  $|0\rangle$  und  $|1\rangle$ , lassen sich ebenfalls als Vektoren schreiben:

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

Der Zustandsvektor eines Qubits kann also als Linearkombination der zweidimensionalen Standardbasisvektoren angegeben werden:

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle$$

Die Länge des Vektors ist dabei 1, welche durch die Bedingung  $|\alpha|^2 + |\beta|^2 = 1$  gegeben ist.

---

<sup>35</sup> Glendinning: The Bloch Sphere, 08.11.2021.

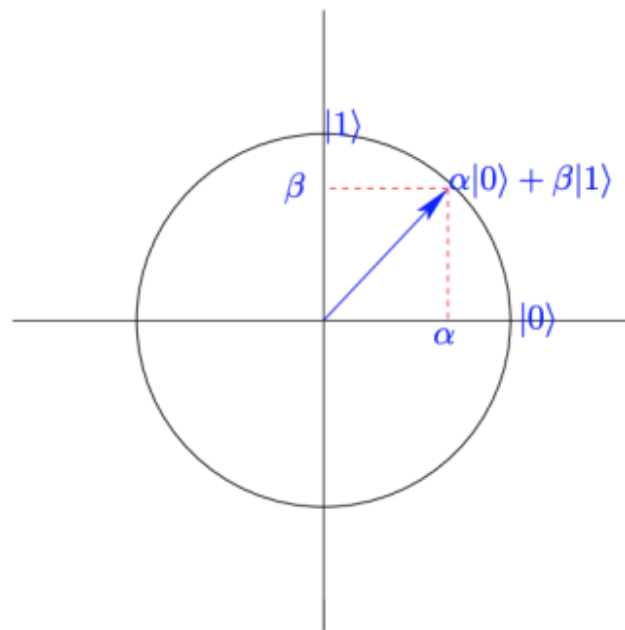


Abbildung 4: Zustand eines Qubits beschrieben mit Vektoren. Wordpress.com: Bloch Sphere, 12.05.2021

Jedoch können Qubits auch in einem dreidimensionalen Raum dargestellt werden. Dabei definieren die Zahlen  $\theta$  und  $\varphi$  einen Punkt auf einer dreidimensionalen Einheitskugel, der Bloch Sphäre. Hinzugefügt muss noch werden, dass die Zahlen  $\theta$  und  $\varphi$  folgende Eigenschaft haben:

$$0 \leq \theta \leq \pi, \quad 0 \leq \varphi \leq 2\pi$$

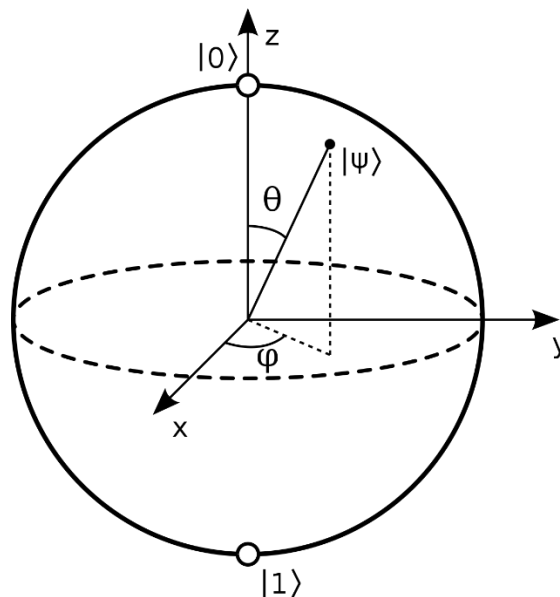


Abbildung 5: Zustand eines Qubits in einer Bloch Sphäre. Wikipedia.com: Bloch Sphere, 08.11.2021

Da man den genauen Zustand eines Qubits nicht wie bei einem klassischen Bit einfach lesen und feststellen kann, muss man ihn messen. Dabei ist das Messergebnis abhängig von den Amplituden  $\alpha$  und  $\beta$ . Durch das Messen des Qubits wird dessen Superposition zerstört und das Qubit nimmt mit einer gewissen Wahrscheinlichkeit  $|\alpha|^2$  den Zustand  $|0\rangle$  an und mit einer gewissen Wahrscheinlichkeit  $|\beta|^2$  den Zustand  $|1\rangle$ . Die Wahrscheinlichkeiten, mit dem sich das Qubit in einem der Zustände befindet,



kann man durch mehrfaches Durchführen und Messen bestimmen. Durch das Messen eines Qubits wird dessen Zustand zerstört. Eine Messung ist somit nur einmal möglich. Für eine neue Messung muss das Qubit wieder durch Operationen in den gleichen Zustand gebracht werden, wobei dabei ein anderes Messergebnis resultieren kann (siehe später).<sup>36</sup>

Ein Beispiel für einen zulässige Superposition eines Zustands eines Qubits ist  $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ . Bei dieser Superposition gilt:

$$\left(\frac{1}{\sqrt{2}}\right)^2 + \left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2} + \frac{1}{2} = 1$$

Würden wir ein Bit in diesem Zustand messen, erhalten wir mit 50% Wahrscheinlichkeit das Ergebnis  $|0\rangle$  und mit 50% Wahrscheinlichkeit  $|1\rangle$ .

Ein Qubit, welches sich in Superposition befindet, kann Operationen gleichzeitig auf dem Zustand  $|0\rangle$  wie auch auf dem Zustand  $|1\rangle$  durchführen. Es können also dadurch alle möglichen Inputs auf einmal berechnet werden, jedoch ist diese Berechnung immer mit einer gewissen Wahrscheinlichkeit falsch. Dies ist der Fall, da Qubits in einzelnen Fällen mit ihrer Umwelt interagieren. So kann durch eine Interaktion mit der Umwelt die Information, die das Qubit gespeichert hat, verloren gehen oder manipuliert werden. Aus diesem Grund werden die Berechnungen auf Quantencomputern nicht nur einmal, wie auf einem klassischen Computer durchgeführt, sondern hundert bis tausend Mal. Aus diesen vielen Berechnungen wird dann das korrekte Ergebnis statistisch berechnet. Quantencomputer bezeichnet man daher als probabilistisch, da man mit wahrscheinlichkeitsbehafteten Zuständen rechnet. Klassische Computer rechnen «deterministisch», also vorhersehbar.<sup>37</sup>

Ein weiterer Unterschied zwischen Qubits und klassischen Bits ist, dass sich Qubits nur für eine bestimmte Dauer, auch Kohärenzzeit genannt, in Superposition befinden können. Wenn man den Zustand eines Qubits ermittelt, indem man misst, verliert das Qubit seine Quanteneigenschaft und es nimmt einen der beiden definierten Zustände (1 oder 0) an.

Daher kann man ein Qubit im Gegensatz zum klassischen Bit nur einmal auslesen, und das Messergebnis kann somit in einem herkömmlichen Bit gespeichert werden.

Wenn das Qubit seine Superposition verliert, wird dies Dekohärenz genannt. Die Bits eines klassischen Computers haben stets einen eindeutig bestimmbar Zustand, während Qubits erst durch eine Messung einen bestimmten Zustand einnehmen. Diese Eigenschaft der Qubits lässt einen Quantencomputer Ergebnisse simultan berechnen.

Eine weitere wichtige Eigenschaft ist das sogenannte No-Cloning-Theorem. Das No-Cloning-Theorem ist eine Folge der Quantenmechanik, die besagt, dass es unmöglich ist, identische Duplikate eines unbekannten Quantenzustands zu erzeugen. Dies verhindert es, klassische Methoden zur Fehlerkorrektur anzuwenden. Das Theorem beruht auf der Annahme, dass alle Quantenoperationen unitäre lineare Zustandstransformationen sein müssen. Eine weitere Erklärung zu dem Theorem, der Beweis wie aber auch die Auswirkung des Theorems findet sich unter <sup>38</sup>.

Die Bra-Ket Notation, auch Dirac-Notation genannt, ist eine Notation, in der man quantenmechanische Zustände schreibt. Dabei beschreibt der Bra-Ausdruck  $\langle \dots |$  einen Zeilenvektor und der Ket-Ausdruck  $|\dots\rangle$  einen Spaltenvektor. Bei Qubits stehen die beiden Kets  $|0\rangle$  und  $|1\rangle$  für die Spaltenvektoren  $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$  und

---

<sup>36</sup> Homeister, Quantencomputing, S. 20.

<sup>37</sup> Homeister, Quantencomputing, S. 22.

<sup>38</sup> The no-cloning theorem, in: Quantiki, 08.11.2021.

(<sup>0</sup><sub>1</sub>). Die dazugehörigen Bras  $\langle 0|$  und  $\langle 1|$  stehen für die Zeilenvektoren  $(1 \ 0)$  und  $(0 \ 1)$ . Eine ausführlichere Erklärung zu der Bra-Ket Notation findet sich unter <sup>39</sup>.

### 2.3.1 Realisierung von Qubits

Die grösste technische Herausforderung bei der Realisierung von Qubits ist es, die Zustände der Qubits und die Verschränkungen zwischen ihnen lange genug aufrechtzuerhalten zu können, um die für einen Algorithmus erforderlichen Rechenschritte auszuführen. Wie bereits diskutiert, genügen schon kleine Störungen, und das Qubit fällt in einen Basiszustand zurück und die quantenmechanische Berechnung schlägt fehl. Zusätzlich ist es wünschenswert, dass ein Quantencomputer das Konzept der Fehlertoleranz umsetzt: "Ein fehlertoleranter Quantencomputer muss daher nicht nur aufwändigst abgeschirmt werden, er braucht zusätzlich auch mehrere Kopien seiner Eingangsdaten und Zwischenstände sowie eine funktionierende Fehlerkorrektur." <sup>40</sup> Experten gehen daher davon aus, dass ein zuverlässiger Quantencomputer bis zu einer Million Qubits benötigt.

Heutige Quantencomputern verwenden grundsätzlich drei verschiedene Arten von Qubits: das supraleitende Qubit, Qubits basierend auf dem Prinzip der Ionenfalle und photonische Qubits. Darüber hinaus existieren drei Ansätze, die vielversprechend sind, aber noch nicht in Quantencomputern eingesetzt werden. Dabei unterscheiden sich die Qubits sowohl in der Art der Herstellung wie aber auch in der Genauigkeit und Schnelligkeit bei der Ausführung von bestimmten Rechnungen.

#### 2.3.1.1 Supraleitendes Qubit

Die heute am häufigsten verwendete Realisierung eines Qubits basiert auf supraleitenden Schleifen. Ein Qubit wird hier durch widerstandslos fließende Ströme in supraleitenden Schleifen realisiert, beispielsweise aus Niob oder Aluminium. Die beiden Zustände können entweder durch Richtung des Stromflusses (sogenannte Flussqubits) oder durch die Verteilung der Ladung (sogenannte Transmonqubits) dargestellt werden. Eine der größten Herausforderung beim Bau und Betrieb von supraleitenden Qubits ist die Aufrechterhaltung von Kohärenz zur Fehlervermeidung.

Flussqubits basieren auf einem Kreisstrom, durch den ein magnetischer Fluss durch die Ebene der Schleife induziert wird (daher der Name Flussqubit). Teilweise sind Flussqubits technisch ausgereift genug, damit lange Kohärenzzeiten und einfache Kopplung zwischen den Qubits umgesetzt werden können. Die Herausforderung besteht darin, konsistente Flussqubits mit vorhersagbaren Eigenschaften zu produzieren. Die Quanteninformation von Transmonqubits wird in Ladungssoszillationen in den supraleitenden Schaltkreisen getragen. Sie haben eine lange Kohärenzzeit und können einfach mit Hilfe von Mikrowellenresonatoren gekoppelt werden.<sup>41</sup>

Supraleitende Qubits können mit bestehenden Fertigungsanlagen für Chips hergestellt werden, was einen grossen Vorteil darstellt. Allerdings bedeutet die extreme Kühlung, die benötigt wird, um die Supraleitung zu ermöglichen, eine grosse Herausforderung, vor allem, wenn man in den oben angesprochenen Grössenordnungen von mehreren Hunderttausenden von Qubits denkt.

#### 2.3.1.2 Qubit basierend auf dem Prinzip der Ionenfalle

Ionenfallen bestehen aus Einzelatomen mit einem Elektronendefizit, die Atome sind also ionisiert, Sie werden durch elektromagnetische Felder fixiert. Die Qubits werden durch einzelne Elektronen in der Schale der Atome realisiert und können mit abgestimmten Laserpulsen manipuliert und verschränkt werden.

Mikrochips aus Ionenfallen-Qubits und solche aus supraleitenden Qubits haben die gleiche Grössenordnung. Auch Ionenfallen benötigen tiefe Temperaturen, dies kann durch eine

---

<sup>39</sup> Bra-Ket Notation, in: Math is fun, 05.11.2021.

<sup>40</sup> Hossenfelder, Rechnen mit Qubits, S.124.

<sup>41</sup> Stollenwerk, Umfeldstudie Quantencomputing, 05.11.2021.

Rundumbestrahlung durch Laserlicht erreicht werden, wodurch ein Ionenfallen-Quantencomputer sogar bei Zimmertemperatur betrieben werden kann.<sup>42</sup>

Der grosse Vorteil der Ionenfallentechnik ist die lange Dekohärenzzeit, die bis zu mehreren Minuten dauern kann. Ausserdem können die Ionenfallen-Qubits relativ leicht mit vielen verschränkt werden. Nachteilig wirkt sich aus, dass Ionenfallen-Qubits langsamer als supraleitende Qubits reagieren. Ausserdem ist es schwieriger, viele Ionenfallen auf einem Chip zu realisieren.<sup>43</sup>

#### *2.3.1.3 Photonisches Qubit*

Bei photonischen Plattformen werden die Qubits durch Photonen, die Quantenteilchen des Lichts, realisiert. Als Quantenzustand kann verwendet werden, dass ein Photon in einem von zwei bestimmten Zeitfenstern vorhanden ist, oder er bezieht sich auf bestimmte variable Eigenschaften eines Photons.

Photonische Qubits können bei Raumtemperatur genutzt werden, was natürlich ein grosser Vorteil ist. Ihre Quantenzustände werden sehr viel länger aufrechterhalten als bei supraleitenden Qubits: Dekohärenzzeiten liegen normalerweise bei einigen Millisekunden, können aber bei idealen Bedingungen sogar mehrere Stunden dauern. Damit ist photonisches Quantencomputing wesentlich billiger und einfacher zu handhaben. Allerdings müssen Systeme, die auf photonischen Qubits basieren, je nach der Anzahl der benötigten Qubits und deren Verschränkungen eigens aus Lasern und optischen Einzelkomponenten aufgebaut werden. Dies benötigt sehr viel Platz. Ein Quantencomputer chinesischer Forscher belegte eine komplette Tischplatte.<sup>44</sup>

#### *2.3.1.4 Topologische Qubits*

Topologische Qubits speichern Informationen in den Eigenschaften von Quasiteilchen. Die Besonderheit besteht darin, dass ihre Eigenschaften physikalisch besonders geschützt sind. Damit sind topologische Qubits gut geschützt gegen Dekohärenz.

Zurzeit gibt es keine lauffähigen Computer mit dieser Technik. Microsoft-Forscher arbeiten an den Grundlagen eines topologischen Quantencomputers. Sie erachten besonders die geringe Fehlerrate der Qubits und ihre Widerstandsfähigkeit gegenüber Umgebungsrauschen überzeugend. Ausserdem hoffen sie, dass die benötigten Strukturen schnell skalieren, d.h. dass man sie rasch auf sehr viele Qubits ausbauen kann.<sup>45</sup>

#### *2.3.1.5 Qubits aus Nanostrukturen in Halbleitern*

Die klassische Halbleitertechnologie ist ein sehr gut erforschter Fachbereich. Qubits aus Nanostrukturen in Halbleitern können wie supraleitende Qubits auf Mikrochips realisiert werden. Das Qubit wird durch die Ladung eines Elektrons oder aus dessen Elektronenspin realisiert. Auch hier ist wieder ein grosser Vorteil, dass im Gegensatz zu supraleitenden Qubits keine so tiefen Temperaturen für die Realisierung benötigt werden.

#### *2.3.1.6 Qubits basierend auf Stickstoff-Fehlstellen in Kohlenstoff-Kristallgittern*

Hier werden Fehlstellen in Kohlenstoff-Kristallgittern, beispielsweise Diamanten, eingesetzt. Ihre Funktionsweise ist ähnlich wie die der Ionenfallen, die Fixierung erfolgt jedoch über die Gitterstruktur des Kohlenstoffs. Vorteil dieser Technologie ist wiederum, dass sich die Qubits bei Raumtemperatur manipulieren und auslesen lassen. Da es jedoch schwierig ist, die Qubits konsistent bereitzustellen, müssen diese über optische Netzwerke angesteuert werden.<sup>46</sup> Daher können sie zurzeit noch nicht in Quantencomputern eingesetzt werden.

---

<sup>42</sup> Marre: Quantencomputer, 08.11.2021.

<sup>43</sup> Hossenfelder, Rechnen mit Qubits, S.126.

<sup>44</sup> Stollenwerk, Umfeldstudie Quantencomputing, 05.11.2021.

<sup>45</sup> Hossenfelder, Rechnen mit Qubits, S.127.

<sup>46</sup> Stollenwerk, Umfeldstudie Quantencomputing, 05.11.2021.

### 2.3.2 Quantenregister

Ein Quantenregister ist ein System, welches aus mehreren Qubits besteht. Zudem ist ein Quantenregister ein Speicherbereich für Qubits, in dem mehrere Qubits zusammengefasst sind. Quantenregister können sich in einem bestimmten Zustand befinden, der als Reihenfolge der Zustände der einzelnen Qubits geschrieben werden kann. Quantencomputer manipulieren die Qubits innerhalb eines Quantenregisters, um Berechnungen durchzuführen. "Ein solches System kombinierter Qubits hat einen eigenen Zustandsraum, der sich aus den Zustandsräumen der beteiligten Qubits ergibt. Mathematisch wird das durch das sogenannte Tensorprodukt beschrieben, das auch festlegt, wie sich Basiszustände des Systems aus den Basiszuständen der einzelnen Qubits ergeben."<sup>47</sup>

Das Tensorprodukt ermöglicht es einen neuen Raum durch die Multiplikation von Dimensionen zu erstellen. Dabei gibt es eine neue Matrix, bei der die Anzahl Zahlen dem Produkt der Multiplikation der Anzahl Zahlen der Anfangsmatrizen entspricht. Somit ist die Bildung des Tensorprodukts  $x \otimes y$  vergleichbar mit der Bildung des kartesischen Produkts von zwei Mengen. Eine detaillierte Erklärung zum Tensorprodukt findet sich unter <sup>48</sup>.

## 2.4 Verschiedene Ansätze zur Implementierung von Quantencomputern

Momentan gibt es zwei verschiedene Arten von Ansätzen zur Implementierung von Quantencomputern: Die Quantengattercomputer und die Quanten-Annealing-Maschinen, eine bestimmte Art eines adiabatischen Quantencomputers, also ein Quantencomputer, bei dem ein Qubit von einem Zustand in einen anderen überführt wird, ohne dabei Wärme mit der Umgebung auszutauschen. Dabei haben diese zwei Arten von Quantencomputern verschiedene Ansätze, um die grössten Probleme bei Quantencomputern, die Fehleranfälligkeit und das Verlieren der Dekohärenz von Qubits, möglichst gering zu halten. Die Fehleranfälligkeit entsteht durch unzuverlässige Verbindungen zwischen den Qubits in einem Quantenchip. Die Dekohärenz wird dabei unterdrückt, in dem man die Wechselwirkungen zwischen einem Qubit und seiner Umgebung nicht zulässt, wodurch das Qubit in der Superposition bleibt. Da diese zwei Arten von Quantencomputer durch unterschiedliche Ansätze implementiert wurden, haben sie somit auch unterschiedliche Stärken und Schwächen und werden deshalb in verschiedenen Anwendungsbereichen jeweils eingesetzt.

### 2.4.1 Quantengattercomputer

Ein Quantengattercomputer, auch Universal-Gate-Quantencomputer genannt, verwendet elementare Logik-Bausteine, sogenannte «Quantengatter» (auch Quanten-Logik-Gatter genannt, engl. «quantum gates» oder «quantum logic gates»), in jedem Rechenschritt. Diese Logik-Gatter sind vergleichbar mit den elektronischen Gattern in klassischen Computern, welche elementare Operationen durchführen. Jedoch sind die Quantengatter quantenmechanische Systeme. Man kann somit komplexe Algorithmen mit Quantengattern bauen, bei welchen man die Möglichkeit hat, diese am Schluss zu messen. Durch das Messen der Quantengatter verliert das Qubit seine Superposition und kollabiert auf einen klassischen Zustand  $|0\rangle$  oder  $|1\rangle$ .

Beim Bilden von Quantenschaltungen werden mehrere Quantenlogikgatter benutzt, die grundlegende Rechenoperationen verarbeiten können. Durch die Eigenschaft von Qubits, dass das Qubit in Superposition Operationen gleichzeitig auf dem Zustand  $|0\rangle$  wie auch auf dem Zustand  $|1\rangle$  durchführen kann, sollen Quantengattercomputer komplexe Algorithmen ausführen können. Zu diesen Algorithmen gehören Shors Algorithmus, um die RSA-Kryptographie zu knacken, und Grovers Algorithmus, zur schnellen Suche in einer unsortierten Datenbank. Zur Berechnung dieser komplexen Algorithmen werden viele verschiedene Quantengatter benutzt. Die verschiedenen Arten der Quantengatter und deren Funktionsweise wird in Abschnitt 6.2 noch genauer erklärt.<sup>49</sup>

---

<sup>47</sup> Neukart: Qubits (2020), S. 145.

<sup>48</sup> The Tensor Product, Demystified, in: math3ma, 08.11.2021.

<sup>49</sup> Marchenkova: Universal gate quantum computers, 19.05.2021.

Das bisher noch grösste Problem bei Quantengattercomputern ist die Beschädigung der Daten, während die Berechnung zum Auslesen der endgültigen Resultate fortschreitet. Dies kann passieren, wenn die Qubits ungewollt mit der klassischen Umgebung in Kontakt kommen. Wenn dies geschieht, tendieren die Qubits ihre Quanteneigenschaften zu verlieren, was die gemessenen Daten stark beeinflussen kann. Somit kann der ungewollte Kontakt mit der Umgebung den Zustand des ganzen Quantenregisters beeinflussen und die Daten verändern. "Für den Rechengang ist es wichtig, die quantenmechanischen Zustände des Systems zu erhalten – also die Superposition der einzelnen Qubits und deren Verschränkung miteinander."<sup>50</sup>

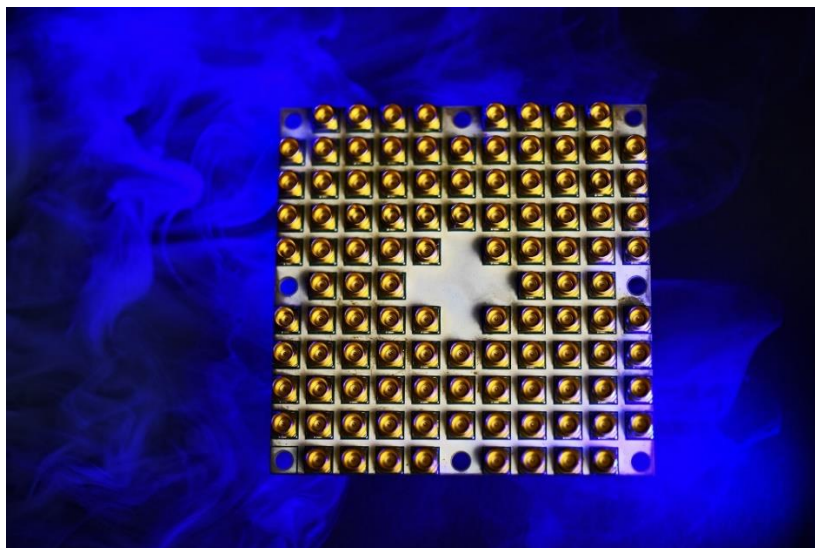
Erste Beispiele für Quantengattercomputer, die schon umgesetzt worden sind, sind:

IBMs Quantencomputer der Serie "Q System One":



*Abbildung 6: Ausgestellter Quantencomputer. IBM: Quantum Computing, 19.05.2021*

Intels 49-Qubit Prozessor "Tangle Lake":



*Abbildung 7: Tangle Lake Quantencomputerchip. techproviderzone.com: Quantum computing, 19.05.2021*

<sup>50</sup> Kusche: Quantencomputer, 21.05.2021.



### 2.4.2 Quanten-Annealing-Maschinen

Quanten-Annealing-Maschinen (engl. «Quantum-annealing-machines») sind Quantencomputer, die sich Quanten-Annealing zunutze machen, eine Klasse von algorithmischen Methoden, die einem natürlicherweise Lösungen mit niedrigen Energien liefern. Einige Anwendungsgebiete der Quanten-Annealing-Maschinen sind Optimierungsprobleme, Anwendungen, die das echte Energieminimum erfordern, und probabilistische Stichprobenprobleme.<sup>51</sup>

Oft werden bei Quanten-Annealing-Maschinen supraleitende Qubits verwendet, wie zum Beispiel bei der Quanten-Annealing-Maschine D-Wave QPU. Somit sind die Qubits die niedrigsten Energiezustände der supraleitenden Schleifen, aus denen die Quanten-Annealing-Maschine besteht. Ausserdem haben die Qubits einen zirkulierenden Strom und ein entsprechendes variables Magnetfeld. Wie schon in Kapitel 1.2 erfahren, kann das Qubit eine Superposition des Zustandes  $|0\rangle$  und des Zustandes  $|1\rangle$  annehmen, wobei es am Ende des Quanten-Annealing Prozesses in einen der klassischen Zustände 0 oder 1 kollabiert. Der Prozess, dass das Qubit von der Superposition in einen klassischen Zustand fällt, lässt sich durch die Zufuhr von Energie beeinflussen.

Dabei kann der physikalische Zustand als Energiediagramm dargestellt werden. Bevor sich das Qubit in der Superposition befindet, gibt es nur ein Tal mit einem einzigen Minimum bzw. Tiefpunkt (a). Während sich das Qubit in der Superposition befindet, der Quanten-Annealing Prozess also läuft, ist die Wahrscheinlichkeit, dass das Qubit den Zustand 0 oder 1 annimmt, je 50 Prozent. Dabei entsteht ein sogenanntes «Double-well potential». In diesem «Double-well potential» entspricht der Tiefpunkt des linken Tals dem Zustand 0 und der Tiefpunkt des rechten Tals dem Zustand 1 (b). Nach dem Quanten-Annealing Prozess fällt das Qubit in eines der Täler, bzw. es nimmt einen dieser Zustände an. Jedoch kann man die Wahrscheinlichkeit, dass das Qubit in eines der Täler fällt, bzw. in einen der Zustände 0 oder 1 fällt beeinflussen und kontrolliert verändern, indem man ein externes Magnetfeld an das Qubit anlegt. Durch dieses Magnetfeld kippt das Potential und es erhöht somit die Wahrscheinlichkeit, dass das Qubit in einer der Täler landet und einen der Zustände annimmt. Dabei wird das externe, veränderbare Magnetfeld von einer programmierbaren Grösse gesteuert, die Bias genannt wird. In der Gegenwart der programmierbaren Grösse minimiert das Qubit seine Energie.<sup>52</sup>

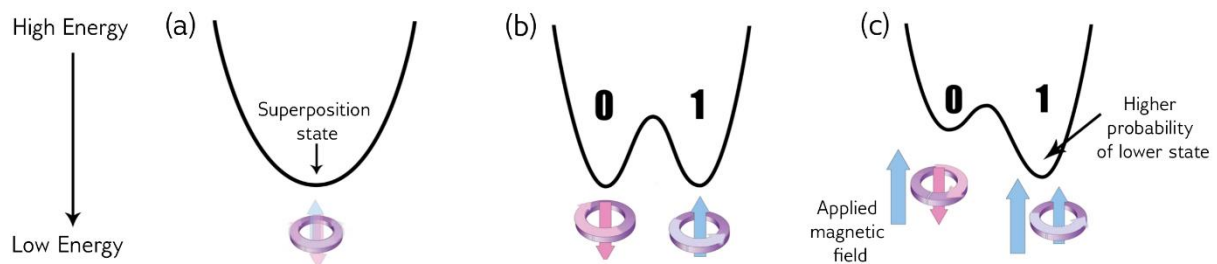


Abbildung 8: Sich über die Zeit veränderndes Energiediagramm. dwavesys.com: Quantum Annealing, 21.05.2021

Jedoch kann man auch mehrere Qubits benutzen und diese so miteinander verbinden, dass sie sich jeweils beeinflussen. Dieser Zustand der Qubits wird wie bereits beschriebene «Verschränkung» (bzw. «Entanglement» im Englischen) oder auch «quantenkohärenter» Zustand genannt. In diesem Zustand beeinflusst die Veränderung eines Qubits gleichzeitig der Zustand aller andern damit verschränkten Qubits. Diese Verschränkung der Qubits kann mit einem Gerät namens «Coupler» oder «Koppler» im Deutschen erreicht werden. Ein Koppler hat zum einen die Eigenschaft, dass er zwei Qubits dazu tendieren lassen kann im gleichen Zustand zu enden, d.h. beide Qubits enden im Zustand 0 oder beide Qubits enden im Zustand 1, oder er kann sie dazu bringen, in entgegengesetzte Zustände zu geraten. Zum anderen kann er Qubits verschränken. Wenn zwei Qubits verschränkt sind, kann man sie sich als

<sup>51</sup> Kobylinska, Martins, Ostler: Quanten-Annealing, 20.05.2021.

<sup>52</sup> Quantum Annealing, in: D-Wave, 20.05.2021.

ein einziges Objekt mit vier möglichen Zuständen vorstellen: (0,0), (0,1), (1,0) und (1,1). Dabei hängt die relative Energie der einzelnen Zustände von den Biases der Qubits und der Kopplung zwischen ihnen ab. Es können somit Energiediagramme erstellt werden, bei denen die beste Antwort gezeigt wird.<sup>53</sup>

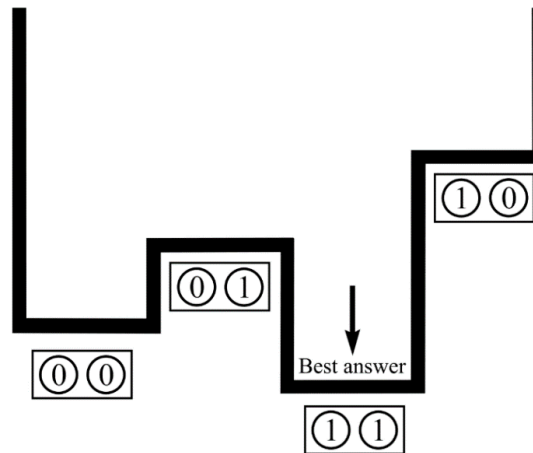


Abbildung 9: Energiediagramm, bei dem die beste Antwort hervorgehoben ist. dwavesys.com: *Quantum Annealing*, 21.05.2021

Beim Quanten-Annealing definieren somit die Base und Kopplungen eine Energielandschaft, wobei die Quanten-Annealing-Maschine die minimale Energie dieser Landschaft findet. Zusätzlich benutzt die Quanten-Annealing-Maschine Quanten-Tunneling. Quanten-Tunneling ist die quantenmechanische Eigenschaft, die es der Quanten-Annealing-Maschine erlaubt dem lokalen Minimum bzw. dem lokalen Tal zu entkommen und dabei die thermische Aktivierungsenergie möglichst gering zu halten durch sogenannte «quantum jumps». Thermische Aktivierungsenergie wird dabei gebraucht, um die Barrieren zwischen den verschiedenen Minima zu überwinden. Da diese Barrieren in den meisten Fällen hoch sind, wird auch eine grosse Menge von thermischer Aktivierungsenergie verwendet, um die Barriere zwischen den Minima zu überwinden. Durch Quanten-Tunneling wird diese thermische Aktivierungsenergie reduziert, da der Pfad zwischen den Minima oftmals sehr viel kleiner ist.<sup>54, 55</sup>

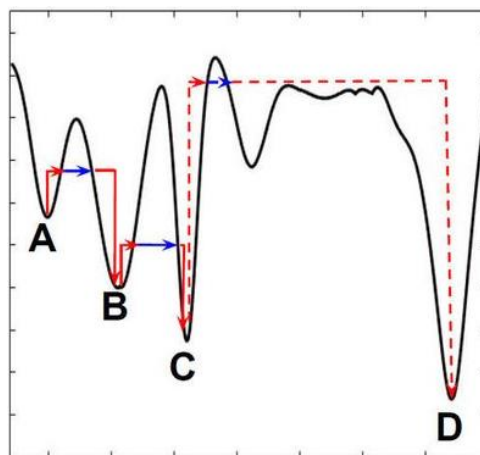


Abbildung 10: Quanten-Tunneling zwischen den Minima A, B, C und D, welches die thermische Aktivierungsenergie deutlich senkt. Google: *Quanten-Annealing*, 22.05.2021

<sup>53</sup> Quantum Annealing, in: D-Wave, 20.05.2021.

<sup>54</sup> Kobylinska, Martins, Ostler: *Quanten-Annealing*, 20.05.2021.

<sup>55</sup> De la Fuente Ruiz: *Quantum Annealing*, 21.05.2021.

Dabei werden die Systeme der Quanten-Annealing-Maschine immer komplizierter, je mehr Qubits zu dem System hinzugefügt werden. So verdoppelt jedes zusätzliche Qubit die Anzahl der Zustände, über die man ein Energiediagramm bzw. beim Quanten-Annealing eine Energielandschaft definieren kann.

Ein Anwendungsgebiet der Quanten-Annealing-Maschine sind die Optimierungsprobleme. Optimierungsprobleme sind Probleme, bei denen man nach der besten Kombination von sehr vielen möglichen Kombinationen sucht. Dazu gehören Planungsprobleme, wie z.B. «Soll ich mein Paket mit diesem oder dem nächsten Schiff versenden?» oder «Was ist der beste und effizienteste Weg für einen Handelsmann, den er nehmen kann, um verschiedene Städte in einem Land zu bereisen?»<sup>56</sup>

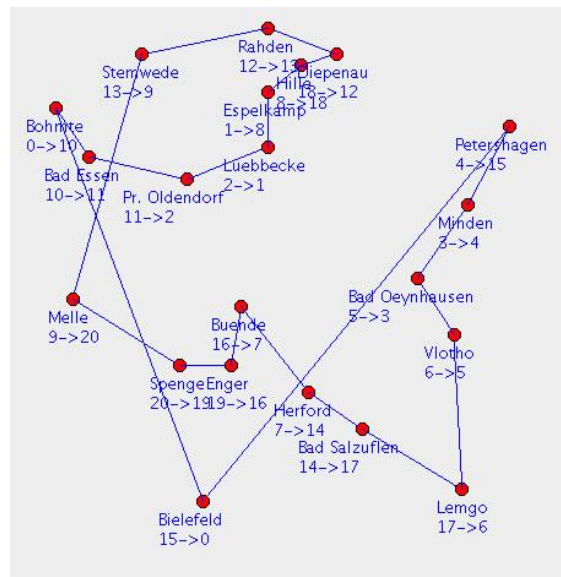


Abbildung 11: Beispiel eines grafisch dargestellten Optimierungsproblems. Helmich: Optimierungsprobleme, 22.05.2021

Die klassische Physik kann bei solchen Problemen helfen, denn mit der klassischen Physik lassen sich diese Probleme als Energieminimierungsprobleme formulieren. Dies ist möglich, da in der Physik alles dazu neigt, einen Zustand mit minimaler Energie anzustreben. Da dieses Verhalten zusätzlich in der Quantenphysik gilt, nutzt die Quanten-Annealing-Maschine die Quantenphysik, um energiearme Zustände eines Problems und damit die optimale oder nahezu optimale Kombination von Elementen zu suchen.<sup>57</sup>

Stichprobenprobleme, auch "sampling problems" genannt, sind Probleme, die gut von Quanten-Annealing-Maschinen gelöst werden können, da dort probabilistische Modelle der Realität erstellt werden müssen. In diesen probabilistischen Modellen sind Schätzung einer räumlichen Grösse, wie z. B. eines Variogramms oder Korrelogramms (welches wiederum die räumliche Abhängigkeit zusammenfasst) oder einer interpolierten Fläche (z. B. einer Konturkarte) dargestellt.<sup>58</sup>

Jedoch ist bei der Quanten-Annealing-Maschine noch die Zerbrechlichkeit der Superposition des Qubits ein sehr grosses Problem. Somit können Einflüsse wie zufällige Wechselwirkungen eines Qubits in einem Quantenregister mit der klassischen Umgebung das gesamte Netzwerk des Quantenregisters in einem System stark beeinflussen, zusammenbrechen lassen oder sogar abtrennen.<sup>59</sup>

<sup>56</sup> Optimierungsproblem, in: Wikipedia, 22.05.2021.

<sup>57</sup> Quantum Annealing, in: D-Wave, 20.05.2021.

<sup>58</sup> De la Fuente Ruiz: Quantum Annealing, 21.05.2021.

<sup>59</sup> Kobylinska, Martins, Ostler: Quanten-Annealing, 20.05.2021.



Der Unterschied zwischen simuliertem Annealing und Quanten-Annealing ist der, dass bei Quanten-Annealing die Tunnelfeldstärke, also die Strecke, in der man zufällig den nächsten Kandidaten bzw. das nächste Minima auswählt, benutzt wird. Bei simuliertem Annealing wird statt der Tunnelfeldstärke ein Temperatur-Parameter benutzt, der eine ähnliche Rolle spielt, wie die Tunnelfeldstärke. Durch die Ersetzung der Tunnelfeldstärke durch den Temperatur-Parameter ist das simulierte Annealing nur ein heuristisches Approximationsverfahren zur Lösung von Optimierungs- und Stichprobenproblemen, währenddessen der Quanten-Annealing Prozess ein exaktes Approximationsverfahren zur Lösung von Optimierungs- und Stichprobenproblemen ist. Durch diese Eigenschaft übertrifft der Quanten-Annealing-Prozess das simulierte Annealing ab ca. 1'000 binären Variablen.<sup>60, 61, 62</sup>

Eine der bekanntesten Realisierungen der Quanten-Annealing-Maschinen, die sich den Quanten-Annealing-Prozess zunutze macht, sind die Quanten-Annealing-Maschinen von D-Wave: D-Wave 2000Q, die vierte der Quanten-Annealing-Maschine von D-Wave Systems, ist mit 2000 Qubits und neuen Steuerungsfunktionen einer der besten, wenn nicht die beste Quanten-Annealing-Maschine, welche grössere Probleme mit einer schnelleren Leistung als bisher lösen kann.



*Abbildung 12: D-Waves 2000Q, die erste Quanten-Annealing-Maschine mit 2000 Qubits. Barb: Quanten-Annealing, 23.05.2021*

---

<sup>60</sup> Quantum annealing, in: Wikipedia, 23.05.2021.

<sup>61</sup> Simulated annealing, in: Software engineering, 23.05.2021.

<sup>62</sup> Koshka, Novoty: Simulated Annealing, 23.05.2021.

### 3 Programmierung von Quantencomputern

Im Abschnitt 2.4 wurden die verschiedenen Ansätze zur Realisierung von Quantencomputern aufgeführt. Im Rahmen dieser Maturaarbeit wird der Fokus auf die Quantengattercomputer und deren Programmierung gelegt.

Der Quantengattercomputer ähnelt herkömmlichen Rechnern, die mithilfe klassischer Logikgatter operieren. Die oben beschriebenen Eigenschaften von Qubits sind dabei aber zu beachten. Das heisst, dass Qubits in eine Superposition gebracht werden und nach Anwendung aller Gatter gemessen werden müssen, was ihren Zustand zerstört. Um die Quantengatter deshalb besser zu verstehen, werden wir uns zuerst die klassischen Logikgatter ansehen, um später einen einfacheren Übergang zu den Quantengattern herstellen zu können.<sup>63, 64</sup>

#### 3.1 Klassische Logikgatter

Logikgatter sind Bausteine der booleschen Algebra und bilden logische Operationen und Mengenoperationen ab. Durch die in den Logikgatter verankerten booleschen Funktionen lassen sich die booleschen Funktionen zu komplexen Schaltungen kombinieren. Die Operationen haben ein oder zwei Bits als Eingänge und jeweils einen Ausgang. Dabei werden die Ausgangsbits abhängig von den Eingangsbits der Schaltung gesetzt. Die Eingangs- und Ausgangsbits können die Werte 0 und 1 annehmen, da sie sich wie herkömmliche Bits verhalten. Man spricht deshalb auch von digitalen Schaltungen. Insgesamt gibt es sieben unterschiedliche Standard-Logikgatter, die durch die grundlegenden Operationen der booleschen Algebra gebildet werden: Negation («NICHT»-Operation, NOT), Konjunktion («UND»-Operation, AND), Disjunktion («ODER»-Operation, OR) und Kontravalenz («Exklusiv-ODER»-Operation, XOR). Darüber hinaus gibt es «NICHT-UND» (NAND), «NICHT-ODER» (NOR) und «Exklusiv-NICHT-ODER» (XNOR). Diese Operationen werden durch jeweils ein bis drei Logikgatter abgebildet. Die wichtigsten vier Logikgatter sind wie folgt aufgebaut:<sup>65, 66, 67</sup>

##### 3.1.1 NOT / NICHT – Gatter

Das NOT-Gatter ist eine der Standard-Logikgatter mit nur einer Operation, der «Nicht»-Operation. Durch die «Nicht»-Operation hat das NOT-Gatter nur ein Eingangsbit und ein Ausgangsbit und setzt den invertierten Wert des Eingangsbit als den Wert des Ausgangsbits. Somit wird das Ausgangsbit auf 1 gesetzt, wenn das Eingangsbit auf 0 gesetzt wurde und das Ausgangsbit wird auf 0 gesetzt, wenn das Eingangsbit auf 1 gesetzt wurde.

##### 3.1.2 AND / UND-Gatter

Das AND-Gatter bildet eine Konjunktion von mindestens zwei Eingangsbits auf ein Ausgangsbit ab und verhält sich wie ein logisches UND. Das heisst, dass nur wenn beide Eingangsbits den Wert 1 haben, das Ausgangsbit auf 1 gesetzt wird. Wenn mindestens ein Eingangsbit den Wert 0 besitzt, wird das Ausgangsbit auf 0 gesetzt.

##### 3.1.3 OR / ODER-Gatter

Das OR-Gatter bildet eine Disjunktion von mindestens zwei Eingangsbits auf ein Ausgangsbit ab und verhält sich wie ein logisches ODER. Somit wird das Ausgangsbit auf 1 gesetzt, wenn mindestens ein Eingangsbit den Wert 1 besitzt. Wenn beide Eingangswerte den Wert 0 haben, wird das Ausgangsbit auf 0 gesetzt.

---

<sup>63</sup> Neukart: Quantenwelt (2020), S. 146.

<sup>64</sup> Logikgatter, in: Wikipedia, 26.05.2021.

<sup>65</sup> Boolesche Algebra, in: ElectronicsTutorials, 26.05.2021.

<sup>66</sup> Logikgatter, in: Wikipedia, 26.05.2021.

<sup>67</sup> Augusten: Logikgatter, 26.05.2021.

### 3.1.4 XOR-Gatter

Das XOR-Gatter bildet eine Kontravalenz von mindestens zwei Eingangsbits auf ein Ausgangsbit ab und verhält sich wie ein logisches ENTWEDER – ODER. Dies bedeutet, dass das Ausgangsbit auf den Wert 1 gesetzt wird, wenn die Eingangsbits einen unterschiedlichen Wert haben. Bei gleichen Werten der Eingangsbits wird das Ausgangsbit auf 0 gesetzt.<sup>68</sup>

### 3.1.5 Wahrheitstabellen

Die oben aufgeführten Gatter sind die Standard-Logikgatter, in welchen jeweils eine der grundlegenden Operationen der booleschen Algebra abgebildet wird. Die vier weiteren Standard-Logikgatter sind eine Kombination von zwei der oben aufgeführten Logikgatter.

Logikgatter können aber auch in einer Wahrheitstabelle aufgelistet werden. In dieser Wahrheitstabelle wird jedem Eingangsbit bzw. jeder Kombination von Eingangsbits gemäss der jeweiligen booleschen Funktion ein eindeutiger Ausgangsbit zugeordnet. Eine Spalte einer Wahrheitstabelle stellt alle möglichen Eingangs- und deren zugeordneten Ausgangskombinationen dar.<sup>69</sup>

Die Wahrheitstabelle eines NOT-Gatters ist:

Eingangsbit	Ausgangsbit
0	1
1	0

Die Wahrheitstabelle eines AND-Gatters ist:

Eingangsbit 1	Eingangsbit 2	Ausgangsbit
0	0	0
1	0	0
0	1	0
1	1	1

Und die Wahrheitstabelle eines OR-Gatters ist:

Eingangsbit 1	Eingangsbit 2	Ausgangsbit
0	0	0
1	0	1
0	1	1
1	1	1

Die Wahrheitstabelle eines XOR-Gatters ist:

Eingangsbit 1	Eingangsbit 2	Ausgangsbit
0	0	0
1	0	1
0	1	1
1	1	0

Logikgatter sind Abbildungen der grundlegenden booleschen Funktionen. Aus diesen Abbildungen der grundlegenden Funktionen lassen sich demnach auch viel komplexere boolesche Funktionen, aber auch beliebige Operationen nachbilden. Mehrere Logikgatter kann man beispielsweise zu verschiedenen

<sup>68</sup> Augusten: Logikgatter, 26.05.2021.

<sup>69</sup> Neukart: Quantenwelt (2020), S. 147.

logischen Verknüpfungen, wie zum Beispiel zu einem Flipflop oder zu einem Multiplex zusammenschalten. Durch Kombination der Gatter lassen sich sämtliche Algorithmen, die durch einen klassischen Computer implementiert werden können, realisieren.<sup>70, 71</sup>

## 3.2 Quantengatter

Im letzten Abschnitt haben wir die klassischen Logikgatter, welche die Basis von klassischen Computern bilden, eingeführt. In diesem Abschnitt werden wir uns den Quantengattern, den grundlegenden Bestandteilen von Quantengattercomputern, widmen. Quantengatter sind, im Gegensatz zu digitalen Gattern, keine Bauelemente. Man kann sie sich als Manipulationen des Zustands (Spins) eines Qubits vorstellen. Dabei wird die vorher beschriebene Grundzustand eines Qubits verändert. Diese Veränderung kann man mittels einer Matrixoperation beschreiben, welche in Abschnitt 2.2.3 erklärt wurden. Somit ist jedes Gatter durch seine Matrixoperation definiert. Diese Matrixoperation muss «unitär» sein. Das bedeutet, dass die Matrixoperationen umkehrbar sein müssen, was mit der Erhaltung der Wellenfunktion zusammenhängt.

Für Quantengattercomputer existieren unzählige Gatter. Im Folgenden werden die wichtigsten Gatter für diese Arbeit beschrieben. Dabei wird neben dem Gatter auch die Matrixoperation und deren Auswirkung auf das Qubit oder die Qubits beschrieben. Es wird immer die Darstellung der einzelnen Gatter in dem Quantencomputersimulator Quirk zur Veranschaulichung gezeigt.<sup>72</sup> Die Ausgangslage ist der Zustand  $|0\rangle$ :

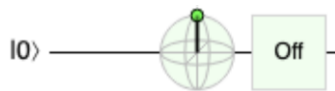


Abbildung 13: Basiszustand eines Qubits in dem Quantencomputersimulator Quirk. Datenquelle: algassert.com, 05.06.2021

### 3.2.1 Pauli-Gatter

#### 3.2.1.1 Pauli-X-Gatter

Das Pauli-X Gatter ist ein 1-Qubit Gatter, also ein Gatter, welches nur auf ein einzelnes Qubit wirkt. Dabei kehrt das Pauli-X-Gatter den Zustandsvektor («Statevektor») eines Qubit um, indem er die Amplitude des jeweiligen Basiszustandes  $|0\rangle$  oder  $|1\rangle$  umkehrt. Somit hat das Pauli-X-Gatter die gleiche Funktion wie das NOT-Gatter bei den klassischen Logikgattern und wird daher auch oft Bit-flip genannt.<sup>73, 74</sup>

Die Matrix des Pauli-X-Gatters sieht so aus:

$$X = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Wenn man nun das Pauli-X-Gatter auf die Basiszustände eines Qubits anwendet, sieht dies wie folgt aus:

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

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

<sup>70</sup> Neukart: Quantenwelt (2020), S. 147.

<sup>71</sup> Logikgatter, in: Wikipedia, 26.05.2021.

<sup>72</sup> Quirk, in: Algassert, 05.06.2021.

<sup>73</sup> Quantum logic gate, in: Wikipedia, 02.06.2021.

<sup>74</sup> Single Qubit Gates, in: Qiskit, 02.06.2021.

Das zweimalige Anwenden des Pauli-X-Gatters, bzw. der Matrix des Pauli-X-Gatters, auf den Basiszustand  $|0\rangle$ , versetzt den Basiszustand wieder in seine Ursprungsform. Dies zeigt, dass das Pauli-X-Gatter zu sich selbst invers ist:  $X^{-1} = X$ , da  $X^{-1}X = E$ .

Das Gatter wird in Quirk wie folgt dargestellt und verändert den Vektor und damit den Basiszustand  $|0\rangle$  zu dem Basiszustand  $|1\rangle$ :

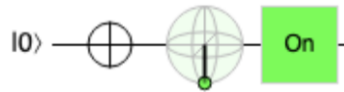


Abbildung 14: Wirken des Pauli-X-Gatters auf den Basiszustand  $|0\rangle$ . Datenquelle: algassert.com, 05.06.2021

### 3.2.1.2 Pauli-Y- und Pauli-Z-Gatter

Die Pauli-Y- und Pauli-Z-Gatter sind analog, wirken sich jedoch auf andere Achsen des Spins aus. Das Pauli-Y-Gatter wirkt dabei so auf die Basiszustände, dass der Basiszustand  $|0\rangle$  abbildet als  $i|1\rangle$  und den Basiszustand  $|1\rangle$  abbildet als  $-i|0\rangle$ . Das Pauli-Z-Gatter wirkt andererseits so auf die Basiszustände, dass der Basiszustand  $|0\rangle$  sich durch das Pauli-Z-Gatter nicht verändert wird, aber der Basiszustand  $|1\rangle$  wird abgebildet  $-|1\rangle$ .<sup>75</sup>

$$Y = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = -i|0\rangle\langle 1| + i|1\rangle\langle 0|$$

$$Z = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1|$$

Die Anwendung des Pauli-Y-Gatters und des Pauli-Z-Gatters auf die Basiszustände eines Qubits, sieht wie folgt aus:

$$Y|0\rangle = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ i \end{pmatrix} = i|1\rangle$$

$$Y|1\rangle = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -i \\ 0 \end{pmatrix} = -i|0\rangle$$

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

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

Bei einer nochmaligen Anwendung des Pauli-Y-Gatters und auch bei einer nochmaligen Anwendung des Pauli-Z-Gatters, wird bei beiden Gattern wie bei dem Pauli-X-Gatter den Ursprungszustand wiedergegeben. Somit ist die Matrix des Pauli-Y-Gatter sowie des Pauli-Z-Gatter invers. Es gilt also:  $Y^{-1} = Y$  und  $Z^{-1} = Z$ .<sup>76</sup>

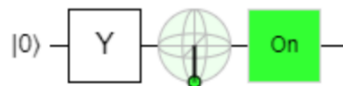


Abbildung 15: Wirken des Pauli-Y-Gatters auf den Basiszustand  $|0\rangle$ . Datenquelle: algassert.com, 12.06.2021



Abbildung 16: Wirken des Pauli-Z-Gatters auf den Basiszustand  $|0\rangle$ . Datenquelle: algassert.com, 12.06.2021

<sup>75</sup> Single Qubit Gates, in: Qiskit, 02.06.2021.

<sup>76</sup> Roell: Quantum Gates, 03.06.2021.

### 3.2.2 Hadamard-Gatter

Das Hadamard-Gatter ist ein Gatter, welches auf ein einzelnes Qubit wirkt. Dabei hebt das Gatter den Basiszustand  $|0\rangle$  oder  $|1\rangle$  in eine Superposition der beiden möglichen Zustände. Dabei sieht die Matrix des Hadamard Gatters wie folgt aus:

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

Wenn man das Hadamard-Gatter auf einem Qubit im Zustand  $\alpha|0\rangle + \beta|1\rangle$  angewendet werden soll, wird ein Rechenschritt auf dem Qubit ausgeführt. Dieser Rechenschritt wird durch eine zweidimensionale Matrix

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

beschrieben. Wenn wir nun den Zustandsvektor mit der Matrix  $A$  multiplizieren erhalten wir:

$$\begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} = A \cdot \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} a \cdot \alpha & b \cdot \beta \\ c \cdot \alpha & d \cdot \beta \end{pmatrix}$$

Wenn wir nun statt der Matrix  $A$  die Matrix des Hadamard Gatters einsetzen, erhalten wir:

$$\begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} = H \cdot \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \cdot \alpha & \frac{1}{\sqrt{2}} \cdot \beta \\ \frac{1}{\sqrt{2}} \cdot \alpha & -\frac{1}{\sqrt{2}} \cdot \beta \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \cdot \alpha & 1 \cdot \beta \\ 1 \cdot \alpha & -1 \cdot \beta \end{pmatrix}$$

Wenn man nun das Hadamard-Gatter auf die Basiszustände  $|0\rangle$  oder  $|1\rangle$  angewendet wird, sieht das so aus:

$$\begin{aligned} |0\rangle &\xrightarrow{H} \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle &\xrightarrow{H} \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned}$$

Dies zeigt, dass wir beim Messen des Qubits zu 50% Wahrscheinlichkeit den Basiszustand  $|0\rangle$  und zu 50% Wahrscheinlichkeit den Basiszustand  $|1\rangle$  messen werden.<sup>77</sup>



Abbildung 17: Wirken des Hadamard-Gatters auf den Basiszustand  $|0\rangle$ . Datenquelle: algassert.com, 05.06.2021

Beim nochmaligen Anwenden des Hadamard-Gatters ergibt sich:

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \xrightarrow{H} |0\rangle$$

<sup>77</sup> Homeister, Quantencomputing, S. 24.

$$\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \xrightarrow{H} |1\rangle$$

Dies zeigt, dass das Hadamard-Gatter zu sich selbst invers ist, also  $H^{-1} = H$  wahr ist, das Hadamard-Gatter macht sich somit bei einer erneuten Anwendung selbst rückgängig.<sup>78</sup> Dies haben wir bereits in den mathematischen Grundlagen bewiesen.

### 3.2.3 Phasenverschiebungs-Gatter

Das Phasenverschiebungs-Gatter, auch  $R_\phi$ -Gatter bezeichnet, ist wie die Pauli-Gatter und das Hadamard-Gatter auch ein 1-Qubit Gatter. Dabei bildet es die Basiszustände bei Anwendung des Phasenverschiebungs-Gatters so ab, dass der Basiszustand  $|0\rangle$  gleich bleibt, der Basiszustand  $|1\rangle$  jedoch als  $e^{i\phi}|1\rangle$  abgebildet wird. Das Phasenverschiebungs-Gatter ist parametrisiert, das heisst, dass es von einer Zahl  $\phi$  abhängt, welche durch Einsetzen von verschiedenen Werten die Berechnungen durch das Gatter bestimmt. Das Phasenverschiebungs-Gatter führt eine Drehung um  $\phi$  um die Z-Achse durch, weshalb es auch oft als  $R_Z$ -Gatter bezeichnet wird. Die Matrix des Phasenverschiebungs-Gatters sieht dabei so aus:<sup>79</sup>

$$R_\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$$

### 3.2.4 Z-, S- und T-Gatter

Das Z-, S- und T-Gatter sind alles Anwendungen des Phasenverschiebungs-Gatters. Bei diesen Gattern, werden für das  $\phi$  im  $e^{i\phi}$  des Phasenverschiebungs-Gatters verschiedene Werte eingesetzt, um den so entstehenden Gattern verschiedene Eigenschaften zu geben. Die einzelnen Gatter führen somit eine Drehung von dem jeweiligen Parameter um die Richtung der Z-Achse durch.

Bei dem Z-Gatter wird  $\phi = \pi$  gesetzt. Das Z-Gatter führt eine Drehung um  $\pi$  durch, also eine halbe Drehung, um die Richtung der Z-Achse. Dabei sieht die Matrix des Z-Gatters so aus:

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = R_\pi$$

, da  $e^{i\pi} = -1$  ist.

Bei dem S-Gatter gilt  $\phi = \frac{\pi}{2}$ . Dabei sieht die Matrix so aus:

$$S = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = R_{\frac{\pi}{2}} = \sqrt{Z}$$

Bei dem T-Gatter gilt  $\phi = \frac{\pi}{4}$ . Dabei sieht die Matrix so aus:

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} \cdot i \end{pmatrix} = R_{\frac{\pi}{4}} = \sqrt{S} = \sqrt[4]{Z}$$

Es wird also bei dem S-Gatter eine Drehung von  $\frac{\pi}{2}$ , also von einem Viertel und bei dem T-Gatter eine Drehung von  $\frac{\pi}{4}$ , also einem Achtel um die Richtung der Z-Achse durchgeführt.<sup>80</sup>

<sup>78</sup> Homeister, Quantencomputing, S. 25.

<sup>79</sup> Single Qubit Gates, in: Qiskit, 02.06.2021.

<sup>80</sup> Quantum logic gate, in: Wikipedia, 02.06.2021.

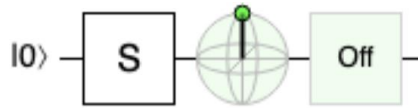


Abbildung 18: Wirken des S-Gatters auf den Basiszustand  $|0\rangle$ . Datenquelle: algassert.com, 05.06.2021

### 3.2.5 CNOT Gatter

Das CNOT-Gatter wirkt auf zwei verschiedenen Qubits, ein Kontroll-Qubit und ein Ziel-Qubit. Wenn sich die Qubits im Basiszustand  $|0\rangle$  oder  $|1\rangle$  befinden, dann verhalten sie sich wie klassische Bits. In diesem Zustand ist das Gatter sehr einfach und intuitiv zu verstehen. Das CNOT-Gatter führt dabei eine NOT-Operation auf dem zweiten Qubit durch, wenn der Basiszustand des ersten Qubits  $|1\rangle$  ist. Wenn der Basiszustand des ersten Qubits jedoch  $|0\rangle$  ist, dann verändert es den Basiszustand des zweiten Qubits nicht.<sup>81</sup>

Um uns das zu veranschaulichen, können wir die klassische Wahrheitstabelle verwenden:<sup>82</sup>

Eingangswert ( $q_1, q_2$ )	Ausgangswert ( $q_1, q_2$ )
$ 00\rangle$	$ 00\rangle$
$ 01\rangle$	$ 01\rangle$
$ 10\rangle$	$ 11\rangle$
$ 11\rangle$	$ 10\rangle$

Dabei ist die Matrix des CNOT-Gatters die folgende:

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

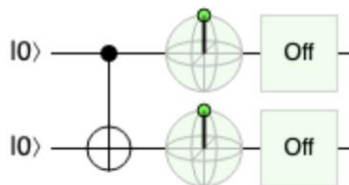


Abbildung 19: Wirken des CNOT-Gatters auf zwei Qubits mit je dem Basiszustand  $|0\rangle$ . Datenquelle: algassert.com, 05.06.2021

Die Kombination von H-Gatter und CNOT-Gatter nennt man Bell Zustand oder Bell State.<sup>83</sup>

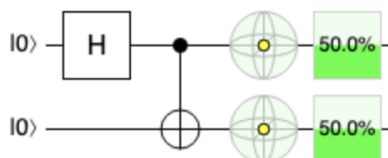


Abbildung 20: Wirken des Bell Zustandes auf zwei Qubits mit je dem Basiszustand  $|0\rangle$ . Datenquelle: algassert.com, 05.06.2021

<sup>81</sup> Quantum logic gate, in: Wikipedia, 02.06.2021.

<sup>82</sup> Single Qubit Gates, in: Qiskit, 02.06.2021.

<sup>83</sup> Bell State, in: Wikipedia, 02.06.2021.



Der Bell Zustand wird mit folgender Formel beschrieben:

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} = |\Phi^+\rangle$$

Die Besonderheit dieses Zustandes ist die in Abschnitt 2.1.4 beschriebene Verschränkung der beiden Qubits, welche im Rahmen der klassischen Physik nicht erklärt werden kann. Die Bell Zustände sind maximal verschränkt.

### 3.3 Quantenschaltungen

Eine Quantenschaltung ist eine Rechenroutine, die aus kohärenten Quantenoperationen auf Qubits und gleichzeitigen klassischen Echtzeitberechnungen besteht. Es handelt sich um eine geordnete Abfolge von Quantengattern, die miteinander verbunden sind.

Eine Menge von Quantengattern wird als universell bezeichnet, wenn jede unitäre Transformation der Quantendaten effizient und beliebig nahe durch eine Folge von Gattern der Menge approximiert werden kann. Jedes Quantenprogramm kann durch eine Folge von Quantenschaltungen und nicht-gleichzeitigen klassischen Berechnungen dargestellt werden.<sup>84, 85</sup>

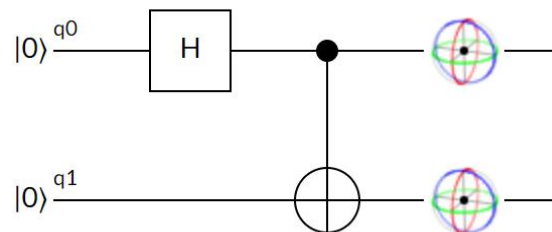


Abbildung 21: Einfache Quantenschaltung mit einem Hadamard-Gatter und einem CNOT-Gatter, auch Bell State genannt. Datenquelle: quantum-circuits.com, 12.06.2021

Wie aufgezeigt wurde, gibt es einige Quantengatter, die sehr ähnlich sind wie ihre klassischen Gegenstücke. So ist zum Beispiel das X-Pauli-Gatter und das CNOT-Gatter sehr ähnlich zu dem NOT-Gatter. Jedoch wurde auch gezeigt, dass mithilfe der Anwendung der Quantenmechanik auf die klassischen Logikgatter komplett neue Gatter entstehen können wie zum Beispiel das Hadamard-Gatter oder das Phasenverschiebungs-Gatter. Diese unterschiedlichen Gatter können nun genutzt und auf Qubits angewendet werden, um diese zu verändern. Eine Reihe von Quantengattern, die eine geordnete Abfolge bilden, nennt man Quantenschaltung.

<sup>84</sup> Quantum Circuits, in: Qiskit, 23.06.2021.

<sup>85</sup> Marinescu, D; Marinescu, G: Quantum Circuits, 23.06.2021.

## 4 Spieltheorie

### 4.1 Klassische Spieltheorie

Die Spieltheorie, auch «Game Theory» genannt, in der Form, wie sie Ökonomen, Sozialwissenschaftlern und Biologen bekannt ist, erhielt 1944 ihre erste allgemeine mathematische Formulierung von John von Neumann und Oskar Morgenstern. Dies ist jedoch nicht die endgültige Version der Spieltheorie; es werden ständig Verbesserungen vorgenommen, um ungelöste Probleme zu lösen.

Die Untersuchung der strategischen Interaktionen verschiedener rationaler Entscheidungsträger in sozialen Konfliktsituationen wird als Spieltheorie bezeichnet. Die Bedeutung der Spieltheorie liegt jedoch darin, dass das Verhalten eines Spielers, d. h. eines strategischen Entscheidungsträgers im Rahmen des Spiels, und seine Handlungen von der Strategie des anderen Spielers abhängig sind. Die Strategie eines Spielers bezieht sich auf einen umfassenden Aktionsplan, den ein Spieler als Reaktion auf die im Spiel auftretenden Situationen ausführen kann.<sup>86</sup>

Nach der Definition der Spieltheorie wirken sich die Handlungen und Entscheidungen aller am Spiel Beteiligten auf das Ergebnis eines jeden einzelnen aus. Es wird auch davon ausgegangen, dass die Spieler rational sind und somit vernünftig handeln und dass jeder Spieler danach strebt mit jedem Zug seinen Gewinn im Spiel zu maximieren.

Das Spiel identifiziert und zeigt somit die Identitäten, Präferenzen und verfügbaren Strategien der Spieler auf und wie diese Strategien das Ergebnis des Spiels beeinflussen können. "Die Spieltheorie hat ein breites Anwendungsspektrum, darunter Psychologie, Evolutionsbiologie, Krieg, Politik, Wirtschaft und Geschäftsleben. Trotz ihrer vielen Fortschritte ist die Spieltheorie eine noch junge und sich entwickelnde Wissenschaft."<sup>87</sup>

#### 4.1.1 Nash-Gleichgewicht

Das Nash-Gleichgewicht, auch «Nash Equilibrium» genannt, ist ein Kernbegriff der Spieltheorie. Es ist ein Entscheidungstheorem in der Spieltheorie, welches besagt, dass das optimale Ergebnis eines Spiels dann erreicht wird, wenn kein Spieler einen Anreiz hat, von seiner ursprünglichen Strategie abzuweichen, nachdem er die Entscheidung eines Gegners berücksichtigt hat.<sup>88</sup> "Strategisch aus der Sicht eines Spielers betrachtet bedeutet dies: Ich tue das Beste, was ich kann, unter Berücksichtigung dessen, was du tust; du tust, unter Berücksichtigung dessen, was ich tue, das Beste, was du tun kannst."<sup>89</sup>

Die Definition und der Beweis für die Existenz des Nash-Gleichgewichts lassen sich auf eine Veröffentlichung des Mathematikers John Forbes Nash Jr. aus dem Jahr 1950 zurückführen.<sup>90</sup>

##### 4.1.1.1 Grundidee und Definition des Nash-Gleichgewichts

Das Hauptziel der mathematischen Spieltheorie besteht darin, rationale Konflikte und kooperative Entscheidungsfindung zu beschreiben und zu charakterisieren. Das Dilemma besteht darin, dass keiner der Entscheidungsträger oder Spieler die Pläne der anderen Spieler kennt oder weiss, wie diese darauf reagieren. Daher ist es schwierig, vorherzusagen, wie sich der strategische Zug eines einzelnen Spielers auswirken wird. Er kann jedoch die Umstände aus der Sicht der anderen Spieler bewerten, um das vorherzusagen, was sie tun werden.<sup>91</sup>

Das Nash Equilibrium basiert auf der Idee, dass man mit allen möglichen Kombinationen von Strategien für jeden Spieler beginnt. Falls dabei kein einzelner Teilnehmer die Motivation hat, von seiner Strategie

---

<sup>86</sup> Ross: Game Theory, 25.08.2021.

<sup>87</sup> Hayes: Game Theory, 25.08.2021.

<sup>88</sup> Chen: Nash Equilibrium, 25.08.2021.

<sup>89</sup> Nash-Gleichgewicht, in: Wikipedia, 25.08.2021.

<sup>90</sup> Chen: Nash Equilibrium, 25.08.2021.

<sup>91</sup> Nash-Gleichgewicht, in: Wikipedia, 25.08.2021.

abzuweichen, kann man davon ausgehen, dass diese Strategiekombination eine gewisse Stabilität besitzt. Diese Strategiekombination wird als Nash-Gleichgewicht bezeichnet.

Durch das Spielen eines Spieles mit einem Nash-Gleichgewicht entstehen sogenannte Auszahlungen. Diese Auszahlungen korrespondieren mit dem jeweiligen Spiel, welches gespielt wird. Dabei kann die Auszahlung eines Spieles Geld sein oder wie bei bekannten Spielen wie bei Gefangenendilemma die Zeit der Haft. Die Grösse der jeweiligen Auszahlung hängt von der Strategie bzw. den Entscheidungen des Spielers ab, die dieser anwendet bzw. trifft. Bei unterschiedlichen Strategien und Entscheidungen gibt es unterschiedliche grosse Auszahlungen. Aufgrund der Grösse der Auszahlung sind bestimmte Strategien dominant oder rezessiv.

Bei dem Nash-Gleichgewicht werden drei Arten von Strategien unterschieden: Dominante Strategien, reine Strategien, gemischte Strategien.

#### *Dominante Strategien:*

Wenn die Strategiewahl des Spielers unabhängig von der Strategiewahl des Gegners ist, sprechen manche Quellen vom Nash-Gleichgewicht in der dominanten Strategie. Das bekannteste Beispiel für das Nash-Gleichgewicht in der dominanten Strategie ist das Gefangenendilemma.

#### *Reine Strategien:*

Wenn der Spieler die von seinem Gegner gewählte Strategie kennt und mit der Antwort antwortet, die ihm am besten passt und somit eine ganz bestimmte Entscheidung trifft, kann man von einem Nash-Gleichgewicht in reiner Strategie sprechen.

#### *Gemischte Strategien:*

Um das Nash-Gleichgewicht in der gemischten Strategie zu bestimmen, wird auch die Wahrscheinlichkeit der Strategiewahl jedes Teilnehmers berücksichtigt. In diesem Fall haben endliche Spiele mit gemischten Strategien im Vergleich zu reinen Strategiespielen immer ein Nash-Gleichgewicht.<sup>92</sup>

#### *4.1.1.2 Prisoner's Dilemma*

Das Prisoner's Dilemma, auch Gefangenendilemma, ist ein Paradoxon in der Entscheidungsanalyse und der Spieltheorie, welches von Merrill Flood und Melvin Dresher 1950 konzipiert wurde. Dabei zeigt das Gefangenendilemma auch die Wirkung des Nash-Gleichgewichts. In diesem Dilemma handeln zwei Personen in ihrem eigenen Interesse und können somit nicht das optimale Ergebnis erzielen.

##### *4.1.1.2.1 Beschreibung des Dilemmas*

Zwei Kriminelle wurden festgenommen und ins Gefängnis gesteckt. Jeder Gefangene wurde in Einzelhaft gehalten und konnte somit nicht mit dem jeweiligen anderen Gefangenen kommunizieren. Die Staatsanwaltschaft hatte nicht genügend Beweise, um sie beide in den Hauptanklagen zu verurteilen, aber sie hatten genügend Beweise, um sie beide in den geringeren Anklagen zu verurteilen. Gleichzeitig unterbreitete der Staatsanwalt jedem Gefangenen ein Verhandlungsangebot. Jeder Gefangene kann wählen, ob er gestehen möchte, indem er behauptet, dass die andere Partei ein Verbrechen begangen hat, oder mit der anderen Partei kooperiert, indem sie schweigt. Die daraus folgenden möglichen Ergebnisse sind:

- Wenn A und B gestehen, muss jeder von ihnen drei Jahre im Gefängnis verbringen.
- Wenn A gesteht und B schweigt, wird A freigelassen und B kommt für vier Jahre ins Gefängnis

---

<sup>92</sup> Nash-Gleichgewicht, in: BWL-Lexikon, 25.08.2021.

- Wenn A schweigt und B gesteht, wird A zu vier Jahren Gefängnis verurteilt und B freigelassen.
- Wenn sowohl A als auch B schweigen, verbüssen beide nur ein Jahr Gefängnis.<sup>93</sup>

Ein Gefangenendilemma ist eine Situation, in der einzelne Entscheidungsträger immer einen Anreiz haben, sich so zu entscheiden, dass das Ergebnis für die Einzelnen als Gruppe nicht optimal ist.

Die Zahlen in der obigen Beschreibung des Dilemmas werden jetzt verwendet, um die möglichen Ergebnisse der Auszahlung jedes Spielers unter verschiedenen Umständen darzustellen. Die Auszahlung eines Spielers ist in diesem Fall die Anzahl Jahre, die der Spieler durch seine Strategie erhält. Wir können die Probleme, mit denen sie konfrontiert sind, in einer Matrix ausdrücken, die die interaktive Art ihrer jeweiligen Entscheidungen erfasst; dies ist die strategische Form des Spiels:<sup>94</sup>

		Spieler A	
		Schweigen	Gestehen
Spieler B	Schweigen	1 / 1	4 / 0
	Gestehen	0 / 4	2 / 2

Abbildung 22: Auszahlungsmatrix beim Gefangenendilemma

Somit gibt es vier mögliche Auszahlungen für die jeweiligen Spieler:

Versuchung (V)	Gestehen, wenn der andere Spieler schweigt	Keine Haft
Entschädigung (E)	Schweigen, wenn der andere Spieler schweigt	Ein Jahr Haft
Bestrafung (B)	Gestehen, wenn der andere Spieler gesteht	Zwei Jahre Haft
Gutgläubigkeit (G)	Schweigen, wenn der andere Spieler gesteht	Vier Jahre Haft

Abbildung 23: Auszahlungsergebnisse beim Gefangenendilemma

Somit lässt sich das Problem auch mit folgender Matrix ausdrücken:

		Spieler A	
		Schweigen	Gestehen
Spieler B	Schweigen	E / E	V / G
	Gestehen	G / V	B / B

Abbildung 24: Vereinfachte Auszahlungsmatrix beim Gefangenendilemma

Daher hängt der Gewinn des Spielers nicht nur von seinen eigenen Entscheidungen ab, sondern auch von den Entscheidungen seines Komplizen. Im Allgemeinen ist es für beide Parteien objektiv vorteilhafter, zu schweigen. Wenn zwei Gefangene kooperieren, muss jede Person nur für ein Jahr ins Gefängnis. Der kombinierte Verlust der beiden beträgt zwei Jahre, und jede andere Kombination von Geständnis und Schweigen führt zu noch grösseren Verlusten. Persönlich erscheint es für beide Parteien vorteilhafter, vor Gericht auszusagen.<sup>95</sup>

Für einzelne Gefangene stellt sich die Situation wie folgt dar: Bekennt sich die andere Partei schuldig, wird die Strafe durch seine Aussage von vier auf zwei Jahre herabgesetzt; Aber wenn die andere Partei aufgrund seiner Aussage nichts sagt, kann er seine Strafe von einem Jahr auf kein Jahr Haft reduzieren!

<sup>93</sup> Prisoners's Dilemma, in: Wikipedia, 25.08.2021.

<sup>94</sup> Ross: Game Theory, 25.08.2021.

<sup>95</sup> Prisoner's Dilemma, in: Wikipedia, 25.08.2021.

Aus persönlicher Sicht ist ein Geständnis definitiv eine empfehlenswerte Strategie. Diese Aussage hängt nicht vom Verhalten der anderen Partei ab, und es scheint immer vorteilhafter zu sein, offen zu sein. Diese Strategie, die nichts mit Gegnern zu tun hat, wird in der Spieltheorie als dominante Strategie bezeichnet.

Das Dilemma beruht darauf, dass kollektive und individuelle Analysen zu unterschiedlichen Handlungsempfehlungen führen.

Die Spielanlage behindert die Kommunikation und löst einseitigen Verrat aus, der Verräter erhofft sich dadurch bessere Ergebnisse für sich selbst, kein Jahr oder zwei Jahre statt vier. Wenn jedoch beide Häftlinge dieser Strategie folgen, wird sich ihre Situation – wie auch der Einzelne – verschlechtern, da jeder von ihnen zu vier Jahren statt zu zwei Jahren Gefängnis verurteilt wird.

Das Gefangenendilemma liegt im Zusammenbruch dieser möglichen Strategie. Die sogenannte rationale, schrittweise Analyse der Situation führte zu einem Geständnis beider Häftlinge, was zu schlechten Ergebnissen führte. Bessere Ergebnisse können durch Zusammenarbeit erzielt werden, aber die Zusammenarbeit wird leicht durch Vertrauen untergraben. Die rationalen Teilnehmer treffen sich, wenn die dominante Strategie zusammentrifft. Dies wird als Nash-Gleichgewicht bezeichnet. Das Paradoxe ist, dass keiner der Teilnehmer einen Grund hat, vom Nash-Gleichgewicht abzuweichen, obwohl bei dem Nash-Gleichgewicht keine weiteren Veränderungen dazu führen könnte, dass es ein Spieler sich einen Vorteil verschafft, ohne dass es dem anderen Spieler dafür einen Nachteil gibt.<sup>96</sup>

## 4.2 Quanten Spieltheorie

Die Quanten Spieltheorie ist eine Erweiterung der klassischen Spieltheorie, bei der die Gesetze der Quantenphysik angewendet werden. Folgende Prinzipien der Quantenmechanik sind auch in den Quanten Spieltheorie von überragender Bedeutung:<sup>97</sup>

*Überlagerung (Superposition):* Die Fähigkeit von Quantenobjekten, an zwei Orten gleichzeitig zu sein.

*Verschränkung (Entanglement):* Das Phänomen, dass entfernte Komponenten eines Quantensystems Korrelationen aufweisen, die nicht erklärt werden können. Dabei wird durch das Beeinflussen einer Komponente, der Zustand der anderen Komponente auch beeinflusst.

*Kollabieren (Collapse):* Das Phänomen, dass die Quantenzustände eines Systems auf klassische Zustände reduziert werden. Zusammenbrüche treten auf, wenn wir den Quantenkreislauf messen und dadurch das System reduzieren.

---

<sup>96</sup> Gefangenendilemma, in: Wikipedia, 25.08.2021.

<sup>97</sup> Sagole; Anurit; Bikash; Prasanta: Quantum Tic-Tac-Toe, 01.09.2021.

## 5 Anwendung der Quanten Spieltheorie

In dem letzten Kapitel wurde die Spieltheorie, wie aber auch die Quanten Spieltheorie.

### 5.1 Coin Flip

Der Coin Flip ist ein Zweispieler Spiel, bei dem man eine Münze, die eine *Kopf* Seite und eine *Zahl* Seite besitzt, abwechselungsweise dreht. Die Münze legt man dann unter eine Schachtel, bei der man nicht sieht, welche Seite der Münze oben liegt. Die Münze wird zuerst mit der *Kopf* Seite nach oben unter die Schachtel gelegt. Dieser Anfangszustand ist beiden Spielern bekannt. Ab jetzt wechseln sich die beiden Spieler ab, wobei Spieler A anfängt, die Münze zu drehen oder sie so zu lassen, wie sie gerade unter der Schachtel liegt, ohne dabei selbst den Zustand der Münze sehen zu können. Spieler A gewinnt das Spiel, falls sich die Münze bei dem Beobachten des Endzustandes mit der *Kopf* Seite oben befindet. Die Reihenfolge, in der die Spieler die Münze drehen bzw. nicht drehen ist folgende: Spieler A, Spieler B, Spieler A. Somit hat Spieler A zwei Züge:  $A_1$  und  $A_2$  und Spieler B nur einen Zug:  $B_1$ .

Dies ist ein strategisches Zwei-Personen-Nullsummenspiel, welches anhand der Auszahlungsmatrix analysiert werden kann:

		Spieler A			
		$A_1: \bullet$ $A_2: \bullet$	$A_1: \bullet$ $A_2: \curvearrowright$	$A_1: \curvearrowright$ $A_2: \bullet$	$A_1: \curvearrowright$ $A_2: \curvearrowright$
Spieler B	$B_1: \bullet$	A gewinnt	B gewinnt	B gewinnt	A gewinnt
	$B_1: \curvearrowright$	B gewinnt	A gewinnt	A gewinnt	B gewinnt
	Gewinnchance	50%	50%	50%	50%

Abbildung 25: Auszahlungsmatrix beim Coin Flip

Dabei indiziert der Punkt  $\bullet$ , dass man die Münze nicht dreht und der Pfeil  $\curvearrowright$ , dass die Münze vom Spieler gedreht wird. Unter dieser Abbildungsmatrix ist die Gewinnchance dargestellt. Wie man sieht, ist die Gewinnchance für Spieler A, wie auch Spieler B für jede der Strategien 50%. Es macht also keinen Unterschied, welche Strategie der jeweilige Spieler auswählt.

Man kann den Ablauf des Spieles folgendermassen als Graph darstellen:

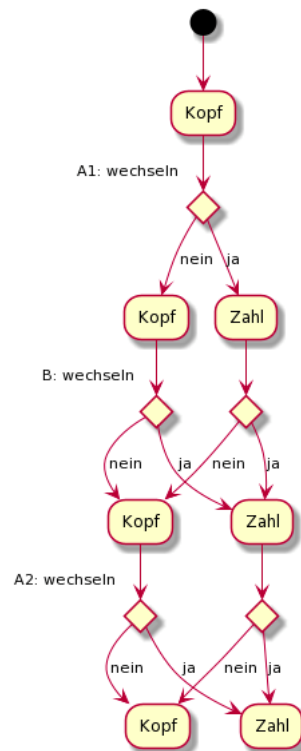


Abbildung 26: Ablauf des Coin Flip

Ein Beispiel eines Spieles: Spieler A dreht die Münze bei seinem Zug und dreht ihn auch bei seinem zweiten. Spieler B hingegen dreht die Münze bei seinem ersten Zug. Somit ist der Zustand der Münze: *Zahl, Kopf, Zahl* somit gewinnt Spieler B.<sup>98,99</sup>

### 5.1.1 Anwendung von Quanten Strategien

Wie bei der klassischen Variante ist auch der Quanten Coin Flip ein Zweispieler Spiel, bei dem eine Münze mit einer *Kopf* und einer *Zahl* Seite abwechselungsweise gedreht wird. Der Startzustand, sowie die Schachtel, die Reihenfolge der Spieler und wie oft jeder Spieler die Münze drehen darf, werden aus der klassischen Variante übernommen. Der einzige Unterschied zum klassischen Coin Flip ist, dass Spieler A Quanten Strategien auf das Spiel anwenden kann. Spieler B hingegen wendet weiterhin eine klassische gemischte, probabilistische Strategie an, bei der er die Münze mit der Wahrscheinlichkeit  $p$  umdreht.

Es gibt noch einen weiteren Unterschied, der beim Untersuchen des Spieles festgestellt wird. Nämlich ist der Quanten Coin Flip kein Zwei-Personen-Nullsummenspiel. Spieler A hat in diesem Spiel eine Strategie, mit dem er eine 100-prozentige Gewinnchance hat. Welche Strategie es ist und wie die 100-prozentige Gewinnchance entsteht wird im Folgenden untersucht.

Der Quanten Coin Flip wird mithilfe von Qubits simuliert. Dabei werden die Veränderungen der Münze, also die Drehung, keine Drehung und der Quantenzug durch Gates abgebildet, welche das Qubit verändern.

<sup>98</sup> Piotrowski: Quantum game theory, 22.09.2021.

<sup>99</sup> Meyer: Quantum strategies, 22.09.2021.

Die Münze nimmt einen einzelnen Zustand eines Qubits

$$|\psi\rangle = [\alpha\beta]|\psi\rangle = [\alpha\beta]$$

an, bei dem

$$\alpha, \beta \in \mathbb{C}$$

wobei

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

gilt.

Spieler B hat mit seiner klassisch gemischten Strategie genau zwei Möglichkeiten die Münze zu beeinflussen: Die Münze zu drehen oder den Zustand der Münze nicht zu ändern. Wenn Spieler B die Münze umdreht, bzw. „flippt“ wird auf die Münze ein Bit-Flip Operator, nämlich das X-Gate angewendet, welche die Matrix

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

besitzt.

Dabei ist der *Kopf* Zustand der Münze definiert als

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

und der Zustand *Zahl* ist definiert als

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

Im Gegensatz zu Spieler B, kann Spieler A Quanten Strategien anwenden, welche beinhalten, dass Spieler A auch einen Quantenzug („Quantum move“) anwenden kann. Somit kann Spieler A zusätzlich noch zu den Operationen, die Spieler B ausführen kann, das Hadamard-Gate anwenden, welches die Matrix

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

besitzt.



Nun können wir wie in der klassischen Variante auch in der Quanten Variante alle Züge in einer Auszahlungsmatrix analysieren. Noch zu dieser Abbildungsmatrix hinzu kommt die Abbildungsmatrix aus dem vorherigen Abschnitt zum klassischen Coin Flip, da Spieler A die Möglichkeit hat auch die Münze zu drehen oder den Zustand der Münze zu lassen. Aus Übersichtlichkeitsgründen wurden diese reinen klassischen Züge nicht erneut aufgenommen. Daneben sind folgende neue Züge möglich:
















		Spieler A				
Spieler B		$A_1$ :  $A_2$ : 	$A_1$ :  $A_2$ : 	$A_1$ :  $A_2$ : 	$A_1$ :  $A_2$ : 	$A_1$ :  $A_2$ : 
	$B_1$ : 	50/50	50/50	50/50	50/50	A gewinnt
	$B_1$ : 	50/50	50/50	50/50	50/50	A gewinnt
	Gewinnchance	50%	50%	50%	50%	100% für A

Abbildung 27: Auszahlungsmatrix beim Quanten Coin Flip

Dabei indiziert der Punkt , dass man die Münze nicht dreht, der Pfeil , dass die Münze vom Spieler gedreht wird und die Kugel  einen Quantenzug.

Man kann den Ablauf des Spieles folgendermassen darstellen:

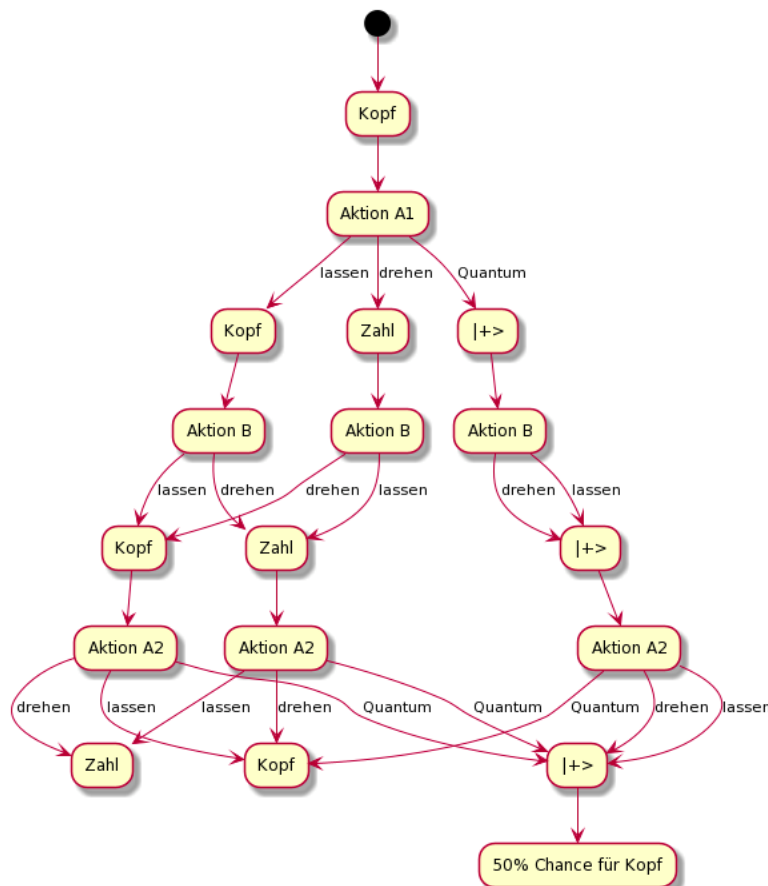


Abbildung 28: Ablauf des Quanten Coin Flip

Man sieht also, dass es somit **kein** Zwei-Personen-Nullsummenspiel mehr ist, im Gegensatz dazu, als man die klassische Spieltheorie auf das Spiel angewendet haben. Der Spieler A hat bei der Strategie,

auf die Münze zwei Mal einen Quantenzug anzuwenden, eine 100-prozentige Gewinnchance. Dies beruht auf der Tatsache, dass Spieler A Quanten Strategien und Spieler B nur klassische gemischte Strategie anwenden kann.

Die Strategie, die Spieler A anwenden kann, um jedes Mal gewinnen zu können, ist folgende:

Am Anfang wird die Münze auf den Startzustand  $|0\rangle$ , also *Kopf* gesetzt. Danach kann Spieler A das Hadamard Gatter auf die Münze anwenden und die Münze in den Zustand

$$|+\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

versetzen. Nun ist Spieler B am Zug. Spieler B hat nun die Möglichkeit die Münze zu drehen oder den Zustand der Münze beizubehalten. Falls sich Spieler B dazu entscheidet, die Münze zu drehen, hat die Münze den Zustand:

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

Sie hat also genau den gleichen Zustand wie vorher. Dies ist der Fall, da das X-Gatter die Matrix flippt. Dies hat aber keine Auswirkungen auf den Zustand der Münze.

Zum Schluss kann Spieler A noch einmal einen Zug machen. In diesem Zug wendet Spieler A wieder das Hadamard Gatter auf die Münze an, welchen die Münze in den Zustand

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

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

zurückversetzt. Somit hat die Münze nun wieder den Zustand *Kopf*, welches auch das Ziel von Spieler A ist. Aus diesem Grund kann Spieler A in allen Fällen, auch wenn Spieler B die Entscheidung trifft, die Münze nicht zu drehen, gewinnen.

### 5.1.2 Prototyp Coin Flip

Der Prototyp des Spieles wurde mit JavaScript programmiert und wurde in der HTML Webseite `coin.html` dargestellt. Um das Spiel programmieren zu können, wurde zuerst eine JavaScript-Library hinzugefügt, welche ermöglicht Quantenbewegungen programmieren zu können. Diese JavaScript-Library heisst `quantum-javascript.js`. Die Erklärung der Library und dessen Funktionen findet sich unter <sup>100</sup>. Mit dieser Library ist es möglich, die im Abschnitt 3.2 eingeführten sowie weitere Quantengatter auf einzelne Qubits zu programmieren. Die Library wurde in die Datei `main.js` eingebettet. In dieser Datei wurde auch der Code für die Funktion des Coin Flips geschrieben.

Damit man versteht, wie sich die Wahrscheinlichkeit zu gewinnen während des Spiels verhält, wurden die verschiedenen Zustände der Bloch Sphäre fotografiert und eingefügt. Diese Bilder wurden als

<sup>100</sup> Quantum Circuit Simulator, in: npmjs, 03.11.2021.

bloch img auf der HTML Webseite eingefügt. Das Layout der HTML Webseite wurde mit einem dem `main.css` Datei stilisiert.

Die einzelnen Dateien sind wie folgt verknüpft:

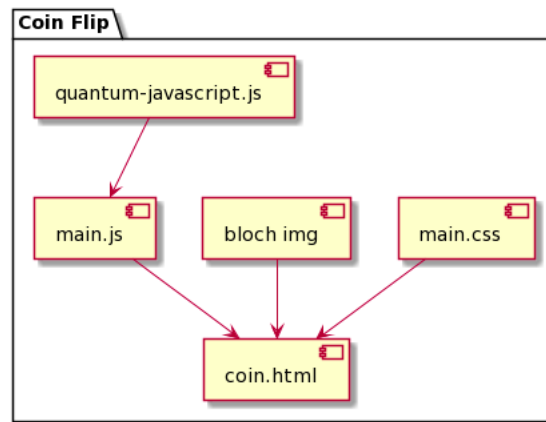


Abbildung 29: Komponenten des Coin Flip Prototyp

Wie vorher beschrieben ist der Code, der für das Funktionieren des Coin Flips essenziell ist, in der Datei `main.js`. Dieser Code implementiert folgende Funktion:

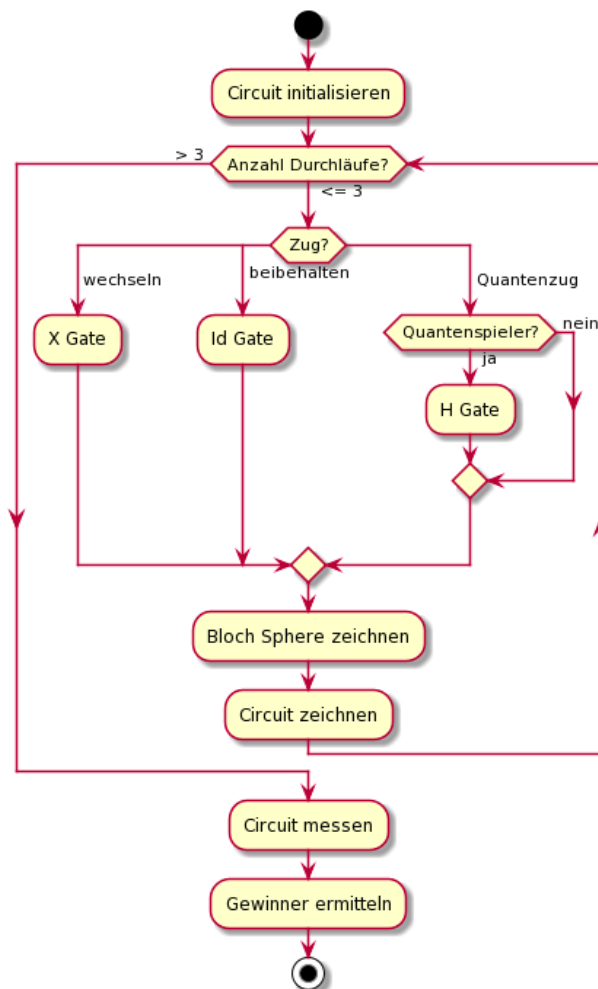


Abbildung 30: Flowchart des Programms des Coin Flip Prototyp

Der Quanten Circuit wird durch folgende Befehle initialisiert:

```
var circuit = new QuantumCircuit(1);  
circuit.createCreg("c", 1);
```

Die Funktion `new QuantumCircuit(1);` initialisiert einen neuen Quanten Circuit mit einem Qubit. Die Funktion `circuit.createCreg("c", 1);` erstellt ein klassisches Register, welches später benutzt wird, um den Quanten Circuit zu messen.

Ein X-Gatter wird folgendermassen erstellt:

```
circuit.appendGate("x", 0, {});  
change_image(1);
```

Durch den Befehl `circuit.appendGate("x", 0, {});` wird ein X-Gatter auf dem Qubit erstellt und durch den Befehl `change_image(1);` wird das Bild der Bloch Sphäre verändert.

Ein H-Gatter wird analog erstellt:

```
circuit.appendGate("h", 0, {});  
change_image(2);
```

Wichtig ist hier noch zu bemerken, dass das H-Gatter, also das Hadamard-Gatter, nur von Spieler 1, dem Spieler, dem Quantenzüge erlaubt sind, zugänglich ist. Dies ist im Programm mit einem Zähler umgesetzt (`counter % 2 == 0`), sodass nur für alle Züge, die einen geraden Counter haben, ein Quantenzug erlaubt ist. Die anderen Züge, also das Drehen der Münze, welches durch ein X-Gatter definiert wurde, und das nicht Drehen der Münze, welches durch das ID-Gatter definiert wird, sind für beide Spieler zugänglich.

Ein ID-Gatter wird analog erstellt:

```
circuit.appendGate("id", 0, {});
```

Die verschiedenen Zustände der Bloch Sphäre wurden schon vorher als Grafiken aufgenommen und in einem eigenen Ordner gespeichert. Dabei gibt es genau vier verschiedene Zustände der Bloch Sphäre, welche unter `1.png` bis `4.png` gespeichert sind. Dabei ist die Kopf Seite nach oben der Anfangszustand der Münze. Dieser Anfangszustand muss mit dem Anfangszustand der Bloch Sphäre korrespondieren, nämlich  $|0\rangle$ . Durch den Code

```
bloch_state = 1;
```

wurde der Anfangszustand der Bloch Sphäre gleich eins gesetzt. Dieser beinhaltet folgenden Zustand der Bloch Sphäre:



Abbildung 31: Zustand  $|0\rangle$  der Bloch Sphere

Die weiteren Zustände der Bloch Sphäre, also  $|1\rangle$ ,  $|+\rangle$  und  $|-\rangle$  werden gleich 2, 3 und 4 gesetzt. Wenn nun ein Spieler den Zustand der Münze und somit den Zustand der Bloch Sphäre verändert, wird das Bild, das den neuen Zustand repräsentiert, folgendermassen abgebildet:

```
function change_image(value) {  
    if (value == 1) {
```

```

    if ((bloch_state % 2) == 0) {
        // wenn gerade => Superposition => keine Anpassung
    }

    else {

        bloch_state += 2; // sonst spiegeln

    }

}

```

Im oberen Code wird gezeigt, dass -falls der Wert der Bloch Sphäre eins ist- geprüft wird, ob der `bloch_state` ohne Rest durch zwei teilbar ist (modulo zwei). Ist dies der Fall, gibt es keine Anpassung, da sich das Qubit in der Superposition befindet. Wenn das nicht der Fall ist, wird zu `bloch_state` zwei addiert, damit der Zustand der Bloch Sphäre gespiegelt wird.

```

    else if (value == 2) {

        if ((bloch_state % 2) == 0) {
            // wenn gerade => Superposition => zurückdrehen

            bloch_state += 1; // zurückdrehen

        }

        else {

            bloch_state -= 1; // sonst um spiegeln

        }

    }

}

```

Auch wenn der Wert der Bloch Sphäre zwei ist, wird der `bloch_state` überprüft, ob dieser durch zwei ohne Rest teilbar ist (modulo zwei). Ist dies der Fall, wird die Bloch Sphäre zurückgedreht. Wenn das aber nicht der Fall ist, wird von `bloch_state` eins subtrahiert, damit der Zustand der Bloch Sphäre gespiegelt wird.

```

    if (bloch_state <1) bloch_state = 4;

    if (bloch_state >4) bloch_state = 1;

```

Dadurch, dass man zu `bloch_state` Beträge summiert, bzw. Beträge davon subtrahiert, kann der Wert von `bloch_state` grösser als 4 bzw. kleiner als 1 werden. Ist dies der Fall, wird `bloch_state` nach dem Wert vier auf den Wert eins gesetzt und nach dem Wert eins auf den Wert vier gesetzt.

Im letzten Schritt wird das Bild in der HTML Website (`<img id="bloch" />`) durch folgenden Code ersetzt:

```

    var name = "img/"+bloch_state+".png";

    document.getElementById("bloch").src=name;

```

Falls drei Züge von den Spielern gespielt wurde, wird das Spiel beendet. Die Funktion `function finish()` beendet das Spiel in dem die Variable `game_over`, welche vor dem Beginn des Spieles definiert wurde und auf falsch gesetzt wurde, auf wahr gesetzt wird.

Jetzt wird das Qubit durch:

```
circuit.addMeasure(0, "c", 0);
```

auf das am Anfang definierte klassische Register abgebildet und somit gemessen. Der Quanten Circuit wird durch den Code:

```
circuit.run();
```

ausgeführt.

Die Variable `result`, welche am Anfang des Spieles definiert wurde, wird durch:

```
if (circuit.getCregValue("c") == 0) {  
    result = "Kopf"; // das Ergebnis "Kopf"  
}
```

auf das Ergebnis Kopf gesetzt, falls die Messung des Circuits das Resultat Bit = 0 ist. Ansonsten wird das Ergebnis auf Zahl gesetzt:

```
else {  
    result = "Zahl"; // das Ergebnis "Zahl"  
}
```

Die Ergebnisse werden wiederum in der HTML Website angepasst:

```
document.getElementById("output").innerHTML = result;  
  
document.getElementById("probability").innerHTML =  
circuit.probability(0);  
  
document.getElementById("drawing").innerHTML =  
circuit.exportSVG(true);
```

Dabei gibt `circuit.probability(0)`; die Wahrscheinlichkeit für Zahl, also  $|1\rangle$  an.  
`circuit.exportSVG(true)`; exportiert eine grafische Darstellung des Quanten Circuits mit allen Gattern.

### 5.1.3 Resultat des Coin Flip

Durch die Anwendung von Quanten Strategien auf das Zwei-Personen-Nullsummenspiel ergibt sich ein Zwei-Personenspiel, welches einem Spieler eine höhere Gewinnchance als dem anderen Spieler ermöglicht. Der Grund liegt darin, dass dieser Spieler eine unterschiedliche Strategie verwendet als der andere Spieler. Dieser Spieler hat über alle Spielvarianten gesehen im Gegensatz zu dem anderen Spieler neu eine 55.56-prozentige Gewinnchance und mit einer Strategie eine 100-prozentige Gewinnchance. Somit ist durch die Anwendung der Quanten Strategien die Gewinnchance von Spieler A um 5.56 Prozent gestiegen. Die theoretische Erklärung in Abschnitt 5.1.1 konnte gezeigt und mit dem Prototyp veranschaulicht und erlebbar gemacht werden.

## 5.2 Quanten Tic-Tac-Toe

Eine weitere Anwendung der Quanten Spieltheorie ist das Quanten Tic-Tac-Toe. Dieses wurde 2002 von Allan Goff et.al in <sup>101</sup> als ein Einstieg in die Quantenmechanik ohne die Mathematik vorgeschlagen. Dabei ist es gleich aufgebaut, wie das klassische Tic-Tac-Toe, welches schon den meisten Personen aus

---

<sup>101</sup> Goff, A; Lehmann, D; Siegel, J: Quantum tic-tac-toe, 27.10.2021.

ihrer Kindheit bekannt sein sollte. Das klassische Tic-Tac-Toe wird auf einem 3x3 Gitter mit festen Raum- und Zeitelementen mit eindeutigen Symbolen gespielt, wobei man als Symbole meistent «X» und «O» verwendet. Jedoch bemerkt man nach einigem spielen, dass man bei dem klassischen Tic-Tac-Toe mit den jeweilig besten Zügen der einzelnen Spieler immer auf ein Unentschieden kommt, welches das Spiel nicht mehr interessant macht.

#### 5.2.1.1 Das klassische Tic-Tac-Toe

Das klassische Tic-Tac-Toe startet damit, dass ein Spieler ein Kästchen mit seiner Markierung («X») markiert. Darauf markiert der andere Spieler ein anderes beliebiges Kästchen mit seiner Markierung («O»). Diese Schritte werden anschliessend alternierend fortgesetzt, bis einer der Spieler drei seiner Markierungen in einer Zeile, Spalte oder Diagonale des Gitters hat. Der Spieler, der diese Bedingung zuerst erfüllt, ist der Gewinner. Für den Fall, dass keiner der Spieler die Gewinnbedingung erfüllt und alle Felder markiert sind, gilt das Spiel als unentschieden.

Der Beweis, dass das klassische Tic-Tac-Toe bei optimaler Strategie immer auf ein Unentschieden endet, kann mithilfe der kombinatorischen Spieltheorie durchgeführt werden. Dabei werden alle Kombinationen aufgelistet. Der Start des Spielbaums sieht wie folgt aus:

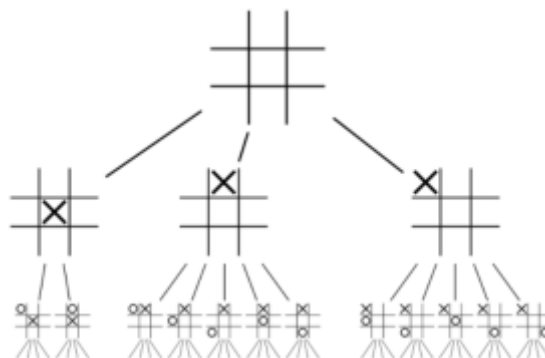


Abbildung 32: Spielbaum für Tic-Tac-Toe, *Intermathematics*, 29.09.2021

Dieser Spielbaum ist erweiterbar, damit man jeden möglichen Ausgang des Spieles abdecken kann. Wenn der Baum vollständig gezeichnet ist, ist es möglich für einen Spieler, den optimalen Zug bei einem bestimmten Zeitpunkt von einer beliebigen Position durch das Durcharbeiten des vollständigen Baumes herauszufinden.

Die Obergrenze für die Anzahl der Positionen und die Anzahl der verschiedenen Spiele ist durch  $3^9 = 19'683$  gegeben. Dies ist die Gesamtzahl von allen möglichen Spielpositionen in einem 3x3-Gitter. Dies ist der Fall, da jedes Kästchen in dem 3x3-Gitter entweder durch ein O oder durch ein X besetzt ist oder es ein leeres Kästchen ist.

Die Gesamtzahl der Möglichkeiten, wie die einzelnen Positionen des Rasters besetzt werden können wird durch  $9! = 362'880$  beschrieben. Dabei wird jedes X und O einzeln als unterschiedliches Symbol von anderen X und Os gezählt.

Die Anzahl der verschiedenen Kombinationen, die sich ergeben, wenn man das Gitter mit der maximalen Anzahl von X und O füllt wird durch  $\frac{9!}{5!(9-5)!} = 126$  beschrieben. Dies folgt daraus, da die maximale Anzahl von X fünf ist und die maximale Anzahl von O vier ist.

Die Erkenntnis, dass viele alternative Stellungen im Wesentlichen Spiegelungen oder Drehungen einer anderen sind, kann die Untersuchung dieses Spielbaums jedoch erheblich vereinfachen. Wir können feststellen, dass es nur drei mögliche Anfangszüge gibt, wenn wir nur nach eindeutigen Positionen suchen, wie in Abbildung 32 dargestellt.



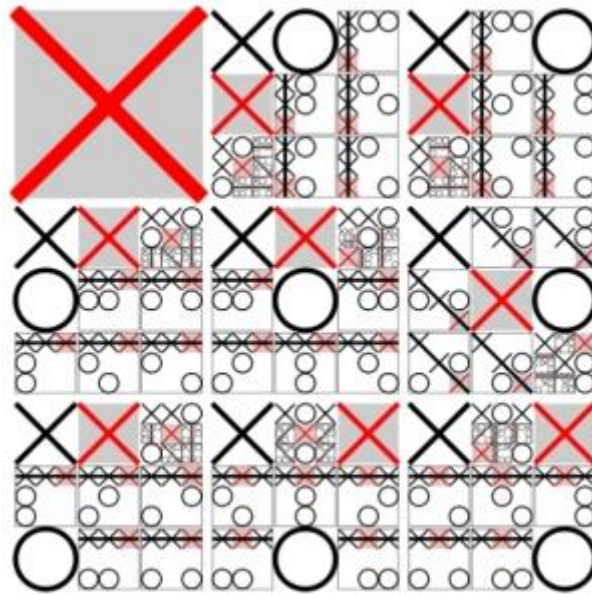


Abbildung 33: Antwort für das beste Ergebnis für X bei einem Tic-Tac-Toe-Spiel, Intermathematics, 27.09.2021

Abbildung 33, welche auf den ersten Blick erschreckend wirkt, verrät in Wirklichkeit die Antwort auf das Tic-Tac-Toe-Spiel. Wenn man diese erfolgreich benutzt, kann man immer das beste Ergebnis für X zu erhalten. Die Strategie funktioniert wie folgt:

- 1) Der beste Anfangszug wird durch das grosse rote X in der oberen linken Ecke angezeigt. Der Spieler, der X als Symbol hat, ist also derjenige, der den ersten Zug macht.
- 2) Danach beobachtet Spieler X die Aktionen des Gegners und wählt das Gitter mit dem grossen schwarzen O an der von ihm angegebenen Stelle.
- 3) In diesem neuen Raster erscheint ein riesiges rotes X, das den nächsten besten Zug anzeigt.
- 4) Danach bleibt Spieler X in seinem Teil des grösseren Gitters und wiederholt den Vorgang.<sup>102</sup>

### 5.2.2 Anwendung von Quanten Strategien

Das Quanten Tic-Tac-Toe wird auf demselben Feld gespielt, wie das klassische Tic-Tac-Toe von den Spielern X und O. Bei dem Quanten Tic-Tac-Toe wird im Gegensatz zum klassischen Tic-Tac-Toe die Quanten Spieltheorie angewendet, weshalb man bei Quanten Tic-Tac-Toe noch Eigenschaften wie die Superposition, das Verschränken und das Kollabieren hat. Dadurch wird das Quanten Tic-Tac-Toe auf der einen Seite anders und komplexer, auf der anderen Seite natürlich auch interessanter als das klassische Tic-Tac-Toe.

Es ist wichtig zu verstehen, dass es mittlerweile viele verschiedene Definitionen des Quanten Tic-Tac-Toe gibt. Die ursprüngliche Definition von Allan Goff, welche unter<sup>103</sup> gefunden werden kann, ist nicht für Quantengatter vorgesehen und soll lediglich die Phänomene der Superposition, des Verschränkens und des Kollabierens aufzeigen. Aus diesem Grund wurde diese Version in der Arbeit nicht weiterverfolgt.

Zwei weitere Definitionen von Quanten Tic-Tac-Toes stammen von Nagy & Nagy, welche unter<sup>104</sup> und<sup>105</sup> gefunden werden können. Beide Varianten wurden im Rahmen dieser Arbeit als Prototypen umgesetzt und analysiert. Im Zusammenhang mit dieser Arbeit und dem Ziel des Aufzeigens von Quanten Strategien haben diese Spielvarianten jedoch den entscheidenden Nachteil, dass sie primär auf

<sup>102</sup> Chambers: Tic Tac Toe, 29.09.2021.

<sup>103</sup> Goff, A; Lehmann, D; Siegel, J: Quantum tic-tac-toe, 27.10.2021.

<sup>104</sup> Nagy, M & Nagy, N: Quantum Tic-Tac-Toe, 27.10.2021.

<sup>105</sup> Sagole & Anurit & Bikash & Prasanta: Quantum Tic-Tac-Toe, 27.10.2021.

die Initialisierung durch Hadamard Gattern (H Gates) basieren. Dadurch wird ein erhöhter Zufallsfaktor eingeführt. Die Überlegenheit einer Quanten Strategie kann zwar in dem Ansatz von Nagy & Nagy gezeigt werden, tritt jedoch nicht deutlich hervor.

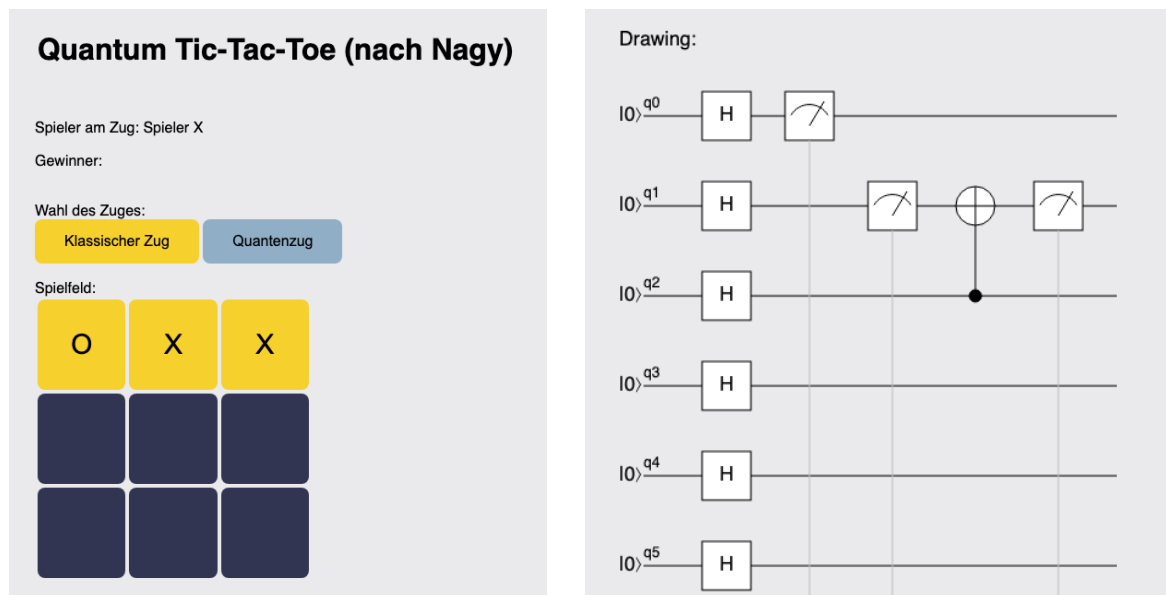


Abbildung 34: Bilder von dem Prototyp nach Nagy & Nagy

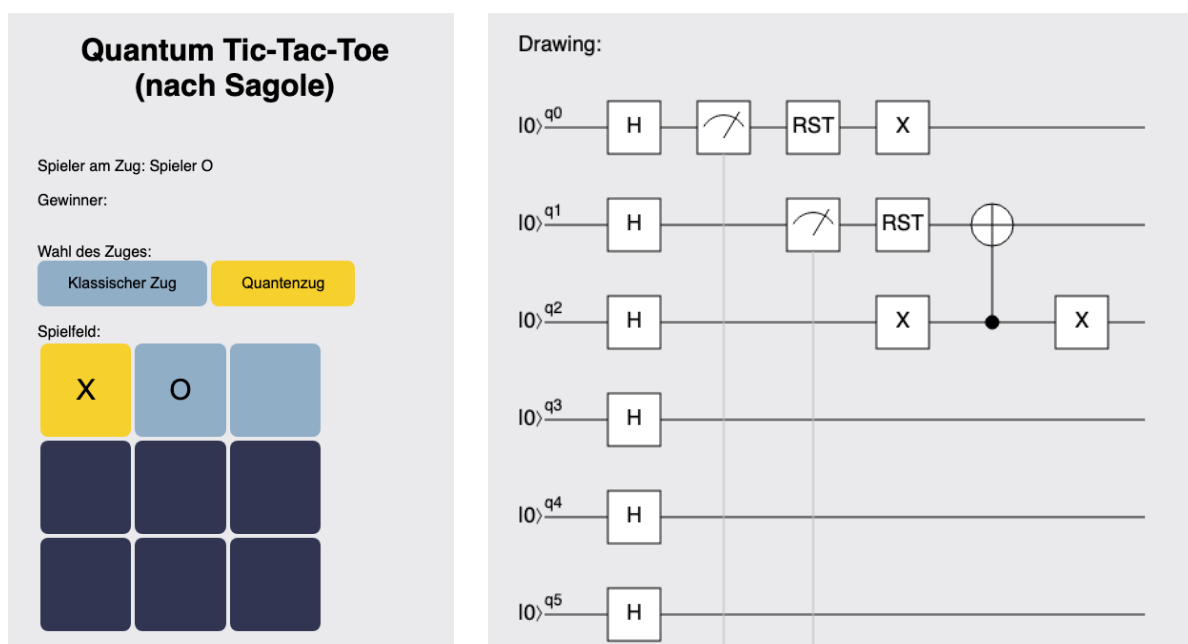


Abbildung 35: Bilder von dem Prototyp nach Sagole

Basierend auf den Erfahrungen aus der Analyse der beiden Spielvarianten aus der Literatur wurde ein «umfassender» Prototyp realisiert. Bei diesem wurden möglichst viele Wahlvarianten umgesetzt, so dass detaillierte Erfahrungen gesammelt werden konnten. Durch die Darstellung der Quanten Gatter konnten Spielzüge und Spiel Strategien ausprobiert werden.

Abbildung 36 zeigt Bilder von diesem Prototyp.

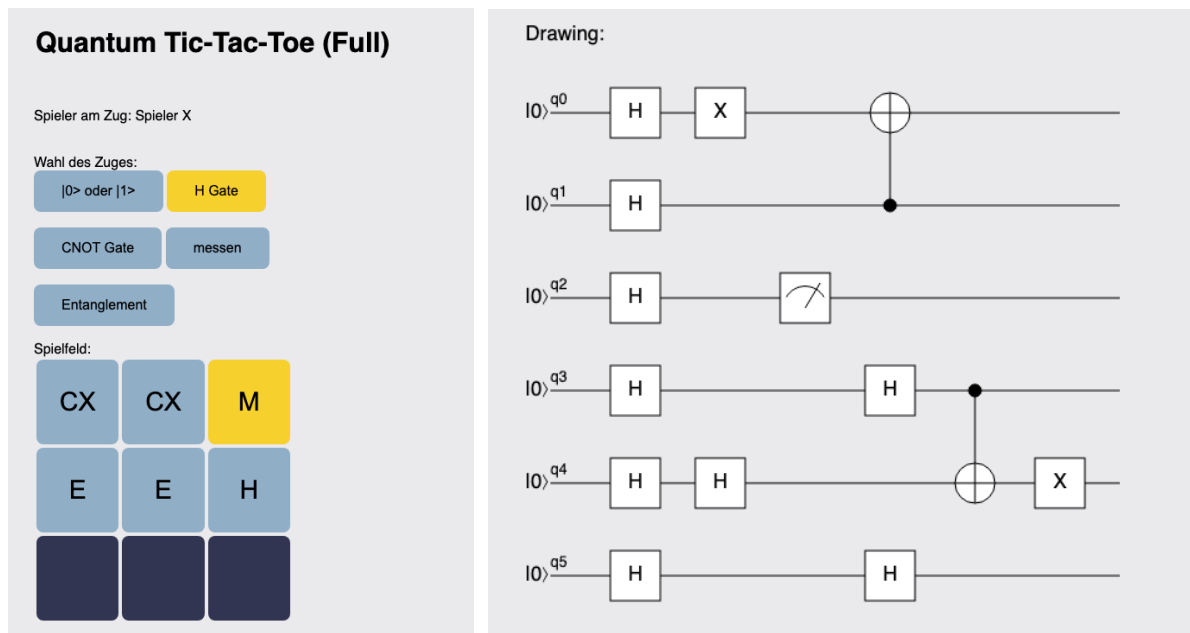


Abbildung 36: Bilder von dem Prototyp mit vollem Funktionsumfang

Es wird nicht weiter auf diesen Prototypen eingegangen, da basierend auf den gemachten Erfahrungen zwei eigene Definitionen der Regeln eingeführt wurden:

#### Variante 1: Quanten Spieler und klassischer Spieler

- I. **Quantengatter:** Das Spielfeld wird durch 9 Qubits repräsentiert. Jedes der Qubits ist mit dem Startzustand  $|0\rangle$  initialisiert.
- II. **Reihenfolge:** Die Spieler wechseln sich mit den Spielzügen ab, wobei X den ersten Zug macht.
- III. **Spielzug:** Spieler X ist der Quanten Spieler, der die Möglichkeit zu einem klassischen und einem Quanten Spielzug hat. Spieler O ist der klassische Spieler, welcher auf den klassischen Spielzug beschränkt ist. Unter einem klassischen Spielzug versteht sich der einzige Spielzug, welcher auch im klassischen Tic-Tac-Toe benutzt wird. Es gibt also einem Spieler die Möglichkeit, das Feld zu besetzen. Unter einem Quanten Spielzug versteht sich ein Spielzug, in welchem man Eigenschaften der Quanten Spieltheorie anwendet.
- IV. **Verschränkung:** Spielzüge können sich verschränken, indem man zwei Felder durch den Zug «Entanglement» verschränkt. Durch die Verschränkung wird ein noch unbesetztes Feld mit einem H Gatter in Superposition versetzt und anschliessend dieses Feld als Kontroll-Qubit eines CNOT Gatters benutzt (siehe Abschnitt 3.2.5). Das Ziel-Qubit des CNOT Gatters muss bereits bespielt worden sein (X oder O).
- V. **Messen der Qubits:** Die 9 Qubits werden erst gemessen und kollabieren in den klassischen Zustand, wenn alle Felder mindestens einmal bespielt wurden. Dies wurde gewählt, da durch die Verschränkung rückwirkend Veränderungen geschehen können. So kann ein vermeintlicher Gewinner rückwirkend durch den Einsatz der Quanten Logik noch verlieren.
- VI. **Gewinner:** Nach dem Messen der Qubits wird der Gewinner ermittelt. Wie beim klassischen Tic-Tac-Toe gilt das Erreichen von drei eigenen Symbolen in einer Reihe. Dabei spielt es keine Rolle, ob diese drei klassischen Zustände horizontal, vertikal oder diagonal in einer Reihe liegen. Es gibt vier Varianten:
  - a. Spieler X hat eine Reihe und Spieler O nicht: Damit hat X gewonnen.
  - b. Spieler O hat eine Reihe und Spieler X nicht: Damit hat O gewonnen.
  - c. Kein Spieler hat eine Reihe: Damit gilt ein unentschieden.
  - d. Beide Spieler haben eine Reihe: Damit gilt ebenfalls ein unentschieden.

## Variante 2: Zwei Quanten Spieler

- I. **Quantengatter:** Das Spielfeld wird durch 9 Qubits repräsentiert. Jedes der Qubits ist mittels X-Gatters mit dem Startzustand  $|1\rangle$  initialisiert.
- II. **Reihenfolge:** Die Spieler wechseln sich mit den Spielzügen ab, wobei X den ersten Zug macht.
- III. **Spielzug:** Spieler X und Spieler O sind Quanten Spieler, welche die Möglichkeit zu mehreren Quanten Spielzügen haben. Diese Spielzüge sind wie folgt:
  - a. **Wechseln:** Bei diesem Spielzug wird ein X-Gatter an das gewählte Qubit angehängt. Damit wechselt der Zustand, sofern es sich in einem Basiszustand befindet.
  - b. **Beibehalten:** Bei diesem Spielzug wird ein ID-Gatter an das gewählte Qubit angehängt. Damit behält es den Zustand, sofern es sich in einem Basiszustand befindet.
  - c. **Superposition:** Bei diesem Spielzug wird ein H-Gatter an das gewählte Qubit angehängt. Damit behält wird es in Superposition versetzt.
- IV. **Verschränkung:** Spielzüge können sich verschränken, indem man zwei Felder durch den Zug «Entanglement» verschränkt. Durch die Verschränkung wird ein Feld mit einem H Gatter in Superposition versetzt und anschliessend dieses Feld als Kontroll-Qubit eines CNOT Gatters benutzt (siehe Abschnitt 3.2.5). Das Ziel-Qubit des CNOT Gatters muss bereits bespielt worden sein.
- V. **Anzahl Operationen pro Qubits:** Es können maximal drei Züge (Quantengatter) pro Qubit gespielt werden.
- VI. **Messen der Qubits:** Die 9 Qubits werden erst gemessen und kollabieren in den klassischen Zustand, wenn alle Felder mindestens einmal bespielt wurden. Dies wurde gewählt, da durch die Verschränkung rückwirkend Veränderungen geschehen können. So kann ein vermeintlicher Gewinner rückwirkend durch den Einsatz der Quanten Logik noch verlieren.
- VII. **Gewinner:** Nach dem Messen der Qubits wird der Gewinner ermittelt. Wie beim klassischen Tic-Tac-Toe gilt das Erreichen von drei eigenen Symbolen in einer Reihe. Dabei spielt es keine Rolle, ob diese drei klassischen Zustände horizontal, vertikal oder diagonal in einer Reihe liegen. Es gibt vier Varianten:
  - a. Spieler X hat eine Reihe und Spieler O nicht: Damit hat X gewonnen.
  - b. Spieler O hat eine Reihe und Spieler X nicht: Damit hat O gewonnen.
  - c. Kein Spieler hat eine Reihe: Damit gilt ein unentschieden.
  - d. Beide Spieler haben eine Reihe: Damit gilt ebenfalls ein unentschieden.

In Abschnitt 5.2.4 werden die Auswirkungen dieser Regelvarianten auf mögliche Quanten Strategien analysiert.

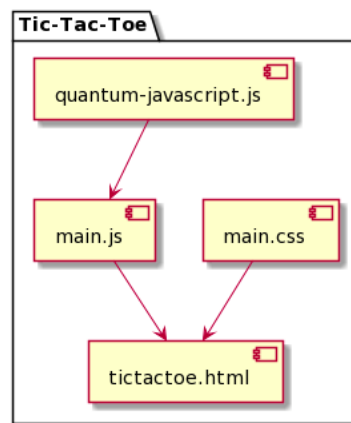
### 5.2.3 Prototyp Quanten Tic-Tac-Toe

Genau gleich wie bei dem Quanten Coin Flip wurde auch beim Quanten Tic-Tac-Toe der Prototyp des Spieles mit JavaScript programmiert und in einem HTML-Textdatei `coin.html` dargestellt. Es wurde auch die JavaScript-Library `quantum-javascript.js` hinzugefügt, damit es möglich ist, Quantenbewegungen programmieren zu können. Die Erklärung der Library und dessen Funktionen findet sich unter <sup>106</sup>. Die HTML Webseite wurde mit einem der `main.css` Datei stilisiert.

---

<sup>106</sup> Quantum Circuit Simulator, in: npmjs, 03.11.2021.

Die einzelnen Dateien sind somit wie folgt verknüpft:



*Abbildung 37: Komponenten des Tic-Tac-Toe Prototyp*

Wie schon vorher beschrieben ist der Code, der für das Funktionieren des Quanten Tic-Tac-Toe essenziell ist, in der Datei `main.js`. Dieser Code hat folgende Funktion:

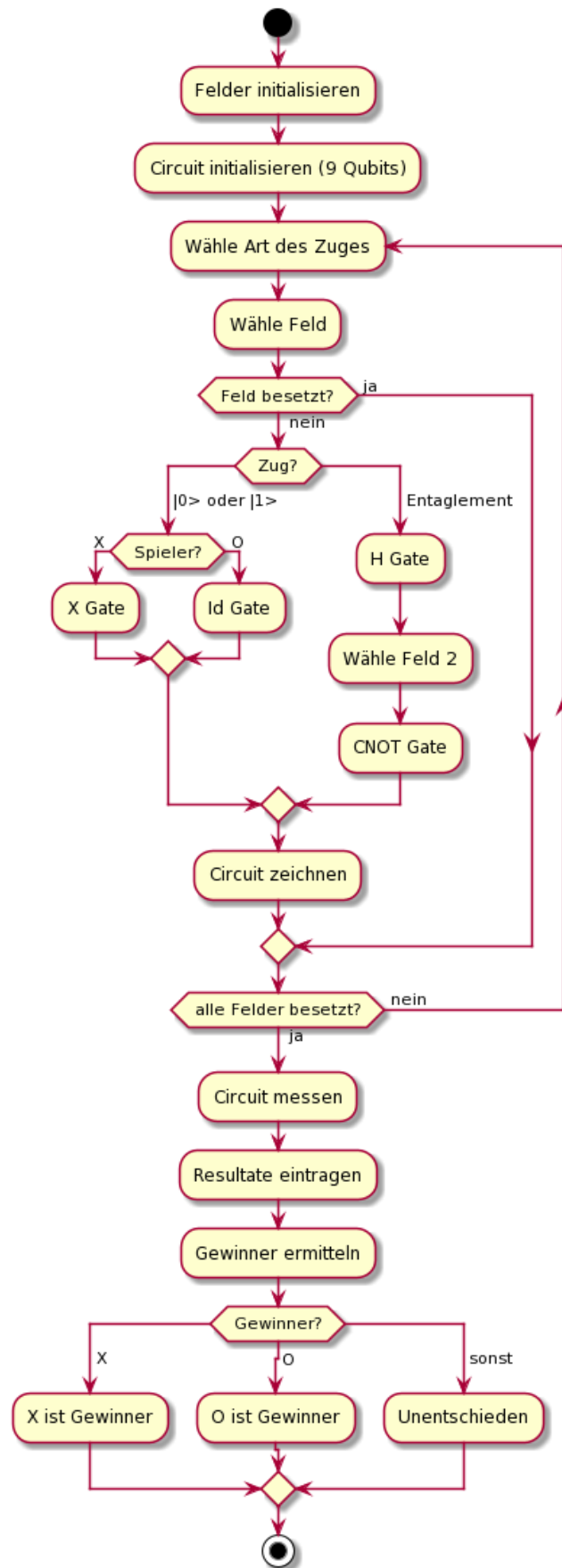


Abbildung 38: Flowchart des Programms des Tic-Tac-Toe Prototyp

Die Felder des Tic-Tac-Toes werden zu Beginn initialisiert. Dabei werden die Züge in dem Array `field_played[k]` und die Resultate in dem Array `field_result[k]` festgehalten. Der Quanten Circuit wird mit 9 Qubits, 1 Qubit für jedes Feld, initialisiert:

```
function init(){
    for (var k=0; k<9; k++) {
        field_played[k] = 0;
        field_result[k] = 0;
    }

    circuit = new QuantumCircuit(9);
    circuit.createCreg("c", 9);
}
```

Die Messmethode unterscheidet sich grundlegend von der des Coin Flips. Sie wird in beiden Varianten erst ausgeführt, wenn alle Felder mindestens einmal gespielt wurden. D.h. wenn `field_played[k] = 1` für alle  $0 \leq k < 9$ :

```
function measure() {
    var name = "";

    Der Circuit wird ebenfalls ausgeführt und gemessen.

    circuit.run();

    var measure = circuit.measureAll();

    Es werden alle 9 Felder durchlaufen und die Messergebnisse zugeordnet.
    1 = X und 0 = O.

    for (var k=0; k<9; k++) {
        name = "p" + (k+1);

        if (measure[k] == 1) {
            field_result[k] = 1;

            Der name entspricht der Id im HTML File und
            der Text wird angepasst.

            document.getElementById(name) .
            firstChild.data = "X";

        }

        else { // measure[k] == 0 !

            In diesem Fall wird der Wert -1 genutzt, um die Prüfung des
            Gewinners effizienter gestalten zu können (siehe unten)

            field_result[k] = -1;

            document.getElementById(name) .
            firstChild.data = "O";
        }
    }
}
```

```

    }

    }

    Nachdem alle Resultate in field_result[k] eingetragen wurden,
    wird der Gewinner ermittelt.

    check_winner();

}

```

Die Prüfung, ob es nach dem Messen einen Gewinner gibt, läuft folgendermassen ab:

```

function check_winner() {

    var x = false;

    var o = false;

    var a = field_result[0] + field_result[1] +
    field_result[2];

    var b = field_result[3] + field_result[4] +
    field_result[5];

    var c = field_result[6] + field_result[7] +
    field_result[8];

    var d = field_result[0] + field_result[3] +
    field_result[6];

    var e = field_result[1] + field_result[4] +
    field_result[7];

    var f = field_result[2] + field_result[5] +
    field_result[8];

    var g = field_result[0] + field_result[4] +
    field_result[8];

    var h = field_result[2] + field_result[4] +
    field_result[6];
}

```

Zuerst werden die beiden Variablen `x` und `o` auf `false` gesetzt. Die Resultate der Felder, die wenn horizontal, vertikal und diagonal durch dasselbe Symbol einen Gewinner gäben, werden addiert und werden als Variablen von `a` bis `h` definiert. Im nächsten Schritt wird überprüft, ob es mindestens in einem der Werte `a` bis `h` einen Gewinner gibt. Somit würde zum Beispiel die Besetzung der Qubits 0, 1 und 2 mit dem Symbol X, zum Sieg des Spielers X führen.



```

if (a == 3 || b == 3 || c == 3 || d == 3 || e == 3 || f
== 3 || g == 3 || h == 3) {
    x = true;
}

if (a == -3 || b == -3 || c == -3 || d == -3 || e == -3
|| f == -3 || g == -3 || h == -3) {
    o = true;
}

```

Falls einer dieser Variablen, also die Summer der einzelnen Felder, 3 gibt, wird die Variable `x` auf `true` gesetzt und falls einer dieser Variablen -3 ergibt, wird die Variable `o` auf `true`.

Somit kann durch:

```

if ( (x && o) || (!x && !o) ) {
    document.getElementById("winner").innerHTML =
    "Unentschieden";
}

else if (x) {
    document.getElementById("winner").innerHTML =
    "Spieler X";
}

else {
    document.getElementById("winner").innerHTML =
    "Spieler O";
}
}

```

überprüft werden, ob es einen Gewinner gibt. Erhält keiner der Spieler eine Reihe oder haben beide Spieler eine Reihe (in Variante 2 möglich), wird das Spiel als Unentschieden gewertet. Falls jedoch `x` true und `o` `false` ist, ist Spieler X der Sieger sowie umgekehrt, falls `o` true und `x` `false` ist, ist Spieler O der Sieger.

Das Resultat der HTML Website mit der beschriebenen Umsetzung des Codes sieht folgendermassen aus:

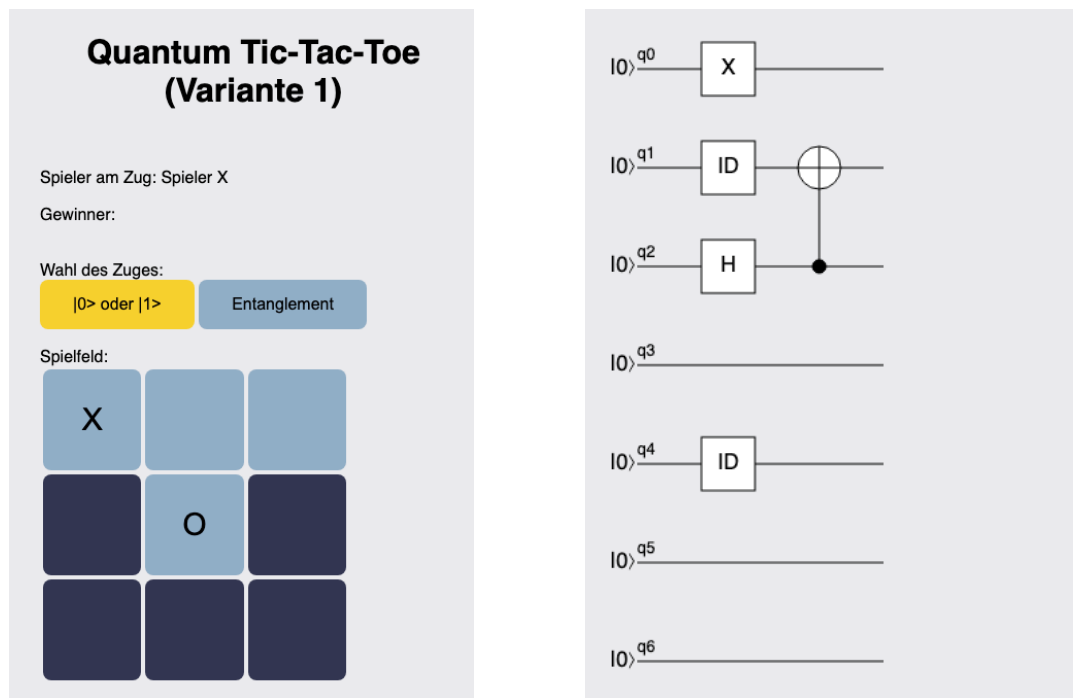


Abbildung 39: Bilder von dem Prototyp der Variante 1

Bei der zweiten Version des Quanten Tic-Tac-Toe ist der Code fast identisch, hat jedoch folgende Änderungen:

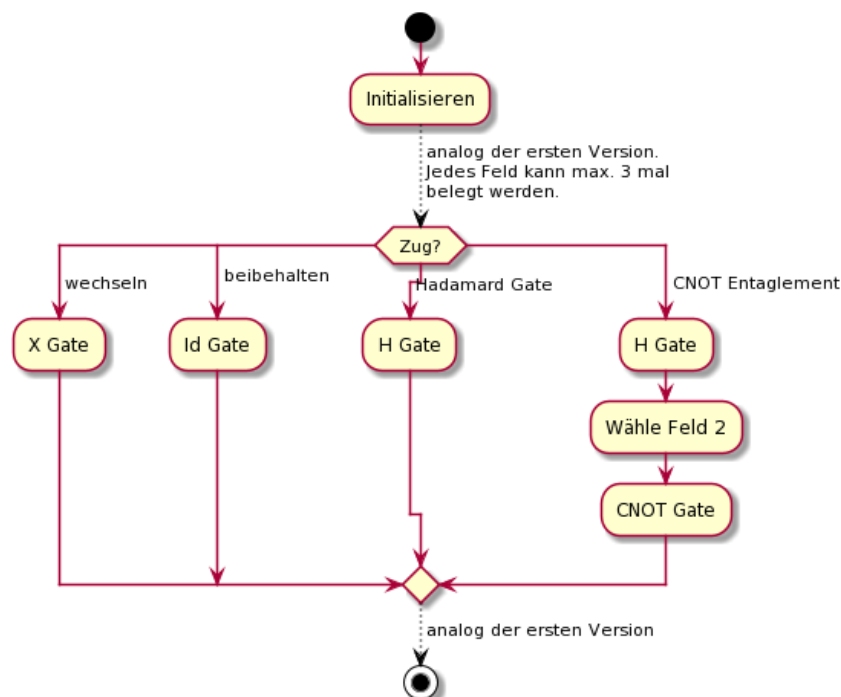


Abbildung 40: Flowchart der zweiten Version des Quanten Tic-Tac-Toe

Der Unterschied bei dieser Variante des Tic-Tac-Toes zur vorherigen wurde schon in den Regeln beschrieben und in dem Flowchart der Abbildung 40 noch einmal abgebildet.

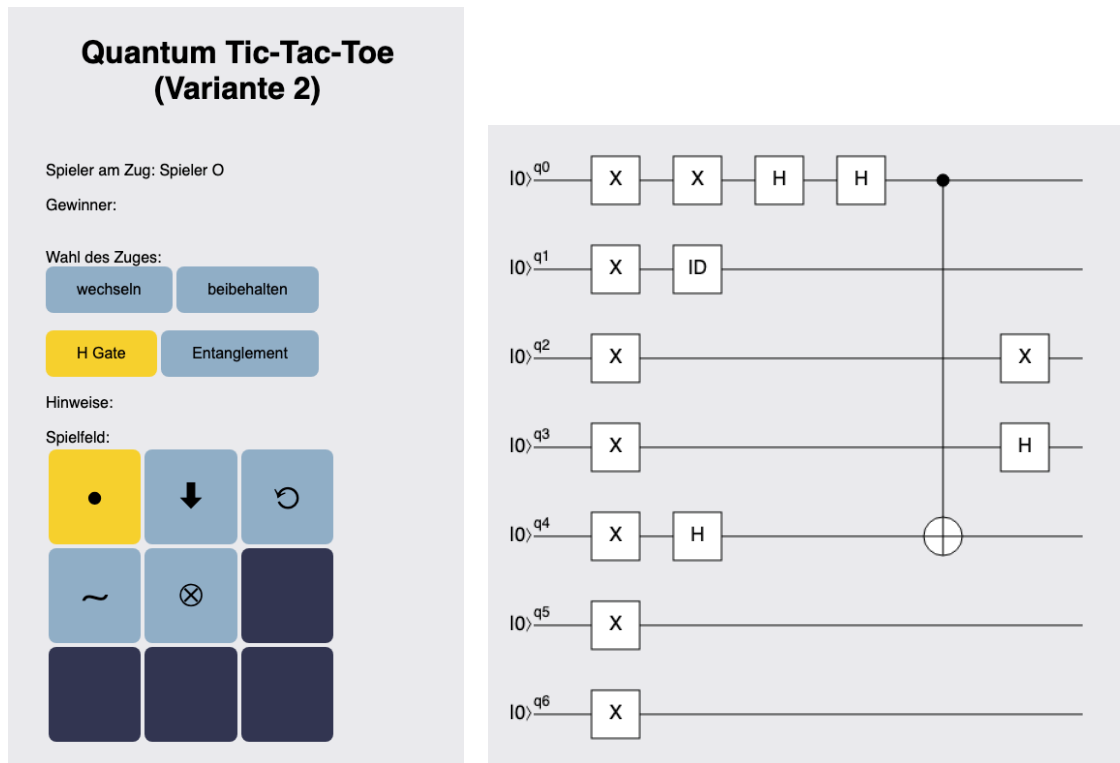


Abbildung 41: Bilder von dem Prototyp der Variante 2

Dabei wurden folgende Symbole verwendet:

- $\downarrow$ : Beibehalten (ID-Gatter)
- $\circlearrowleft$ : Wechseln (X-Gatter)
- $\sim$ : Superposition (Hadamard-Gatter)
- $\otimes$ : Entanglement (Zielgatter)
- $\bullet$ : Entanglement (Kontrollgatter)

## 5.2.4 Resultate der Experimente zum Quanten Tic-Tac-Toe

Beide Tic-Tac-Toe Varianten zeigen unterschiedliche Verhaltensmuster und ermöglichen unterschiedliche Spiel Strategien. Diese werden in den folgenden Abschnitten genauer analysiert.

### 5.2.4.1 Resultate Variante 1

Bei der ersten Variante der Umsetzung des Tic-Tac-Toe hat man, falls man als Spieler X optimal spielt, eine 50-prozentige Chance einen Sieg zu erzielen und eine 50-prozentige Chance auf ein Unentschieden. Dabei sind mit den optimalen Spielzügen die Spielzüge gemeint, welche Spieler X treffen muss, damit er mit einem Quantenzug sich diese Gewinnchancen erspielen kann. Ein derartiger Ablauf, bei dem Spieler X nur optimale Spielzüge trifft, wird nun geschildert:

Der erste Spielzug, welcher X machen muss, ist ein X-Symbol in das obere linke Kästchen zu setzten. Dies ist erzwungen, da es auch im klassischen Tic-Tac-Toe dem Spieler X die grösste Gewinnchance gibt. Die Gewinnstrategie für Spieler X ist in Abbildung 33 in Abschnitt 5.2.1.1 abgebildet.

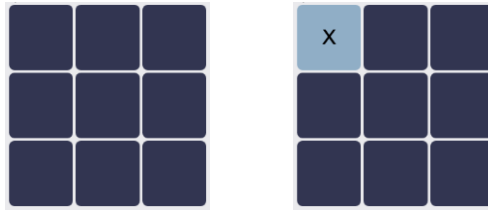


Abbildung 42: Leeres Spielfeld und erster Zug von Spieler X

Ab jetzt ist es egal, wohin Spieler O sein Symbol setzt. Zur Veranschaulichung wird eine Situation vorgespielt, bei der Spieler O sein Symbol in die Mitte setzt:

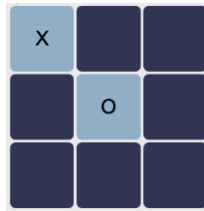


Abbildung 43: Erster Zug von Spieler O

Der nächste Zug von Spieler X ist es in dieser Situation, ein Entanglement zwischen dem unteren rechten Feld mit dem Feld in der Mitte zu errichten. Somit gibt bei diesen Feldern, wenn das ganze Spielfeld gemessen wird, eine 50-prozentige Wahrscheinlichkeit, dass der Zustand  $|1\rangle$  gemessen wird oder der Zustand  $|0\rangle$  gemessen wird:

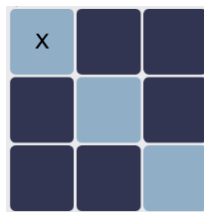


Abbildung 44: Zweiter Zug von Spieler X

Spieler O setzt nun sein nächstes Symbol in die Ecke unten links, worauf Spieler X mit seinem Symbol in der rechten oberen Ecke antwortet, um eine allfällige diagonale Verbindung der O Symbole nach dem Messen der Felder zu verhindern.

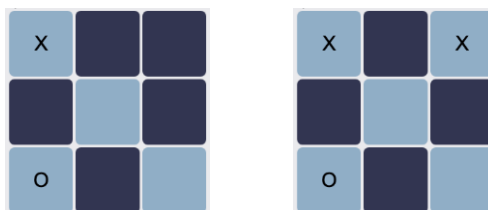


Abbildung 45: Nächste Züge von Spieler X und O

Spieler O wird nun die obere horizontale Verbindung der X Symbole verhindern und Spieler X noch einmal eine allfällige horizontale Verbindung der O Symbole in der unteren Reihe verhindern. Dies spielt keine Rolle mehr, da O es nicht schaffen würde, in diesem Fall eine horizontale, vertikale oder diagonale Verbindung herzustellen.

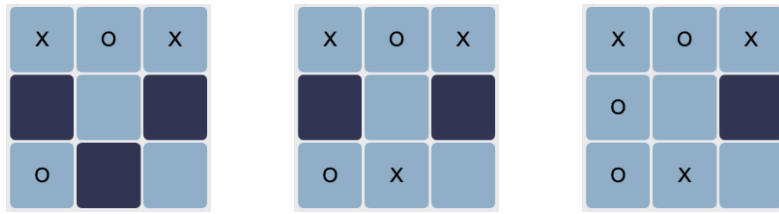


Abbildung 46: Nächsten Züge von Spieler X und O

Das letzte Symbol von X wird in das letzte verbleibende Feld gesetzt und dadurch wird das Spielfeld gemessen:

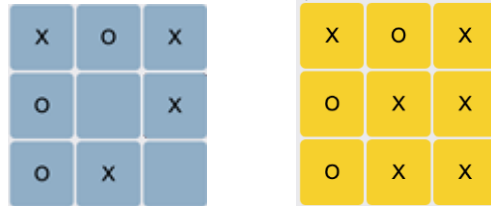


Abbildung 47: Letzter Zug von Spieler X und gemessenes Spielfeld mit dem gemessenen Zustand  $|1\rangle$

Die Felder, die ein Entanglement haben, wurden gemessen und es wurde der Zustand  $|1\rangle$  gemessen und Spieler X hat dadurch drei seiner Symbole in einer diagonalen Linie und somit gewonnen. Jedoch würde in der Hälfte der Fälle statt dem Zustand  $|1\rangle$  der Zustand  $|0\rangle$  gemessen werden:

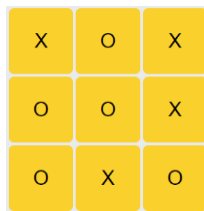


Abbildung 48: Letzter Zug von Spieler X und gemessenes Spielfeld mit dem gemessenen Zustand  $|0\rangle$

In dieser Situation würde es auf ein Unentschieden herauslaufen. Somit würde Spieler X in 50% der Fälle gewinnen und in 50% der Fälle ein Unentschieden erspielen, je nach dem welches Resultat nach dem Messen herauskommt. Spieler X würde mit dieser Strategie also **nie** verlieren, sondern nur ein Unentschieden oder einen Sieg erspielen. Dabei würde die Quantenschaltung bei dieser Spielsituation so aussehen:

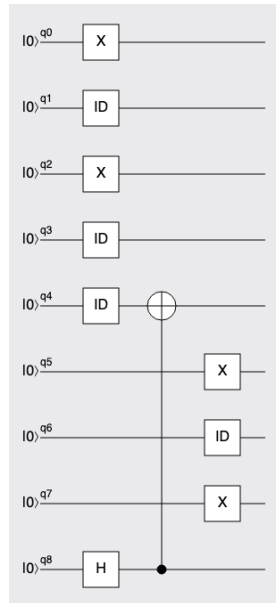


Abbildung 49: Quantenschaltung des Tic-Tac-Toe Spiels

Allerdings ist O nicht gezwungen sein erstes Symbol in die Mitte zu setzen. Somit könnte O sein erstes Symbol zum Beispiel auch in die Mitte links setzen. Jedoch könnte Spieler X auch in dieser Situation gewinnen oder ein Unentschieden erspielen, in dem er folgende Züge spielt:

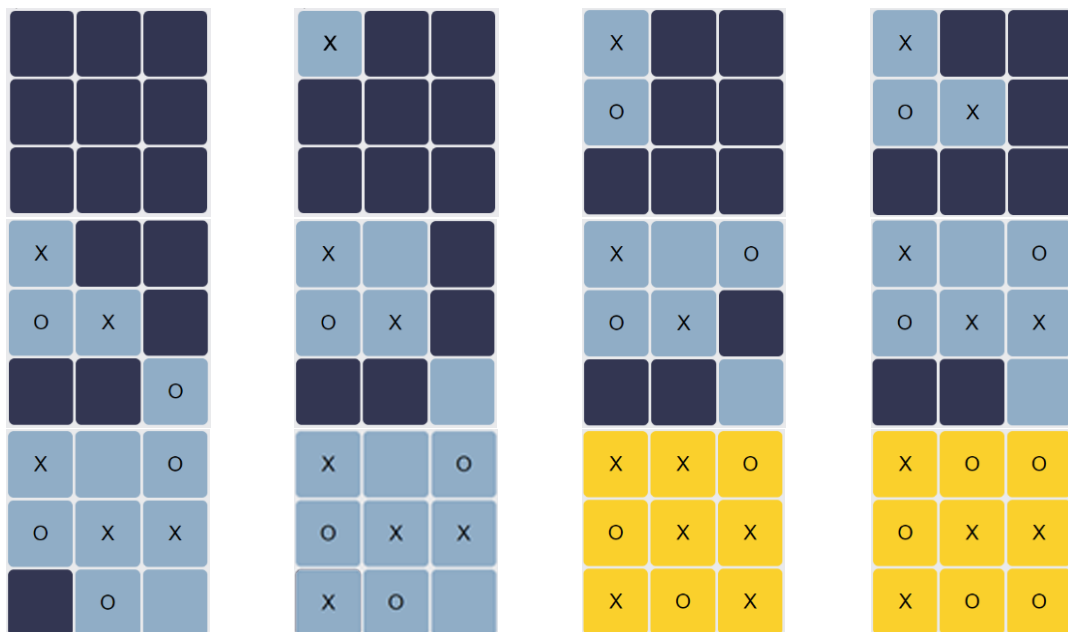


Abbildung 50: Anderer Spielverlauf mit den optimalen Zügen für Spieler X

Es konnte anhand der Analyse gezeigt werden, dass Spieler X mit seiner Quanten Strategie einen eindeutigen Vorteil vor dem Spieler O mit seiner klassisch gemischten Strategie hat. Wie beim Coin Flip kann gezeigt werden, dass Spieler, welche die Quanten Strategie anwenden, einen signifikanten Vorteil erhalten gegenüber den Spielern, welche eine klassisch gemischte Strategie anwenden.

### 5.2.4.2 Resultate Variante 2

Die Variante 2 bietet deutlich komplexere Spielzüge an. Durch die Gleichstellung der beiden Spieler (beide können Quantenzüge durchführen), wird eine Ausgeglichenheit erzeugt. Die Qubits werden zwar mit den Werten des Spielers X vorbelegt (auf den Basiszustand des Qubits  $|0\rangle$  wird ein X-Gatter gesetzt, aus dem somit der Zustand  $|1\rangle$  wird), durch die maximal drei Züge wird dies jedoch wieder ausgeglichen.

Die Resultate der Basiszüge «wechseln» und «beibehalten» können wie folgt beschrieben werden:



ergibt  $|0\rangle = O$  für 100% aller Fälle.



ergibt  $|1\rangle = X$  für 100% aller Fälle.



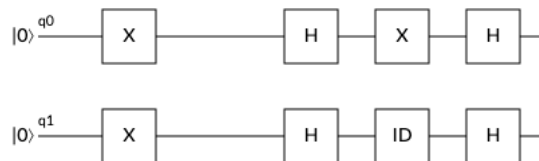
ergibt  $|0\rangle = O$  für 100% aller Fälle.

Das H-Gatter erzeugt den Zufallswurf:



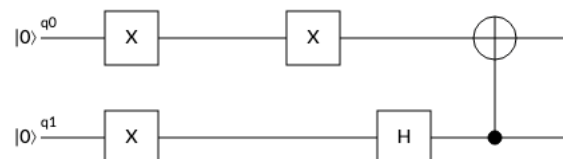
ergibt  $|1\rangle = X$  oder  $|0\rangle = O$  mit jeweils 50% Wahrscheinlichkeit.

Analog zum Coin Flip eliminieren zwei H-Gatter dazwischenliegende Werte:



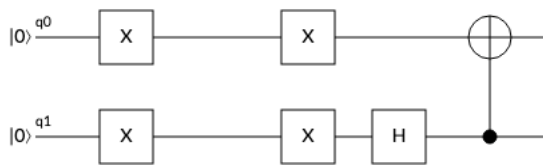
ergibt  $|1\rangle = X$  für 100% aller Fälle.

Interessant wird eine Zugkombination mit einem Bell State (Entanglement):



ergibt  $|11\rangle = XX$  oder  $|00\rangle = OO$  mit jeweils 50% Wahrscheinlichkeit.

Dabei ist es unerheblich, ob vor dem H-Gatter ein weiteres X-Gatter vorhanden ist:



ergibt  $|11\rangle = XX$  oder  $|00\rangle = OO$  mit jeweils 50% Wahrscheinlichkeit.

Durch Entfernen des X-Gatters wird diese Logik umgekehrt:



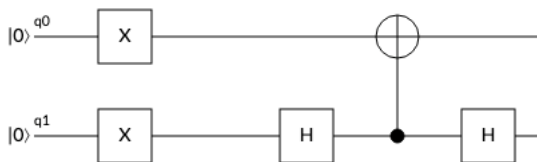
ergibt  $|01\rangle = OX$  oder  $|10\rangle = XO$  mit jeweils 50% Wahrscheinlichkeit.

Fügt man in beiden Qubits hinter dem Bell State H-Gatter an, so wird das gleiche Ergebnis erzielt:



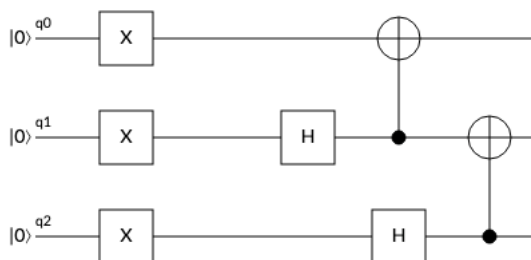
ergibt  $|01\rangle = OX$  oder  $|10\rangle = XO$  mit jeweils 50% Wahrscheinlichkeit.

In der folgenden Kombination (ohne H-Gatter in q0) wird jedoch wieder die Gleichverteilung erzielt:



ergibt  $|00\rangle = OO$ ,  $|01\rangle = OX$ ,  $|10\rangle = XO$  oder  $|11\rangle = XO$  mit 25% Wahrscheinlichkeit.

Eine spannende Kombination ist der verknüpfte Bell State, welcher einer für Tic-Tac-Toe relevante Konstellation erzeugt:

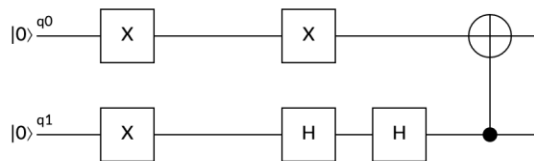


ergibt  $|001\rangle = OOX$ ,  $|010\rangle = OXO$ ,  $|100\rangle = XOO$  oder  $|111\rangle = XXX$  mit 25% Wahrscheinlichkeit.



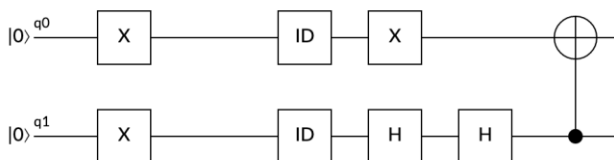
Dies bedeutet, dass in dieser Konstellation X zu 25% gewinnt und nie verlieren kann. Werden vorher die X durch zusätzliche X-Gatter umgekehrt, so gilt dies umgekehrt für O.

Noch extremer wird das Ergebnis, wenn folgende Konstellation erzeugt wird:



ergibt  $|11\rangle = XX$  mit 100% Wahrscheinlichkeit.

Das bedeutet, dass die Kombination H-Gatter, X-Gatter und Bell State zu einer 100% Chance für X führt, wenn danach kein Gatter durch O mehr eingebaut wird, oder wenn zuvor noch Identity Gatter verwendet wurde:



Zur Verdeutlichung dieser Einzelstrategien kann ein konkreter Spielverlauf wie folgt aussehen:

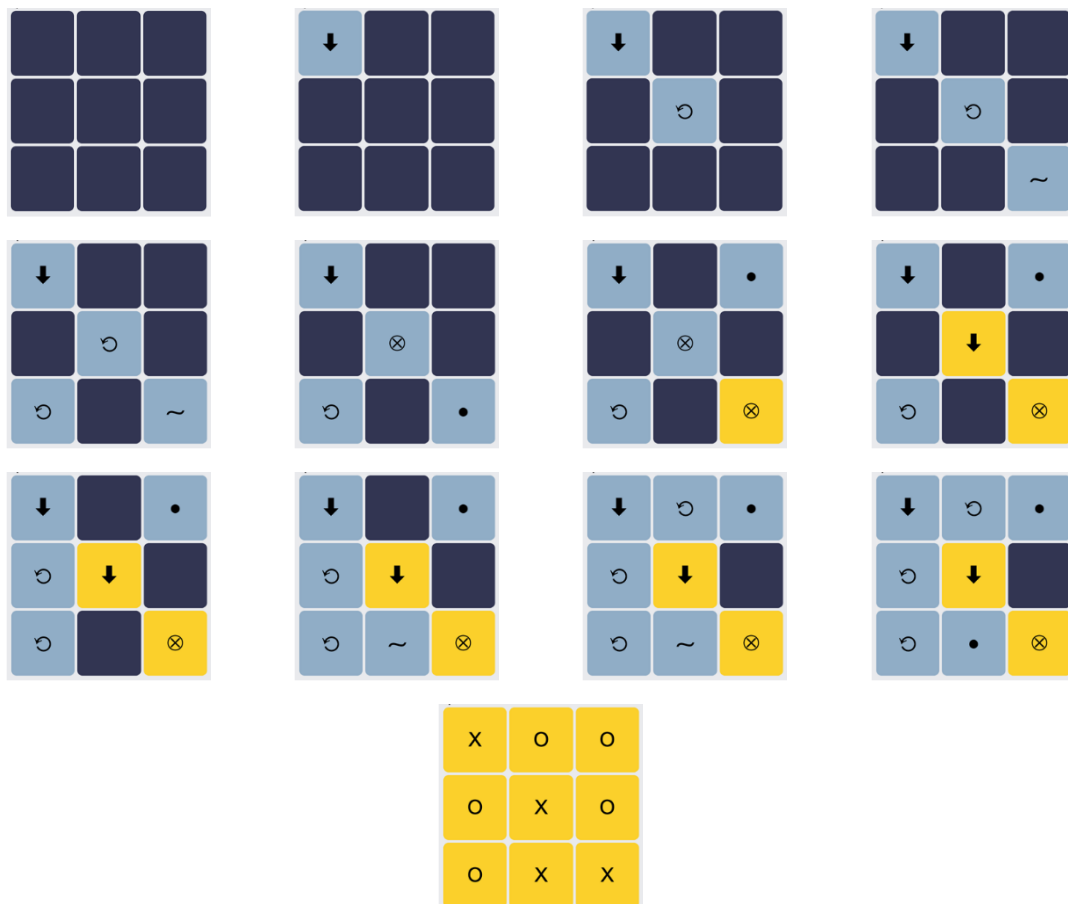


Abbildung 51: Spielverlauf der Variante 2 des Quanten Tic-Tac-Toe

Dabei wurden folgende Symbole benutzt:

- $\downarrow$ : Beibehalten (ID-Gatter)
- $\odot$ : Wechseln (X-Gatter)
- $\sim$ : Superposition (Hadamard-Gatter)
- $\otimes$ : Entanglement (Zielgatter)
- $\bullet$ : Entanglement (Kontrollgatter)

Spieler X gewinnt auf Grund des strategischen Entanglements auf der Diagonale, welche ihm eine 100% Chance für das mittlere und eine 50% Chance für das rechtsuntere Feld gibt. Gleichzeitig konnte Spieler X die Situation ausnutzen, dass er im letzten Zug über den Zeitpunkt der Messung entscheiden konnte.

Die resultierende Quantenschaltung sieht folgendermassen aus:

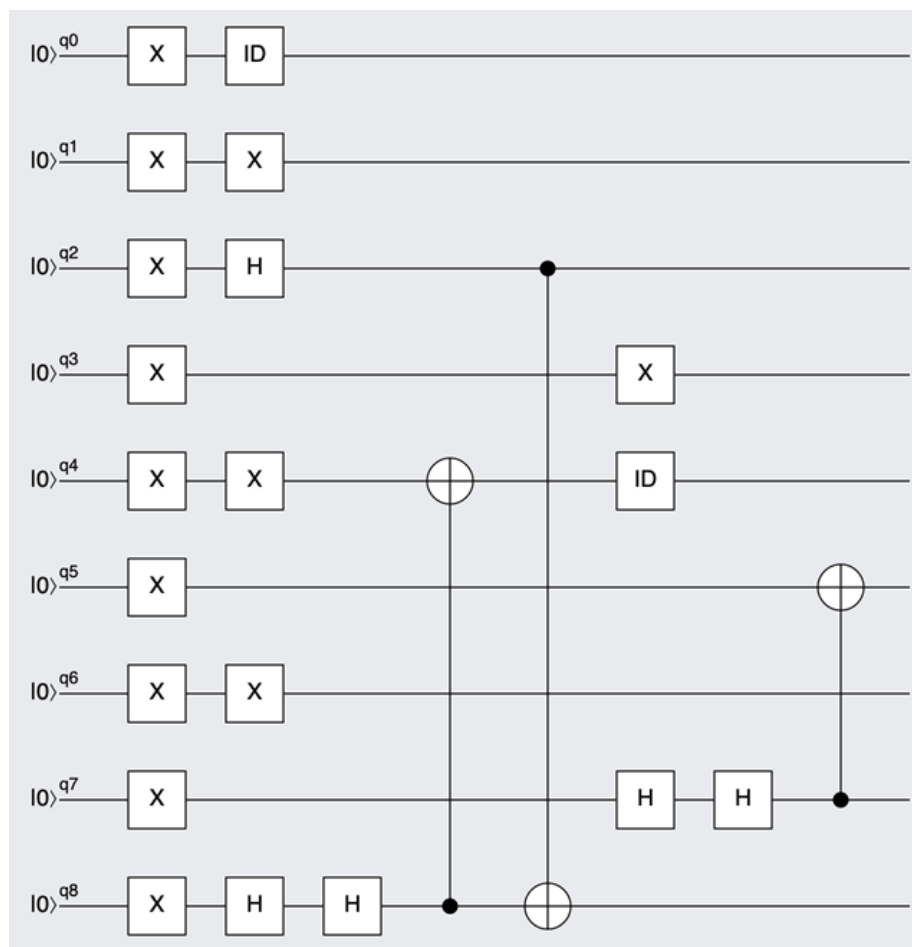


Abbildung 52: Quantenschaltung des obigen Spielverlaufs der Variante 2 des Quanten Tic-Tac-Toe

## 6 Zusammenfassung und Ausblick

Das Ziel der Arbeit war es, die Unterschiede der Logik in der Quantenphysik und der klassischen Physik zu zeigen und diese anhand von verschiedenen Beispielspielen zu verdeutlichen. Weiterhin war es in meinem Interesse dem Leser die Grundlagen der Quantenphysik und der zugrundeliegenden Mathematik näherzubringen. Somit wurde der Leserin und dem Leser der Einstieg in die Quantencomputer ermöglicht und gleichzeitig die quantenphysikalischen Hintergründe vermittelt.

Quantenmechanik an sich ist schon in vielen anderen Büchern sehr detailliert erklärt. In dieser Arbeit habe ich diese Grundlagen, welche für Quantencomputer essenziell sind, komprimiert aufgezeigt und in einem Kapitel erklärt. Somit wurde die Basis für die darauf aufbauenden Arbeiten geschaffen. Mit diesen Grundlagen wurde ein sehr wichtiges Bauteil eines Quantencomputers erklärt: Das Qubit. Das Qubit, auch Quantenbit genannt, wurde als kleinste Informationseinheit von Quantencomputern, welches ein quantenmechanisches System ist, beschrieben. Es wurde auch erklärt, dass ein Qubit zwei verschiedene Zustände,  $|0\rangle$  und  $|1\rangle$ , annehmen kann. Eine Operation auf einem Qubit in Superposition wird sowohl auf dem Zustand  $|0\rangle$  als auch auf dem Zustand  $|1\rangle$  ausgeführt. Zudem wurde erklärt, dass der interne Zustand eines Qubits vor der Messung:

$$\alpha \cdot |0\rangle + \beta \cdot |1\rangle$$

annimmt, wobei  $\alpha$  und  $\beta$  komplexe Zahlen sind. Man kann aber den Zustand eines Qubits auch als Vektor beschreiben. Dabei wird der Zustand als Vektor in einem zweidimensionalen Vektorraum über zwei komplexen Zahlen betrachtet:  $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ .

Mit den Grundlagen der Qubits konnten die verschiedenen Realisierungsansätze der Quantencomputer erklärt werden. Dabei wurden zuerst die verschiedenen Qubits: Das supraleitende Qubit, das photonische Qubit und das Qubit, basierend auf der Ionenfalle vorgestellt. Mit diesen verschiedenen Qubits lassen sich zwei verschiedene Arten von Quantencomputern bauen: Der Quantengattercomputer und die Quanten-Annealing-Maschine. Der Quantengattercomputer verwendet elementare Logik-Bausteine, sogenannte «Quantengatter», in jedem Rechenschritt, welche von komplexen Algorithmen ausführen können. Die zweite Art von Quantencomputern, die Quanten-Annealing-Maschine, ist auf einem ganz anderen System aufgebaut: Quanten-Annealing. Quanten-Annealing ist eine Klasse von algorithmischen Methoden, die einem natürlicherweise Lösungen mit niedrigen Energien liefern. Somit sind einige Anwendungsgebiete der Quanten-Annealing-Maschinen sind Optimierungsprobleme, Anwendungen, die das echte Energieminimum erfordern, und probabilistische Strichprobenprobleme.

Da ich im Rahmen dieser Maturaarbeit den Fokus auf die Quantengattercomputer und deren Programmierung gelegt habe, wurde auch die Programmierung der Quantengattercomputer aufgezeigt. Beim Aufzeigen der verschiedenen Quantengatter, mit denen man Quantengattercomputer programmiert, ist mir aufgefallen, dass es viele Quantengatter gibt, welche ähnlich zu Logikgatter sind, wie zum Beispiel das CNOT-Gatter zum NOT-Gatter. Jedoch gibt es unzählige Gatter, welche komplett neu sind, wie zum Beispiel, das Hadamard-Gatter, welches ein Qubit in eine Superposition versetzt.

Nachdem ich die Programmierung von Quantengattercomputern veranschaulicht habe, hatte ich meinen Fokus auf die Unterschiede der Logik in der klassischen Physik und der Quantenmechanik gelegt. Diese Theorie stellt die Grundlagen für neue Spiele dar. Um die Spieltheorie und den späteren Coin Flip gut erklären zu können, hatte ich das Nash-Gleichgewicht eingeführt und dieses Gleichgewicht anhand von dem Gefangenenden Dilemma erklärt. Zudem habe ich die Quanten Spieltheorie eingeführt, zum besseren Verständnis der Unterschiede der Logik der Spieltheorie und der Quanten Spieltheorie.

Um die Prinzipien der Unterschiede der Spieltheorie und Quanten Spieltheorie deutlich darzustellen zu können, habe ich diese anhand von zwei Spielen veranschaulicht: Dem Coin Flip und dem klassischen Tic-Tac-Toe. Mit dem Beispiel des Coin Flip konnte ich zeigen, dass ein Quantenspieler einen

entscheidenden Vorteil und mit einer gewissen Strategie eine 100-prozentige Gewinnchance hat. Dies konnte ich zeigen, indem ich zuerst den klassischen Coin Flip anhand einer Auszahlungsmatrix analysiert habe, wie ich es auch schon bei dem Gefangenen Dilemma getan habe. Anschliessend habe ich den Quanten Coin Flip anhand einer Auszahlungsmatrix analysiert und eine Strategie entdeckt, die dem Quantenspieler eine 100-prozentige Gewinnchance garantiert. Abschliessend habe ich die einzelnen Quantenzüge in Gatter umgewandelt und bei diesen Gattern die Matrizen miteinander multipliziert, um die Aussage der Auszahlungsmatrix zu bestätigen und die spätere Programmierung des Coin Flips zu vereinfachen. Den Quanten Coin Flip habe ich mittels eines HTML und JavaScript Prototypen umgesetzt und so eine reale Experimentierumgebung geschaffen.

Mit der Anwendung von Quanten Strategien auf das Spiel Tic-Tac-Toe konnte ich aufzeigen, dass auch in komplexeren Spielsituationen die Quanten Spieltheorie deutlich überlegen ist. Um dies verdeutlichen, hatte ich zuerst die Spielstrategie des klassischen Tic-Tac-Toe erklärt, damit ich ein generelles Verständnis zur bisherigen Spielstrategie darstellen konnte. Danach hatte ich die Überlegenheit von Quanten Strategien in den Ansätzen von Nagy & Nagy und Sagole gezeigt, welche jedoch nicht deutlich hervortraten. Somit habe ich einen umfassenderen Prototyp gebaut und diesen als Stützpunkt für meine Überlegungen verwendet. Aus diesem Prototyp habe ich zwei verschiedene Arten von Quanten Tic-Tac-Toe entwickelt: Eine Variante (1), in der ein Quantenspieler gegen einen normalen Spieler spielt und eine Variante (2), in der zwei Quantenspieler gegeneinander spielen. Für diese Variante habe ich Spielregeln definiert und deren Prototypen gebaut. Dabei wurden folgende Strategien und Erkenntnisse gewonnen: Bei der ersten Variante kann der Quantenspieler einen entscheidenden Vorteil erzielen, bei dem er eine Gewinnchance von 50% und eine Chance ein Unentschieden zu erzielen von 50% hat. Bei der zweiten Variante, kann durch Quanten Strategien die Komplexität des Spieles deutlich erhöht werden und dadurch der Spielspass auch erhöht werden kann.

Im Rahmen dieser Maturaarbeit konnte ich mich vertieft mit Quantenphysik, Quantenmechanik, Quantencomputern sowie deren Programmierung, klassischer Spieltheorie und Quanten Spieltheorie und Web-Programmierung auseinandersetzen. In allen Bereichen konnte ich mein Wissen fundamental verbessern. Die wichtigsten Erkenntnisse waren dabei für mich, dass die Programmierung von Quantencomputern sich komplett von der klassischen Programmierung unterscheidet und Qubits durch Manipulationen in Zustände überführt werden. Weiterhin verlangt es ein Umdenken, wenn logische Operationen in der Quantenmechanik genutzt werden. Faszinierend war die Tatsache, dass es in der Quantenmechanik Zustände und Logikabfolgen gibt welche darauffolgenden Aktionen komplett aushebeln. Die Entwicklung der Prototypen hat viel Spass gemacht und hat einen komplett neuen Blickwinkel auf die Quantenmechanik eröffnet. Meine zweite Version des Quanten Tic-Tac-Toe macht mir selbst immer noch viel Spass zu spielen und verlangt komplexe strategische Überlegungen.

Als möglichen Ausbau meiner Arbeit sehe ich folgende spannende Erweiterungen: Die weitere Untersuchung der zweiten Variante des Quanten Tic-Tac-Toes. Somit kann man bei dieser Variante zum Beispiel noch weitere Spielzüge suchen, bei dem ein Spieler nur gewinnen oder bzw. nicht verlieren kann. Zudem kann man die Unterschiede zwischen der klassischen Spieltheorie und der Quanten Spieltheorie auf weitere Spiele ausgebaut, wie zum Beispiel das Spiel Vier-Gewinnt. Dieses Spiel weist noch eine höhere Komplexität auf als das Tic-Tac-Toe und ist bereits bei der Anwendung der klassischen Spieltheorie sehr schwer zu meistern.

Quantencomputer stecken noch in ihren Kinderschuhen, jedoch glaube ich fest daran, dass mit der zunehmenden Entwicklung und Faszination dieser Maschinen, die Technologie von morgen sich von der von heute drastisch unterscheiden wird.

## 7 Anhang

### 7.1 Source Code des Coin Flip

#### 7.1.1 HTML

```
<html>
  <head>
    <title>Quantum Coin Flip</title>
    <link rel="stylesheet" href="css/main_Coin_01.css">
    <meta charset="utf-8">
  </head>

  <body>
    <h1>Quantum Coin Flip</h1> <br><br>
    Spieler am Zug: <span id="Player"></span> <br><br>
    Art des Zuges: <span id="Zug"></span> <br><br><br>
    Startzustand: <span id="start"></span> <br><br>
    <button id="Wechseln" onclick="select(1);">Wechseln</button>
    <button id="Beibehalten" onclick="select(0);">Beibehalten</button>
    <button id="Quantenzug" onclick="select(2);">Quantenzug</button> <br><br><br>
    Ergebnis: <span id="output"></span> <br><br>
    Wahrscheinlichkeit für Zahl: <span id="probability"></span> <br><br>
    Bloch Sphere:  <br><br>
    Quanten Circuit:<br> <span id="drawing"></span> <br><br><br>
    <button id="NewGame" onClick="refreshPage()">Neues Spiel</button>
  </body>

  <script src="js/quantum-circuit.min.js"></script>
  <script src="js/main.js"></script>
</html>
```

#### 7.1.2 JavaScript

```
var i = 3; // Anzahl Durchläufe
var counter = 0; //Neue Variable "Zähler" wird auf 0 gesetzt
var circuit = new QuantumCircuit(1); //Variable, dass es ein QuantumCircuit erstellt werden kann
var game_over = false; // beendet das spiel
var bloch_state = 1; // status der bloch sphere (1 = oben, 2 = hinten, 3 = unten, 4 = vorn)

circuit = new QuantumCircuit(1); // erstellen eines neues QuantumCircuits
circuit.createCreg("c", 1); // ein klassischer Register
console.log("Quantum Circuit created"); //in der Konsole wird gesagt, dass ein neuer Quantum Circuit erstellt worden ist

circuit.appendGate("id", 0, {}); // setze ein Identitäts-Gate auf das Qubit
document.getElementById("start").innerHTML = "Kopf"; // Startzustand
document.getElementById("Player").innerHTML = "Spieler 1"; // Spieler 1 wird definiert
document.getElementById("Zug").innerHTML = "Quantenzug möglich"; // Zug ist "Normal"
document.getElementById("drawing").innerHTML = circuit.exportSVG(true); // der Circuit wird gezeichnet

// wird aufgerufen, wenn ein Button gedrückt wird
function select(selection){
  if (game_over === false) { // solange nicht zu Ende
    if (selection == 1) {
      // Qubit Zustand wechseln
      circuit.appendGate("x", 0, {}); // X-Gate wird erstellt
      console.log("Normal added "); // In der Konsole wird geschrieben, dass ein normaler Zug hinzugefügt wurde
      change_image(1);
    }
    else if (selection == 2 && (counter % 2) == 0) { // nur für Quantenspieler möglich
      // Player 1 = Quantum player
      circuit.appendGate("h", 0, {}); // H-Gate wird erstellt
      console.log("Quantum added "); // In der Konsole wird geschrieben, dass ein quantum Zug hinzugefügt wurde
      change_image(2);
    }
    else if (selection == 0) {
      // Quanten Zustand beibehalten
      circuit.appendGate("id", 0, {}); // ID-Gate wird erstellt
      console.log("Identity added "); //In der Konsole wird geschrieben, dass ein identitäts Zug hinzugefügt wurde
    }
  }
}
```

```

    }
    else { // sonst => ungültiger Zug
        console.log("Ungültiger Zug "); //In der Konsole wird geschrieben, dass es ein ungültiger
Zug war
        return; // abbruch
    }

    if ((counter % 2) == 0) { //Falls der Counter gerade
        document.getElementById("Player").innerHTML = "Spieler 1"; //Spieler 1 ist am Zug
        document.getElementById("Zug").innerHTML = "Quantenzug möglich"; //Sein Zug ist ein
"Qunatum" Zug
    }
    else { //Falls der Counter ungerade ist
        document.getElementById("Player").innerHTML = "Spieler 2"; //Spieler 2 ist am Zug
        document.getElementById("Zug").innerHTML = "Nur normale Züge möglich"; //Sein Zug ist ein
normaler Zug
    }
    counter++; //counter wird erhöht
    document.getElementById("drawing").innerHTML = circuit.exportSVG(true); // der Circuit wird
gezeichnet

    if (counter >= i) { //falls der Counter grösser gleich vier ist
        finish(); //Spiel wird beendet
    }
}

// ändert die Bloch Sphere
function change_image(value){
    if (value == 1) {
        if ((bloch_state % 2) == 0) { // wenn gerade => H state => keine Anpassung
            // do nothing
        }
        else {
            bloch_state += 2; // sonst um spiegeln
        }
    }
    else if (value == 2) {
        if ((bloch_state % 2) == 0) { // wenn gerade => H state => zurückdrehen
            bloch_state += 1; // zurückdrehen
        }
        else {
            bloch_state -= 1; // sonst um spiegeln
        }
    }
    if (bloch_state < 1) bloch_state = 4; // overflow => zurückwechseln
    if (bloch_state > 4) bloch_state = 1; // overflow andere Richtung
    var name = "img/"+bloch_state+".png"; // Bild auswählen
    document.getElementById("bloch").src=name; // Bild zeichnen
}

//die Funktion "finish" wird nach Ablauf der Züge gewählt
function finish(){
    game_over = true;
    circuit.addMeasure(0, "c", 0); //misst den Quantum Circuit und bildet es auf einem Bit ab
    circuit.run(); //Circuit wird ausgeführt

    var result = ""; //Die Variable "Resultat"
    if (circuit.getCregValue("c") == 0) { //Ist falls das Bit 0 ist
        result = "Kopf"; //das Ergebnis "Kopf"
    }
    else { //Sonst
        result = "Zahl"; //ist es das Ergebnis "Zahl"
    }
    console.log("Ergebnis: ", result); //In der Konsole wird das Resultat geschrieben
    console.log("Wahrscheinlichkeit für Zahl: ", circuit.probability(0)); //In der Konsole wir
die Wahrscheinlichkeit angegeben, dass der QuantumCircuit das Ergebnis hat

    // Ergebnisfelder setzen
    document.getElementById("output").innerHTML = result; //
    document.getElementById("probability").innerHTML = circuit.probability(0); //
    document.getElementById("drawing").innerHTML = circuit.exportSVG(true); //
}

// Die Funktion "refreshPage" bewirkt einen Neutart
function refreshPage(){
    game_over = false;

```

```

    window.location.reload(); //Die Seite wird refresht
}

```

## 7.2 Source Code des Prototypen Version 1 – Tic-Tac-Toe

### 7.2.1 HMTL

```

<html>
  <head>
    <title>Tic Tac Toe</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="css/main.css">
  </head>

  <body>
    <h1>Quantum Tic-Tac-Toe (Variante 1)</h1> <br><br>
    Spieler am Zug: <span id="Player"></span> <br><br>
    Gewinner: <span id="winner"></span> <br><br><br>
    Wahl des Zuges:<br>
    <button id="1" onclick="define_zug(1);">|0> oder |1></button>
    <button id="2" onclick="define_zug(2);">Entanglement</button> <br><br>
    Spielfeld: <br>
    <table>
      <col width="95px" />
      <col width="95px" />
      <col width="95px" />
      <tr>
        <th><button class="fieldbutton" onClick="select(1);" id="p1"> </button></th>
        <th><button class="fieldbutton" onClick="select(2);" id="p2"> </button></th>
        <th><button class="fieldbutton" onClick="select(3);" id="p3"> </button></th>
      </tr>
      <tr>
        <th><button class="fieldbutton" onClick="select(4);" id="p4"> </button></th>
        <th><button class="fieldbutton" onClick="select(5);" id="p5"> </button></th>
        <th><button class="fieldbutton" onClick="select(6);" id="p6"> </button></th>
      </tr>
      <tr>
        <th><button class="fieldbutton" onClick="select(7);" id="p7"> </button></th>
        <th><button class="fieldbutton" onClick="select(8);" id="p8"> </button></th>
        <th><button class="fieldbutton" onClick="select(9);" id="p9"> </button></th>
      </tr>
    </table> <br><br>
    Circuit: <br> <span id="drawing"></span> <br><br><br>
    <button id="NewGame" onclick="init()">Neues Spiel</button>
  </body>

  <script src="js/quantum-circuit.min.js"></script>
  <script src="js/main.js"></script>
</html>

```

### 7.2.2 JavaScript – main.js

```

var winner = ""; // Gewinner Variable
var game_over = false; // Spielende
var circuit = new QuantumCircuit(1); // Quanten Circuit
var field_played = []; // Dokumentation, welche Felder gespielt wurden.
var field_result = []; // Resultat pro Feld
var player = 1; // Player 1 / X / q1 / result = 1 // Player 0 / O / q0 / result = -1
var counter = 0; // Zähler für Spielzüge
var erster_zug = true; // Indikator für ersten Zug, da kein Quantezug möglich

var second_field_needed = false; // braucht es ein weiteres Feld, um den Zug abzuschliessen
(z.B. bei CNOT)
var helper_field = 0; // merkt sich das letzte Feld, falls zwei Felder selektiert werden müssen
var zug = 1; // Art des Zuges ( Klassisch / CNOT = "cx" 2 Felder)

// Selektierter Zug-Button
document.getElementById("1").style.backgroundColor = "#FAD02C";

// wird aufgerufen beim laden der Seite und initialisiert alle Felder
init();

// initialisieren der Felder
function init(){
  // Liste der gespielten Felder und fixierten Felder werden auf 0 gesetzt
  var i_name = "";
  // Felder initialisiert

```

```

    for (var k=0; k<9; k++) {
        field_played[k] = 0;
        field_result[k] = 0;
        i_name = "p" + (k+1);
        document.getElementById(i_name).style.backgroundColor = "#333652";
        document.getElementById(i_name).firstChild.data = "\xa0";
    }
    // Variablen initialisieren
    game_over = false;
    erster_zug = true;
    counter = 0; // Zähler auf 0
    winner = ""; // Gewinner leer
    player = 1; // Erster Spieler ist 1 = X
    document.getElementById("winner").innerHTML = " ";
    document.getElementById("Player").innerHTML = "Spieler X";
    // Quanten Circuit initialisiert
    init_quantum();
    // Quanten Circuit zeichnen
    document.getElementById("drawing").innerHTML = circuit.exportSVG(true);
}

// initialisieren des Quanten Circuit
function init_quantum(){
    circuit = new QuantumCircuit(9); // Quanten circuit initialisieren / 9 Qubits
    circuit.createCreg("c", 9); // 9 klass. Register initialisieren
    console.log("Quantum created "); // status
}

// Definieren der Art des Zuges
function define_zug(value){
    if (player == 1) {
        // alle Zuelemente zurücksetzen
        for (var k=1; k<3; k++) {
            document.getElementById(k.toString()).style.backgroundColor = "#90ADC6";
        }
        // Art des Zuges ( |0> = "id", |1> = "x", H = "h", CNOT = "cx" 2 Felder)
        document.getElementById(value.toString()).style.backgroundColor = "#FAD02C";
        zug = value;
    }
}

// Aufruf beim Klicken auf ein Feld
function select(field){
    var name = "p" + field;
    console.log("Feld " + name + " , " + player);
    var text = "";
    if (game_over) {
        console.log("Das Spiel ist schon zu Ende.");
    }
    else {
        // Wenn ein Quanten-Zug => das zweite Feld (Target Feld) muss bereits gemessen sein (hat Result)
        if (second_field_needed && field_played[field-1] == 1) { // 2. Feld (target) muss schon gespielt sein
            // CX gate
            console.log("Zweiter Zug von Quantenzug");
            circuit.appendGate("cx", [helper_field-1, field-1], {}); // CNOT Gate
            field_played[field-1] = 1; // Feld gespielt
            document.getElementById(name).style.backgroundColor = "#90ADC6";
            change_player(); // Spieler wechseln
            second_field_needed = false; // zurücksetzen für nächsten Zug
        }
        // Sonst => klassischer Zug oder Quantenzug (Control Feld)
        else if (!second_field_needed && field_played[field-1] == 0) {
            switch (zug) {
                // klassischer Zug
                case 1:
                    // setzt je nach Spieler Feld fix
                    if (player == 0) {
                        circuit.appendGate("id", field-1, {}); // Identity Gate
                        text = "0"; // Feldtext
                    }
                    else { // player == 1
                        circuit.appendGate("x", field-1, {}); // X Gate
                        text = "X"; // Feldtext
                    }
                    field_played[field-1] = 1;

```



```

        second_field_needed = false;
        erster_zug = false;
        document.getElementById(name).style.backgroundColor = "#90ADC6";
        change_player();
        break;
        // Quantenzug
        case 2:
            // zuerst Control gate => Muss H Gate sein
            if (erster_zug) {
                document.getElementById("Player").innerHTML = "Erster Zug muss klassischer
Zug sein";
            }
            else {
                circuit.appendGate("h", field-1, {}); // H Gate
                // CNOT wird erst im nächsten Zug angehängt
                // circuit.appendGate("cx", [field-1, helper_field-1], {});
                helper_field = field; // Control Feld merken
                second_field_needed = true; // zweite Selektion des gleichen Spielers
                field_played[field-1] = 1;
                document.getElementById(name).style.backgroundColor = "#90ADC6";
            }
            break;
        default:
            console.log("Error 1: Zug falsch gewählt");
    }
}
else {
    console.log("Error 2: Feld falsch gewählt");
    return;
}
// Feldtext setzen und Quantum Circuit zeichnen
document.getElementById(name).firstChild.data = text;
document.getElementById("drawing").innerHTML = circuit.exportSVG(true);

// wenn alles gespielt ist, dann wird gemessen
var all_played = true;
// alle felder prüfen, ob gespielt
for (var k=0; k<9; k++) {
    if (field_played[k] == 0) {
        all_played = false;
    }
}
// wenn alle gespielt => messen
if (all_played) {
    measure(); // messen
    game_over = true; // Game Over => keine Selektion mehr möglich
}
}

// alle Qbits messen
function measure() {
    var m_name = "";
    circuit.run(); // Circuit starten
    var measure = circuit.measureAll(); // alle Qubits messen
    // alle Felder durchlaufen und prüfen
    for (var k=0; k<9; k++) {
        m_name = "p" + (k+1);
        if (field_played[k] == 1 && measure[k] == 1) {
            // Erbesnis == 1 => X
            field_result[k] = 1; // Ergebnis setzen
            // Resultate zeichnen
            document.getElementById(m_name).firstChild.data = "X";
            document.getElementById(m_name).style.backgroundColor = "#FAD02C";
        }
        else if (field_played[k] == 1 && measure[k] == 0) {
            // Erbesnis == 0 => 0
            field_result[k] = -1; // Ergebnis setzen
            // Resultate zeichnen
            document.getElementById(m_name).firstChild.data = "0";
            document.getElementById(m_name).style.backgroundColor = "#FAD02C";
        }
        else {
            console.log("Error3 : sollte nicht vorkommen");
        }
    }
}
check_winner(); // Prüfen, ob es einen Gewinner gibt

```

```

// Ergebnis ausgeben
document.getElementById("Player").innerHTML = "SPIEL IST ZU ENDE";
}

// Prüft ob es einen Gewinner gibt
function check_winner() {
    var x = false;
    var o = false;
    var a = field_result[0] + field_result[1] + field_result[2];
    var b = field_result[3] + field_result[4] + field_result[5];
    var c = field_result[6] + field_result[7] + field_result[8];
    var d = field_result[0] + field_result[3] + field_result[6];
    var e = field_result[1] + field_result[4] + field_result[7];
    var f = field_result[2] + field_result[5] + field_result[8];
    var g = field_result[0] + field_result[4] + field_result[8];
    var h = field_result[2] + field_result[4] + field_result[6];

    // alle Varianten prüfen
    if (a == 3 || b == 3 || c == 3 || d == 3 || e == 3 || f == 3 || g == 3 || h == 3) {
        x = true;
    }
    if (a == -3 || b == -3 || c == -3 || d == -3 || e == -3 || f == -3 || g == -3 || h == -3) {
        o = true;
    }

    if ( (x && o) || (!x && !o) ) { // unentschieden
        document.getElementById("winner").innerHTML = "Unentschieden";
    }
    else if (x) { // X Gewinner
        document.getElementById("winner").innerHTML = "Spieler X"; // Spieler X hat gewonnen
    }
    else { // O Gewinner
        document.getElementById("winner").innerHTML = "Spieler O"; // Spieler O hat gewonnen
    }
}

// Welchselt den Spieler
function change_player(){
    define_zug(1); // Zug zurücksetzen
    if (game_over) { // Anzeigen, dass neu gestartet werden muss
        document.getElementById("Player").innerHTML = "DAS SPIEL IST ZU ENDE";
    }
    else {
        if (player == 1) { // Spieler 1 war dran => Spieler 0
            player = 0;
            document.getElementById("Player").innerHTML = "Spieler O";
        }
        else { // Spieler 0 war dran => Spieler 1
            player = 1;
            document.getElementById("Player").innerHTML = "Spieler X";
        }
    }
}
}

```

## 7.3 Source Code des Prototypen Version 2 – Tic-Tac-Toe

### 7.3.1 HTML

```

<html>
<head>
    <title>Tic Tac Toe</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="css/main.css">
</head>

<body>
    <br>
    <h1>Quantum Tic-Tac-Toe (Variante 2)</h1> <br><br>
    Spieler am Zug: <span id="Player"></span> <br><br>
    Gewinner: <span id="winner"></span> <br><br><br>
    Wahl des Zuges:<br>
    <button id="1" onclick="define_zug(1);">wechseln</button>
    <button id="4" onclick="define_zug(4);">beibehalten</button> <br><br>
    <button id="2" onclick="define_zug(2);">H Gate</button>
    <button id="3" onclick="define_zug(3);">Entanglement</button> <br><br>

```

```

Hinweise: <span id="Tipp"></span> <br><br>
Spielfeld: <br>
<table>
  <col width="95px" />
  <col width="95px" />
  <col width="95px" />
  <tr>
    <th><button class="fieldbutton" onClick="select(1);" id="p1"> </button></th>
    <th><button class="fieldbutton" onClick="select(2);" id="p2"> </button></th>
    <th><button class="fieldbutton" onClick="select(3);" id="p3"> </button></th>
  </tr>
  <tr>
    <th><button class="fieldbutton" onClick="select(4);" id="p4"> </button></th>
    <th><button class="fieldbutton" onClick="select(5);" id="p5"> </button></th>
    <th><button class="fieldbutton" onClick="select(6);" id="p6"> </button></th>
  </tr>
  <tr>
    <th><button class="fieldbutton" onClick="select(7);" id="p7"> </button></th>
    <th><button class="fieldbutton" onClick="select(8);" id="p8"> </button></th>
    <th><button class="fieldbutton" onClick="select(9);" id="p9"> </button></th>
  </tr>
</table> <br><br>
Circuit:<br> <span id="drawing"></span> <br><br><br>
<button id="NewGame" onClick="init()">Neues Spiel</button>

<script src="js/quantum-circuit.min.js"></script>
<script src="js/main.js"></script>
</html>

```

## 7.3.2 JavaScript

```

var winner = ""; // Gewinner Variable
var game_over = false; // Spielende
var circuit = new QuantumCircuit(1); // Quanten Circuit
var field_played = []; // Dokumentation, welche Felder gespielt wurden.
var field_result = []; // Resultat pro Feld
var field_fixed = []; // Feld fixiert (anzahl Züge gespielt)
var player = 1; // Player 1 / X / q1 / result = 1 // Player 0 / O / q0 / result = -1
var counter = 0; // Zähler für Spielzüge
var erster_zug = true; // Indikator für ersten Zug, da kein Quantezug möglich

var second_field_needed = false; // braucht es ein weiteres Feld, um den Zug abzuschliessen
(z.B. bei CNOT)
var helper_field = 0; // merkt sich das letzte Feld, falls zwei Felder selektiert werden müssen
var zug = 1; // Art des Zuges ( Klassisch / CNOT = "cx" 2 Felder)
var anzahl_zuege = 3; // Züge pro Feld

// Selektierter Zug-Button
document.getElementById("1").style.backgroundColor = "#FAD02C";

// wird aufgerufen beim laden der Seite und initialisiert alle Felder
init();

// initialisieren der Felder
function init(){
  // Liste der gespielten Felder und fixierten Felder werden auf 0 gesetzt
  var i_name = "";
  for (var k=0; k<9; k++) {
    field_played[k] = 0;
    field_result[k] = 0;
    field_fixed[k] = 0;
    i_name = "p" + (k+1);
    document.getElementById(i_name).style.backgroundColor = "#333652";
    document.getElementById(i_name).firstChild.data = "\xa0";
  }
  game_over = false;
  erster_zug = true;
  counter = 0; // Zähler auf 0
  winner = ""; // Gewinner leer
  player = 1; // Erster Spieler ist 1 = X
  document.getElementById("winner").innerHTML = " ";
  document.getElementById("Player").innerHTML = "Spieler X";
  // Quanten Circuit initialisiert
  init_quantum();
}

```

```

// Quanten Circuit zeichnen
document.getElementById("drawing").innerHTML = circuit.exportSVG(true);
}

// initialisieren des Quanten Circuit
function init_quantum(){
    circuit = new QuantumCircuit(9); // Quanten circuit initialisieren
    circuit.createCreg("c", 9); // 9 Qbits initialisieren
    // alle Felder mit X-Gate vorinitialisieren
    for (var k=0; k<9; k++) {
        circuit.appendGate("x", k, {});
    }
    console.log("Quantum created "); // status
}

// Definieren der Art des Zuges
function define_zug(value){
    if (!second_field_needed) {
        // alle Zügelemente zurücksetzen
        for (var k=1; k<5; k++) {
            document.getElementById(k.toString()).style.backgroundColor = "#90ADC6";
        }
        // Art des Zuges ( wechseln "x", H = "h", CNOT = "cx" 2 Felder, lassen = "id", )
        document.getElementById(value.toString()).style.backgroundColor = "#FAD02C";
        zug = value;
    }
}

// Aufruf beim Klicken auf ein Feld
function select(field){
    var name = "p"+field;
    console.log("Feld "+name+ " , "+ player);
    var text = "";
    document.getElementById("Tipp").innerHTML = " ";

    if (game_over) {
        document.getElementById("Tipp").innerHTML = "Das Spiel ist schon zu Ende.";
    }
    else if (field_fixed[field-1] == 1) { // Feld fixiert
        document.getElementById("Tipp").innerHTML = "Das Feld kann nicht gewählt werden. Es ist schon fixiert.";
    }
    else {
        // Wenn ein Quanten-Zug => das zweite Feld (Target Feld) muss bereits gemessen sein (hat Result)
        if (second_field_needed) {
            // CX gate
            console.log("Zweiter Zug von Quantenzug");
            circuit.appendGate("cx", [helper_field-1, field-1], {}); // CNOT Gate
            field_played[field-1] += 1; // Zähler für Feld hochsetzen
            text = "\u2A02"; // Zeichen für Feld
            document.getElementById(name).style.backgroundColor = "#90ADC6";
            change_player(); // Spieler wechseln
            second_field_needed = false; // zurücksetzen für nächsten Zug
        }
        // Sonst => andere Züge
        else if (!second_field_needed) {
            switch (zug) {
                // klassischer Zug => Wechseln
                case 1:
                    // Qubit wird gewechselt mit X Gate
                    circuit.appendGate("x", field-1, {}); // X Gate
                    text = "\u27F2"; // X Zeichen
                    field_played[field-1] += 1; // Zähler für Feld hochsetzen
                    second_field_needed = false; // kein zweiter Zug des Spielers
                    document.getElementById(name).style.backgroundColor = "#90ADC6";
                    change_player(); // Spieler wechseln
                    break;
                // H Gate => Superposition
                case 2:
                    circuit.appendGate("h", field-1, {}); // H Gate
                    text = "\u2053"; // Wellen Zeichen
                    second_field_needed = false; // kein zweiter Zug des Spielers
                    field_played[field-1] += 1; // Zähler für Feld hochsetzen
                    document.getElementById(name).style.backgroundColor = "#90ADC6";
                    change_player(); // Spieler wechseln
            }
        }
    }
}

```

```

        break;
    // Entanglement ( H CNOT )
    case 3:
        // zuerst Control gate => Muss H Gate sein
        circuit.appendGate("h", field-1, {}); // H CNOT => / zweites X wird im nächsten Zug
angehängt
        // CNOT wird erst im nächsten Zug angehängt
        // circuit.appendGate("cx", [field-1, helper_field-1], {});
        text = "\u25CF"; // control Zeichen
        helper_field = field; // Control Feld merken
        second_field_needed = true; // zweiter Zug des Spielers
        field_played[field-1] += 1; // Zähler für Feld hochsetzen
        document.getElementById(name).style.backgroundColor = "#90ADC6";
        break;
    // Klassischer Zug => beibehalten
    case 4:
        circuit.appendGate("id", field-1, {}); // identity Gate
        text = "\u2B07"; // identity Zeichen
        field_played[field-1] += 1; // Zähler für Feld hochsetzen
        second_field_needed = false; // kein zweiter Zug des Spielers
        document.getElementById(name).style.backgroundColor = "#90ADC6";
        change_player(); // Spieler wechseln
        break;
    default:
        console.log("Error 1: Zug falsch gewählt");
    }
}
else {
    console.log("Error 2: Feld falsch gewählt");
    document.getElementById("Tipp").innerHTML = "Falsches Feld gewählt.";
}
// Feldtext setzen und Quantum Circuit zeichnen
document.getElementById(name).firstChild.data = text;
document.getElementById("drawing").innerHTML = circuit.exportSVG(true);

// wenn max. Anzahl Züge erreicht, dann Feld einfärben und fixieren
if (field_played[field-1] == anzahl_zuege) {
    field_fixed[field-1] = 1; // Feld fixieren
    document.getElementById(name).style.backgroundColor = "#FAD02C";
}
// wenn alles gespielt ist, dann wird gemessen
var all_played = true;
// alle felder prüfen, ob gespielt
for (var k=0; k<9; k++) {
    if (field_played[k] == 0) {
        all_played = false;
    }
}
// wenn alle gespielt => messen
if (all_played) {
    measure(); // messen
    game_over = true; // Game Over => keine Selektion mehr möglich
}
}

// alle Qbits messen
function measure() {
    var m_name = "";
    circuit.run(); // Circuit starten
    var measure = circuit.measureAll(); // alle Qubits messen
    // alle Felder durchlaufen und prüfen
    for (var k=0; k<9; k++) {
        m_name = "p" + (k+1);
        console.log("name: ", m_name);
        if (measure[k] == 1) {
            // Erbesnis == 1 => X
            field_result[k] = 1; // Ergebnis setzen
            // Resultate zeichnen
            document.getElementById(m_name).firstChild.data = "X";
            document.getElementById(m_name).style.backgroundColor = "#FAD02C";
        }
        else if (measure[k] == 0) {
            // Erbesnis == 0 => O
            field_result[k] = -1; // Ergebnis setzen
            // Resultate zeichnen
            document.getElementById(m_name).firstChild.data = "O";
        }
    }
}

```

```

        document.getElementById(m_name).style.backgroundColor = "#FAD02C";
        console.log("Feld "+m_name+" O");
    }
    else {
        console.log("Error3 : sollte nicht vorkommen");
    }
}
check_winner(); // Prüfen, ob es einen Gewinner gibt
// Ergebnis ausgeben
document.getElementById("Player").innerHTML = "SPIEL IST ZU ENDE";
}

// Prüft ob es einen Gewinner gibt
function check_winner() {
    var x = false;
    var o = false;
    var a = field_result[0] + field_result[1] + field_result[2];
    var b = field_result[3] + field_result[4] + field_result[5];
    var c = field_result[6] + field_result[7] + field_result[8];
    var d = field_result[0] + field_result[3] + field_result[6];
    var e = field_result[1] + field_result[4] + field_result[7];
    var f = field_result[2] + field_result[5] + field_result[8];
    var g = field_result[0] + field_result[4] + field_result[8];
    var h = field_result[2] + field_result[4] + field_result[6];

    if (a == 3 || b == 3 || c == 3 || d == 3 || e == 3 || f == 3 || g == 3 || h == 3) {
        x = true;
    }
    if (a == -3 || b == -3 || c == -3 || d == -3 || e == -3 || f == -3 || g == -3 || h == -3) {
        o = true;
    }

    if ( (x && o) || (!x && !o) ) { // unentschieden
        document.getElementById("winner").innerHTML = "Unentschieden";
    }
    else if (x) { // X Gewinner
        document.getElementById("winner").innerHTML = "Spieler X"; // Spieler X hat gewonnen
    }
    else { // O Gewinner
        document.getElementById("winner").innerHTML = "Spieler O"; // Spieler O hat gewonnen
    }
}

// Welchselt den Spieler
function change_player(){
    if (game_over) { // Anzeigen, dass neu gestartet werden muss
        document.getElementById("Player").innerHTML = "DAS SPIEL IST ZU ENDE";
    }
    else {
        if (player == 1) { // Spieler 1 war dran => Spieler 0
            player = 0;
            document.getElementById("Player").innerHTML = "Spieler O";
        }
        else { // Spieler 0 war dran => Spieler 1
            player = 1;
            document.getElementById("Player").innerHTML = "Spieler X";
        }
    }
}
}

```

## 8 Literaturverzeichnis

### 8.1 Literaturverzeichnis mit Angabe des Autors

Akfeld, Meike: Die Vektorraumaxiome,

<https://metaphor.ethz.ch/x/2016/hs/401-1151-00L/dokumente/VektorraumAxiome.pdf>,

heruntergeladen am 16.06.2021.

Augusten, Stephan: Was ist ein Logikgatter? – Definition «Logic gate»,

<https://www.dev-insider.de/was-ist-ein-logikgatter-a-798738/>,

heruntergeladen am 26.05.2021.

Burgard, Wolfram: Einführung in die Informatik - Vom Problem zum Algorithmus und zum Programm,

[http://ais.informatik.uni-freiburg.de/teaching/ws06/info1/material/14\\_alg/14\\_alg-4up-new.pdf](http://ais.informatik.uni-freiburg.de/teaching/ws06/info1/material/14_alg/14_alg-4up-new.pdf),

heruntergeladen am 29.05.2021.

Chambers, Andrew: Game Theory and Tic Tac Toe,

<https://ibmathsresources.com/2013/11/24/game-theory-and-tic-tac-toe/>,

heruntergeladen am 29.09.2021.

Chen, James: Nash Equilibrium – What Is Nash Equilibrium?,

<https://www.investopedia.com/terms/n/nash-equilibrium.asp>,

heruntergeladen am 25.08.2021.

De la Fuente Ruiz, Alfonso: Quantum Annealing – A Brief description on the state of the art of some local optimization methods,

<https://arxiv.org/ftp/arxiv/papers/1404/1404.2465.pdf>,

heruntergeladen am 21.05.2021.

Fufaev, Alexander: Was ist der Unterschied zwischen Teilchen und Quasiteilchen?,

<https://de.universaldenker.org/fragen/145>,

heruntergeladen am 16.06.2021.

Glendinning, Ian: The Bloch Sphere,

<http://www.vcpc.univie.ac.at/~ian/hotlist/qc/talks/bloch-sphere.pdf>,

heruntergeladen am 08.11.2021.

Goff, A; Lehmann, D; Siegel, J: Quantum tic-tac-toe, spooky-coins & magic-envelopes, as metaphors for relativistic quantum physics,

<https://arc.aiaa.org/doi/abs/10.2514/6.2002-3763>,

heruntergeladen am 27.10.2021.

Haken, Hermann, Wolf Hans Christian: Atom- und Quantenphysik, Einführung in die experimentellen und theoretischen Grundlagen, Springer Verlag, 1990 (4. Auflage).

Hayes, Adam: Game Theory – What Is Game Theory?,

<https://www.investopedia.com/terms/g/gametheory.asp>,

heruntergeladen am 25.08.2021.

Helge, Michael: Quantencomputer versus PC – Wie unterscheiden sich Quantencomputer von herkömmlichen Computern,

<https://main-incubator.com/wie-unterscheiden-sich-quantencomputer-von-herkoemmlichen-pcs/#:~:text=Der%20gro%C3%9Fe%20Unterschied%20besteht%20im,und%20kann%20Ergebnisse%20simultan%20berechnen>,

heruntergeladen am 18.05.2021.

Homeister, Matthias: Quantencomputing verstehen, Grundlagen. Anwendungen – Perspektiven, Wiesbaden: Springer Vieweg, 2018 (5. Auflage).

Hossenfelder, Sabine: Rechnen mit Qubits im Jahr 2021. Sechs technische Konzepte für Quantencomputer, in: ct magazin für computer technik 17 (31.07.2021), S. 124-127.

Hufsky, Franziska: Die fleissigen Biber der Informatik,  
<http://bioinfowelten.uni-jena.de/2016/06/21/die-fleissigen-biber-der-informatik/>,  
heruntergeladen am 29.05.2021.

Hufsky, Franziska: Von leichten, schweren und vollständigen Problemen,  
<http://bioinfowelten.uni-jena.de/2016/08/09/von-leichten-schweren-und-vollstaendigen-problemen/>,  
heruntergeladen am 29.05.2021.

Jordan, Stephen: Quantum Algorithm Zoo,  
<https://quantumalgorithmzoo.org/>,  
heruntergeladen am 30.05.2021.

Kagermann, Henning, Süßenguth, Florian, Körner, Jorg, Liepold, Annka: Innovationspotenziale der Quantentechnologien der zweiten Generation,  
file:///C:/Users/a21313/Downloads/aca\_IMP\_Quantentechnologie\_de\_web.pdf,  
heruntergeladen am 30.05.2021.

Knopf, Ingo; Funk, Sebastian: Technik – So funktioniert ein Quantencomputer,  
<https://www.quarks.de/technik/faq-so-funktioniert-ein-quantencomputer/>,  
heruntergeladen am 19.05.2021.

Kobylinkska, Anna; Martins, Filipe; Ostler, Ulrike: Quanten-Gate-Computer versus Annealer – Probleme, Lösungen, Mathe und Materialien von Quantencomputern,  
<https://www.datacenter-insider.de/quanten-gate-computer-versus-annealer-a-782545/>,  
heruntergeladen am 20.05.2021.

Kobylinkska, Anna; Martins, Filipe; Ostler, Ulrike: Was ist Quanten-Annealing? – Vorglühen für Optimierungsaufgaben,  
<https://www.datacenter-insider.de/was-ist-quanten-annealing-a-782266/>,  
heruntergeladen am 20.05.2021.

Koshka Yaroslav; Novotny, M.A., Comparison of D-Wave Quantum Annealing and Classical Simulated Annealing for Local Minima Determination,  
<https://arxiv.org/abs/1911.03338>,  
heruntergeladen am 23.05.2021.

Kreye, Manuel: Es können nicht beide recht haben,  
<https://www.weltderphysik.de/gebiet/teilchen/news/2018/es-koennen-nicht-beide-recht-haben/>,  
heruntergeladen am 16.06.2021

Kusche, Nora: Wie funktioniert ein Quantencomputer,  
[Welt der Physik: Wie funktioniert ein Quantencomputer?](https://www.weltderphysik.de/gebiet/teilchen/news/2018/es-koennen-nicht-beide-recht-haben/),  
heruntergeladen am 21.05.2021.

Lässig, Jörg: Angewandtes Quantencomputing,  
<https://link.springer.com/article/10.1007/s00287-020-01309-9>,  
heruntergeladen am 09.06.2021.

Marchenkova, Anastasia: What's the difference between quantum annealing and universal gate quantum computers?. Universal Gate Quantum Computing,



<https://medium.com/quantum-bits/what-s-the-difference-between-quantum-annealing-and-universal-gate-quantum-computers-c5e5099175a1>,  
heruntergeladen am 19.05.2021.

Marinescu, Dan; Marinescu, Gabriela: Quantum Circuits,  
<https://www.sciencedirect.com/topics/computer-science/quantum-circuit>,  
heruntergeladen am 23.06.2021.

Marre, Jens: Welche Quantencomputer gibt es jetzt schon? – Quantencomputer durch Ionenfallen,  
[http://www.quantencomputer-info.de/quantencomputer/welche-quantencomputer-gibt-es-jetzt-schon/#Quantencomputer\\_durch\\_Ionenfallen](http://www.quantencomputer-info.de/quantencomputer/welche-quantencomputer-gibt-es-jetzt-schon/#Quantencomputer_durch_Ionenfallen),  
heruntergeladen am 08.11.2021.

Meyer, David: Quantum Strategies,  
<https://arxiv.org/abs/quant-ph/9804010>,  
heruntergeladen am 22.09.2021.

Nagy, M., & Nagy, N: Quantum Tic-Tac-Toe: A Genuine Probabilistic Approach,  
[https://www.scirp.org/html/4-7400990\\_24755.html](https://www.scirp.org/html/4-7400990_24755.html),  
heruntergeladen am 27.10.2021.

Neukart, Florian: Hallo, Quantenwelt!. Wie man mit Quantengatter programmiert, in: c't Magazin für Computertechnik (2020), S. 146-147.

Ostler, Ulrike: Teilchen für Quantencomputer – Was ist ein Qubit,  
<https://www.datacenter-insider.de/was-ist-ein-qubit-a-715873/>,  
heruntergeladen am 18.05.2021.

Piotrowski, Edward: The next stage: quantum game theory,  
[https://www.researchgate.net/publication/2190925\\_The\\_next\\_stage\\_quantum\\_game\\_theory](https://www.researchgate.net/publication/2190925_The_next_stage_quantum_game_theory),  
heruntergeladen am 22.09.2021.

Press, Gill: 27 Milestones In The History Of Quantum Computing, in: Forbes,  
<https://www.forbes.com/sites/gilpress/2021/05/18/27-milestones-in-the-history-of-quantum-computing/?sh=3b3a668c7b23>,  
heruntergeladen am 06.05.2021.

Roell, Jason: Demystifying Quantum Gates – One Qubit At A Time. The Pauli Y and Z-Gates,  
<https://towardsdatascience.com/demystifying-quantum-gates-one-qubit-at-a-time-54404ed80640>,  
heruntergeladen am 03.06.2021.

Romero-Isart, Oriol; Pflanzner, Anika, Quantenmechanik am Limit: Superpositionen massiver Objekte,  
[https://www.mpg.de/4981741/Quantenphysik\\_am\\_Limit](https://www.mpg.de/4981741/Quantenphysik_am_Limit),  
heruntergeladen am 16.09.2021.

Ross, Don: Game Theory – Solution Concepts and Equilibria,  
<https://plato.stanford.edu/entries/game-theory/#PD>,  
heruntergeladen am 25.08.2021.

Sagole, Sai; Dey, Anurrit; Behera, Bikash; Panigrahi, Prasanta: Quantum Tic-Tac-Toe: A Hybrid of Quantum and Classical Computing,  
[https://www.researchgate.net/publication/338113536\\_Quantum\\_Tic-Tac-Toe\\_A\\_Hybrid\\_of\\_Quantum\\_and\\_Classical\\_Computing](https://www.researchgate.net/publication/338113536_Quantum_Tic-Tac-Toe_A_Hybrid_of_Quantum_and_Classical_Computing),  
heruntergeladen am 06.11.2021.

Spentzouris, Panagiotis: Quantum Computing: Advancing Fundamental Physics,  
<https://link.springer.com/article/10.1007/s41781-020-00043-x#auth-Panagiotis-Spentzouris>,  
heruntergeladen am 06.05.2021.

Stollenwerk, Tobias: Umfeldstudie Quantencomputing,  
[https://www.dlr.de/content/de/downloads/publikationen/broschueren/2021/umfeldstudie-quantencomputing.pdf?\\_\\_blob=publicationFile&v=2](https://www.dlr.de/content/de/downloads/publikationen/broschueren/2021/umfeldstudie-quantencomputing.pdf?__blob=publicationFile&v=2),  
heruntergeladen am 05.11.2021.

Stroh, Iris: Multi-Chip-Technologien für Leistungssteigerung?,  
<https://www.elektroniknet.de/halbleiter/prozessoren/multi-chip-technologien-fuer-leistungssteigerung.149755.html>,  
heruntergeladen am 05.05.2021.

## 8.2 Literatur ohne Angabe des Autors

Adiabatische Zustandsänderung, in: chemie.de,  
[https://www.chemie.de/lexikon/Adiabatische\\_Zustands%C3%A4nderung.html](https://www.chemie.de/lexikon/Adiabatische_Zustands%C3%A4nderung.html),  
heruntergeladen am 16.06.2021.

Bra-Ket Notation, in: Math ist fun,  
<https://www.mathsisfun.com/physics/bra-ket-notation.html>,  
heruntergeladen am: 05.11.2021.

Bell State, in: Wikipedia,  
[https://en.wikipedia.org/wiki/Bell\\_state](https://en.wikipedia.org/wiki/Bell_state),  
heruntergeladen am 02.06.2021.

Boolesche Algebra Wahrheitstabellen, in: ElectronicsTutorials,  
<https://www.electronics-tutorials.ws/de/boolsche/boolsche-algebra-wahrheitstabellen.html>,  
heruntergeladen am 26.05.2021.

China vermeldet Rekord-Leistung bei Quantencomputern, in: dw.com,  
<https://www.dw.com/de/china-vermeldet-rekord-leistung-bei-quantencomputern/a-59654510>,  
heruntergeladen am 09.06.2021.

Defining Quantum Circuits, in: Qiskit,  
<https://qiskit.org/textbook/ch-algorithms/defining-quantum-circuits.html>,  
heruntergeladen am 06.06.2021.

Elektronenspin, in: Wikipedia,  
<https://de.wikipedia.org/wiki/Elektronenspin>,  
heruntergeladen am 16.06.2021.

Gefangenendilemma, in: Wikipedia:  
<https://de.wikipedia.org/wiki/Gefangenendilemma>,  
heruntergeladen am 25.08.2021.

Heisenbergsche Unschärferelation, in: chemie.de  
[https://www.chemie.de/lexikon/Heisenbergsche\\_Unschärferelation.html](https://www.chemie.de/lexikon/Heisenbergsche_Unschärferelation.html),  
heruntergeladen am 09.06.2021.

Heisenbergsche Unbestimmtheitsrelation, in: Schülerlexikon  
<https://www.lernhelfer.de/schuelerlexikon/physik-abitur/artikel/heisenbergsche->

[unbestimmtheitsrelation#](#),

heruntergeladen am 09.06.2021.

Kombinatorische Optimierung, in: Wikipedia,

[https://de.wikipedia.org/wiki/Kombinatorische\\_Optimierung](https://de.wikipedia.org/wiki/Kombinatorische_Optimierung),

heruntergeladen am 05.05.2021.

Logikgatter, in: Wikipedia,

<https://de.wikipedia.org/wiki/Logikgatter>,

heruntergeladen am 26.05.2021.

Nash-Gleichgewicht, in: BWL-Lexikon,

<https://www.bwl-lexikon.de/wiki/nash-gleichgewicht/>,

heruntergeladen am 25.08.2021.

Nash-Gleichgewicht, in: Wikipedia,

<https://de.wikipedia.org/wiki/Nash-Gleichgewicht>,

heruntergeladen am 25.08.2021.

Notes on the Bloch Sphere, in: Wordpress,

<https://dmm613.wordpress.com/2018/11/10/notes-on-the-bloch-sphere/>

heruntergeladen am 29.05.2021.

Optimierungsproblem, in: Wikipedia,

<https://de.wikipedia.org/wiki/Optimierungsproblem>,

heruntergeladen am 22.05.2021.

Prinzipien der Quantenphysik, in: Uni Hannover

[https://qig.itp.uni-hannover.de/quant/index.php/A1/Prinzipien\\_der\\_Quantenphysik](https://qig.itp.uni-hannover.de/quant/index.php/A1/Prinzipien_der_Quantenphysik),

heruntergeladen am 09.06.2021.

Prisoner's Dilemma, in: Wikipedia,

[https://en.wikipedia.org/wiki/Prisoner%27s\\_dilemma](https://en.wikipedia.org/wiki/Prisoner%27s_dilemma),

heruntergeladen am 25.08.2021.

Quantenphysik, in: Chemie.de,

<https://www.chemie.de/lexikon/Quantenphysik.html>,

heruntergeladen am 05.11.2021.

Quantenverschränkung: »Spukhafte Fernwirkung« – einfach erklärt, in: chip.de,

[https://praxistipps.chip.de/quantenverschraenkung-spukhafte-fernwirkung-einfach-erklart\\_104842](https://praxistipps.chip.de/quantenverschraenkung-spukhafte-fernwirkung-einfach-erklart_104842),

heruntergeladen am 16.06.2021.

Quantenverschränkung, in: Wikipedia,

<https://de.wikipedia.org/wiki/Quantenverschr%C3%A4nkung>,

heruntergeladen am 16.06.2021.

Quantenteleportation, in: Lexikon der Physik

<https://www.spektrum.de/lexikon/physik/quantenteleportation/11885>

heruntergeladen am 29.05.2021.

Quantum annealing, in: Wikipedia,

[https://en.wikipedia.org/wiki/Quantum\\_annealing](https://en.wikipedia.org/wiki/Quantum_annealing),

heruntergeladen am 23.05.2021.

Quantum Circuit, in: Qiskit,  
<https://qiskit.org/documentation/stubs/qiskit.circuit.QuantumCircuit.html>,  
heruntergeladen am 23.06.2021.

Quantum Circuit Simulator, in: npmjs,  
<https://www.npmjs.com/package/quantum-circuit?activeTab=readme>,  
heruntergeladen am 03.11.2021.

Quantum logic gate, in: Wikipedia,  
[https://en.wikipedia.org/wiki/Quantum\\_logic\\_gate](https://en.wikipedia.org/wiki/Quantum_logic_gate),  
heruntergeladen am 02.06.2021.

Quasiteilchen, in: Lexikon der Physik,  
<https://www.spektrum.de/lexikon/physik/quasiteilchen/11939>,  
heruntergeladen am 19.06.2021.

Quirk, in: Algassert,  
[https://algassert.com/quirk#circuit={%22cols%22:\[ \]}](https://algassert.com/quirk#circuit={%22cols%22:[ ]}),  
heruntergeladen am 05.06.2021.

Roman Stanisław Ingarden, in: Wikipedia,  
[https://de.wikipedia.org/wiki/Roman\\_Stanis%C5%82aw\\_Ingarden](https://de.wikipedia.org/wiki/Roman_Stanis%C5%82aw_Ingarden),  
heruntergeladen am 07.05.2021.

Schrödingers Katze – ein Gedankenexperiment, in LEIFIphysik,  
<https://www.leifiphysik.de/atomphysik/quantenmech-atommodell/versuche/schroedingers-katze-ein-gedankenexperiment>,  
heruntergeladen am 16.06.2021.

Single Qubit Gates, in: Qiskit,  
<https://qiskit.org/textbook/ch-states/single-qubit-gates.html#xgate>,  
heruntergeladen am 02.06.2021.

Supraleitende Qubits: Traum vom Quantencomputer wird wahr, in: Cordis,  
<https://cordis.europa.eu/article/id/85447-superconducting-qubits-quantum-computing-gets-connected/de>,  
heruntergeladen am 13.05.2021.

Supraleiter, in: Wikipedia,  
<https://de.wikipedia.org/wiki/Supraleiter>,  
heruntergeladen am 16.06.2021.

The no-cloning theorem, in: Quantiki,  
<https://www.quantiki.org/wiki/no-cloning-theorem>,  
heruntergeladen am 08.11.2021.

The Tensor Product, Demystified, in: math3ma:  
<https://www.math3ma.com/blog/the-tensor-product-demystified>,  
heruntergeladen am 08.11.2021.

Tic-Tac-Toe, in: Wikipedia:  
<https://de.wikipedia.org/wiki/Tic-Tac-Toe>,  
heruntergeladen am 06.11.2021.

Was ist Zufall?, in: Physikalische Soiree

<https://www.phyx.at/was-ist-zufall/>,

heruntergeladen am 09.06.2021.

What is quantum computing?, in: IBM,

<https://www.ibm.com/quantum-computing/what-is-quantum-computing/>,

heruntergeladen am 05.05.2021.

What is Quantum Annealing?, in: D-Wave,

[https://docs.dwavesys.com/docs/latest/c\\_gs\\_2.html](https://docs.dwavesys.com/docs/latest/c_gs_2.html),

heruntergeladen am 20.05.2021.

What is the difference between Quantum annealing and simulated annealing?, in: Software engineering,

<https://softwareengineering.stackexchange.com/questions/194552/what-is-the-difference-between-quantum-annealing-and-simulated-annealing>,

heruntergeladen am 23.05.2021.

## 9 Abbildungsverzeichnis

Titelbild: Quantencomputer. Aspen-8 Quantum computer, <https://www.rigetti.com/>, heruntergeladen am 29.08.2021.

Abb. 1: Projizierte Entwicklung des Marktpotenzials für Quantencomputer (2025–2050) mit Grund- und Positivszenarios. Actech.de: Innovationspotenziale der Quantentechnologien der zweiten Generation, <https://www.acatech.de/publikation/innovationspotenziale-der-quantentechnologien/download-pdf?lang=de>, S.70, heruntergeladen am 21.11.2021.

Abb. 2: Schema eines Experiments. Uni Hannover: Prinzipien der Quantenphysik, [https://qig.itp.uni-hannover.de/quanth/index.php/A1/Prinzipien\\_der\\_Quantenphysik](https://qig.itp.uni-hannover.de/quanth/index.php/A1/Prinzipien_der_Quantenphysik), heruntergeladen am 21.11.2021.

Abb. 3: Katze in überlagertem Zustand, Wikipedia: Schrödingers Katze, [https://de.wikipedia.org/wiki/Schr%C3%B6dingers\\_Katze](https://de.wikipedia.org/wiki/Schr%C3%B6dingers_Katze), heruntergeladen am: 09.06.2021.

Abb. 4: Zustand eines Qubits beschrieben mit Vektoren. Wordpress.com: Notes on the Bloch Sphere. Complex plane representation of a single qubit, <https://dmm613.wordpress.com/2018/11/10/notes-on-the-bloch-sphere/>, heruntergeladen am 12.05.2021.

Abb. 5: Zustand eines Qubits in einer Bloch Sphäre. Wikipedia.com: Bloch sphere, [https://en.wikipedia.org/wiki/Bloch\\_sphere#/media/File:Bloch\\_sphere.svg](https://en.wikipedia.org/wiki/Bloch_sphere#/media/File:Bloch_sphere.svg), heruntergeladen am 08.11.2021.

Abb. 6: Ausgestellter Quantencomputer. IBM: IBM Quantum, Quantum Computing: tomorrow's computing, today, <https://www.ibm.com/quantum-computing/>, heruntergeladen am 19.05.2021.

Abb. 7: Tangle Lake Quantencomputerchip. Techproviderzone.com: Quantum computing gets closer with Intel's Tangle Lake CPU, <https://www.techproviderzone.com/advanced-technologies/quantum-computing-gets-closer-with-intel-s-tangle-lake-cpu>, heruntergeladen am 19.05.2021.

Abb. 8: Sich über die Zeit veränderndes Energiediagramm. dwavesys.com: What is Quantum Annealing?, How Quantum Annealing Works in D-Wave Systems, [https://docs.dwavesys.com/docs/latest/c\\_gs\\_2.html](https://docs.dwavesys.com/docs/latest/c_gs_2.html), heruntergeladen am 21.05.2021.

Abb. 9: Energiediagramm, bei dem die beste Antwort hervorgehoben ist. dwavesys.com: What is Quantum Annealing?, How Quantum Annealing Works in D-Wave Systems, [https://docs.dwavesys.com/docs/latest/c\\_gs\\_2.html](https://docs.dwavesys.com/docs/latest/c_gs_2.html), heruntergeladen am 21.05.2021.

Abb. 10: Quanten-Tunneling zwischen den Minima A, B, C und D, welches die thermische Aktivierungsenergie deutlich senkt. Google: Was ist Quanten-Annealing?, Quanten-Tunneling in einem Quanten-Annealer, <https://www.datacenter-insider.de/was-ist-quanten-annealing-a-782266/>, heruntergeladen am 22.05.2021.

Abb. 11: Beispiel eines grafisch dargestellten Optimierungsproblems. Helmich, U/u-helmich.de: Optimierungsprobleme, Die Klasse StadtApp, <http://www.u-helmich.de/inf/TSP/TSP-NN2.html>, heruntergeladen am 22.05.2021.

Abb. 12: D-Waves 2000Q, die erste Quanten-Annealing-Maschine mit 2000 Qubits. Barb, Bruce: Was ist Quanten-Annealing?, Quantum-Annealing, <https://www.datacenter-insider.de/was-ist-quanten-annealing-a-782266/>, heruntergeladen am 23.05.2021.

Abb. 13: Basiszustand eines Qubits in dem Quantencomputersimulator Quirk. Datenquelle: [https://algassert.com/quirk#circuit={%22cols%22:\[\]%22}](https://algassert.com/quirk#circuit={%22cols%22:[]%22}), heruntergeladen am 05.06.2021.

Abb. 14: Wirken des Pauli-X-Gatters auf den Basiszustand  $|0\rangle$ , Datenquelle: [https://algassert.com/quirk#circuit={%22cols%22:\[\]%22}](https://algassert.com/quirk#circuit={%22cols%22:[]%22}), heruntergeladen am 05.06.2021.

Abb. 15: Wirken des Pauli-Y-Gatters auf den Basiszustand  $|0\rangle$ , Datenquelle: [https://algassert.com/quirk#circuit={%22cols%22:\[\]%22}](https://algassert.com/quirk#circuit={%22cols%22:[]%22}), heruntergeladen am 12.06.2021.

Abb. 16: Wirken des Pauli-X-Gatters auf den Basiszustand  $|0\rangle$ , Datenquelle: [https://algassert.com/quirk#circuit={%22cols%22:\[\]%22}](https://algassert.com/quirk#circuit={%22cols%22:[]%22}), heruntergeladen am 12.06.2021.

Abb. 17: Wirken des Hadamard-Gatters auf den Basiszustand  $|0\rangle$ , Datenquelle: [https://algassert.com/quirk#circuit={%22cols%22:\[\]%22}](https://algassert.com/quirk#circuit={%22cols%22:[]%22}), heruntergeladen am 05.06.2021.

Abb. 18: Wirken des S-Gatters auf den Basiszustand  $|0\rangle$ , Datenquelle: [https://algassert.com/quirk#circuit={%22cols%22:\[\]%22}](https://algassert.com/quirk#circuit={%22cols%22:[]%22}), heruntergeladen am 05.06.2021.

Abb. 19: Wirken des CNOT-Gatters auf zwei Qubits mit je dem Basiszustand  $|0\rangle$ , Datenquelle: [https://algassert.com/quirk#circuit={%22cols%22:\[\]%22}](https://algassert.com/quirk#circuit={%22cols%22:[]%22}), heruntergeladen am 05.06.2021.

Abb. 20: Wirken des Bell Zustandes auf zwei Qubits mit je dem Basiszustand  $|0\rangle$ , Datenquelle: [https://algassert.com/quirk#circuit={%22cols%22:\[\]%22}](https://algassert.com/quirk#circuit={%22cols%22:[]%22}), heruntergeladen am 05.06.2021.

Abb. 21: Einfache Quantenschaltung mit einem Hadamard-Gatter und einem CNOT-Gatter, auch Bell State genannt. Datenquelle: <https://quantum-circuits.com>, heruntergeladen am: 12.06.2021.

Abb. 22: Auszahlungsmatrix beim Gefangenendilemma (Eigene Darstellung)

Abb. 23: Auszahlungsergebnisse beim Gefangenendilemma (Eigene Darstellung)

Abb. 24: Vereinfachte Auszahlungsmatrix beim Gefangenendilemma (Eigene Darstellung)

Abb. 25: Auszahlungsmatrix beim Coin Flip (Eigene Darstellung)

Abb. 26: Ablauf des Coin Flip (Eigene Darstellung)

Abb. 27: Auszahlungsmatrix beim Quanten Coin Flip (Eigene Darstellung)

Abb. 28: Ablauf des Quanten Coin Flip (Eigene Darstellung)

Abb. 29: Komponenten des Coin Flip Prototyp (Eigene Darstellung)

Abb. 30: Flowchart des Programms des Coin Flip Prototyp (Eigene Darstellung)

Abb. 31: Zustand 0 der Bloch Sphere (Eigene Darstellung)

Abb. 32: Spielbaum für Tic-Tac-Toe. Intermathematics: Game Theory and Tic Tac Toe: Game Theory and Tic Tac Toe,  
 Datenquelle: <https://ibmathsresources.com/2013/11/24/game-theory-and-tic-tac-toe/>,  
 heruntergeladen am: 29.09.2021.

Abb. 33: Antwort für das beste Ergebnis für X bei einem Tic-Tac-Toe-Spiel: Game Theory and Tic Tac Toe: Game Theory and Tic Tac Toe,  
 Datenquelle: <https://ibmathsresources.com/2013/11/24/game-theory-and-tic-tac-toe/>,  
 heruntergeladen am: 29.09.2021.

Abb. 34: Bilder von dem Prototyp nach Nagy & Nagy (Eigene Darstellung)

Abb. 35: Bilder von dem Prototyp nach Sagole (Eigene Darstellung)

Abb. 36: Bilder von dem Prototyp mit vollem Funktionsumfang (Eigene Darstellung)

Abb. 37: Komponenten des Tic-Tac-Toe Prototyp (Eigene Darstellung)

Abb. 38: Flowchart des Programms des Tic-Tac-Toe Prototyp (Eigene Darstellung)

Abb. 39: Bilder von dem Prototyp der Variante 1 (Eigene Darstellung)

Abb. 40: Flowchart der zweiten Version des Quanten Tic-Tac-Toe (Eigene Darstellung)

Abb. 41: Bilder von dem Prototyp der Variante 2 (Eigene Darstellung)

Abb. 42: Leeres Spielfeld und erster Zug von Spieler X (Eigene Darstellung)

Abb. 43: Erster Zug von Spieler O (Eigene Darstellung)

Abb. 44: Zweiter Zug von Spieler X (Eigene Darstellung)

Abb. 45: Nächste Züge von Spieler X und O (Eigene Darstellung)

Abb. 46: Nächsten Züge von Spieler X und O (Eigene Darstellung)

Abb. 47: Letzter Zug von Spieler X und gemessenes Spielfeld mit dem gemessenen Zustand 1 (Eigene Darstellung)

Abb. 48: Letzter Zug von Spieler X und gemessenes Spielfeld mit dem gemessenen Zustand 0 (Eigene Darstellung)

Abb. 49: Quantenschaltung des Tic-Tac-Toe Spiels (Eigene Darstellung)

Abb. 50: Anderer Spielverlauf mit den optimalen Zügen für Spieler X (Eigene Darstellung)

Abb. 51: Spielverlauf der Variante 2 des Quanten Tic-Tac-Toe (Eigene Darstellung)

Abb. 52: Quantenschaltung des obigen Spielverlaufs der Variante 2 des Quanten Tic-Tac-Toe (Eigene Darstellung)