

Quantum Penny Flip - Spielen mit dem Quantencomputer

Rainer Müller & Franziska Greinert

1. Einleitung

Die neuen Quantentechnologien finden derzeit große Beachtung in der Öffentlichkeit und den Medien. Auf besonderes Interesse stößt dabei die nachdrücklich vorangetriebene Entwicklung von Quantencomputern. In Zeitschriften und Internet-Newskanälen wird regelmäßig über die Fortschritte berichtet. Die hohe öffentliche Aufmerksamkeit zeigt, dass Quantencomputer als ein interessantes Thema empfunden werden. Wir wollen diese motivierende Wirkung für das Lernen der Quantenphysik nutzbar machen. Speziell möchten wir die Möglichkeiten nutzen, die sich durch den Zugang zu echten Quantencomputern bieten, den verschiedene Anbieter kostenlos zur Verfügung stellen.

Auf verschiedenen Plattformen (z. B. IBM Quantum [1] oder Quantum Inspire der TU Delft [2]) kann man sich für einen cloudbasierten Quantencomputer-Zugang registrieren und Quantenalgorithmien auf realer Quantenhardware erproben. Daneben gibt es benutzerfreundliche Simulatoren wie Quirk [3], [4] und Umgebungen mit teilweise umfangreichen Lernmaterialien, in denen man hardwarenahe Programmiersprachen erlernen kann (z. B. IBM Qiskit [5], Microsoft Q# [6] oder Google Cirq [7]). Regelmäßig kommen neue Anbieter und Ansätze dazu [8]. Eine Übersicht über frei zugängliche Quanten-Programmierbibliotheken und andere Angebote bietet die fortlaufend aktualisierte GitHub-Sammlung „Open-Source Quantum Software Projects“ [9].

Ein Vorteil des Zugangs zur Quantenphysik über die Quantentechnologien liegt darin, dass die hier betrachteten Systeme prinzipiell einfach sind. Qubits lassen sich als Zwei-Zustands-Systeme beschreiben – es sind die einfachst denkbaren Quantensysteme. Auch die Polarisationsfreiheitsgrade von Licht, die in der Quantenkommunikation eine große Rolle spielen, lassen sich formal sehr einfach beschreiben. Als weiterer didaktischer Vorteil kommt dazu, dass die Quantentechnologien direkt an den nichtklassischen Merkmalen der Quantenphysik ansetzen – wichtige Themen wie Superposition, Messung und Verschränkung sind wesentlich für ihre Funktion. Die Physik, die den neuen Quantentechnologien zugrunde liegt, ist nicht neu: Es ist unverändert die gleiche Quantenphysik, die in den Zeiten von Heisenberg und Schrödinger entwickelt wurde. Aber ihre technische Anwendung in Quantencomputern, mit neuen Begriffen wie Qubits und Quantengattern, erlaubt einen neuen Zugang, neue Aufgabenstellungen und neue inhaltliche Schwerpunkte in der Lehre der Quantenphysik. In diesem Artikel möchten wir einen möglichst direkten und unkomplizierten Weg aufzeigen, Quantencomputer in einem einführenden Kurs zur Quantenphysik zu nutzen. Die technischen Möglichkeiten dafür sind bereits gegeben: Die oben aufgeführten Plattformen geben Lernenden die Möglichkeit, mit echten Quantencomputern umzugehen und sie in intuitiv bedienbaren Entwicklungsumgebungen eigenständig zu programmieren.

Es gibt jedoch ein Hindernis bei diesem neuen und motivierenden Zugang zur Quantenphysik: Die etablierten Quantenalgorithmien sind kompliziert und bleiben deshalb für Lernende oft unverständlich. Um dieses Problem zu lösen, greifen wir auf einen Algorithmus zurück, der ein einfaches Problem löst und dabei einen Quantenvorteil bietet. Es handelt sich allerdings nicht um ein klassisches Problem aus der Informatik, sondern um ein simples Spiel, bei dem die Ausnutzung der Quantenphysik einen Gewinnvorteil bietet. Es heißt Quantum Penny Flip und wurde bereits 1999 von David A. Meyer vorgeschlagen [10]. Das Spiel lässt sich auf einfache Weise so umformulieren, dass es mit Quantengattern beschreibbar und damit auf einem Quantencomputer implementierbar ist.

Das „Quantum Penny Flip“-Spiel wurde bereits als Videospiel umgesetzt, einzeln [11], [12] oder in einer Kombination mit weiteren kleinen Quantenspielen [13]. Dabei steht jedoch das Spielen und Erleben der Quantenphänomene im Fokus, nicht die Umsetzung als Algorithmus auf einem Quantencomputer. Zudem müssen für die Videospiele Programme installiert werden, was für den Einsatz in der Lehre eine Hürde darstellen kann.

In diesem Artikel beschreiben wir, wie sich das „Quantum Penny Flip“-Spiel für einen Einstieg in die gatterbasierte Quantenprogrammierung nutzen lässt. Mit dem Spiel lässt sich eine Form der Quantenüberlegenheit einfach und überzeugend demonstrieren. Wir nutzen zur Umsetzung den Quantum Composer von IBM [14], der keine Installation erfordert. Die grafische Programmierumgebung lässt sich ohne Registrierung nutzen. Lediglich wenn ein realer Quantencomputers zur Ausführung des Algorithmus verwendet werden soll, ist ein kostenloser Account nötig.

Unsere Zielgruppe sind Studierende in Einführungskursen zur Quantenphysik und auch Schülerinnen und Schüler, die sich für das aktuelle Forschungsgebiet interessieren. Mathematische Voraussetzung ist ein elementares Verständnis des Dirac'schen Bra-Ket-Formalismus, von dem allerdings nur die einfachsten Grundlagen benötigt werden. Das Ziel ist, die Motivation und die Lernmöglichkeiten zu nutzen, die der Zugang zu einem echten Quantencomputer mit sich bringt. Wir beschreiben im Folgenden den Quantum-Penny-Flip-Algorithmus und lernen bei seiner Umsetzung einige der wichtigsten Quantengatter kennen.

2. Ein einfacher Münzwurf-Algorithmus

Beim Quantum Penny Flip handelt es sich um ein Spiel mit zwei Teilnehmern, die abwechselnd eine Folge von Zügen ausführen. Ausgangssituation ist der klassische Münzwurf mit zwei Spielern (Alice und Bob): Eine Münze wird in die Luft geworfen. Wenn sie bei der Landung „Kopf“ zeigt, hat Alice gewonnen. Zeigt sie „Zahl“, hat Bob gewonnen. Um das Spiel anschlussfähig an den Quantenalgorithmus zu machen, formulieren wir die Regeln für die klassische Variante des Spiels etwas um:

1. Alice legt die Münze in einem Zustand ihrer Wahl (Kopf oder Zahl oben) in eine nicht einsehbare Kiste. Auch durch Ertasten ist nicht festzustellen, welche Seite oben liegt.
2. Bob greift in die Kiste und hat die Wahl, die Münze umzudrehen oder nicht. Alice kann nicht sehen, was er tut.
3. Alice greift in die Kiste und führt eine Operation ihrer Wahl durch (Umdrehen oder Nicht-Umdrehen).
4. Die Münze wird aufgedeckt und das Ergebnis abgelesen.

Wie in Abb. 1 gezeigt, haben wir das Spiel mit diesen Regeln schon in der klassischen Version nach dem Schema „Präparation – Wechselwirkung – Messung“ formuliert, das oft zur Beschreibung quantenmechanischer Prozesse verwendet wird [15]. Die die beiden Operationen von Alice und Bob in der Mitte von Abb. 1 sind hier die Wechselwirkung.

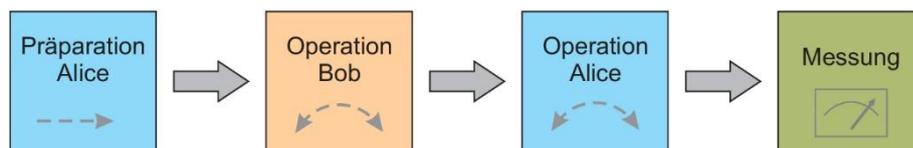


Abbildung 1: Das Spiel Quantum Penny Flip in dem Schema Präparation – Wechselwirkung – Messung.

Auch wenn die Rollen von Bob und Alice nicht ganz symmetrisch verteilt sind, ist es doch klar, dass es für keinen der beiden Spieler eine Gewinnstrategie gibt. Die Chancen sind gleich verteilt. Umso erstaunlicher ist es, dass es in der Quantenversion eine klare Gewinnstrategie für Alice gibt. Diese überraschende Existenz eines Quantenvorteils in einer so einfachen und übersichtlichen Situation möchten wir in unserem Zugang für didaktische Zwecke nutzbar machen. Unser Algorithmus verwendet nur ein einzelnes Qubit mit zwei Zuständen. Es handelt sich also um das einfachst denkbare Quantensystem.

3. Quantenversion des Algorithmus: Quantum Penny Flip

Die Quantenversion des Algorithmus läuft ganz gemäß den gerade aufgestellten Regeln ab. Der wesentliche Unterschied: An die Stelle der klassischen Münze tritt ein Qubit. Qubits sind die kleinste Grundeinheit eines Quantencomputers. Sie werden durch zwei Basiszustände $|0\rangle$ und $|1\rangle$ beschrieben. Zur grundsätzlichen Diskussion von Quantenalgorithmus muss man über die physikalische Realisierung dieser Zustände zunächst nichts wissen. Es kann sich um Polarisationsfreiheitsgerade von Licht, um Hyperfeinzustände von gefangenen Ionen oder um Elektronenzustände in Supraleitern handeln – wir können zunächst davon abstrahieren. Die Anforderungen an die physikalische Realisierung von Qubits, von denen hier hauptsächlich die Existenz langlebiger Überlagerungszustände relevant ist, wurde von DiVincenzo in einem Satz von fünf Kriterien zusammengefasst [16].

Für unser Spiel nehmen wir an, dass Bob auf die Operationen beschränkt ist, die auch in der klassischen Physik möglich sind. Die Auswahl ist für ihn klein: Er kann die Münze umdrehen oder nicht umdrehen. In der Informatik wird das „Umdrehen“ eines Bits durch ein NOT-Gatter beschrieben, das 0 und 1 vertauscht. Im Quantencomputing tritt an dessen Stelle das Pauli-X-Gatter, das unten näher erläutert wird. Das Nicht-Umdrehen wird durch die Identität $\mathbb{1}$ beschrieben.

Anders als Bob kennt Alice die Gesetze der Quantenphysik und kann an dem Qubit alle denkbaren Quantenoperationen vornehmen. Um dies zu diskutieren, müssen wir das Qubit und die daran vorgenommenen

Operationen in der Sprache der Quantenphysik beschreiben. Es gibt dazu drei Möglichkeiten: (1) im Diracschen Bra-Ket-Formalismus, (2) durch Vektoren mit zwei Komponenten, die Gatteroperationen werden dann durch 2×2 -Matrizen beschrieben, und (3) geometrisch mit Hilfe der Blochkugel [17]. Alle drei Wege sind leicht gangbar; in dieser Arbeit wählen wir den ersten Zugang.

Die orthogonalen Basiszustände $|0\rangle$ und $|1\rangle$ des Qubits lassen sich zum allgemeinen Zustand

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

überlagern. Dabei sind α und β komplexe Koeffizienten mit $|\alpha|^2 + |\beta|^2 = 1$. Im Quantencomputing wird generell angenommen, dass jedes Qubit zu Beginn im Zustand $|0\rangle$ präpariert ist.

4. Qubit-Operationen

Nach dem Vorbild der klassischen Informatik werden Operationen an Qubits durch Gatter beschrieben. Sie beschreiben wohldefinierte Aktionen, die den Zustand eines Qubits auf gezielte Weise verändern. Physikalisch geschieht das zum Beispiel durch Mikrowellenpulse von bestimmter Dauer und Feldstärke. Die Beschreibung durch Quantengatter abstrahiert von der konkreten physikalischen Realisierung.

Für den Quantum-Penny-Flip-Algorithmus sind die in Abb. 2 dargestellten Gatter relevant. Neben der Identitäts-Operation $\mathbb{1}$, die keine Änderung am Zustand vornimmt, sind es das quantenmechanische NOT-Gatter (das, wie schon erwähnt, auch als Pauli-X-Gatter bezeichnet wird) und das Hadamard-Gatter. Weiterführende Informationen zu Quantengattern findet man in Büchern zum Quantencomputing (wie [17]) und zahlreichen Webseiten (z. B. [18]).

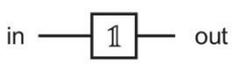
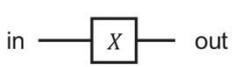
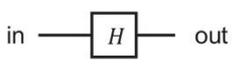
Identität		<table style="border-collapse: collapse; border: none;"> <thead> <tr> <th style="border-right: 1px solid black; padding: 5px;">in</th> <th style="padding: 5px;">out</th> </tr> </thead> <tbody> <tr> <td style="border-right: 1px solid black; padding: 5px;">$0\rangle$</td> <td style="padding: 5px;">$0\rangle$</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$1\rangle$</td> <td style="padding: 5px;">$1\rangle$</td> </tr> </tbody> </table>	in	out	$ 0\rangle$	$ 0\rangle$	$ 1\rangle$	$ 1\rangle$
in	out							
$ 0\rangle$	$ 0\rangle$							
$ 1\rangle$	$ 1\rangle$							
Pauli-X		<table style="border-collapse: collapse; border: none;"> <thead> <tr> <th style="border-right: 1px solid black; padding: 5px;">in</th> <th style="padding: 5px;">out</th> </tr> </thead> <tbody> <tr> <td style="border-right: 1px solid black; padding: 5px;">$0\rangle$</td> <td style="padding: 5px;">$1\rangle$</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$1\rangle$</td> <td style="padding: 5px;">$0\rangle$</td> </tr> </tbody> </table>	in	out	$ 0\rangle$	$ 1\rangle$	$ 1\rangle$	$ 0\rangle$
in	out							
$ 0\rangle$	$ 1\rangle$							
$ 1\rangle$	$ 0\rangle$							
Hadamard		<table style="border-collapse: collapse; border: none;"> <thead> <tr> <th style="border-right: 1px solid black; padding: 5px;">in</th> <th style="padding: 5px;">out</th> </tr> </thead> <tbody> <tr> <td style="border-right: 1px solid black; padding: 5px;">$0\rangle$</td> <td style="padding: 5px;">$\frac{1}{\sqrt{2}} (0\rangle + 1\rangle)$</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$1\rangle$</td> <td style="padding: 5px;">$\frac{1}{\sqrt{2}} (0\rangle - 1\rangle)$</td> </tr> </tbody> </table>	in	out	$ 0\rangle$	$\frac{1}{\sqrt{2}} (0\rangle + 1\rangle)$	$ 1\rangle$	$\frac{1}{\sqrt{2}} (0\rangle - 1\rangle)$
in	out							
$ 0\rangle$	$\frac{1}{\sqrt{2}} (0\rangle + 1\rangle)$							
$ 1\rangle$	$\frac{1}{\sqrt{2}} (0\rangle - 1\rangle)$							

Abbildung 2: Die für den Quantum-Penny-Flip-Algorithmus benötigten Quantengatter.

Pauli-X-Gatter: Das Pauli-X-Gatter wandelt den Basiszustand $|0\rangle$ in $|1\rangle$ um und umgekehrt den Basiszustand $|1\rangle$ in $|0\rangle$. Es beschreibt somit das Umdrehen der Münze. In Dirac-Notation wird es durch den Ausdruck

$$X = |0\rangle \langle 1| + |1\rangle \langle 0|$$

dargestellt. Wir überprüfen, dass sich das Gatter gemäß der Wahrheitstabelle in Abb. 2 verhält, indem wir es zum Beispiel auf den Basiszustand $|0\rangle$ anwenden. Es ergibt sich:

$$|0\rangle = [|0\rangle \langle 1| + |1\rangle \langle 0|] |0\rangle.$$

Das formale Ausmultiplizieren der eckigen Klammern führt zu den beiden Skalarprodukten $\langle 1|0\rangle = 0$ und $\langle 0|0\rangle = 1$. Wenn wir dies einsetzen, erhalten wir wie erwartet:

$$X |0\rangle = |0\rangle \langle 1|0\rangle + |1\rangle \langle 0|0\rangle = |1\rangle.$$

Analog ergibt sich bei Anwendung von X auf den Basisvektor $|1\rangle$:

$$X |1\rangle = [|0\rangle \langle 1| + |1\rangle \langle 0|] |1\rangle = |0\rangle \underbrace{\langle 1|1\rangle}_{=1} + |1\rangle \underbrace{\langle 0|1\rangle}_{=0} = |0\rangle$$

Der Operator X wirkt also auf die Basisvektoren so wie in der Wahrheitstabelle in Abb. 2 dargestellt. Anders als das klassische NOT-Gatter ist das Pauli-X-Gatter auch für quantenmechanische Überlagerungszustände definiert. Das Ergebnis folgt unmittelbar aus der Linearität der Operation. Wenn $|\psi_{in}\rangle = \alpha |0\rangle + \beta |1\rangle$ ist, dann ist:

$$|\psi_{out}\rangle = X |\psi_{in}\rangle = \alpha |1\rangle + \beta |0\rangle.$$

Hadamard-Gatter: Das Hadamard-Gatter H überführt die Basiszustände $|0\rangle$ und $|1\rangle$ in Überlagerungszustände:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad \text{bzw.} \quad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

Es ist eines der in Quantenalgorithmen am häufigsten verwendeten Gatter, denn es erzeugt Überlagerungszustände. Sie sind das kennzeichnende Merkmal von Quantencomputern und bilden die Voraussetzung für das Ausnutzen des Quantenvorteils. Es wird daher meist schon ganz am Anfang eines Quantenalgorithmus eingesetzt.

In Dirac-Notation lautet der Ausdruck für das Hadamard-Gatter wie folgt:

$$H = \frac{1}{\sqrt{2}}[|0\rangle\langle 0| + |1\rangle\langle 0| + |0\rangle\langle 1| - |1\rangle\langle 1|]$$

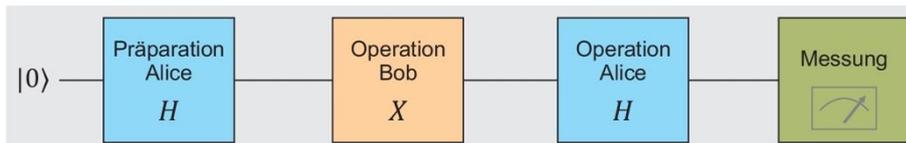
Die Wirkung auf die Basiszustände lässt sich in gleicher Weise wie beim X -Gatter überprüfen. Mit diesen beiden Quantengattern, die auch für komplexere Algorithmen wichtig sind, können wir den Gewinnalgorithmus beim Quantum Penny Flip diskutieren.

5. Die Gewinnstrategie beim Quantum Penny Flip

Wir hatten schon erwähnt, dass Alice in der Quantenversion des Spiels alle Quantengatter benutzen darf. Das können wir nun präzisieren: Ihr steht insbesondere das Hadamard-Gatter zur Verfügung. Bob bleibt weiterhin nur die Wahl zwischen Umdrehen (entspricht Gatter X) oder Nichtumdrehen (entspricht Gatter $\mathbb{1}$). Die Gewinnstrategie von Alice beim Quantum Penny Flip lässt sich ganz einfach formulieren: Immer wenn sie an der Reihe ist, führt sie die Operation H an der Münze durch. Wenn sie dieser Strategie folgt, gewinnt sie – egal was Bob tut.

Abb. 3 zeigt die Abfolge der möglichen Qubit-Operationen. Zu Beginn wird das Qubit im Zustand $|0\rangle$ präpariert. Wie zuvor ist zuerst Alice an der Reihe, dann Bob, dann wieder Alice. Wenn Alice ihrer Gewinnstrategie folgt und immer H einsetzt, gibt es zwei Möglichkeiten für den Spielverlauf: Bob kann die Münze umdrehen (Operation X) oder nicht (Operation $\mathbb{1}$). Mit den zuvor angegebenen expliziten Ausdrücken für die Quantengatter können wir das Ergebnis in beiden Fällen bestimmen.

a) Bob dreht um



b) Bob dreht nicht um

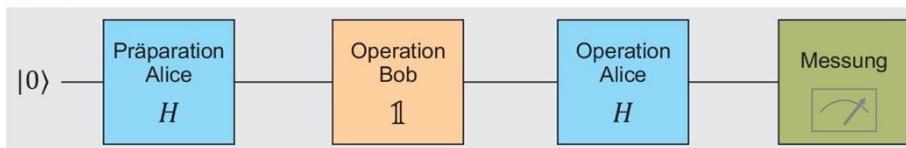


Abbildung 3: Die Abfolge der möglichen Qubit-Operationen beim Quantum Penny Flip.

Fall a) Bob dreht um

Für den ersten Fall (Bob dreht um) ergibt sich der Endzustand des Qubits $|\psi_{out}\rangle$ durch die aufeinanderfolgende Anwendung der Operatoren H , X und H auf den Anfangszustand $|\psi_{in}\rangle = |0\rangle$:

$$\begin{aligned} |\psi_{out}\rangle &= H X H |0\rangle = \frac{1}{\sqrt{2}}[|0\rangle\langle 0| + |1\rangle\langle 0| + |0\rangle\langle 1| - |1\rangle\langle 1|] \cdot \frac{1}{\sqrt{2}}[|0\rangle\langle 1| + |1\rangle\langle 0|] \\ &\quad \cdot \frac{1}{\sqrt{2}}[|0\rangle\langle 0| + |1\rangle\langle 0| + |0\rangle\langle 1| - |1\rangle\langle 1|]|0\rangle \end{aligned}$$

Das Ausmultiplizieren der Terme wird erleichtert, wenn man $\langle 0|1\rangle = 0$ berücksichtigt und die entsprechenden Terme erst gar nicht aufschreibt. Auf diese Weise ergibt sich recht schnell:

$$|\psi_{out}\rangle = [|0\rangle\langle 0| - |1\rangle\langle 1|]|0\rangle = |0\rangle.$$

Es ergibt sich also $|0\rangle = \text{Kopf}$, und Alice hat gewonnen.

Fall b) Bob dreht nicht um

Die zweite Möglichkeit (Bob dreht nicht um) ist noch einfacher zu behandeln. Da der Einheitsoperator $\mathbb{1} = |0\rangle\langle 0| + |1\rangle\langle 1|$ das Nichts-Tun repräsentiert, hat seine Anwesenheit keinen Einfluss auf das Endergebnis. Auf das Qubit wirken also zwei H -Gatter hintereinander.

$$|\psi_{out}\rangle = H \mathbb{1} H |0\rangle = HH |0\rangle = \frac{1}{\sqrt{2}} [|0\rangle\langle 0| + |1\rangle\langle 0| + |0\rangle\langle 1| - |1\rangle\langle 1|] \\ \cdot \frac{1}{\sqrt{2}} [|0\rangle\langle 0| + |1\rangle\langle 0| + |0\rangle\langle 1| - |1\rangle\langle 1|]|0\rangle$$

Die Berechnung erfolgt nach dem gleichen Schema wie oben, und wir finden:

$$|\psi_{out}\rangle = [|0\rangle\langle 0| + |1\rangle\langle 1|]|0\rangle.$$

Die zweimalige Anwendung des Hadamard-Gatter führt wieder zum Ausgangszustand zurück. Das Hadamard-Gatter ist sein eigenes Inverses. Als Ergebnis tritt auch in diesem Fall Kopf auf, und wieder hat Alice gewonnen.

Insgesamt haben wir also eine sichere Gewinnstrategie für Alice gefunden. Einerlei, was Bob tut – umdrehen oder nicht umdrehen: Sofern Alice bei beiden Zügen das Hadamard-Gatter anwendet, kann sie nicht verlieren. Das ist ein augenfälliger Unterschied zur klassischen Spielvariante, wo für keinen der beiden Spieler eine Gewinnstrategie existiert.

Es ist aufschlussreich, den quantenmechanischen Hintergrund dieses Erfolges aufzudecken: Superpositionszustände sind ein Charakteristikum von Quantenzuständen, das sie von der klassischen Physik unterscheidet. Der durch Anwendung des Hadamard-Gatters erzeugte Überlagerungszustand $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ist invariant gegenüber der Vertauschung von $|0\rangle$ und $|1\rangle$ – also gegenüber der Anwendung des X -Gatters. Das Ausnutzen von dieser Eigenschaft des Superpositionszustands ist entscheidend für den Erfolg von Alice und bildet die Basis für ihre Gewinnstrategie.

6. Implementation auf einem echten Quantencomputer

Zur Implementation des Quantum-Penny-Flip-Algorithmus nutzen wir den Quantum Composer von IBM [14]. Die Programmierung erfolgt hier grafisch mit Symbolen für die Quantengatter (oben in Abb. 4). Die einzelnen Qubits werden durch horizontale Linien dargestellt, die mit q_0, q_1, \dots beschriftet sind (im vorliegenden Fall benötigen wir nur ein Qubit q_0). Die verschiedenen Gatter können auf diesen „Drähten“ per drag-and-drop platziert werden. Für jedes Qubit wird der Anfangszustand $|0\rangle$ angenommen.

Für Lernende ist es vielleicht zunächst verwirrend, dass der Algorithmus in der grafischen Darstellung von links nach rechts gelesen wird, während die Reihenfolge der Operatoren im im Bra-Ket-Formalismus umgekehrt ist. Beispielsweise werden die Operationen $|\psi_0\rangle = XH|0\rangle$ in der grafischen Darstellung durch $q_0 - [H] - [X]$ – dargestellt.

Die abschließende Messung am Qubit, mit der das Endergebnis ausgelesen wird, muss im Quantum Composer explizit angegeben werden. Sie wird durch ein stilisiertes Messgerät symbolisiert und kann ebenso wie die Gatter platziert werden. Zur Speicherung des Messergebnisses dient das zusätzliche klassische Bit, das durch eine horizontale Doppellinie dargestellt wird (in Abb. 4 mit c_1 bezeichnet).

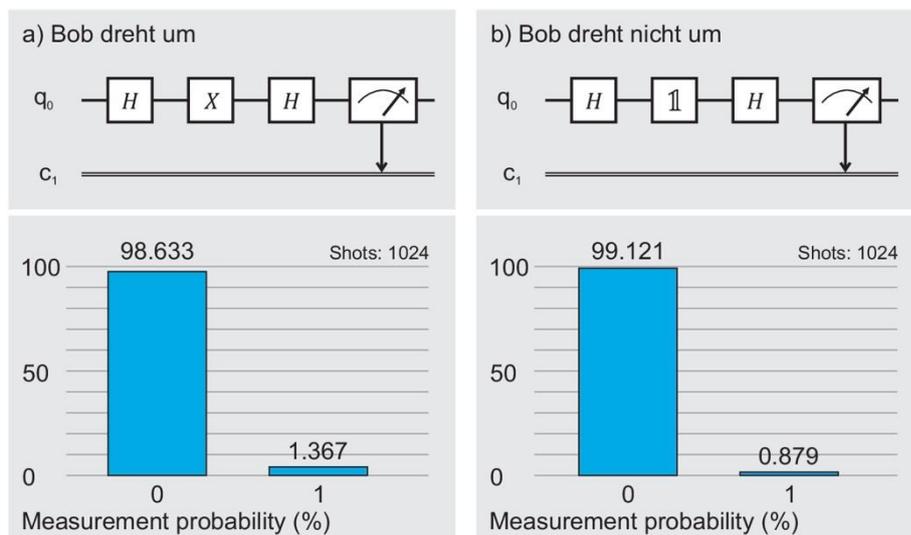


Abbildung 4: Ergebnis der Berechnung auf einem Quantencomputer. Das klassische Bit c_1 enthält am Ende das Ergebnis der Quantenmessung am Qubit q_0 .

Die symbolische Darstellung der Programmierung für die beiden Fälle unseres Spiels ist in Abb. 4 oben gezeigt. Im Quantum Composer werden der simulierte Zustandsvektor und die simulierten Messergebnisse unmittelbar angezeigt. Man kann aber die Berechnung auch von einem echten Quantencomputer durchführen lassen. Unten in Abb. 4 sind die Ergebnisse einer solchen Berechnung dargestellt. Aufgrund von Dekohärenz sind die Effekte rauschbehaftet, so dass bei der Messung nicht immer das erwartete Ergebnis gefunden wird. Deshalb wurden für jeden der beiden Fälle 1024 Durchläufe durchgeführt. Das Resultat bestätigt die Gewinnstrategie für Alice: Abgesehen vom Rauschen ergibt sich in allen Fällen bei der Messung am Qubit q_0 der Output 0 (= Kopf).

An den Ergebnissen ist auch zu erkennen, dass jede Bitoperation mit zusätzlichem Rauschen verbunden ist. Wenn Bob umdreht, müssen drei Operationen am Qubit durchgeführt werden, beim Nicht-Umdrehen nur zwei (Gatter $\mathbb{1}$ = Nichts-Tun). An den Zahlenwerten in Abb. 4 lässt sich ablesen, dass im ersten Fall die Abweichung vom theoretisch erwarteten Ergebnis größer ist als im zweiten.

7. Literatur

- [1] „IBM Quantum“, 2021. <https://quantum-computing.ibm.com/> (zugegriffen 11.08.2021).
- [2] „Quantum Inspire“, QuTech, 2018. <https://www.quantum-inspire.com/> (zugegriffen 11.08.2021).
- [3] „Quirk“, Quantum Circuit Simulator, Version 2.3. <https://algassert.com/quirk> (zugegriffen 29.07.2021).
- [4] C. Gidney, „How to use Quirk“, GitHub: Strilanc/Quirk Wiki. <https://github.com/Strilanc/Quirk> (zugegriffen 29.07.2021).
- [5] „Qiskit: An Open-source Framework for Quantum Computing“, Qiskit Development Team, 2021. <https://qiskit.org/> (zugegriffen 29.07.2021).
- [6] „Q# und das Quantum Development Kit“, Microsoft Azure, 2021. <https://azure.microsoft.com/de-de/resources/development-kit/quantum-computing/> (zugegriffen 29.07.2021).
- [7] „Cirq“, Cirq Developers, Google Quantum AI, 2019. <https://quantumai.google/cirq?hl=de> (zugegriffen 29.07.2021).
- [8] D. Matthews, „How to get started in quantum computing“, Nature, Bd. 591, Nr. 7848, S. 166–167, März 2021, doi: 10.1038/d41586-021-00533-x.
- [9] M. Fingerhuth und Contributors, „Curated list of open-source quantum software projects“, GitHub, 14. Juli 2021. <https://github.com/qosf/awesome-quantum-software> (zugegriffen 29. Juli 2021).
- [10] D. A. Meyer, „Quantum Strategies“, Phys. Rev. Lett., Bd. 82, Nr. 5, S. 1052–1055, Feb. 1999, doi: 10.1103/PhysRevLett.82.1052.

- [11] E. Zapirain, S. M. Massa, und J. P. Cardoso, „Quantum Penny Flip: an open source serious game for quantum computing assessment“, gehalten auf der XIV Congreso Nacional de Tecnología en Educación y Educación en Tecnología (TE&ET 2019), (Universidad Nacional de San Luis, 1 y 2 de julio de 2019), 2019.
- [12] E. Zapirain und J. P. Cardoso, „Quantum Penny Flip“, GitHub: [estebanzapirain/qpenny-flip](https://github.com/estebanzapirain/qpenny-flip), 14. Januar 2020. <https://github.com/estebanzapirain/qpenny-flip> (zugegriffen 30.07.2021).
- [13] M. S. Alam, J. Wootton, und J. DuBois, „A Link to Quantum“, 2019. Zugegriffen: 30.07.2021. [Online]. Verfügbar unter: https://github.com/msohaibalam/Link_to_Quantum_game
- [14] „IBM Quantum Composer“, 2021. <https://quantum-computing.ibm.com/composer> (zugegriffen 29.07.2021).
- [15] L. E. Ballentine, Quantum mechanics: a modern development. Singapore; River Edge, NJ: World Scientific, 1998.
- [16] D. P. DiVincenzo, „The Physical Implementation of Quantum Computation“, Fortschritte Phys., Bd. 48, Nr. 9–11, S. 771–783, 2000,
- [17] E. G. Rieffel und W. H. Polak, Quantum Computing: A Gentle Introduction. Cambridge, MA, USA: MIT Press, 2011.
- [18] „Qiskit Textbook“, Learn Quantum Computation Using Qiskit., 2020. <https://community.qiskit.org/textbook/> (zugegriffen 29.07.2021).

Affiliation

Rainer Müller, Technische Universität Braunschweig, Institut für Fachdidaktik der Naturwissenschaften, Physik und Physikdidaktik

Franziska Greinert, Technische Universität Braunschweig, Institut für Fachdidaktik der Naturwissenschaften, Physik und Physikdidaktik