The Joy of Quantum Computing

A Concise Introduction

Jed Brody

The Joy of Quantum Computing

The Joy of Quantum Computing

A Concise Introduction

Jed Brody

Princeton University Press
Princeton and Oxford

Published by Princeton University Press
41 William Street, Princeton, New Jersey 08540
99 Banbury Road, Oxford OX2 6JX

press.princeton.edu

GPSR Authorized Representative: Easy Access System Europe - Mustamäe tee 50,
10621 Tallinn, Estonia, gpsr.requests@easproject.com

ISBN 9780691264493
ISBN (pbk.) 9780691264486
ISBN (e-book) 9780691264479

Library of Congress Control Number: 2025930793

British Library Cataloging-in-Publication Data is available

Editorial: Hallie Stebbins and Chloe Coy
Production Editorial: Mark Bellis
Text and Cover Design: Wanda España
Production: Erin Suydam
Publicity: William Pagdatoon
Copyeditor: Jennifer Harris
Cover Credit: Maryna Stryzhak / Alamy Stock Vector

This book has been composed in Sabon

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Dedicated to Katherine Johnson, Dorothy Vaughan,
and Mary Jackson, the hidden figures of ***Hidden Figures***.
In the ongoing scientific revolution, may the contributions
of all people be welcomed, celebrated, and no longer hidden.

From this are & do come admirable adaptations.

—*The Emerald Tablet*, translated by Isaac Newton

Contents

Preface

Almost Too Much Awesome

I love quantum computing so much, I don’t know where to start. I’m tongue-tied. I’m delirious. I can’t contain my joy. I’m doing backflips off dumpsters. I’m riding shopping carts down stairs. I’m swinging by vines over rivers of lava. As in a song lyric I liked in high school, “I’m so bloated up happy I could throw things around me.” (“Heavenly Pop Hit” by The Chills.)

In high school, I had a book called *The Secret Guide to Computers*. It taught BASIC programming. The book opened a misty passageway to a world of almost mystical union with silicon circuitry. It was an initiation to a fellowship of advanced nerds. Because if you’re going to be a nerd, you might as well be an advanced one. Well, I’ve got news for you, advanced nerds. It’s time to take a quantum leap.

Quantum information science, the broader discipline that contains quantum computing, stands at a grand conjunction of computer science, digital electronics, engineering, quantum mechanics, linear algebra, number theory, and even philosophy. It’s a bustling crossroads of all my favorite nerdy pursuits. It’s almost too much awesome.

Yes, even philosophy is relevant to quantum information science, as we will see in the chapter about the 2022 Nobel Prize in Physics. As we continue to expand the frontiers of knowledge, the rigorous mysteries of quantum physics remain stubbornly unsolved. I wonder if this is a salutary check on human hubris, a reminder of our place in a world we never made. Physics achieves the goal of medieval alchemy and astrology, to illuminate the invisible forces that govern the destinies of all things. But something always scurries away from the light, and our thirst for complete understanding is forever unquenched.

Philosophical questions aside, quantum technology is advancing all the time, and the potential uses for quantum computers are exciting and fun to explore. Quantum computing is a new and growing field that students hunger to learn about, and instructors who are new to the field are desperate for books they can understand. (I speak from experience as an instructor who is new to the field.) I had to carefully study a dozen quantum computing books before I understood any of them. New instructors, as well as autodidactic hobbyists, need a large pool of resources. I hope I'm contributing to this pool.

I'm keeping the math as simple as I possibly can, and I'm avoiding matrices entirely, until the completely optional final two chapters. You do need to know some precalculus (algebra and occasional trigonometry). On the other hand, you don't need to know any quantum physics at all. I hope that our leap to the farthest Shor, over the howling abyss of quantum phase estimation, is not too daunting. Please don't feel bad if you have to skim some sections, or chew over them slowly like cud. It's also okay to skip some passages out of sheer boredom. Not every sentence can be a thrill, and the parts you skip are always there if you ever want to go back to them.

So without any further ado, onward to the awesome.

The Joy of Quantum Computing

Chapter 1

Forging the Quantum Key

There are a lot of reasons to keep data secret, accessible only to intended viewers. Examples include credit card numbers intended only for a seller, medical information intended only for health care providers, military intelligence intended only for allies, proprietary industrial processes intended only for collaborators, and photos from a meeting of the Nude Headstand Enthusiasts Club intended only for fellow club members (you *said* the site was password protected, Steve).

One way to keep data secure is to seal it in a bank vault, or in a safe wrapped with padlocked chains buried in a cobra-infested island in a sea swarming with sharks. The trouble with this kind of security is that data often needs to be shared. So we need a convenient way to share data remotely with intended recipients, and only with intended recipients.

All electronic data, whether text, images, videos, or anything else, is stored as combinations of 0's and 1's. 0 and 1 represent two different voltages in electronic circuits. The two voltages could be 0 volts and 1 volt, but that's not the only choice. The two voltages could be 0 volts and 5 volts; we simply use 1 to represent 5 volts. The two voltages could be –4 volts and 3.5 volts; we arbitrarily pick one of these to call 0, and the other to call 1. The point is that we can analyze the 0's and 1's in data without paying any attention to the physical details of how they're stored.

In fact, 0's and 1's can represent more than just voltages. The 0's and 1's in bar codes and QR codes are black and white stripes or squares. The 0's and 1's in CDs and DVDs are different thicknesses of a layer of plastic. As long as there are two, and only two, distinct physical conditions, we have 0's and 1's, and we can do classical computation.

Our electronic devices know how to convert 0's and 1's to videos, images, sounds, text, and so on. The details of this conversion are not our focus. We wish only to securely transmit 0's and 1's from a sender to a recipient, over a perilous distance fraught with eavesdroppers. In fact, we assume that eavesdroppers will be greedily poring over our data transmissions, combing through our 0's and 1's for valuable secrets.

So we have little choice, then, but to encrypt our data. We transmute our sequence of 0's and 1's into meaningless gibberish, a cipher, which only the intended recipient can decipher. There are many ways of achieving this. Near the end of our journey, we will meet the RSA cryptosystem, which is vulnerable to the quantum attack of Shor's algorithm. For now, we will consider a simpler cryptosystem: the private, or secret, key.

It's convenient to give names to the sender and receiver of data. The traditional names are Alice and Bob. But I think Alice and Bob deserve a vacation. So as Alice and Bob settle into their cozy rooms overlooking waves booming against a rocky shore silvered by moonlight, let's meet our new heroes, Odysseus and Penelope. Odysseus is rightly regarded as the most cunning of warriors. Less well known is that his wife Penelope is the most cunning of quantum engineers.

A 0 or 1 is called a *bit*. For each bit of the message that Penelope wants to send to Odysseus, she needs a secret *key* bit. The message bit is combined with the key bit to form an encrypted bit, according to these rules:

0 combined with 0 is 0.
0 combined with 1 is 1.
1 combined with 1 is 0.

In other words, if the message bit and the key bit are the same, the encrypted bit is 0. If the message bit and the key bit are different, the key bit is 1. There's a mathematical symbol, ⊕, called "exclusive OR," that represents these rules:

$0 \oplus 0 = 0$
$0 \oplus 1 = 1$ (also, $1 \oplus 0 = 1$)
$1 \oplus 1 = 0$

Let's represent the message bit by M, the key bit by K, and the encrypted bit by E. So $E = M \oplus K$. Penelope sends encrypted bit E to Odysseus. How can Odysseus recover the message bit M? Odysseus knows the key bit K; this is the secret information known only to Odysseus and Penelope. To recover the message bit M, all Odysseus has to do is combine the encrypted bit E with the key bit K according to the same rule: $E \oplus K$. Since $E = M \oplus K$, Odysseus is really computing $E \oplus K = M \oplus K \oplus K$. Now, K is either 0 or 1. Since $0 \oplus 0 = 0$ and $1 \oplus 1 = 0$,

$$K \oplus K = 0, \tag{1.1}$$

whether K is 0 or 1. So Odysseus computes $M \oplus K \oplus K = M \oplus 0$. Because M is either 0 or 1, and because $0 \oplus 0 = 0$ and $1 \oplus 0 = 1$,

$$M \oplus 0 = M. \tag{1.2}$$

So Odysseus recovers the message bit, but only because he knows the key bit. A potential eavesdropper like Hector doesn't know the key bit and cannot compute the message bit even if he glimpses the encrypted bit.

Let's take an example. Suppose Penelope wants to send Odysseus the message 0010. Before Odysseus began his voyage, with masts creaking and 10-foot waves slapping the hull, he and Penelope agreed to use the secret key 1011. Penelope combines each bit of the message with the corresponding bit of the secret key to obtain the cipher, as shown in Table 1.1. The first encrypted bit is $0 \oplus 1 = 1$, the second is $0 \oplus 0 = 0$, the third is $1 \oplus 1 = 0$, and the fourth is $0 \oplus 1 = 1$. So the cipher is 1001, which Penelope sends to Odysseus. Hector spies on this message but can't make heads or tails of it because he doesn't know the secret key.

Now, Odysseus receives the cipher 1001, and he combines each of its bits with the corresponding bit of the secret key, 1011, as shown in Table 1.2. The first bit becomes $1 \oplus 1 = 0$, the second bit becomes $0 \oplus 0 = 0$, the third bit becomes $0 \oplus 1 = 1$, and the fourth bit becomes $1 \oplus 1 = 0$. Thus, Odysseus has restored the (lurid and poignant) message, 0010.

So far, there's nothing quantum about this. Suppose, however, that Penelope and Odysseus decide they need to periodically change their secret key to keep Hector from guessing it. How can Penelope and Odysseus establish a secret key remotely? This is where Penelope's quantum genius comes in.

Three thousand years ahead of her time, Penelope has perfected a single-atom version of an experiment that normally requires a beam of atoms. (The real experiment, with a beam of atoms, is called the Stern-Gerlach experiment.) Penelope launches silver atoms through a magnetic field and observes that each atom is deflected toward either the magnet's north pole or south pole; no atom passes straight through. If the magnetic field is vertical, each atom is deflected either UP or DOWN. If the magnetic field is horizontal, each atom is deflected either RIGHT or LEFT.

Table 1.1

	Message Bit	Key Bit	Encrypted Bit
First Bit	0	1	$0 \oplus 1 = 1$
Second Bit	0	0	$0 \oplus 0 = 0$
Third Bit	1	1	$1 \oplus 1 = 0$
Fourth Bit	0	1	$0 \oplus 1 = 1$

Table 1.2

	Encrypted Bit	Key Bit	Message Bit
First Bit	1	1	$1 \oplus 1 = 0$
Second Bit	0	0	$0 \oplus 0 = 0$
Third Bit	0	1	$0 \oplus 1 = 1$
Fourth Bit	1	1	$1 \oplus 1 = 0$

Penelope observes that if an atom is deflected UP and then immediately enters another vertical magnetic field, it will again be deflected UP:

$$\begin{gathered}\text{atom} \rightarrow \boxed{\text{vertical magnetic field}} \rightarrow \text{deflected UP} \\ \rightarrow \boxed{\text{vertical magnetic field}} \rightarrow \text{deflected UP}\end{gathered}$$

We could send the atom through a hundred vertical magnetic fields in a row, and it would get deflected UP every time. The atom apparently has an enduring property that determines its behavior in vertical magnetic fields.

Similarly, an atom deflected DOWN is again deflected DOWN when it immediately enters another vertical magnetic field. If an atom is deflected RIGHT in a horizontal magnetic field, it is again deflected RIGHT in another horizontal magnetic field; the same rule applies to an atom deflected LEFT.

Penelope further observes that if an atom is deflected UP, and then enters a horizontal magnetic field, it is equally likely to be deflected RIGHT or LEFT. If the atom then enters a vertical magnetic field, it is no longer certain to go UP; it is equally likely to go DOWN:

$$\begin{gathered}\text{atom} \rightarrow \boxed{\text{vertical magnetic field}} \rightarrow \text{deflected UP} \rightarrow \boxed{\text{horizontal magnetic field}} \\ \rightarrow \text{deflected LEFT or RIGHT} \rightarrow \boxed{\text{vertical magnetic field}} \\ \rightarrow \text{deflected UP or DOWN}\end{gathered}$$

The horizontal magnetic field apparently erased the atom's vertical-field property: The atom lost its reliable UP-ness and has become just as likely to deflect DOWN.

Similarly, an atom initially deflected DOWN is equally likely to be deflected RIGHT or LEFT in a horizontal magnetic field, after which it is equally likely to go UP and DOWN in a vertical magnetic field. An atom initially deflected either RIGHT or LEFT is equally likely to be deflected UP or DOWN in a vertical magnetic field, after which it is equally likely to go either direction in a horizontal field, regardless of its initial deflection.

This is 100% of the quantum physics we need to understand quantum key distribution. To summarize, a silver atom deflected in a magnetic field will be deflected the same way if it subsequently enters a magnetic field in the same direction—if it hasn't been in any other magnetic fields. If the atom enters a magnetic field perpendicular to the field it initially passed through, it has a 50% chance of going either way, and if it later enters a magnetic field in the same direction as the original field it traversed, it has a 50% chance of going either way.

In effect, when a silver atom passes through a magnetic field, it is endowed with one bit of information about how it behaves in that field: UP or DOWN in a vertical field, and RIGHT or LEFT in a horizontal field. But when the atom passes through a field perpendicular to the original field, the original information is erased and replaced with information about how the atom behaves in the new field.

So, Penelope's plan is this. She will represent a 0 by a silver atom initially deflected either UP or RIGHT. She will represent a 1 by a silver atom initially deflected either DOWN or LEFT. She launches the selected atom to Odysseus, across the azure tides of sea-roiling Poseidon. Odysseus randomly sets his magnetic field either vertical or horizontal, and he observes the deflection of the atom.

For example, suppose Penelope wants to transmit a 1 by sending Odysseus a DOWN atom. Suppose Odysseus chooses to set his magnetic field vertical. Then, he will observe the atom deflected DOWN. He knows that Penelope uses DOWN to represent 1, so he guesses that Penelope wanted to transmit a 1.

However, if Odysseus instead chooses a horizontal magnetic field for this atom, it equally likely deflects RIGHT or LEFT. If it deflects RIGHT, Odysseus guesses incorrectly that Penelope wanted to transmit a 0.

Suppose that the choices and results for the first four atoms are as shown in Table 1.3. After Odysseus measures all the atoms, he and Penelope reveal the directions of their magnetic fields in all cases. They don't need to encode this announcement; eavesdroppers can do no harm now. Odysseus discards his guesses whenever he chose a different magnetic field direction than Penelope. So in the example in Table 1.3, he discards his guesses for the second and fourth atoms. He knows that his guesses for the first and third atoms were correct, so he and Penelope have now established two bits of their secret key: 11. They repeat with as many atoms as necessary to generate a sufficiently long key.

Now, how do the laws of quantum physics guarantee that their key is secure? In other words, how can they be *certain* that no eavesdropper copied the data as it traveled from Penelope to Odysseus? If Hector tries to intercept the silver atom, he has to choose whether to set his magnetic field horizontal or vertical, just as Odysseus does. He observes the atom and passes it on to Odysseus, but his attempt at espionage is thwarted by quantum physics. Let's see how.

Table 1.3

	First atom	Second atom	Third atom	Fourth atom
Penelope's bit	1	1	1	0
Penelope's magnetic field	vertical	horizontal	horizontal	vertical
Penelope's atom	DOWN	LEFT	LEFT	UP
Odysseus's magnetic field	vertical	vertical	horizontal	horizontal
Odysseus's observation	DOWN	UP	LEFT	RIGHT
Odysseus's guess	1	0	1	0

Consider this sequence of choices and outcomes:

Penelope's bit	1
Penelope's magnetic field	vertical
Penelope's atom	DOWN
Hector's magnetic field	horizontal
Hector's observation	RIGHT
Odysseus's magnetic field	vertical
Odysseus's observation	UP
Odysseus's guess	0

Penelope chooses a vertical magnetic field, and Hector chooses a horizontal magnetic field. The silver atom is equally likely to deflect RIGHT or LEFT in Hector's magnetic field. Odysseus has chosen the same magnetic field as Penelope, but the silver atom, having been deflected RIGHT, is equally likely to deflect UP and DOWN. If it deflects UP, Odysseus's guess, 0, differs from Penelope's bit, even though they chose the same magnetic field direction.

To detect Hector's meddling, Penelope and Odysseus sacrifice some of their key bits by revealing them to each other (and unavoidably to any eavesdropper monitoring their communication). If their key bits disagree, when they chose the same magnetic field direction, they must conclude that an eavesdropper meddled with their attempt to generate a secret key. So they have to abandon this attempt at a secret key, and maybe try again later.

Penelope and Odysseus have to compare a sufficiently large number of key bits, perhaps 10, to have a high probability of detecting an eavesdropper. This is because the eavesdropper corrupts only 25% of the key bits. Half of the time, the eavesdropper chooses the same magnetic field direction as Penelope. In this case, the eavesdropper observes the silver atom without changing it and passes it unaltered on to Odysseus. The other half of the time, the eavesdropper chooses a different magnetic field direction than Penelope. This effectively erases the information about deflection in the direction of Penelope's magnetic field. So when Odysseus sets his magnetic field in the same direction as Penelope's, he's only 50% likely to re-create Penelope's original deflection. In summary: Half of the time, Hector chooses a different magnetic field direction than Penelope, and when this occurs, the key bit is corrupted half of the time. Half of one half is 25%, the rate of key bit corruption.

If Penelope and Odysseus compare a subset of their key bits and find that they all agree, they conclude that no eavesdropper was present, and all their *other* key bits remain secret and secure. (They have to discard the bits they reveal because an eavesdropper could be eavesdropping on this communication, even if no eavesdropper intercepted the silver atoms.) This is a successful instance of quantum key distribution. Quantum key distribution can't stop eavesdroppers from eavesdropping, but it reveals the presence of an eavesdropper if there is one.

Now, let's rewrite UP, DOWN, RIGHT, and LEFT in the language of quantum computing. Let's use the symbol $|0\rangle$ to represent a silver atom deflected UP. This symbol, $|0\rangle$, is called a *ket*, which is the second syllable of brac*ket*. $|0\rangle$ is often pronounced "ket zero." We'll use $|1\rangle$ to represent an atom deflected DOWN. $|0\rangle$ and $|1\rangle$ are two possible states of a quantum bit, or *qubit*.

Remember that classical bits, 0 and 1, can represent two voltages in a circuit, or black and white stripes in a bar code, or different thicknesses of a plastic layer in CDs and DVDs. Similarly, a qubit can be constructed of many different physical systems. A silver atom is only one possibility, and not a very feasible one; not all quantum engineers are as cunning as Penelope. A qubit can be made of a photon, such that $|0\rangle$ and $|1\rangle$ represent two different polarization directions. In IBM's quantum processors that we'll use throughout this book, $|0\rangle$ and $|1\rangle$ represent two different states of a superconducting circuit. In fact, we'd rather *not* specify how our qubits are constructed: We want to establish rules and algorithms that work for *any* qubits, however they are made.

I once asked Matthias Steffen, IBM's chief quantum architect, how to think about the $|0\rangle$ and $|1\rangle$ states of a superconducting circuit. He told me that he'd given up on visualizing it. So let's follow the lead of IBM's chief quantum architect. We will establish rules that allow us to predict the results when qubits are measured. But we will not stumble far along the rocky path of wondering what qubits are doing when we're not measuring them.

Whereas a classical bit is either 0 or 1, a qubit can be in some combination of $|0\rangle$ and $|1\rangle$, written $\alpha|0\rangle+\beta|1\rangle$. α and β are called *probability amplitudes*, and they are related to the probabilities of different measurements. Now, there are different ways of measuring qubits, analogous to the different magnetic field directions for the silver atoms. If we do a measurement that results in either $|0\rangle$ and $|1\rangle$, this is called a measurement in the *computational basis*. (The computational basis is sometimes called the z basis by association with the vertical, or z, direction.) The probability of measuring $|0\rangle$ is $|\alpha|^2$, and the probability of measuring $|1\rangle$ is $|\beta|^2$. The total probability of measuring *something* is 1, which means

$$|\alpha|^2+|\beta|^2=1. \tag{1.3}$$

This condition is called *normalization*. If α and β are real numbers, then $|\alpha|^2=\alpha^2$ and $|\beta|^2=\beta^2$. However, α and β are allowed to be complex numbers. In this case, $|\alpha|^2=\alpha\alpha^*$, where α^* is the complex conjugate of α. We will work exclusively with real numbers for most of our journey.

We assigned UP $=|0\rangle$ and DOWN $=|1\rangle$. What about RIGHT and LEFT? Atoms deflected RIGHT and LEFT are equally likely to subsequently deflect UP or DOWN in a vertical magnetic field. This means α^2 and β^2 should both be 1/2. We'll choose $\text{RIGHT}=\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle=\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)$

and $\text{LEFT} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$. When we write $\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$, the probability amplitude of $|0\rangle$ is $\frac{1}{\sqrt{2}}$, and the probability amplitude of $|1\rangle$ is $-\frac{1}{\sqrt{2}}$.

It's convenient to define

$$|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \tag{1.4a}$$

and

$$|-\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right). \tag{1.4b}$$

In the language of qubits, we can now say that deflection in a horizontal magnetic field is a case of a measurement that yields either $|+\rangle$ or $|-\rangle$. This is called a measurement in the x basis by association with the horizontal, or x, direction.

We can combine Eqs. (1.4a) and (1.4b) to write $|0\rangle$ and $|1\rangle$ in terms of $|+\rangle$ and $|-\rangle$. The ket symbols can be manipulated exactly like algebraic symbols such as x and y. We can add Eqs. (1.4a) and (1.4b) together, to find $|+\rangle + |-\rangle = \frac{2}{\sqrt{2}}|0\rangle$. Solving for $|0\rangle$, we obtain

$$|0\rangle = \frac{1}{\sqrt{2}}\left(|+\rangle + |-\rangle\right), \tag{1.5a}$$

using $\frac{\sqrt{2}}{2} = \frac{\sqrt{2}}{2}\left(\frac{\sqrt{2}}{\sqrt{2}}\right) = \frac{2}{2\sqrt{2}} = \frac{1}{\sqrt{2}}$. Similarly, subtracting Eq. (1.4b) from Eq. (1.4a) yields $|+\rangle - |-\rangle = \frac{2}{\sqrt{2}}|1\rangle$. Solving for $|1\rangle$,

$$|1\rangle = \frac{1}{\sqrt{2}}\left(|+\rangle - |-\rangle\right). \tag{1.5b}$$

Whereas Eq. (1.4) gives probability amplitudes of $|0\rangle$ and $|1\rangle$, Eq. (1.5) gives probability amplitudes of $|+\rangle$ and $|-\rangle$: probability amplitudes for measurements in the x basis. Remembering to square probability amplitudes to find probabilities, we see that a qubit in state $|0\rangle$ or $|1\rangle$ is equally likely to be found in $|+\rangle$ or $|-\rangle$ when measured in the x basis. This is a generalization of the fact that a silver atom deflected UP or DOWN is equally likely to deflect RIGHT or LEFT when entering a horizontal magnetic field.

When a qubit is measured, the state *becomes* whatever was measured. For example, if a qubit, initially in state $|1\rangle$, is measured in the x basis, it is equally likely to *become* $|+\rangle$ or $|-\rangle$. Effectively, its original state is erased and replaced by the new one. This is a generalization of the rule we saw for the silver atoms: If an atom is initially deflected UP or DOWN, and then traverses a horizontal magnetic field, it will deflect RIGHT or LEFT without retaining any information about whether it had been deflected UP or DOWN. This is sometimes called the *collapse* of the state due to measurement.

Actually, this effect of measurement is not significant in most of the later chapters. Measurements will occur only at the end of our quantum circuits. And we will almost always measure in the computational basis, so the result of measuring a qubit will be either $|0\rangle$ or $|1\rangle$. In fact, the result of the measurement will be recorded as a classical bit, 0 or 1. All we have to remember going forward is that if a qubit in state $\alpha|0\rangle+\beta|1\rangle$ is measured, then the probability of measuring 0 is $|\alpha|^2$, and the probability of measuring 1 is $|\beta|^2$.

To review, let's repeat our example with Penelope, Hector, and Odysseus, but now using ket notation:

Penelope's bit	1
Penelope's basis	computational, also called z (measurement yields $|0\rangle$ or $|1\rangle$)
Penelope's atom	$|1\rangle$
Hector's basis	x (measurement yields $|+\rangle$ or $|-\rangle$)
Hector's measurement	$|+\rangle$
Odysseus's basis	computational, also called z (measurement yields $|0\rangle$ or $|1\rangle$)
Odysseus's measurement	$|0\rangle$
Odysseus's guess	0

Penelope's initial state is $|1\rangle$, which equals $|1\rangle=\frac{1}{\sqrt{2}}\left(|+\rangle-|-\rangle\right)$, given by Eq. (1.5b). Hector measures this qubit in the x basis, so the result will be $|+\rangle$ or $|-\rangle$. The probability amplitude of $|+\rangle$ is $\frac{1}{\sqrt{2}}$, and the probability amplitude of $|-\rangle$ is $-\frac{1}{\sqrt{2}}$. We square these amplitudes to determine probabilities, and we find that the probability of measuring $|+\rangle$ is 1/2, and so is the probability of measuring $|-\rangle$. Hector's measurement happens to yield $|+\rangle$.

Next, Odysseus measures this qubit in the computational basis, so we have to write $|+\rangle$ in terms of computational basis states: $|+\rangle=\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)$, as given in Eq. (1.4a). The probability amplitude is $\frac{1}{\sqrt{2}}$ for both $|0\rangle$ and $|1\rangle$, so $\left(\frac{1}{\sqrt{2}}\right)^2=\frac{1}{2}$ is the probability of obtaining either result. Odysseus happens to find $|0\rangle$, which is different from the state that Penelope sent him. If they share these facts with each other, they will know that Hector has meddled with their qubit.

Chapter 2

The First Quantum Algorithm

“Collaboration between Parallel Universes”?

Classical computers are very good at solving a wide range of problems. Some problems, at least in theory, can be solved more efficiently by quantum computers than by classical computers. Deutsch’s algorithm was the first quantum algorithm shown to surpass a classical computer in solving a specific problem. The problem solved by Deutsch’s algorithm doesn’t have a lot of practical importance, but it’s a warmup to other applications of quantum computing.

Deutsch’s algorithm was invented by David Deutsch, who happens to be a huge advocate of the many worlds interpretation of quantum mechanics. The many worlds interpretation asserts that the universe branches into parallel worlds in which all possible outcomes may occur. Deutsch wrote that quantum computers “will be the first technology that allows useful tasks to be performed in collaboration between parallel universes.” Sadly for Deutsch, almost everyone who agrees with him lives in other universes. In the mainstream version of the many worlds interpretation, the different universes have no influence on one another. (We can but gaze wistfully across the widening chasm toward the universes in which we never made any mistakes.) Still, Deutsch is one of the pioneers of quantum computing, and it’s interesting to know what he believes.

Before we develop Deutsch’s algorithm, we need to become familiar with a handful of *quantum gates*. Quantum gates transform qubits, from one state to another. (Quantum gates are also called quantum *operators*. The two terms mean exactly the same thing for our purposes.) Just as we’re not concerned with the physical details of how qubits are made, we’re not concerned with the physical details of how gates are constructed, either. Let’s start with the identity gate, I, which has no effect on any qubit:

$$\mathrm{I}|0\rangle = |0\rangle \tag{2.1a}$$

and

$$\mathrm{I}|1\rangle = |1\rangle. \tag{2.1b}$$

Next, we have the quantum NOT gate, called X. The classical NOT operation turns 0 to 1, and 1 to 0. The quantum NOT gate, X, turns $|0\rangle$ to $|1\rangle$, and $|1\rangle$ to $|0\rangle$:

$$\mathrm{X}|0\rangle = |1\rangle \tag{2.2a}$$

and

$$\mathrm{X}|1\rangle = |0\rangle. \tag{2.2b}$$

Let's represent both cases with a single equation. We need a variable to represent either 0 or 1, a single bit. This kind of variable is called a *Boolean* variable. We also need a symbol to represent the classical NOT operation. Let's use a bar over the Boolean value to represent NOT, so $\overline{0} = 1$ and $\overline{1} = 0$. Now we can combine Eqs. (2.2a) and (2.2b) into

$$\mathrm{X}|\mathrm{j}\rangle = |\overline{\mathrm{j}}\rangle, \tag{2.2c}$$

where j is 0 or 1.

What's the difference between a classical NOT and a quantum NOT? A classical NOT acts only on a 0 or 1. On the other hand, a quantum NOT gate can act on a generic qubit $\alpha|0\rangle + \beta|1\rangle$, which is a combination, called a *superposition*, of $|0\rangle$ and $|1\rangle$.

How do quantum gates act on superpositions? Let's take an analogy. Suppose I have four nearly identical deer, and seven nearly identical sheep. I want to know the approximate total weight of all my animals, but I can't fit them all on the scale at the same time. (I could if I stacked them, but they don't like that.) So I simply weigh one deer, and multiply by four, and then weigh one sheep, and multiply by seven, and then add it all together. In other words,

$$\begin{aligned}\mathrm{WEIGHT}(4|\mathrm{DEER}\rangle + 7|\mathrm{SHEEP}\rangle) = 4 \times \mathrm{WEIGHT}|\mathrm{DEER}\rangle \\ + 7 \times \mathrm{WEIGHT}|\mathrm{SHEEP}\rangle.\end{aligned}$$

This is called *linearity*: I can let WEIGHT act separately on $|\mathrm{DEER}\rangle$ and $|\mathrm{SHEEP}\rangle$, and I can pull the numbers 4 and 7 outside the WEIGHT operation. Quantum gates obey exactly the same linearity rule, so that

$$\mathrm{X}(\alpha|0\rangle + \beta|1\rangle) = \alpha\mathrm{X}|0\rangle + \beta\mathrm{X}|1\rangle = \alpha|1\rangle + \beta|0\rangle. \tag{2.3}$$

Our next quantum gate is called Z. It has no effect on $|0\rangle$, but it multiplies $|1\rangle$ by -1:

$$\mathrm{Z}|0\rangle = |0\rangle \tag{2.4a}$$

and

$$\mathrm{Z}|1\rangle = -|1\rangle. \tag{2.4b}$$

Can we combine these into a single equation? Yes. Since $(-1)^0 = 1$ and $(-1)^1 = -1$,

$$Z|j\rangle = (-1)^j|j\rangle. \tag{2.4c}$$

Our final quantum gate for this chapter is the Hadamard gate, H. The Hadamard gate turns computational basis states $(|0\rangle$ and $|1\rangle)$ into superpositions $|+\rangle$ and $|-\rangle$:

$$H|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) = |+\rangle \tag{2.5a}$$

and

$$H|1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) = |-\rangle \tag{2.5b}$$

We can combine these into a single equation by using the same trick in Eq. (2.4c):

$$H|j\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^j|1\rangle\right). \tag{2.5c}$$

Let's show that two H gates applied in a row cancel each other out. In other words, $H^2 = HH = I$. (We say that H is its own *inverse*, written H^{-1}.) We apply H to Eq. (2.5c):

$$H^2|j\rangle = H\left(H|j\rangle\right) = H\left[\frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^j|1\rangle\right)\right].$$

We then use the linearity rule to bring H inside the parentheses, to act on $|0\rangle$ and $|1\rangle$:

$$H^2|j\rangle = \frac{1}{\sqrt{2}}\left[H|0\rangle + (-1)^j H|1\rangle\right].$$

Then we use Eqs. (2.5a) and (2.5b) to replace $H|0\rangle$ and $H|1\rangle$:

$$H^2|j\rangle = \frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) + (-1)^j\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)\right].$$

Collecting probability amplitudes of $|0\rangle$ and $|1\rangle$, we find

$$H^2|j\rangle = \frac{1}{2}[1 + (-1)^j]|0\rangle + \frac{1}{2}[1 - (-1)^j]|1\rangle.$$

We see that when $j = 0$, the right-hand side becomes $|0\rangle$, and when $j = 1$, the right-hand side becomes $|1\rangle$. In other words,

$$H^2|j\rangle = |j\rangle. \tag{2.6}$$

Since $|+\rangle = H|0\rangle$ and $|-\rangle = H|1\rangle$ from Eq. (2.5), Eq. (2.6) implies

$$H|+\rangle = HH|0\rangle = H^2|0\rangle = |0\rangle \tag{2.7a}$$

and

$$H|-\rangle = HH|1\rangle = H^2|1\rangle = |1\rangle. \tag{2.7b}$$

A qubit can be acted on by one gate after another. For example, if H acts on $|0\rangle$, and then Z acts on the result, we get $ZH|0\rangle = Z(H|0\rangle) = Z\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Notice that the first gate to act in $ZH|0\rangle$ is written on the right, closest to the ket. $HZ|0\rangle$ is something different: $HZ|0\rangle = H(Z|0\rangle) = H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. So $HZ \neq ZH$. The order of gates matters. This is called *noncommutativity*, which is a good word to use when you want people to think you're smart. On a first date, for instance.

We can practice transforming qubits to our heart's content. For example, $HZH|1\rangle = HZ\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = H\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. At this point, we could let H act separately on $|0\rangle$ and $|1\rangle$, according to Eq. (2.5). But it's simpler to recognize that we now have $H|+\rangle$, which lets us use Eq. (2.7a) to arrive at the final result of $|0\rangle$.

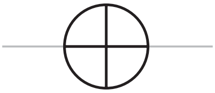
Let's take a moment to define *relative phase factor*. The only difference between $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ is a factor of −1 on $|1\rangle$. In $|-\rangle$, the factor of −1 (on $|1\rangle$ but not on $|0\rangle$) is called a relative phase factor, and it distinguishes $|-\rangle$ from $|+\rangle$. We can perform manipulations to detect the difference between $|+\rangle$ and $|-\rangle$. For example, we can apply the H gate to the qubit, which will convert $|+\rangle$ to $|0\rangle$, and $|-\rangle$ to $|1\rangle$. Then we can measure in the computational basis and obtain either 0 or 1, indicating that the initial state was either $|+\rangle$ or $|-\rangle$, respectively.

However, is there any measurable difference between $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $-|+\rangle = -\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$? In this case, no. There is no measurement or manipulation that can distinguish between $|+\rangle$ and $-|+\rangle$. The −1 in this case (which multiplies the entire expression) is called a *global phase factor*, which can be simply ignored because it has no physical meaning. $|+\rangle$ and $-|+\rangle$ are two different names for the same state.

Let's briefly make a futile effort to distinguish between $|+\rangle$ and $-|+\rangle$. We can apply H to the qubit, which converts it to $|0\rangle$ in the first case and to $-|0\rangle$ in the second. If we then measure the qubit in the computational basis, we are 100% likely to obtain 0, in either case. The minus sign does not affect the probability, and no combination of gates will enable the minus sign to affect any probability.

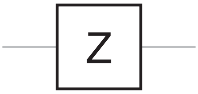
Now, we're ready for quantum circuit diagrams. These are exactly equivalent to the equations we've been writing, but diagrams are often easier to work with.

The X gate is represented by either an X in a box or this symbol:

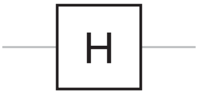


Quantum circuit diagrams are always read from left to right. So if we start with $|0\rangle$ on the left of the X gate, we will have $|1\rangle$ on the right.

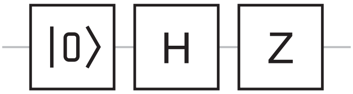
This is the Z gate:



And last, this is the H gate:



Since we read quantum circuit diagrams from left to right, the gate that acts first appears on the left. Recall, however, that $ZH|0\rangle = Z(H|0\rangle)$: The H gate, written closest to the ket, acts first. The gate acting *last* appears on the left in an equation, but on the *right* in a circuit diagram. So the circuit diagram for $ZH|0\rangle$ exactly reverses the order of the three parts of the expression $ZH|0\rangle$:



Classical circuit diagrams often have loops and branches. In a way, quantum circuit diagrams are simpler. You always read from left to right, applying the gates in order.

Now we're ready to apply our basic rules to Deutsch's algorithm. Deutsch's algorithm seeks to investigate a function f(x). f(x) is simply a number that may depend on the value of x. In Deutsch's algorithm, x is a single bit, a Boolean variable, either 0 or 1. f(x) is also a single bit, either 0 or 1. For example, one possible function is $f(x)=x$: When $x=0$, $f(x)=0$, and when $x=1$, $f(x)=1$. This information is restated in Table 2.1. Another possible function is $f(x) = \overline{x}$, shown in Table 2.2. These two functions are called *balanced* because 0 and 1 each appear once in the f(x) column.

There are two other possible functions f(x). There is $f(x)=0$, shown in Table 2.3. The last possible function is $f(x)=1$, shown in Table 2.4. These two

Table 2.1

value of x	value of f(x) = x
0	0
1	1

Table 2.2

value of x	value of $f(x) = \bar{x}$
0	1
1	0

Table 2.3

value of x	value of f(x) = 0
0	0
1	0

Table 2.4

value of x	value of f(x) = 1
0	1
1	1

functions are called *constant* because each of these f(x) functions has a constant value, independent of x.

Deutsch's algorithm solves this problem: Determine whether f(x) is constant or balanced. To solve this with a classical computer, we would have to input both values of x, one after the other. Let's simplify the classical computer to a circuit with a single input wire, x, and a single output wire, f(x). To determine whether f(x) is constant or balanced, we have no choice but to apply the two possible values of x in sequence. There are two steps. In the jargon of quantum computing, we say that we *query* the classical circuit twice.

Deutsch's algorithm uses a single query, which is an improvement over the two queries in the classical solution. To implement Deutsch's algorithm, we first need a quantum gate that incorporates the function f(x). A quantum gate is not the same thing as a function. A function inputs a number x, and outputs another number (or possibly the same number), f(x). A quantum gate inputs a qubit in one state, and outputs a qubit in another state (or possibly the same state).

A quantum gate that incorporates a function is called a *quantum oracle*. In mythology, an oracle is someone who will answer a question, but in a confusing and cryptic way. You have to be careful when you try to interpret what the oracle tells you. A quantum oracle is similar. It answers a question, but in a cryptic way. If you thought the Delphic oracle was cryptic, wait until you meet the quantum oracle.

Traditionally, the quantum oracle used in Deutsch's algorithm acts on two qubits. But, there's a simpler oracle, called a *phase oracle*, that works exactly as well as the traditional oracle. The phase oracle, U_f, is defined by its action on a computational basis state $|x\rangle$, where x is 0 or 1:

$$U_f|x\rangle = (-1)^{f(x)}|x\rangle. \tag{2.8}$$

The only thing that the phase oracle does is multiply $|x\rangle$ by either -1 or $+1$, depending on the value of f(x).

We're ready to construct U_f out of gates we've already seen, for each of the four functions f(x).

- If $f(x) = 0$, Eq. (2.8) becomes $U_f|x\rangle = |x\rangle$: U_f doesn't do anything, so it's the identity gate I, or simply no gate at all.
- If $f(x) = 1$, Eq. (2.8) becomes $U_f|x\rangle{=}{-}|x\rangle$: U_f multiplies any qubit by -1. This is a global phase factor, which can be ignored. So U_f again is I, or no gate at all.
- If $f(x) = x$, Eq. (2.8) becomes $U_f|x\rangle = (-1)^x|x\rangle$. This is exactly Eq. (2.4c), replacing j with x. So $U_f = Z$. Let's see what U_f does to a general superposition of basis states, $\alpha|0\rangle + \beta|1\rangle$: $U_f(\alpha|0\rangle + \beta|1\rangle) = Z(\alpha|0\rangle + \beta|1\rangle) = \alpha Z|0\rangle + \beta Z|1\rangle$ using the linearity rule. Applying Eq. (2.4), we arrive finally at $\alpha|0\rangle - \beta|1\rangle$. We see that Z creates a relative phase factor, which we cannot ignore.
- If $f(x) = \bar{x}$, Eq. (2.8) becomes $U_f|x\rangle = (-1)^{\bar{x}}|x\rangle$. Let's see what this does to a generic qubit, $\alpha|0\rangle + \beta|1\rangle$: $U_f(\alpha|0\rangle + \beta|1\rangle) = \alpha U_f|0\rangle + \beta U_f|1\rangle$, using the linearity rule. Then using $U_f|x\rangle = (-1)^{\bar{x}}|x\rangle$ on each term, we obtain $\alpha(-1)^1|0\rangle + \beta(-1)^0|1\rangle = -\alpha|0\rangle + \beta|1\rangle$, This is exactly -1 times what we found earlier for $U_f = Z$. So $U_f = -Z$ for $f(x) = \bar{x}$, but we can drop the global phase factor of -1 and simply use Z.

In summary, we've found that $U_f = I$ for constant functions, and $U_f = Z$ for balanced functions.

Deutsch's algorithm is simply this: $HU_fH|0\rangle$, and then we measure the final result in the computational basis. For constant functions, $U_f = I$, the two H gates cancel each other out, and our final result is $|0\rangle$.

For balanced functions, $U_f = Z$, and we have just a little more work to do. $HZH|0\rangle = HZ|+\rangle = H|-\rangle$ because $Z|+\rangle = Z\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) = \frac{1}{\sqrt{2}}\left(Z|0\rangle + Z|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) = |-\rangle$. The last step is that H acting on $|-\rangle$ yields $|1\rangle$, from Eq. (2.7b).

So that's it. We get 0 if the function is constant, and 1 if the function is balanced. The lingering question is this: How do you construct U_f if you don't

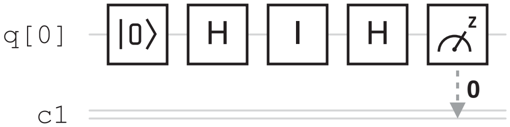


Figure 2.1. Deutsch's algorithm with a phase oracle for a constant function, created using IBM Quantum.

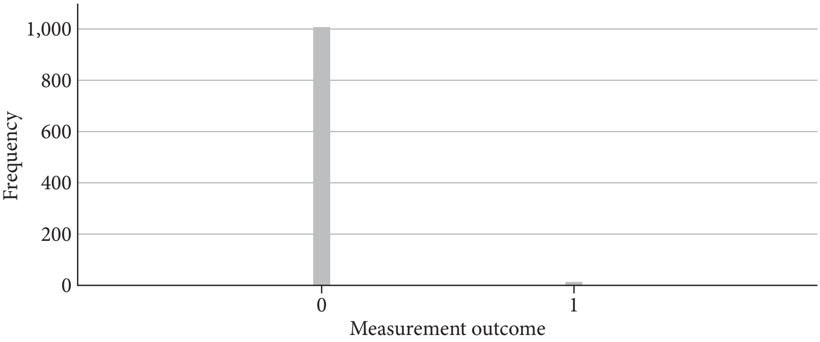


Figure 2.2. Results from the circuit in Fig. 2.1, created using IBM Quantum.

already know what f(x) is? And if you *do* already know what f(x) is, then you know if it's constant or balanced, so there's no need to use Deutsch's algorithm. Well, that's all true. To come up with a practical application of Deutsch's algorithm, we need an unlikely scenario. Maybe we already constructed our U_f gates, but forgot to label them, so we have to use Deutsch's algorithm to determine which oracles are which.

Let's see how Deutsch's algorithm works on real IBM quantum processors, which are accessible remotely online for free. Figure 2.1 shows the circuit for a constant function. In Fig. 2.1, q[0] is an arbitrary label for the qubit, and c1 is an arbitrary label for the classical bit obtained when the qubit is measured in the computational basis. The symbol on the right indicates a measurement. The 0 on the arrow is an additional label on the classical bit. You can construct the circuit yourself at quantum.ibm.com. Simply choose the Quantum Composer (currently found in the "Learning app") and grab the gates from the graphical menu.

I ran the circuit 1024 times on a 5-qubit processor called ibmq_quito. The results are shown in Fig. 2.2. Only one qubit is measured, yielding a measurement of either 0 or 1. Nearly 100% of the time, I got the expected result, 0. Due to error, the result of 1 did occur, but only 11 times out of 1024. (This error does *not* occur in a simulation but is a consequence of the nonideal behavior of the real quantum processor that I accessed. Deutsch's algorithm is theoretically expected to give the correct result 100% of the time. We

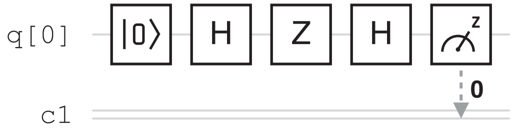


Figure 2.3. Deutsch's algorithm with a phase oracle for a balanced function, created using IBM Quantum.

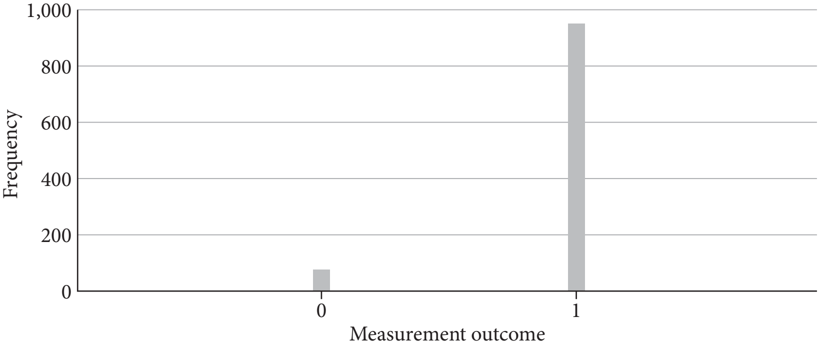


Figure 2.4. Results from the circuit in Fig. 2.3, created using IBM Quantum.

will see later that some quantum algorithms are expected to give the correct result less than 100% of the time, even in theory and simulations. In any case, the correct result is expected to be the most likely outcome.)

Next, Fig. 2.3 shows the circuit for the balanced function. And the results are in Fig. 2.4. Most of the time, the result was 1, as expected. The error rate this time was a little higher, 75 out of 1024.

Chapter 3

Qubit? Cube It

One qubit at a time is all we need for quantum key distribution and Deutsch's algorithm. However, to accomplish most tasks, we usually require two or more qubits. If we have two qubits, both in state $|0\rangle$, we represent the state of the two qubits simply as $|0\rangle|0\rangle$. Later, it will be convenient to shorten this to $|00\rangle$. For now, we will stick with $|0\rangle|0\rangle$.

When we measure two qubits in the computational basis, we obtain $|0\rangle$ or $|1\rangle$ for each of them. So there are a total of four possible results: $|0\rangle|0\rangle$, $|0\rangle|1\rangle$, $|1\rangle|0\rangle$, and $|1\rangle|1\rangle$. These are the four computational basis states for a system of two qubits. The general state of two qubits is a superposition of these four basis states: $\alpha|0\rangle|0\rangle+\beta|0\rangle|1\rangle+\gamma|1\rangle|0\rangle+\delta|1\rangle|1\rangle$. If we measure the two qubits in the computational basis, the probability of measuring $|0\rangle|0\rangle$ is $|\alpha|^2$, the probability of measuring $|0\rangle|1\rangle$ is $|\beta|^2$, the probability of measuring $|1\rangle|0\rangle$ is $|\gamma|^2$, and the probability of measuring $|1\rangle|1\rangle$ is $|\delta|^2$. The sum of the four probabilities must be 1, so $|\alpha|^2+|\beta|^2+|\gamma|^2+|\delta|^2=1$.

Notice that $|0\rangle|1\rangle$ is not the same as $|1\rangle|0\rangle$. There are two qubits, and the one represented by the ket on the left is different from the one represented by the ket on the right.

Suppose we have two qubits, one in state $A|0\rangle+B|1\rangle$, and the other in state $C|0\rangle+D|1\rangle$. How do we write the state of the two-qubit system? We simply multiply together the two single-qubit states: $(A|0\rangle+B|1\rangle)(C|0\rangle+D|1\rangle)$. This is a called a *product state* because it is the product of two single-qubit states. How do we write this product state in terms of the four computational basis states ($|0\rangle|0\rangle$, $|0\rangle|1\rangle$, $|1\rangle|0\rangle$, and $|1\rangle|1\rangle$)? We follow the normal rules of multiplication, except that we have to distinguish $|0\rangle|1\rangle$ from $|1\rangle|0\rangle$. (AC, on the other hand, is the same as CA. C and A represent *numbers*, and the order of multiplication doesn't matter when numbers are multiplied. The order does matter when we multiply kets or gates.)

We recall the FOIL (First, Outside, Inside, Last) rule, which means, for example, that $(x+y)(w+z)=xw+xz+yw+yz$; each term in the first pair of parentheses multiplies each term in the second pair of parentheses. Similarly,

$$(A|0\rangle + B|1\rangle)(C|0\rangle + D|1\rangle) = AC|0\rangle|0\rangle + AD|0\rangle|1\rangle + BC|1\rangle|0\rangle + BD|1\rangle|1\rangle. \quad (3.1)$$

For example, suppose we start with two qubits in the state $|0\rangle|0\rangle$. Then, we apply the H gate to each qubit: $H|0\rangle H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{2}(|0\rangle\,|0\rangle + |0\rangle|1\rangle + |1\rangle|0\rangle + |1\rangle|1\rangle)$. The factors of $\frac{1}{\sqrt{2}}$ are just numbers and can be multiplied together.

If we're given an expression like $\frac{1}{2}(|0\rangle|0\rangle - |0\rangle|1\rangle + |1\rangle|0\rangle - |1\rangle|1\rangle)$, with a little work we can factor it into $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. What happens if just one qubit in a product state is measured? The measurement of one qubit does not affect the other. If our two qubits are in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, and we measure the qubit on the left and find it in state $|0\rangle$, the two qubit state becomes $|0\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle\,|0\rangle - |0\rangle\,|1\rangle)$.

However, there are some two-qubit states that *cannot* be factored into a product state. For example, consider the state $\frac{1}{\sqrt{2}}(|0\rangle\,|0\rangle + |1\rangle\,|1\rangle)$. This is a superposition of two computational basis states. Each computational basis state, in isolation, is a product state. But the superposition $\frac{1}{\sqrt{2}}(|0\rangle\,|0\rangle + |1\rangle|1\rangle)$ cannot be factored into a product state $(A|0\rangle + B|1\rangle)(C|0\rangle + D|1\rangle)$, for any values of A, B, C, and D. A state that cannot be factored is called an *entangled* state.

Quantum entanglement has that straightforward mathematical definition. However, the physical meaning is complex, unresolved, and even controversial. Consider the entangled state $\frac{1}{\sqrt{2}}(|0\rangle\,|0\rangle + |1\rangle|1\rangle)$. If the qubit on the left is measured, and the result is $|0\rangle$, the state of both qubits collapses to $|0\rangle|0\rangle$. Apparently, the measurement of one qubit affects both. Or does it? Were both qubits in state $|0\rangle$ all along, only we didn't know it for sure? These questions have haunted physicists like a half-dead cat. We'll revisit these questions in later chapters.

As difficult as entanglement is to explain, it's easy to create with quantum gates. Besides the Hadamard gate, we need just one new gate: the controlled NOT, or CNOT. CNOT acts on two qubits, one of which is the control, and one of which is the target. If the control is $|1\rangle$, then a NOT, or X, is applied to the target. If the control is $|0\rangle$, nothing happens.

If we write the control first,

$$\mathrm{CNOT}|0\rangle|0\rangle = |0\rangle|0\rangle, \quad (3.2a)$$

$$\mathrm{CNOT}|0\rangle|1\rangle = |0\rangle|1\rangle, \quad (3.2b)$$

$$\mathrm{CNOT}|1\rangle|0\rangle = |1\rangle|1\rangle, \quad (3.2c)$$

and

$$\text{CNOT}|1\rangle|1\rangle = |1\rangle|0\rangle. \tag{3.2d}$$

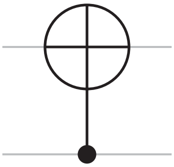
In Eqs. (3.2a) and (3.2b), the control is $|0\rangle$, and the CNOT gate has no effect. In Eqs. (3.2c) and (3.2d), the control is $|1\rangle$, and a NOT is applied to the target. In Eq. (3.2), we see that the control (the left qubit) is the same on both sides of the equations: the control never changes. So if the control is initially $|\text{control}\rangle$, the control remains $|\text{control}\rangle$ after the CNOT operation.

In Eqs. (3.2a) and (3.2d), the target is initially in the same state as the control, and the target ends up as $|0\rangle$. In Eqs. (3.2b) and (3.2c), the target and control start in different states, and the target ends up as $|1\rangle$. We recall that the exclusive OR operation produces an output of 1 when its two input bits differ. So if the control is initially $|\text{control}\rangle$, and the target is initially $|\text{target}\rangle$, the target becomes $|\text{control} \oplus \text{target}\rangle$. So all four cases of Eq. (3.2) can all be represented by

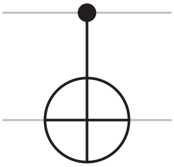
$$\text{CNOT}|\text{control}\rangle|\text{target}\rangle = |\text{control}\rangle|\text{control} \oplus \text{target}\rangle. \tag{3.2e}$$

Like all quantum gates, the CNOT gate can be applied to superpositions. So, for example, $\text{CNOT}(\alpha|0\rangle|0\rangle + \beta|1\rangle|1\rangle) = \alpha\text{CNOT}|0\rangle|0\rangle + \beta\text{CNOT}|1\rangle|1\rangle = \alpha|0\rangle|0\rangle + \beta|1\rangle|0\rangle$.

In quantum circuit diagrams, each horizontal line represents one qubit. If the control is the bottom qubit, and the target is the top qubit, the CNOT gate looks like this:



If the control is the top qubit, and the target is the bottom qubit, the CNOT gate looks like this:



In either case, the control is represented by a dot, and the target looks like crosshairs, which is the symbol for the X, or NOT, gate.

Now, we have a decision to make: If we want to represent $|0\rangle|1\rangle$ with a circuit diagram, is $|0\rangle$ the top qubit or the bottom qubit? Different authors choose differently. I will follow IBM Quantum and *write the bottom qubit on* *the left*: $|0\rangle|1\rangle$ means that the bottom qubit is $|0\rangle$, and the top qubit is $|1\rangle$. In other words, we will read our circuit diagrams *from the bottom up*. It's like entering a building: Typically, we enter at the ground floor, and then we go up. We're not descending into a dungeon; we're climbing up a building, or climbing up a mountain.

We're ready to analyze quantum circuits with two or more qubits. Consider Fig. 3.1. The initial state is $|0\rangle|0\rangle$. The H gate transforms the bottom qubit from $|0\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$. Writing the bottom qubit first (on the left), the state of the two qubits is now $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|0\rangle$. Before applying the CNOT, it's convenient to multiply each term in parentheses by the $|0\rangle$ on the right: $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|0\rangle)$. Next, the CNOT acts independently on each of the two terms, $|0\rangle|0\rangle$ and $|1\rangle|0\rangle$. Using Eq. (3.2a), we see that CNOT has no effect on $|0\rangle|0\rangle$ because the control is $|0\rangle$. Using Eq. (3.2c), we see that CNOT transforms $|1\rangle|0\rangle$ because to $|1\rangle|1\rangle$ because the control is $|1\rangle$. So the final state of the circuit is $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|1\rangle)$: an entangled state.

Equation (3.2) applies when the control is written first (on the left), which, according to our convention, means that the control is on the bottom in the circuit diagram. We're just as likely to encounter CNOT gates with the control on the top, as shown in Fig. 3.2. In this case, rather than using Eq. (3.2), we can just use the rule: When the control is $|1\rangle$, apply NOT to the target.

Going through Fig. 3.2 gate by gate, the H gate comes first. After the H gate, the state of the qubits is $|0\rangle\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)=\frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|0\rangle|1\rangle)$. Next, the CNOT acts on each of the two terms in parentheses. In the first term, the control is $|0\rangle$, so the CNOT has no effect on the first term. In the second term, $|0\rangle|1\rangle$, the control is $|1\rangle$ because the top qubit is written last (on the right), and the top qubit is the control. So the CNOT transforms $|0\rangle|1\rangle$ into $|1\rangle|1\rangle$, and the final state of the qubits is again $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|1\rangle)$.

The analysis of all quantum circuits follows these same steps: Read the circuit diagram from left to right, applying one gate at a time, until you obtain the final state at the end of the circuit.

Let's look again at the four computational basis states for two qubits:

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

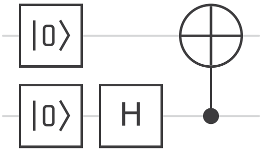


Figure 3.1. A circuit to produce an entangled state, created using IBM Quantum.

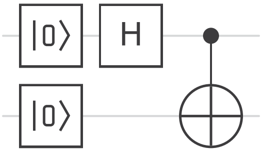


Figure 3.2. A circuit that produces the same entangled state as the one in Fig. 3.1, created using IBM Quantum.

Table 3.1

value of x	value of f(x)
00	0
01	1
10	1
11	0

It's often convenient to write the two numbers inside a single ket:

$$|0\rangle|0\rangle = |00\rangle$$

$$|0\rangle|1\rangle = |01\rangle$$

$$|1\rangle|0\rangle = |10\rangle$$

$$|1\rangle|1\rangle = |11\rangle$$

Now that we know how to work with two qubits, we can generalize Deutsch's algorithm to multiple qubits. This generalization is called the Deutsch-Jozsa algorithm. Our function f(x) still equals 0 or 1, depending on x. But now, x can be two or more bits. For example, if x is two bits, there are four possible values of x: 00, 01, 10, and 11. If f(x) = 0 for all four values of x, f(x) is constant. If f(x) = 1 for all four values of x, again f(x) is constant. If f(x) = 0 for exactly two values of x, and f(x) = 1 for the other two values of x, then we say that f(x) is balanced. Table 3.1 gives an example of a balanced function.

It's possible for f(x) to be neither constant nor balanced: For example, f(x) could be 0 for three values of x and 1 for the other value of x. In the Deutsch-Jozsa algorithm, we know that f(x) is either constant or balanced, and we want to know which.

Just as in the original Deutsch algorithm, we incorporate f(x) into the quantum phase oracle U_f defined in Eq. (2.8), $U_f|x\rangle=(-1)^{f(x)}|x\rangle$. The only difference is that now, x consists of two or more bits, as in Table 3.1. So, U_f acts on two or more qubits. We will see specific examples of U_f later.

The Deutsch-Jozsa algorithm, using the oracle defined earlier, is as follows:

1. Apply H to all qubits (initially in state $|0\rangle$).
2. Apply U_f, which acts on all qubits.
3. Apply H to all qubits.
4. Measure all qubits.

We will now show that if the measurement yields $|0\rangle$ for every qubit, the function is constant. Otherwise, it is balanced. For simplicity, we will show this for the specific case of two qubits; the result is true for any number of qubits.

In the first step, we apply H to both qubits initially in state $|0\rangle$: $H|0\rangle H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)=\frac{1}{2}(|0\rangle|0\rangle+|0\rangle|1\rangle+|1\rangle|0\rangle+|1\rangle|1\rangle)$. Let's combine each pair of kets into a single ket: $\frac{1}{2}(|00\rangle+|01\rangle+|10\rangle+|11\rangle)$.

In the second step, U_f acts on the qubits. It acts separately on each of the four terms. So, for example $U_f|00\rangle=(-1)^{f(00)}|00\rangle$. After U_f acts on all four terms, the total state is $\frac{1}{2}((-1)^{f(00)}|00\rangle+(-1)^{f(01)}|01\rangle+(-1)^{f(10)}|10\rangle+(-1)^{f(11)}|11\rangle)$.

Before we again apply H to both qubits, let's separately consider constant and balanced functions. If f(x) is constant, then $f(00)=f(01)=f(10)=f(11)=f(x)$, and our expression is $\frac{1}{2}((-1)^{f(x)}|00\rangle+(-1)^{f(x)}|01\rangle+(-1)^{f(x)}|10\rangle+(-1)^{f(x)}|11\rangle)$. All four terms contain $(-1)^{f(x)}$, so we can factor it out: $(-1)^{f(x)}\frac{1}{2}(|00\rangle+|01\rangle+|10\rangle+|11\rangle)$. $(-1)^{f(x)}$ is now a global phase factor, which we can ignore. So our state is effectively $\frac{1}{2}(|00\rangle+|01\rangle+|10\rangle+|11\rangle)$, which is what is was before we applied U_f. We know that this state can be written $H|0\rangle|H|0\rangle$. Next, in the third step, we apply H to each qubit: $H^2|0\rangle H^2|0\rangle=|0\rangle|0\rangle$ because $H^2=I$; H cancels itself out. So when we measure the qubits, we're guaranteed to obtain $|0\rangle|0\rangle$ when the function is constant. But are we guaranteed to obtain something different when the function is balanced?

So let's consider the case of balanced f(x). Just after applying U_f, the state is $\frac{1}{2}((-1)^{f(00)}|0\rangle|0\rangle+(-1)^{f(01)}|0\rangle|1\rangle+(-1)^{f(10)}|1\rangle|0\rangle+(-1)^{f(11)}|1\rangle|1\rangle)$. The next step is to again apply H to both qubits. H acts on each qubit in each term, so the state is $\frac{1}{2}((-1)^{f(00)}H|0\rangle H|0\rangle+(-1)^{f(01)}H|0\rangle H|1\rangle+(-1)^{f(10)}H|1\rangle H|0\rangle+(-1)^{f(11)}H|1\rangle H|1\rangle)$. Using the definition of H, we can show that

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

$$H|0\rangle H|1\rangle = \frac{1}{2}(|0\rangle|0\rangle - |0\rangle|1\rangle + |1\rangle|0\rangle - |1\rangle|1\rangle)$$

$$H|1\rangle H|0\rangle = \frac{1}{2}(|0\rangle|0\rangle + |0\rangle|1\rangle - |1\rangle|0\rangle - |1\rangle|1\rangle)$$

$$H|1\rangle H|1\rangle = \frac{1}{2}(|0\rangle|0\rangle - |0\rangle|1\rangle - |1\rangle|0\rangle + |1\rangle|1\rangle)$$

We could plug all of these expressions into $\frac{1}{2}\big((-1)^{f(00)}H|0\rangle H|0\rangle + (-1)^{f(01)}H|0\rangle H|1\rangle + (-1)^{f(10)}H|1\rangle H|0\rangle + (-1)^{f(11)}H|1\rangle H|1\rangle\big)$. But that's no fun. All we want to show now is that the probability of measuring $|0\rangle|0\rangle$ is 0. So we only have to pay attention to the $|0\rangle|0\rangle$ terms. In all four cases, $|0\rangle|0\rangle$ is multiplied by 1/2:

$$H|0\rangle H|0\rangle = \frac{1}{2}|0\rangle|0\rangle + \text{other terms}$$

$$H|0\rangle H|1\rangle = \frac{1}{2}|0\rangle|0\rangle + \text{other terms}$$

$$H|1\rangle H|0\rangle = \frac{1}{2}|0\rangle|0\rangle + \text{other terms}$$

$$H|1\rangle H|1\rangle = \frac{1}{2}|0\rangle|0\rangle + \text{other terms}$$

When we plug these into $\frac{1}{2}\big((-1)^{f(00)}H|0\rangle H|0\rangle + (-1)^{f(01)}H|0\rangle H|1\rangle + (-1)^{f(10)}H|1\rangle H|0\rangle + (-1)^{f(11)}H|1\rangle H|1\rangle\big)$, we get $\frac{1}{2}\left((-1)^{f(00)}\frac{1}{2}|0\rangle|0\rangle + (-1)^{f(01)}\frac{1}{2}|0\rangle|0\rangle + (-1)^{f(10)}\frac{1}{2}|0\rangle|0\rangle + (-1)^{f(11)}\frac{1}{2}|0\rangle|0\rangle\right) + \text{other terms}$. Factoring out $\frac{1}{2}|0\rangle|0\rangle$, the expression becomes $\frac{1}{4}|0\rangle|0\rangle\big((-1)^{f(00)} + (-1)^{f(01)} + (-1)^{f(10)} + (-1)^{f(11)}\big) + \text{other terms}$.

So if $\big((-1)^{f(00)} + (-1)^{f(01)} + (-1)^{f(10)} + (-1)^{f(11)}\big) = 0$, the probability of measuring $|0\rangle|0\rangle$ is 0 for a balanced function. And indeed, for a balanced function, exactly two of the exponents are 0, and two of the exponents are 1. So we get $(-1)^0 = 1$ twice, and $(-1)^1 = -1$ twice. When we add these four numbers together, we get $1 + 1 + (-1) + (-1) = 0$.

Now, we're ready to test the Deutsch-Jozsa algorithm on a real quantum processor. We saw that when f(x) is constant, U_f is effectively the identity operator, and the Deutsch-Jozsa circuit for two qubits is therefore the circuit in Fig. 3.3.

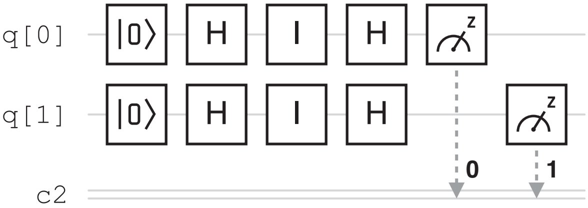


Figure 3.3. The Deutsch-Jozsa algorithm for a constant function, created using IBM Quantum.

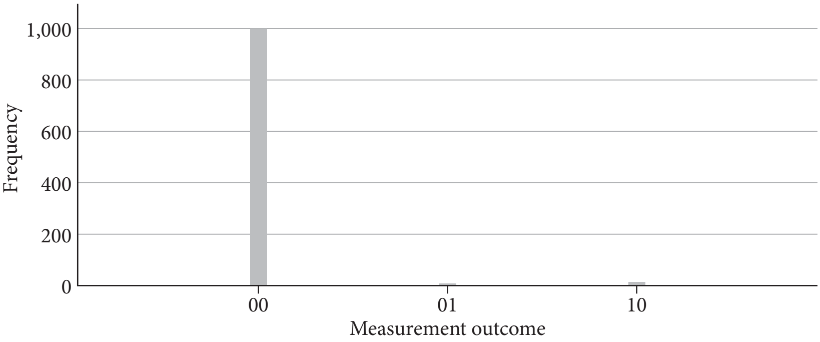


Figure 3.4. Results from the circuit in Fig. 3.3, created using IBM Quantum.

Running the circuit on ibm_oslo, I obtained Fig. 3.4. Since the function is constant, the result ideally is 0 every time. Due to error, other results occur 2% of the time. As in the previous chapter, this error does not occur in theory or simulations. The error we see here is due only to nonideal performance of the available technology; it messes up some of the time.

Next, when f(x) is balanced, we have several choices. We might choose the balanced function in Table 3.2, where f(x) equals the second bit in x (the bit on the right, so for example, 1 in 01). If we write the two bits of x as x_1x_0, we see that $f(x)=x_0$ in Table 3.2. Since $U_f|x\rangle=(-1)^{f(x)}|x\rangle$, the necessary U_f multiplies the state by −1 whenever $x_0=1$. Since $Z|0\rangle=|0\rangle$ and $Z|1\rangle=-|1\rangle$, U_f is a Z gate acting on $|x_0\rangle$, the top qubit, as shown in Fig. 3.5 (where $|x_0\rangle$ is labeled q[0], the default label for the top qubit in IBM Quantum). The results from ibm_oslo are in Fig. 3.6. Since the function is balanced, 00 = 0 should never be measured. Due to error, 0 is measured 1% of the time.

What if we have the balanced function in Table 3.1? This function f(x) is 1 whenever x_1 is different from x_0. So U_f needs to multiply by −1 whenever x_1 is different from x_0. Suppose we multiply by $(-1)^{x_0+x_1}$. When x_0 is different from x_1, one of these bits is 0 and the other is 1, so $x_0+x_1=1$, and $(-1)^{x_0+x_1}=-1$. And when $x_0=x_1$, x_0+x_1 is either 0 or 2, which means $(-1)^{x_0+x_1}=+1$. So multiplying by $(-1)^{x_0+x_1}$ does exactly what we need: It mul-

Table 3.2

value of x	value of f(x)
00	0
01	1
10	0
11	1

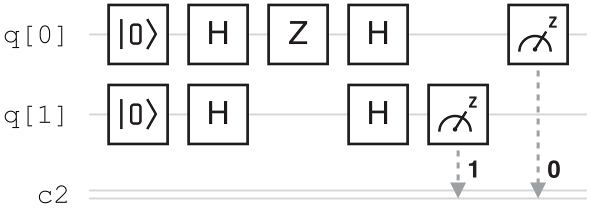


Figure 3.5. The Deutsch-Jozsa algorithm for the function $f(x) = x_0$, created using IBM Quantum.

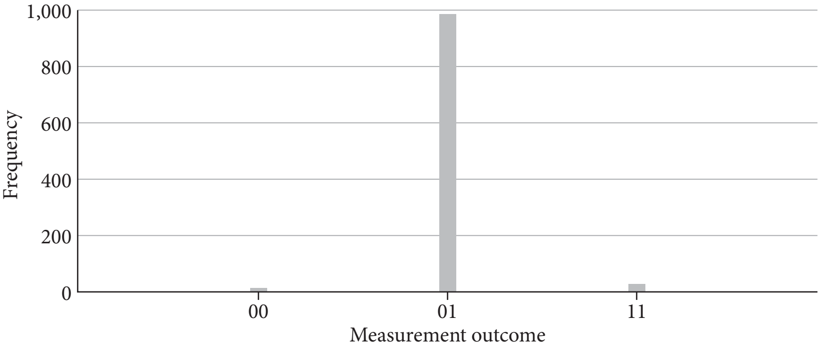


Figure 3.6. Results from the circuit in Fig. 3.5, created using IBM Quantum.

tiplies by −1 when the two bits of x are different, and it has no effect when the two bits of x are the same. Multiplying by $(-1)^{x_0 + x_1}$ is the same as multiplying by both $(-1)^{x_0}$ and $(-1)^{x_1}$, which we achieve with a Z gate on each qubit, shown in Fig. 3.7. Figure 3.8 shows the results from ibm_oslo. Again, since the function is balanced, 00 = 0 should never be measured. The histogram in Fig. 3.8 appears ideal, but actually, 0 was measured once out of 1024 runs.

As a final example of the Deutsch-Jozsa algorithm, let's take the balanced function of three bits in Table 3.3. There are now eight possible values of x. In this example, $f(x) = 1$ whenever exactly one of the bits is 1 (001, 010, and

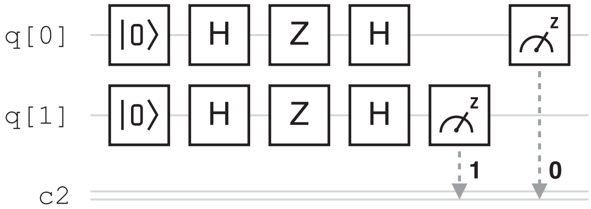


Figure 3.7. The Deutsch-Jozsa algorithm for the function f(x) = 1 if and only if the two bits of x differ, created using IBM Quantum.

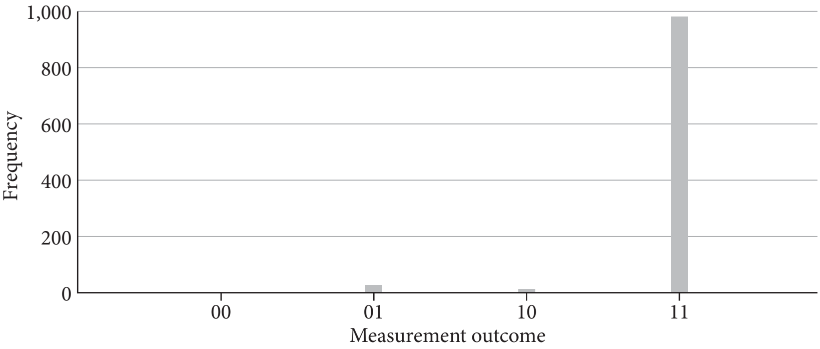


Figure 3.8. Results from the circuit in Fig. 3.7, created using IBM Quantum.

Table 3.3

value of x	value of f(x)
000	0
001	1
010	1
011	0
100	1
101	0
110	0
111	1

100) or all three bits are 1 (111). How do we construct U_f, which multiplies by –1 whenever either one qubit is $|1\rangle$, or all three are $|1\rangle$? If the three bits are $x_2x_1x_0$, we get the desired result if we multiply by $(-1)^{x_0+x_1+x_2}$, which is achieved by a Z gate on all three qubits (Fig. 3.9).

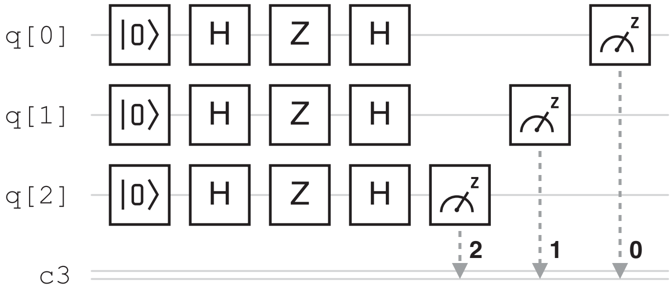


Figure 3.9. The Deutsch-Jozsa algorithm for a balanced function of three bits, created using IBM Quantum.

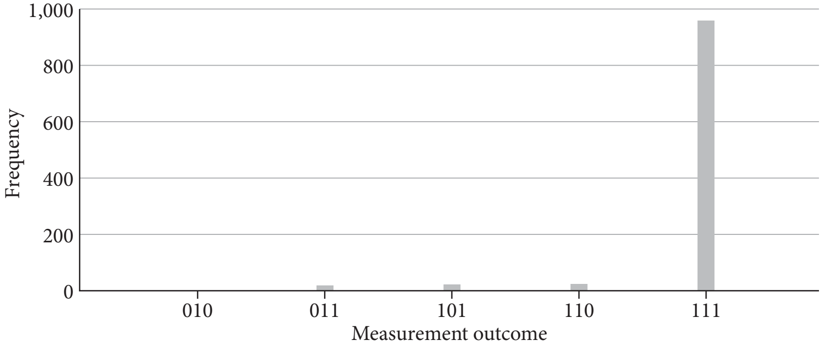


Figure 3.10. Results from the circuit in Fig. 3.9, created using IBM Quantum.

Let's see how the circuit fares on ibm_oslo (Fig. 3.10). Since the function is balanced, 000 = 0 should never be measured. Indeed, it is not.

Chapter 4

Quantum Teleportation

Too Awesome to Require a Pun

The last couple chapters were a bit dry, so let's splash into the Aegean Sea, which separates our lovelorn heroes, Penelope and Odysseus. Penelope has a qubit in a special state, $\alpha|0\rangle+\beta|1\rangle$, which she wishes to send to Odysseus as a token of her love. This qubit is so special that Penelope doesn't want to send it off to face the sirens, the sorceresses, Scylla and Charybdis, and a hundred other perils haunting the route toward Odysseus. So Penelope devises a means to *teleport* the qubit directly to Odysseus.

In quantum teleportation, only information is teleported, not mass. So Odysseus needs to have a spare qubit that will be transformed into Penelope's qubit. In fact, quantum teleportation requires some preparation. Before leaving Penelope in the first place, Odysseus and Penelope need to create a pair of entangled qubits. Odysseus takes one with him on his arduous voyage, and Penelope keeps one home, close to her heart.

The process of quantum teleportation is shown in Fig. 4.1. Let's start with the top two qubits, initially $|0\rangle|0\rangle$. Penelope applies an H gate to the middle qubit, resulting in $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|0\rangle$. Next, a CNOT gate is applied. Since the CNOT gate acts on both qubits, the two qubits must be close together; this step must occur before Odysseus departs with his qubit. The CNOT gate transforms $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|0\rangle=\frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|0\rangle)$ to $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|1\rangle)$, an entangled state.

Now, Odysseus sets sail, taking his qubit with him, while Penelope keeps hers at home. The qubits remain entangled, no matter how widely they're separated. An apt metaphor for Penelope's and Odysseus's love.

At some point, Penelope selects a third qubit, $\alpha|0\rangle+\beta|1\rangle$. She can select this qubit after Odysseus departs, or before (though if she selects it before he departs, she might as well just give it to him then, instead of teleporting it to

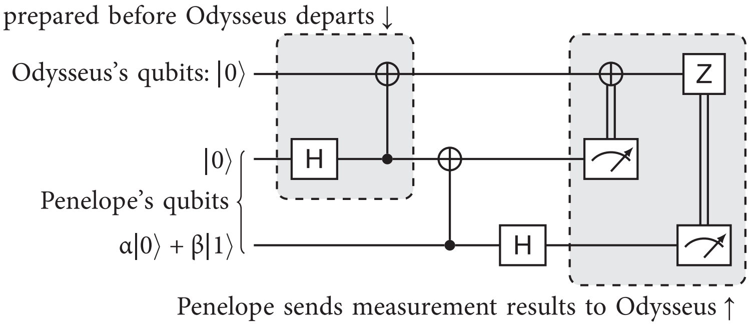

prepared before Odysseus departs ↓
Odysseus's qubits: |0⟩
|0⟩
Penelope's qubits
α|0⟩ + β|1⟩
H
H
Z
Penelope sends measurement results to Odysseus ↑

Figure 4.1. Quantum teleportation, created using the Quantikz LaTeX package.

him later). Since this third qubit is the bottom qubit in the diagram, we write it on the left: $(\alpha|0\rangle+\beta|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|1\rangle)$. Moving $\frac{1}{\sqrt{2}}$ all the way to the left, and applying the FOIL multiplication rule, the expression becomes $\frac{1}{\sqrt{2}}(\alpha|0\rangle|0\rangle|0\rangle+\alpha|0\rangle|1\rangle|1\rangle+\beta|1\rangle|0\rangle|0\rangle+\beta|1\rangle|1\rangle|1\rangle)$.

Penelope will next apply a CNOT to the bottom two qubits (the two she has). When the bottom qubit (written on the left) is $|1\rangle$, a NOT is applied to the middle qubit. So the state becomes $\frac{1}{\sqrt{2}}(\alpha|0\rangle|0\rangle|0\rangle+\alpha|0\rangle|1\rangle|1\rangle+\beta|1\rangle|1\rangle|0\rangle+\beta|1\rangle|0\rangle|1\rangle)$.

Next, Penelope applies an H gate to the bottom qubit. The H acts on the left qubit in all four terms in the preceding expression. The two $|0\rangle$ qubits become $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$, and the two $|1\rangle$ qubits become $\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)$. So, we end up with eight terms. It's a little tedious, but the final result is $\frac{1}{2}(\alpha|0\rangle|0\rangle|0\rangle+\alpha|1\rangle|0\rangle|0\rangle+\alpha|0\rangle|1\rangle|1\rangle+\alpha|1\rangle|1\rangle|1\rangle+\beta|0\rangle|1\rangle|0\rangle-\beta|1\rangle|1\rangle|0\rangle+\beta|0\rangle|0\rangle|1\rangle-\beta|1\rangle|0\rangle|1\rangle)$.

To prepare for Penelope's measurement of her two qubits, we will reorder the eight terms according to common states of the first two qubits (the left and middle qubits). For example, $\alpha|0\rangle|0\rangle|0\rangle$ and $\beta|0\rangle|0\rangle|1\rangle$ both have $|0\rangle|0\rangle$ for the first two qubits, so we will place them side by side in the sequence of eight terms. Doing the same for $|0\rangle|1\rangle$, $|1\rangle|0\rangle$, and $|1\rangle|1\rangle$, the sequence of terms is written $\frac{1}{2}(\alpha|0\rangle|0\rangle|0\rangle+\beta|0\rangle|0\rangle|1\rangle+\alpha|0\rangle|1\rangle|1\rangle+\beta|0\rangle|1\rangle|0\rangle+\alpha|1\rangle|0\rangle|0\rangle-\beta|1\rangle|0\rangle|1\rangle+\alpha|1\rangle|1\rangle|1\rangle-\beta|1\rangle|1\rangle|0\rangle)$.

Next, we look at each pair in the sequence, and factor out the common states of the first two qubits. So:

$$\alpha|0\rangle|0\rangle|0\rangle + \beta|0\rangle|0\rangle|1\rangle = |0\rangle|0\rangle\big(\alpha|0\rangle + \beta|1\rangle\big)$$

$$\alpha|0\rangle|1\rangle|1\rangle + \beta|0\rangle|1\rangle|0\rangle = |0\rangle|1\rangle\big(\alpha|1\rangle + \beta|0\rangle\big)$$

$$\alpha|1\rangle|0\rangle|0\rangle - \beta|1\rangle|0\rangle|1\rangle = |1\rangle|0\rangle\big(\alpha|0\rangle - \beta|1\rangle\big)$$

$$\alpha|1\rangle|1\rangle|1\rangle - \beta|1\rangle|1\rangle|0\rangle = |1\rangle|1\rangle\big(\alpha|1\rangle - \beta|0\rangle\big)$$

The full expression is then $\frac{1}{2}\big[|0\rangle|0\rangle\big(\alpha|0\rangle + \beta|1\rangle\big) + |0\rangle|1\rangle\big(\alpha|1\rangle + \beta|0\rangle\big) + |1\rangle|0\rangle\big(\alpha|0\rangle - \beta|1\rangle\big) + |1\rangle|1\rangle\big(\alpha|1\rangle - \beta|0\rangle\big)\big]$. The $\frac{1}{2}$ indicates a probability of $\left(\frac{1}{2}\right)^2 = \frac{1}{4}$ for each possible result of Penelope's measurements: $|0\rangle|0\rangle$, $|0\rangle|1\rangle$, $|1\rangle|0\rangle$, and $|1\rangle|1\rangle$. Let's consider each possibility separately.

If Penelope measures $|0\rangle|0\rangle$, the state collapses to $|0\rangle|0\rangle(\alpha|0\rangle+\beta|1\rangle)$. The three other terms in the full expression are anchored to the first two qubits having a state other than $|0\rangle|0\rangle$. So when Penelope measures $|0\rangle|0\rangle$, Odysseus's qubit is in the state $(\alpha|0\rangle+\beta|1\rangle)$, exactly what Penelope wanted to teleport to him! But there's only a 25% chance of this lucky result. Odysseus has to be told that Penelope obtained the lucky measurement $|0\rangle|0\rangle$. Since it's easier to transport classical bits than qubits, Penelope sends Odysseus the two bits 00. She could simply write these values on parchment and send it to Odysseus via a trusted messenger. In the circuit diagram, classical bits are represented by the double lines extending upward from the measurement symbols. The classical bits act as controls on the NOT and Z gates on Odysseus's qubit. Since both controls are 0, neither the NOT nor the Z gate acts on Odysseus's qubit, which retains the desired state, $(\alpha|0\rangle+\beta|1\rangle)$.

If Penelope measures $|0\rangle|1\rangle$, the state collapses to $|0\rangle|1\rangle(\alpha|1\rangle+\beta|0\rangle)$. Odysseus's qubit is in the state $(\alpha|1\rangle+\beta|0\rangle)$, which isn't quite what Penelope was trying to convey. So Penelope sends Odysseus the bits 01. The 1 is the measurement result from the middle qubit, which is the control on Odysseus's NOT gate. So the NOT is applied to Odysseus's qubit, transforming $(\alpha|1\rangle+\beta|0\rangle)$ into $(\alpha|0\rangle+\beta|1\rangle)$.

If Penelope measures $|1\rangle|0\rangle$, the state collapses to $|1\rangle|0\rangle(\alpha|0\rangle-\beta|1\rangle)$. Odysseus's qubit is in the state $(\alpha|0\rangle-\beta|1\rangle)$. Penelope sends Odysseus the bits 10. This time, the 1 is the measurement result from the bottom qubit, which is the control on Odysseus's Z gate. So the Z is applied to Odysseus's qubit, transforming $(\alpha|0\rangle-\beta|1\rangle)$ into $(\alpha|0\rangle+\beta|1\rangle)$.

Last, if Penelope measures $|1\rangle|1\rangle$, the state collapses to $|1\rangle|1\rangle(\alpha|1\rangle-\beta|0\rangle)$. Odysseus's qubit is in the state $(\alpha|1\rangle-\beta|0\rangle)$. Penelope sends Odysseus the bits 11.

Both control bits are 1. First, the NOT transforms Odysseus's qubit into $(\alpha|0\rangle - \beta|1\rangle)$, and at last the Z transforms it into $(\alpha|0\rangle + \beta|1\rangle)$.

An important detail is that quantum teleportation cannot transmit information faster than the speed of light. Even though Penelope's measurements instantaneously collapse the state of all three qubits, including Odysseus's distant qubit, the teleportation is not complete until Odysseus receives Penelope's measurement results. Penelope's measurement results cannot travel faster than the speed of light. Even Hermes of the winged sandals, messenger of the gods, cannot exceed the speed of light.

Another important fact is that quantum teleportation does not duplicate Penelope's qubit, $\alpha|0\rangle + \beta|1\rangle$. Her measurement collapses this qubit to a computational basis state, $|0\rangle$ or $|1\rangle$. Her original qubit, $\alpha|0\rangle + \beta|1\rangle$, is reconstituted by Odysseus only after the original vanishes from the initial location. This is why we say *teleportation*, not duplication. The impossibility of duplicating a generic qubit is called the no-cloning theorem, which we will see in the next chapter.

In summary, quantum teleportation allows Penelope to transmit a qubit to Odysseus by sending him two classical bits. The exact opposite process is possible: Penelope can transmit two classical bits to Odysseus by sending him a single qubit. This reverse process is called quantum dense coding, or superdense coding. Figure 4.2 shows the circuit diagram for superdense coding.

Penelope and Odysseus again prepare an entangled pair of qubits before Odysseus's departure: The state of the qubits is initially $|0\rangle|0\rangle$, and then Penelope applies an H gate to her qubit, resulting in $\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)|0\rangle$. Next, the CNOT gate transforms $\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|0\rangle\right)$ into $\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|1\rangle\right)$. At this point, Odysseus sets sail upon the wine-dark sea.

Next, Penelope chooses a two-bit value to send to Odysseus. There are only four possibilities: 00, 01, 10, and 11. She applies to her qubit a gate that

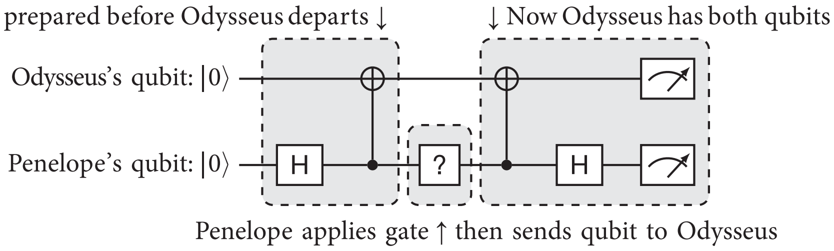

prepared before Odysseus departs ↓
↓ Now Odysseus has both qubits
Odysseus's qubit: |0⟩
Penelope's qubit: |0⟩
H
?
H
Penelope applies gate ↑ then sends qubit to Odysseus

Figure 4.2. Quantum dense coding, created using the Quantikz LaTeX package.

depends on the value she chooses. This variable gate is represented by a question mark in the circuit diagram. Let’s go through all four possible gates.

If Penelope wants to send Odysseus the value 00, she applies the identity gate, or equivalently no gate at all. Then she sends her qubit to her beloved. When Odysseus receives the qubit, he has both qubits, which are still in the state $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|1\rangle)$. Next Odysseus applies a CNOT, with the control on the bottom qubit. So the $|1\rangle|1\rangle$ term becomes $|1\rangle|0\rangle$. The state of the two qubits is now $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|0\rangle)=\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|0\rangle=|+\rangle|0\rangle$. Next, Odysseus applies an H gate to the bottom qubit, obtaining $|0\rangle|0\rangle$. Last, Odysseus measures both qubits. He is certain to get $|0\rangle|0\rangle$, or 00, exactly what Penelope wanted to send him. She was able to send him two bits via a single qubit.

If Penelope wants to send 01 to Odysseus, the question mark in the circuit diagram should be replaced by a NOT gate. This will transform the state of the qubits from $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|1\rangle)$ into $\frac{1}{\sqrt{2}}(|1\rangle|0\rangle+|0\rangle|1\rangle)$. Then she sends her qubit to Odysseus, and he applies a CNOT. The state of the two qubits becomes $\frac{1}{\sqrt{2}}(|1\rangle|1\rangle+|0\rangle|1\rangle)=\frac{1}{\sqrt{2}}(|1\rangle+|0\rangle)|1\rangle=|+\rangle|1\rangle$. After Odysseus applies an H gate to the bottom qubit, the state is $|0\rangle|1\rangle$. When Odysseus measures both qubits, he gets 01.

Penelope uses a Z gate when she wants to send 10. The Z gate transforms the qubits from $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|1\rangle)$ into $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle-|1\rangle|1\rangle)$. After Odysseus applies a CNOT, the state of the two qubits is $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle-|1\rangle|0\rangle)$ $=\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)|0\rangle=|-\rangle|0\rangle$. Then Odysseus applies an H gate to the bottom qubit, and the state becomes $|1\rangle|0\rangle$. Odysseus measures both qubits and obtains 10.

Last, to send 11, Penelope applies an X gate, then a Z gate. The X gate transforms the qubits from $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|1\rangle)$ into $\frac{1}{\sqrt{2}}(|1\rangle|0\rangle+|0\rangle|1\rangle)$, and the Z gate transforms this into $\frac{1}{\sqrt{2}}(-|1\rangle|0\rangle+|0\rangle|1\rangle)$. Odysseus receives Penelope’s qubit and then applies a CNOT, transforming the state of the two qubits into $\frac{1}{\sqrt{2}}(-|1\rangle|1\rangle+|0\rangle|1\rangle)=\frac{1}{\sqrt{2}}(-|1\rangle+|0\rangle)|1\rangle=|-\rangle|1\rangle$. Then Odysseus applies an H gate to the bottom qubit, the state of the qubits becomes $|1\rangle|1\rangle$, and Odysseus measures 11.

Chapter 5

The No-Cloning Theorem

Why We Can't Send Messages to the Past

As Odysseus contemplates the dying embers of a cheerless campfire on the Trojan coast, let us contemplate the entangled state $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle)$. We see that the qubits are certain to be found in matching states when they are measured in the computational basis: There's a 50% chance of obtaining $|0\rangle|0\rangle$, and a 50% chance of obtaining $|1\rangle|1\rangle$. This is true regardless of whether the qubits are measured at the same time. Penelope could measure her qubit days or weeks before Odysseus measures his. But as soon as Penelope measures her qubit, we know exactly what result Odysseus will obtain when he finally gets around to measuring his qubit (after pulling Trojan arrows of out his shield and collapsing in exhaustion in a ragged tent). Does Penelope's measurement physically alter the state of Odysseus's distant qubit? Is there a way for Penelope to manipulate her qubit to send instantaneous messages to Odysseus, faster than the speed of light?

We will see that Penelope *would* be able to send messages faster than the speed of light, *if* Odysseus were able to duplicate a qubit in an unknown state. However, the *no-cloning theorem* prohibits the duplication of qubits in unknown states. We will now prove the no-cloning theorem.

First, suppose there's a process, CLONE, that copies the state of one qubit $|A\rangle$ onto another qubit originally in state $|B\rangle$:

$$\mathrm{CLONE}[|A\rangle|B\rangle] = |A\rangle|A\rangle. \tag{5.1}$$

We want to prove that this process is impossible. First, we want to write the state $|A\rangle$ of the qubit we want to copy, in terms of $|0\rangle$ and $|1\rangle$. As usual, we write the generic qubit state in terms of two unknown variables, α and β:

$$|A\rangle = \alpha|0\rangle + \beta|1\rangle.$$

Now we just substitute $\alpha|0\rangle + \beta|1\rangle$, twice, on the right side of $\text{CLONE}[|A\rangle|B\rangle] = |A\rangle|A\rangle$:

$$\text{CLONE}[|A\rangle|B\rangle] = (\alpha|0\rangle + \beta|1\rangle)(\alpha|0\rangle + \beta|1\rangle).$$

Using the FOIL multiplication rule,

$$\text{CLONE}[|A\rangle|B\rangle] = \alpha^2|0\rangle|0\rangle + \alpha\beta|0\rangle|1\rangle + \beta\alpha|1\rangle|0\rangle + \beta^2|1\rangle|1\rangle. \tag{5.2}$$

Now, let's evaluate Eq. (5.1) a different way. Let's first substitute $|A\rangle = \alpha|0\rangle + \beta|1\rangle$ into $\text{CLONE}[|A\rangle|B\rangle]$:

$$\text{CLONE}[|A\rangle|B\rangle] = \text{CLONE}[(\alpha|0\rangle + \beta|1\rangle)|B\rangle].$$

Ordinary algebra lets us multiply both terms in parentheses by $|B\rangle$:

$$\text{CLONE}[|A\rangle|B\rangle] = \text{CLONE}[\alpha|0\rangle|B\rangle + \beta|1\rangle|B\rangle]. \tag{5.3}$$

Now recall the linearity rule,

$$\text{WEIGHT}(4|\text{DEER}\rangle + 7|\text{SHEEP}\rangle) = \\ 4 \times \text{WEIGHT}|\text{DEER}\rangle + 7 \times \text{WEIGHT}|\text{SHEEP}\rangle.$$

The total weight of 4 identical deer and 7 identical sheep is four times the weight of one deer plus seven times the weight of one sheep. All quantum gates obey this linearity rule. This allows us to pull the α and β outside of the brackets in Eq. (5.3), and allows CLONE to act separately on the two terms:

$$\text{CLONE}[|A\rangle|B\rangle] = \alpha\text{CLONE}[|0\rangle|B\rangle] + \beta\text{CLONE}[|1\rangle|B\rangle]. \tag{5.4}$$

On the right side, $\text{CLONE}[|0\rangle|B\rangle] = |0\rangle|0\rangle$: $|0\rangle$ is cloned. Similarly, $\text{CLONE}[|1\rangle|B\rangle] = |1\rangle|1\rangle$. So Eq. (5.4) becomes

$$\text{CLONE}[|A\rangle|B\rangle] = \alpha|0\rangle|0\rangle + \beta|1\rangle|1\rangle.$$

But! This is different from what we found earlier in Eq. (5.2),

$$\text{CLONE}[|A\rangle|B\rangle] = \alpha^2|0\rangle|0\rangle + \alpha\beta|0\rangle|1\rangle + \beta\alpha|1\rangle|0\rangle + \beta^2|1\rangle|1\rangle.$$

The two equations agree only if either $\alpha = 1$ and $\beta = 0$, or $\alpha = 0$ and $\beta = 1$. But we wanted to clone an unknown state, not a very special case ($|0\rangle$ or $|1\rangle$). Other than these two special cases, the CLONE process generates contradictory results. And since the CLONE process generates contradictory results, it must be impossible, not a realizable process at all.

Now let's see how the no-cloning theorem prevents Penelope from sending instantaneous messages to Odysseus. Penelope and Odysseus each have one qubit in the entangled pair, $\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|1\rangle\right)$. If Penelope measures $|0\rangle$, she immediately knows that Odysseus, who in principle may be light-years away, will measure $|0\rangle$. But this is not instantaneous communication: Penelope's result is random. She could just as easily obtain $|1\rangle$, which means Odysseus will measure $|1\rangle$. Penelope is not choosing the result of Odysseus's measurement.

What if Penelope measures $|0\rangle$, but Odysseus does a measurement that results in either $|+\rangle$ or $|-\rangle$? (We called this a measurement in the x basis, as opposed to the computational basis, which is also called the z basis.) Penelope's measurement of $|0\rangle$ puts Odysseus's qubit in the state $|0\rangle$. Odysseus's qubit then has a 50% chance of being measured as $|+\rangle$, and a 50% chance of being measured as $|-\rangle$.

Now Penelope has an idea: She will try to send a message to Odysseus by choosing to measure her qubit in either the z basis or the x basis. So the choice of basis is a kind of code. She decides that if she measures in the z basis, she's trying to send Odysseus the value 0. If she measures in the x basis, she's trying to send the value 1. Can Odysseus determine the value that Penelope is trying to send?

To understand what happens when Penelope measures in the x basis, let's use Eq. (1.5), $|0\rangle = \frac{1}{\sqrt{2}}\left(|+\rangle + |-\rangle\right)$ and $|1\rangle = \frac{1}{\sqrt{2}}\left(|+\rangle - |-\rangle\right)$. Let's substitute these expressions for each $|0\rangle$ and each $|1\rangle$ in $\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|1\rangle\right)$:

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

$$\frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{2}}\left(|+\rangle + |-\rangle\right)\frac{1}{\sqrt{2}}\left(|+\rangle + |-\rangle\right) + \frac{1}{\sqrt{2}}\left(|+\rangle - |-\rangle\right)\frac{1}{\sqrt{2}}\left(|+\rangle - |-\rangle\right)\right]$$

$$= \frac{1}{2\sqrt{2}}\left[\left(|+\rangle + |-\rangle\right)\left(|+\rangle + |-\rangle\right) + \left(|+\rangle - |-\rangle\right)\left(|+\rangle - |-\rangle\right)\right],$$

where I combined all the factors of $\frac{1}{\sqrt{2}}$. Next, we apply FOIL multiplication to each of the two products of terms in parentheses:

$$\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|1\rangle\right) = \frac{1}{2\sqrt{2}}\left(|+\rangle|+\rangle + |+\rangle|-\rangle + |-\rangle|+\rangle + |-\rangle|-\rangle\right)$$
$$+ \left(|+\rangle|+\rangle - |+\rangle|-\rangle - |-\rangle|+\rangle + |-\rangle|-\rangle\right).$$

The $|+\rangle|-\rangle$ and $|-\rangle|+\rangle$ terms are subtracted off, and the doubled $|+\rangle|+\rangle$ and $|-\rangle|-\rangle$ terms eliminate a factor of 1/2, so finally

$$\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|+\rangle|+\rangle + |-\rangle|-\rangle\right).$$

The right side of the equation means that if Penelope measures in the x basis and finds her qubit in state $|+\rangle$, then Odysseus's qubit is also in state $|+\rangle$. Interestingly, no matter what basis Penelope chooses, her measurement causes both qubits to collapse into matching states.

Now, suppose that Penelope wants to send the value 1. She and Odysseus agreed in advance that in this case, she will do a measurement in the x basis. Suppose she finds her qubit in the state $|+\rangle$, which means that Odysseus's qubit is also in the state $|+\rangle$. Is there any measurement Odysseus can do to determine whether Penelope measured in the x basis or z basis? Sadly, no. If Odysseus measures in the x basis, he will obtain $|+\rangle$, but all he knows for sure is that Penelope's qubit was not $|-\rangle$. He doesn't know if Penelope's qubit was found in state $|+\rangle$, $|0\rangle$, or $|1\rangle$. Penelope could have done a measurement in the z basis, putting Odysseus's qubit in a $|0\rangle$ or $|1\rangle$ state. Then, Odysseus's measurement would have had a 50% chance of obtaining the state $|+\rangle$. So Odysseus's result, no matter what it is, cannot determine which basis Penelope chose, or which bit value she tried to send him.

But now, suppose that Odysseus can clone his qubit. Suppose he makes a hundred copies. He measures each of them in the z basis. So if Odysseus's original qubit was $|+\rangle$, he now measures a hundred $|+\rangle$ qubits in the z basis. About half of them will convert to $|0\rangle$, and the others will convert to $|1\rangle$. This tells him that his original photon was neither $|0\rangle$ nor $|1\rangle$, and so must have been $|+\rangle$ or $|-\rangle$. Odysseus is able to conclude that Penelope measured in the x basis, so Penelope successfully transmitted the value 1, instantaneously, over any distance.

And when Penelope wants to transmit 0, her measurement (in the z basis) yields $|0\rangle$ or $|1\rangle$, which puts Odysseus's qubit in the state $|0\rangle$ or $|1\rangle$. In either case, all 100 of Odysseus's cloned qubits are found in the same state when measured in the z basis. So Odysseus's rule is simply to measure in the z basis, and if all his clones are found in an identical state, Penelope has sent 0. If about half of his clones are $|0\rangle$ and half are $|1\rangle$, Penelope has sent 1.

So, if qubit cloning were possible, the following situation could ensue. Odysseus clones his qubit after Penelope measures hers, so that Penelope can transmit one bit (a 0 or 1) instantaneously with each pair of entangled qubits she shares with Odysseus. If they have eight pairs, Penelope can transmit one byte of data (8 bits). If they have 8000 pairs, Penelope can transmit 8 kilobytes. With many pairs of entangled qubits, Penelope can send a lot of information instantaneously over large distances, faster than the speed of light.

So, if we were not bound by the no-cloning theorem, we would be able to send messages faster than the speed of light. Now let's see how this would enable us to actually send messages back in time.

One of the most surprising results of special relativity is that simultaneity depends on the observer: To determine whether two events occur at the same time, we have to specify who's observing! To understand this, we imagine a flashbulb in the center of a train car. When the bulb flashes, the light simultaneously reaches the front and back walls of the car—but only according to a passenger in the train. To an observer outside the train, the back wall of the train car is racing toward the bulb, so the back wall reaches the incoming light first. The front wall is receding from the bulb, so the light reaches the front wall later. The two events that are genuinely simultaneous, as observed by the train passengers, are genuinely *not* simultaneous, as observed from the ground. All observers are correct. Simultaneity depends on the observer, just as "the train's speed relative to me" depends on who's speaking.

Einstein did not say that everything is relative. In fact, the speed of light is absolute. To rigidly maintain the speed of light for all observers, time and space melt, becoming as fluid as Dali's clocks. Simultaneity becomes fluid: Simultaneity is not absolute but depends on the observer.

The speed of light is so reliable, the passengers in the train decide to use it to synchronize their stopwatches. There's a stopwatch at the front wall of the train car, and another stopwatch at the back wall. The stopwatches are programmed to start counting as soon as they receive a pulse of light. As seen by the passengers in the train, light from the flashbulb reaches both stopwatches at the same time. So the stopwatches are perfectly synchronized.

But, seen from outside the train, the light reaches the stopwatch at the back wall first. So the stopwatch at the back wall has more time to count; it shows a later time than the stopwatch at the front wall. This rule applies to all clocks on the train, regardless of how they are synchronized. So the clock in the back is ahead (in time) of the clock in the front. In other words, the chasing clock is ahead of the fleeing clock. For example, when the clock in the caboose shows 3 pm, the clock in the front car might show noon—according to observers outside the train. The passengers on the train insist their clocks are perfectly synchronized. Again, *all observers are correct!* When we ask whether two events are simultaneous, *or* whether two clocks are synchronized, we have to ask: According to whom?

These relativistic effects, of course, are significant only at relativistic speeds: speeds approaching the speed of light. For our thought experiment, we imagine that the train is moving at a relativistic speed relative to the ground.

Next, we'll assume that we can send messages instantaneously, which means that the message travels infinitely fast (faster than the speed of light) from the sender to the receiver. Figure 5.1 shows two train passengers, Caboose Carl and Engineer Emma. The two observers on the ground are Stationary Steve and Stationary Stella. According to the ground observers, Caboose Carl's clock shows a later time (say, 3 pm) than Engineer Emma's clock (noon). According to the train passengers, however, the two clocks are synchronized.

Caboose Carl sends a message to Stationary Steve just as they pass each other. For example, Caboose Carl could hold a written note up to the window for Stationary Steve to read. Perhaps the note is, "Quantum computing is awesome."

Next, Stationary Steve uses his instantaneous communication technology (which exists only in this thought experiment). Stationary Steve transmits Caboose Carl's message to Stationary Stella. Suppose further that the message appears on Stationary Stella's tablet, which she's holding so that Engineer Emma can read it.

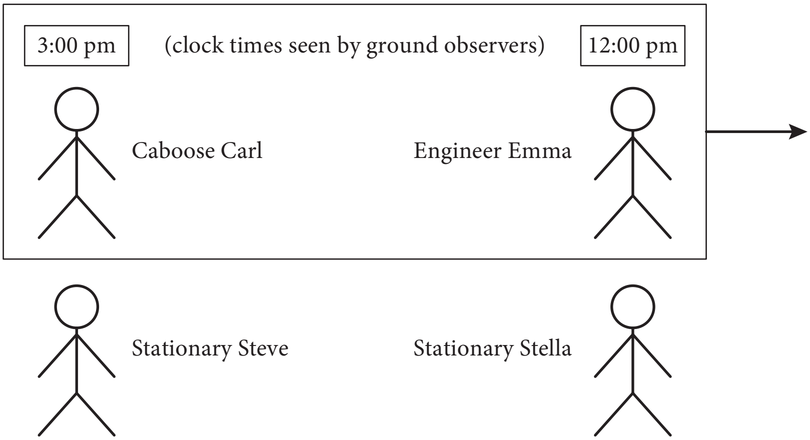


Figure 5.1. A relativistic train as seen by observers on the ground.

So, according to the train clocks, Engineer Emma, at noon, is reading the message that Caboose Carl won't send until 3! If Engineer Emma has instantaneous communication technology, she can, at noon, send Caboose Carl his own future message, three hours before he sends it to Stationary Steve. (Of course, instantaneous communication—*simultaneous* transmission and reception—depends on the observer, since simultaneity depends on the observer. If instantaneous communication is available on the train, then the communication is instantaneous as seen by people on the train.) If the message sent to the past is harmless (and perhaps even beneficial), like "Quantum computing is awesome," then there's not necessarily a paradox. The events can be self-consistent, even self-reinforcing: Caboose Carl could go out of his way to send Stationary Steve the message that traveled back in time to Caboose Carl, before he sent it in the first place.

However, what if Caboose Carl's message was, "At all cost, avoid sending this message"? Then he'd be motivated, at 3 pm, to do something other than what Stationary Steve already saw him do. So the fabric of reality unravels if we send signals to the past. And this scheme, to send signals to the past, relies on instantaneous communication. So we should hope, for the preservation of reality, that instantaneous communication is impossible. We saw that if Odysseus were able to clone his qubit, then Penelope would be able to transmit messages instantaneously. And the no-cloning theorem prevents Odysseus from cloning his qubit, so instantaneous communication is prevented, so sending signals to the past is prevented. So the no-cloning theorem preserves the fabric of reality and saves us all. Thank you, no-cloning theorem. You have saved the entire universe.

Chapter 6

A Nobel Prize in Experimental Philosophy

The 2022 Nobel Prize in Physics was awarded to Alain Aspect, John Clauser, and Anton Zeilinger "for experiments with entangled photons, establishing the violation of Bell inequalities and pioneering quantum information science." In this chapter, we will explore exactly what a Bell inequality is and why it merits a Nobel Prize. We will examine some of the work of the three Nobel laureates. If you're antsy to get to quantum algorithms, you can skip this chapter without loss of continuity. This long chapter is dense with both math and philosophy and may forever change your understanding of reality. Proceed with caution.

The 2022 Nobel Prize has its roots in a 1935 paper by Einstein and two less-famous coauthors (Podolsky and Rosen). The original paper involved the position and momentum of entangled particles. We can make the same point with our entangled pair of qubits, $\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|1\rangle\right)$. As we've seen, if both qubits are measured in the computational basis, we're guaranteed to get matching results: $|0\rangle|0\rangle$ or $|1\rangle|1\rangle$.

In quantum theory, a measurement of either qubit causes the state to collapse from $\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|1\rangle\right)$ to either $|0\rangle|0\rangle$ or $|1\rangle|1\rangle$. But, the two qubits could be very far apart, perhaps separated by light-years. Can the measurement of one qubit affect the other qubit instantaneously, over any distance, faster than the speed of light? Of course not, Einstein argued. Nothing travels faster than the speed of light. This assumption is given the technical name *locality*: An object can be affected only by its local environment, including anything that can travel to its location at speeds up to the speed of light. In other words, an object *cannot* be affected *instantaneously* by something that happens far away.

(Actually, Einstein didn't mention locality or the speed of light at all in the 1935 paper. He simply wrote that "since at the time of measurement the two systems no longer interact, no real change can take place in the second system in consequence of anything that may be done to the first system." An elaborate exegesis has been constructed around the 1935 paper. The exegesis may deviate mildly from Einstein's original position, but I think the exegesis is easier to understand than the original paper, so I'm presenting a version of the exegesis.)

If one qubit in the entangled pair is measured as $|0\rangle$, we *instantaneously* know that the other qubit will be $|0\rangle$ whenever it is measured (in the computational basis), even though, according to locality, the measurement of the first qubit cannot affect the second qubit so quickly. Einstein's conclusion is that the qubits must have been in the state $|0\rangle|0\rangle$ all along, or at least somehow preprogrammed to turn out as $|0\rangle|0\rangle$ when measured in the computational basis. *Realism* is what we call the idea that the qubits are preprogrammed with their measurement results. Realism simply means that measurement reveals properties that the objects already have; the measurements do not conjure the properties out of thin air.

So Einstein's argument is that common sense seems to demand locality, and locality, applied to our entangled pair, demands realism. The combination of locality and realism is called *local realism*: Objects have well-defined properties regardless of whether anyone's measuring or observing them, and regardless of whether anyone's measuring any distant objects. Local realism is an everyday, commonsense assumption, and we might call it a philosophical assumption. If we believe that an apple has a weight even when we're not weighing it, then we believe in realism. If we believe that weighing an apple has no effect on the weight of a pear, then we believe in locality.

We will see that it's possible to do an experiment to test these philosophical assumptions. In physics, philosophy is more than theory. Philosophy can make specific, quantitative predictions that we can test in the lab: We can do experimental philosophy. In fact, from the comfort of your web browser, you can do the philosophical experiment—a version of a Nobel Prize–winning experiment—with IBM Quantum.

But first, let's return to 1935. Einstein argued that the entangled pair all along has properties causing the eventual measurement to yield either $|0\rangle|0\rangle$ or $|1\rangle|1\rangle$. But quantum theory only predicts a 50% chance of either outcome. Therefore, Einstein concluded, quantum theory is *incomplete*: Quantum theory is ignorant of the properties that determine, with certainty, the results of measurement. In fact, the title of the 1935 paper is "Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?" Einstein's answer is a resounding no. A complete theory, he argued, would replace quantum-mechanical probabilities with certainties. The certainties establish realism (all the properties that might be measured are predetermined), and realism guarantees locality (the two qubits independently are preprogrammed with the results of all possible measurements, so the measurement of one qubit doesn't affect the other; in fact, the measurement of one qubit doesn't even affect *that* qubit; the measurement merely reveals a preexisting property). Local realism banishes the mystery as to why the two qubits end up in matching states ($|0\rangle|0\rangle$ or $|1\rangle|1\rangle$): They were in well-defined matching states from the beginning; they were never in a fuzzy blend of probabilities.

Einstein never said quantum mechanics was wrong. Even in 1935, there was plenty of data to prove that it was accurate. Einstein said only that it was incomplete, and he hoped that someone would complete it, preserving its accurate predictions while eliminating its ignorance, replacing its probabilities with certainties.

For nearly three decades, Einstein's hope was perfectly legitimate. But then, in 1964, John Bell proved that quantum theory is inherently inconsistent with local realism. So if existing quantum theory is accurate, local realism is impossible. And if local realism is valid, then existing quantum theory must be flawed. Bell showed that *local realism imposes a mathematical constraint on measurable results*. The constraint imposed by local realism is called a *Bell inequality*. Bell showed that quantum theory disobeys Bell inequalities. But the real question is this: Does reality obey Bell inequalities, in conformity with local realism? Or does reality disobey Bell inequalities, in compliance with quantum theory?

Here's a personal anecdote to emphasize that fact that *local realism imposes constraints*. I was at a meeting of professors teaching interdisciplinary courses. I was there because I was teaching an interdisciplinary course about quantum entanglement and Bell inequalities. Another professor was there because he was teaching a course about local and sustainable agriculture. Making the point that not all food can be grown locally, he said, "Localism imposes constraints." How uncanny! The point of his course was that localism imposes constraints, and the point of my course was that locality imposes constraints.

Clauser, one of the recipients of the 2022 Nobel Prize, published a variation of the original Bell inequality in 1969. This variation is called the CHSH inequality, after Clauser and his coauthors (Horne, Shimony, and Holt). Then in 1972, Clauser and his student, Stuart Freedman, did the first experimental test of a Bell inequality. They showed that reality disobeyed the Bell inequality, proving that local realism is invalid, or at least that it doesn't apply to entangled particles.

We will now derive the CHSH inequality and test it experimentally with IBM Quantum. The CHSH inequality is usually written in terms of variables A and B, which conventionally represent the measurements of Alice and Bob. It would be confusing for A and B to represent the measurements of Penelope and Odysseus, so we will release Penelope and Odysseus to their customary tasks (fending off suitors and defeating Trojans, respectively). We now meet our new heroes, Athena and Bellerophon, who became experts in entanglement when working together to lasso Pegasus (who attempted to remain unentangled).

IBM's qubits are made of superconducting circuits, but the CHSH inequality applies to any pair of entangled objects. In fact, we want the CHSH inequality to apply to as many scenarios as possible: entangled photons, entangled electrons, etc. We want to describe our experiment as generically as possible so that it applies as broadly as possible.

We imagine a system consisting of two objects, A and B, that are measured separately, as shown in Fig. 6.1. Each measurement device has two settings, labeled 1 and 2. We do not need to specify the nature of the objects or the measurements, except that each measurement yields one of two results, which we will designate as +1 and −1. These two numbers are arbitrary, analogous to assigning +1 to "heads" and −1 to "tails" when recording the results of a coin toss. Similarly, if our objects are silver atoms of the type used in Chapter 1, +1 could indicate "deflected toward the north pole of a magnet," −1 could indicate "deflected toward the south pole of a magnet," and the measurement settings control the orientation of the magnet.

The measured result (±1) for object A is called A_1 when the measurement setting is 1, and the measured result is called A_2 when the measurement setting is 2. B_1 and B_2 are defined similarly for the measurements of object B.

For each pair of objects A and B, Athena chooses one measurement setting for object A (so she's measuring either A_1 or A_2 but not both), and Bellerophon chooses one measurement setting for object B (so he's measuring either B_1 or B_2 but not both). Suppose they measure A_1 and B_1 for many pairs of objects. For each pair of objects, they choose to calculate the product A_1B_1, which must be ±1. They can then find the average of A_1B_1. Similarly, they can find the averages of A_1B_2, A_2B_1, and A_2B_2.

Next, we define the quantity

$$S = A_1B_1 - A_1B_2 + A_2B_1 + A_2B_2. \qquad (6.1)$$

S does not have any obvious physical significance; it's just something we can calculate if we know all four variables on the right side of the equation. Since each of these four variables is ±1, S must be ±2: $S = A_1(B_1 - B_2) + A_2(B_1 + B_2)$, and one of the quantities in parentheses must be 0 and the other must be ±2,

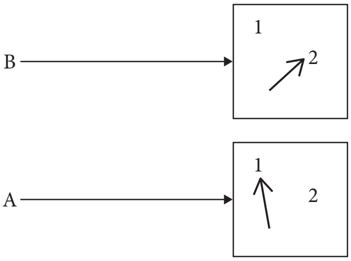


Figure 6.1. Detectors with two settings. Adapted from Jed Brody and Robert Avram, "Testing a Bell Inequality with a Remote Quantum Processor," *The Physics Teacher*, March 2023, https://doi.org/10.1119/5.0069073.

and the variables in front of the parentheses are ±1. If that's not clear, we list in Table 6.1 all possible combinations of values of A_1, A_2, B_1, and B_2.

Since S must be ±2, the average of S for many pairs of objects must be between −2 and +2: $-2 \leq S_{\text{average}} \leq 2$. This is the CHSH inequality. Restated in terms of Eq. (6.1), and using angle brackets to represent averages,

$$<S> = <A_1B_1> - <A_1B_2> + <A_2B_1> + <A_2B_2> \tag{6.2}$$

must be between −2 and 2. That's it! That's the CHSH inequality, a constraint imposed by the philosophical assumption of local realism. The assumption of local realism is subtle: S depends on both A_1 and A_2, so Eq. (6.1) implicitly assumes that A_1 and A_2 both exist for every object A, even though only one of the two variables is measured; this is realism. We also implicitly assume that A_1 and A_2 do not depend on the measurement setting for object B; this is locality. In other words, Bellerophon's measurement setting does not affect the result of Athena's measurement.

In contrast with local realism, quantum theory predicts that the CHSH inequality can be disobeyed. To disobey the CHSH inequality, we need quantum entanglement. As in previous chapters, we create the entangled pair $\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|1\rangle\right)$ with the circuit in Fig. 6.2.

Table 6.1

A_1	A_2	B_1	B_2	$S = A_1B_1 - A_1B_2 + A_2B_1 + A_2B_2$
+1	+1	+1	+1	+2
+1	+1	+1	−1	+2
+1	+1	−1	+1	−2
+1	+1	−1	−1	−2
+1	−1	+1	+1	−2
+1	−1	+1	−1	+2
+1	−1	−1	+1	−2
+1	−1	−1	−1	+2
−1	+1	+1	+1	+2
−1	+1	+1	−1	−2
−1	+1	−1	+1	+2
−1	+1	−1	−1	−2
−1	−1	+1	+1	−2
−1	−1	+1	−1	−2
−1	−1	−1	+1	+2
−1	−1	−1	−1	+2

Next, we need to establish two ways to measure each of the two qubits so that Athena can perform measurements of A_1 and A_2 (on the bottom qubit), and Bellerophon can perform measurements of B_1 and B_2 (on the top qubit). Regardless of how a qubit is made, $|0\rangle$ can represent "spin up" (an arrow in the +z direction), and $|1\rangle$ can represent "spin down" (an arrow in the –z direction). A detailed familiarity with spin is unnecessary. We need to know only that spin is a property that can be measured in different directions. Figure 6.3 gives the geometric representation of $|0\rangle$ and $|1\rangle$.

The figure shows the xz-plane of something called the *Bloch sphere*, and the arrows are known as *Bloch vectors*. To represent an arrow in an arbitrary direction in the xz-plane, we can use the qubit

$$|\theta\rangle = \cos(\theta/2)|0\rangle + \sin(\theta/2)|1\rangle. \tag{6.3}$$

Let's understand why this equation agrees with the figure. We have to remember some basic trigonometry:

$$\begin{aligned} \cos(0°) &= 1 & \sin(0°) &= 0 \\ \cos(45°) &= \frac{1}{\sqrt{2}} & \sin(45°) &= \frac{1}{\sqrt{2}} \\ \cos(90°) &= 0 & \sin(90°) &= 1 \end{aligned}$$

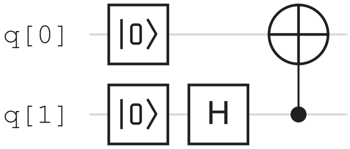


Figure 6.2. A circuit to produce an entangled state, created using IBM Quantum.

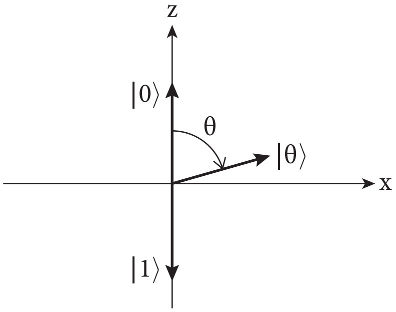


Figure 6.3. The xz-plane of the Bloch sphere. Originally published in Jed Brody and Robert Avram, "Testing a Bell Inequality with a Remote Quantum Processor," *The Physics Teacher*, March 2023, https://doi.org/10.1119/5.0069073.

Let's examine Eq. (6.3) when $\theta = 0°$. According to the Fig. 6.3, when θ is 0°, $|\theta\rangle$ should coincide with $|0\rangle$: an arrow in the +z direction. Does Eq. (6.3) agree? If we plug $\theta = 0°$ into Eq. (6.3), we get $|\theta = 0°\rangle = \cos(0°)|0\rangle + \sin(0°)|1\rangle = |0\rangle$. Equation (6.3) works!

Let's try $\theta = 180°$. According to the figure, $|\theta\rangle$ should now point straight down, coinciding with $|1\rangle$, in the –z direction. Plugging $\theta = 180°$ into Eq. (6.3), we get $|\theta = 180°\rangle = \cos(180°/2)|0\rangle + \sin(180°/2)|1\rangle = \cos(90°)|0\rangle + \sin(90°)|1\rangle = |1\rangle$. Again, success!

Recall that a measurement that yields $|0\rangle$ or $|1\rangle$ is called a measurement in the computational basis, which is also called the z basis. It's called the z basis because the two possible measurement results correspond with the +z and –z directions of the Bloch sphere. Similarly, a measurement in the x basis yields either $|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$ or $|-\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$ because the corresponding arrows point in either the +x or –x directions.

So let's examine $\theta = 90°$, which should make $|\theta\rangle$ point in the +x direction. Plugging $\theta = 90°$ into Eq. (6.3), we get $|\theta = 90°\rangle = \cos(90°/2)|0\rangle + \sin(90°/2)|1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) = |+\rangle$. As promised, an arrow pointing in the +x direction corresponds with a possible result of measuring in the x basis.

Every measurement in IBM Quantum yields 0 or 1, representing a measurement of spin along the z axis. To effectively measure spin in any other direction in the xz-plane, we need to rotate the arrow from the direction we want, onto the z axis. There's a quantum gate that accomplishes this: $R_y(-\theta)$. The subscript indicates rotation around the y axis, which is perpendicular to the plane of Fig. 6.3. $R_y(-\theta)$ affects $|0\rangle$ and $|1\rangle$ as follows:

$$R_y(-\theta)|0\rangle = \cos(\theta/2)|0\rangle - \sin(\theta/2)|1\rangle, \tag{6.4}$$

and

$$R_y(-\theta)|1\rangle = \sin(\theta/2)|0\rangle + \cos(\theta/2)|1\rangle. \tag{6.5}$$

Positive θ represents clockwise rotation in the figure, which is why we apply $R_y(-\theta)$, which effects a counterclockwise rotation from the measurement direction onto the z axis. It requires some algebra, but we could combine Eqs. (6.3)–(6.5) to show that $R_y(-\theta)|\theta\rangle = |0\rangle$: The $R_y(-\theta)$ gate, applied to $|\theta\rangle$, rotates the arrow counterclockwise until it coincides with $|0\rangle$.

Athena measures the first qubit along the direction specified by an angle α, and Bellerophon measures the second qubit along a direction specified by an angle β. This means that Athena has to apply $R_y(-\alpha)$ to her qubit before measuring in the computational basis, and Bellerophon has to apply $R_y(-\beta)$. Applying $R_y(-\alpha)$ to the left qubit and $R_y(-\beta)$ to the right qubit in our entangled state, $\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|1\rangle\right)$, we get $\frac{1}{\sqrt{2}}\left[R_y(-\alpha)|0\rangle R_y(-\beta)|0\rangle + R_y(-\alpha)|1\rangle R_y(-\beta)|1\rangle\right]$.

Now can use Eqs. (6.4) and (6.5), replacing θ with α or β. We obtain $\frac{1}{\sqrt{2}}\left[\left(\cos\frac{\alpha}{2}|0\rangle - \sin\frac{\alpha}{2}|1\rangle\right)\left(\cos\frac{\beta}{2}|0\rangle - \sin\frac{\beta}{2}|1\rangle\right) + \left(\sin\frac{\alpha}{2}|0\rangle + \cos\frac{\alpha}{2}|1\rangle\right)\right.$ $\left.\left(\sin\frac{\beta}{2}|0\rangle + \cos\frac{\beta}{2}|1\rangle\right)\right]$. Using FOIL multiplication, we arrive next at $\frac{1}{\sqrt{2}}\left[\left(\cos\frac{\alpha}{2}\cos\frac{\beta}{2}|0\rangle|0\rangle - \cos\frac{\alpha}{2}\sin\frac{\beta}{2}|0\rangle|1\rangle - \sin\frac{\alpha}{2}\cos\frac{\beta}{2}|1\rangle|0\rangle + \sin\frac{\alpha}{2}\sin\frac{\beta}{2}|1\rangle|1\rangle\right)\right.$ $\left.+\left(\sin\frac{\alpha}{2}\sin\frac{\beta}{2}|0\rangle|0\rangle + \sin\frac{\alpha}{2}\cos\frac{\beta}{2}|0\rangle|1\rangle + \cos\frac{\alpha}{2}\sin\frac{\beta}{2}|1\rangle|0\rangle + \cos\frac{\alpha}{2}\cos\frac{\beta}{2}|1\rangle|1\rangle\right)\right]$. Combining terms with the same kets gives us $\frac{1}{\sqrt{2}}\left[\left(\cos\frac{\alpha}{2}\cos\frac{\beta}{2} + \right.\right.$ $\left.\sin\frac{\alpha}{2}\sin\frac{\beta}{2}\right)|0\rangle|0\rangle + \left(-\cos\frac{\alpha}{2}\sin\frac{\beta}{2} + \sin\frac{\alpha}{2}\cos\frac{\beta}{2}\right)|0\rangle|1\rangle + \left(-\sin\frac{\alpha}{2}\cos\frac{\beta}{2} + \right.$ $\left.\left.\cos\frac{\alpha}{2}\sin\frac{\beta}{2}\right)|1\rangle|0\rangle + \left(\sin\frac{\alpha}{2}\sin\frac{\beta}{2} + \cos\frac{\alpha}{2}\cos\frac{\beta}{2}\right)|1\rangle|1\rangle\right]$.

Next we have to apply the trigonometry identities $\cos(\alpha-\beta)=\cos\alpha\cos\beta + \sin\alpha\sin\beta$, and $\sin(\alpha-\beta)=\sin\alpha\cos\beta-\cos\alpha\sin\beta$. Who would have thought that trigonometry has relevance in an experimental investigation of the ultimate nature of reality! Using these identities, the expression simplifies to $\frac{1}{\sqrt{2}}\cos\frac{\alpha-\beta}{2}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}\sin\frac{\alpha-\beta}{2}|0\rangle|1\rangle + \frac{1}{\sqrt{2}}\sin\frac{\beta-\alpha}{2}|1\rangle|0\rangle +$ $\frac{1}{\sqrt{2}}\cos\frac{\alpha-\beta}{2}|1\rangle|1\rangle$.

The probability of each result is found by squaring the probability amplitudes in the preceding final expression. So the probabilities of measuring the results 00, 01, 10, and 11 are

$$P(00) = \frac{1}{2}\cos^2\frac{\alpha-\beta}{2}, \tag{6.6}$$

$$P(01) = \frac{1}{2}\sin^2\frac{\alpha-\beta}{2}, \tag{6.7}$$

$$P(10) = \frac{1}{2}\sin^2\frac{\alpha-\beta}{2}, \tag{6.8}$$

and

$$P(11) = \frac{1}{2}\cos^2\frac{\alpha-\beta}{2}, \tag{6.9}$$

respectively.

In the derivation of the CHSH inequality, the two possible measurement results are +1 and −1, not 0 and 1. To match the assumptions made in the derivation of the CHSH inequality, we must map the Boolean labels (0 and 1) to the "spin" values used in calculations (+1 and −1). This seems a little fishy, but both pairs of labels, 0/1 and +1/–1, are arbitrary. $|0\rangle$ doesn't mean there's 0 of something, and $|1\rangle$ doesn't mean there's 1 of something. We could easily substitute $|0\rangle = |+1\rangle$ and $|1\rangle = |-1\rangle$, or, for that matter, $|0\rangle = |\text{cat}\rangle$ and $|1\rangle = |\text{dog}\rangle$, or $|0\rangle = |\text{pancreas}\rangle$ and $|1\rangle = |\text{argyle}\rangle$. The physical reality is simply that there are two possible measurement results, which we can label any way we like. We can map one pair of labels to another pair of labels, as long as we're consistent. Now, we want to map 0 to +1, and 1 to –1.

So when we measure 00, we map each 0 to +1 and obtain the product $(+1)(+1) = +1$; recall that Athena and Bellerophon want to multiply their results together for each pair of qubits. Proceeding to 01, 10, and 11, we obtain Table 6.2.

The product of the two spin values is +1 for 00 and 11, and the product of the two spin values is –1 for 01 and 10. So $P(00) + P(11)$ is the probability that the product is +1, and $P(01) + P(10)$ is the probability that the product is –1. Recall that Athena and Bellerophon wanted to find that average of the product of the results, <AB>. One way to find an average is to find the sum of each possible result multiplied by its probability. For example, if 1/5 of the people in a group are 5 feet tall, 2/3 are 6 feet tall, and 2/15 are 15 feet tall, then the average height is (5 feet) × (1/5) + (6 feet) × (2/3) + (15 feet) × (2/15) = 7 feet tall. (The giants really bring up the average.) Following the same logic, the average of the product of the Athena's result and Bellerophon's result is

$$<AB> = (+1) \times P(00) + (-1) \times P(01) + (-1) \times P(10) + (+1) \times P(11). \qquad (6.10)$$

<AB> is called a quantum correlation, or a spin correlation. If A and B are always the same (00 or 11), the quantum correlation is +1. If A and B are always different from each other (01 or 10), the quantum correlation is –1. If A and B are equally likely to be the same or different, the quantum correlation is 0.

Combining Eqs. (6.6)–(6.10),

$$\begin{aligned} <AB> &= \frac{1}{2}\cos^2\frac{\alpha-\beta}{2} - \frac{1}{2}\sin^2\frac{\alpha-\beta}{2} - \frac{1}{2}\sin^2\frac{\alpha-\beta}{2} + \frac{1}{2}\cos^2\frac{\alpha-\beta}{2} \\ &= \cos^2\frac{\alpha-\beta}{2} - \sin^2\frac{\alpha-\beta}{2} = \cos(\alpha-\beta), \qquad (6.11) \end{aligned}$$

using a final trigonometry identity, $\cos(2\theta) = \cos^2\theta - \sin^2\theta$.

Table 6.2

Result (Boolean labels)	Product of spin values
00	$(+1)(+1) = +1$
01	$(+1)(-1) = -1$
10	$(-1)(+1) = -1$
11	$(-1)(-1) = +1$

At long last, we can calculate <S> in the CHSH inequality. If α_1, α_2, β_1, and β_2 are the measurement angles for measurements of A_1, A_2, B_1, and B_2, respectively, then Eqs. (6.2) and (6.11) give the quantum prediction for <S>: $\cos(\alpha_1 - \beta_1) - \cos(\alpha_1 - \beta_2) + \cos(\alpha_2 - \beta_1) + \cos(\alpha_2 - \beta_2)$. If we choose $\alpha_1 = 0°$, $\alpha_2 = 90°$, $\beta_1 = 45°$, and $\beta_2 = 135°$, then we find $<S> = 2\sqrt{2}$, contradicting the CHSH inequality, $-2 \leq <S> \leq 2$. So we have two conflicting predictions for <S>, and we need to do an experiment to see which, if either, is correct. Is the commonsense philosophical assumption of local realism correct? Or is quantum theory correct? A lot is riding on this experiment!

Actually, experimental error tends to reduce the magnitude of <S>, so the quantum prediction of $2\sqrt{2}$ is never perfectly achieved. The goal of the experiment is really to test the CHSH inequality, $-2 \leq <S> \leq 2$, which is the commonsense prediction of local realism.

Figure 6.4 shows the circuit to determine $<A_1B_1>$. When Athena makes an A_1 measurement, her measurement angle is $\alpha_1 = 0°$: Her measurement is in the z direction, the computational basis, so she doesn't need to use a rotation gate. Bellerophon is making a B_1 measurement, so he uses the measurement angle $\beta_1 = 45°$, which is $\pi/4$ radians. This is shown in the R_y gate.

Figure 6.5 shows the circuit to determine $<A_1B_2>$, with $\alpha_1 = 0°$ and $\beta_2 = 135°$. Figure 6.6 shows the circuit for $<A_2B_1>$, with $\alpha_2 = 90°$ and $\beta_1 = 45°$. And last, Fig. 6.7 shows the circuit for $<A_2B_2>$, with $\alpha_2 = 90°$ and $\beta_2 = 135°$. I ran each circuit 1024 times on ibmq_belem. Table 6.3 shows the number of times 00, 01, 10, and 11 occurred. For each circuit, I want to estimate the probabilities of measuring 00, 01, 10, 11. Since 00 occurs 312 times out of the 1024 times I ran the A_1B_1 circuit, I estimate the probability as $P(00) = 312/1024 = 0.305$. Calculating all probabilities this way, we obtain Table 6.4. The final column uses Eq. (6.10) to calculate spin correlations from probabilities.

At last we can use Eq. (6.2) to calculate <S>: $<S> = 0.473 - (-0.506) + 0.621 + 0.625 = 2.225$. It's greater than 2, so the CHSH inequality is *disobeyed*! Let's take a moment to stand agog and aghast. We showed that the only pos-

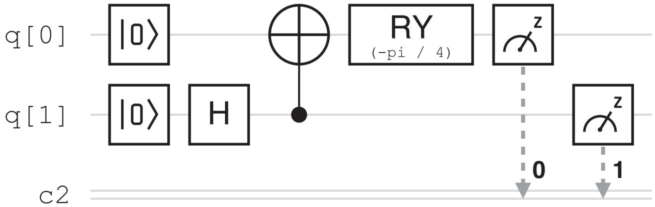


Figure 6.4. A circuit to measure $<A_1B_1>$, created using IBM Quantum. Adapted from Jed Brody and Robert Avram, "Testing a Bell Inequality with a Remote Quantum Processor," *The Physics Teacher*, March 2023, https://doi.org/10.1119/5.0069073.

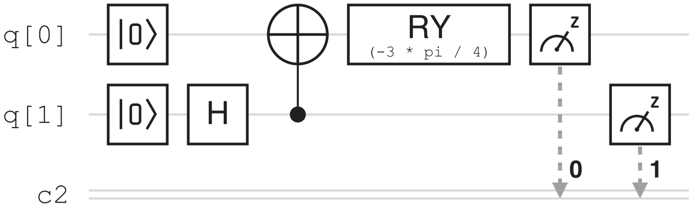


Figure 6.5. A circuit to measure <A_1B_2>, created using IBM Quantum. Adapted from Jed Brody and Robert Avram, "Testing a Bell Inequality with a Remote Quantum Processor," *The Physics Teacher*, March 2023, https://doi.org/10.1119/5.0069073.

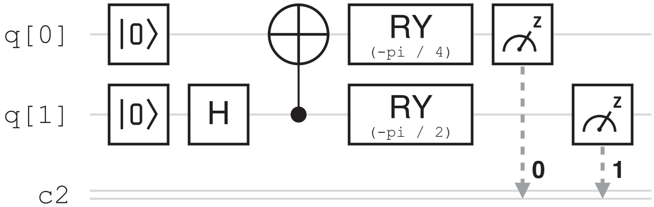


Figure 6.6. A circuit to measure <A_2B_1>, created using IBM Quantum. Adapted from Jed Brody and Robert Avram, "Testing a Bell Inequality with a Remote Quantum Processor," *The Physics Teacher,* March 2023, https://doi.org/10.1119/5.0069073.

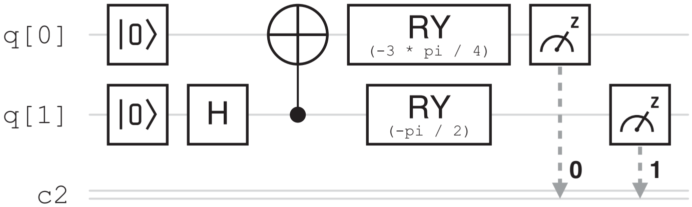


Figure 6.7. A circuit to measure <A_2B_2>, created using IBM Quantum. Adapted from Jed Brody and Robert Avram, "Testing a Bell Inequality with a Remote Quantum Processor," *The Physics Teacher*, March 2023, https://doi.org/10.1119/5.0069073.

Table 6.3

	00	01	10	11
A_1B_1 circuit	312	211	59	442
A_1B_2 circuit	65	461	310	188
A_2B_1 circuit	251	68	126	579
A_2B_2 circuit	241	69	123	591

Table 6.4

	00	01	10	11	
A_1B_1 circuit	0.305	0.206	0.058	0.432	$<A_1B_1>=0.473$
A_1B_2 circuit	0.063	0.450	0.303	0.184	$<A_1B_2>=-0.506$
A_2B_1 circuit	0.245	0.066	0.123	0.565	$<A_2B_1>=0.621$
A_2B_2 circuit	0.235	0.067	0.120	0.577	$<A_2B_2>=0.625$

sible values of S, for each pair of qubits, are +2 or −2. Since the only possible values of S are +2 and −2, surely the average value of S is between −2 and +2. This is a constraint mandated by common sense. And yet *nature disobeys this constraint!* Experimentally, we find that the average value of S exceeds 2!

Where did we go wrong? All we used was basic logic and arithmetic. Did we just invalidate arithmetic? Do we need to inform elementary math teachers, so they stop spreading their insidious lies? No, arithmetic isn't the problem here. But some false assumption, whatever it may be, leads to a constraint flagrantly disregarded by nature.

Let's look again at our equation:

$$S = A_1B_1 - A_1B_2 + A_2B_1 + A_2B_2.$$

If A_1, A_2, B_1, and B_2 are physical properties that exist all along, independent of measurement, and each of these properties is represented by +1 or −1, then S must indeed be either +2 or −2, and the average of S must indeed be between −2 and +2. Since this constraint is disobeyed, we can only conclude that the particles' properties (A_1, A_2, B_1, and B_2) do *not* exist all along, independent of measurement.

In quantum theory, S exists *only* as an average over many qubit pairs. (In other words, <S> exists, but S is undefined for a single qubit pair.) We define the average of S as the average of $(A_1B_1 - A_1B_2 + A_2B_1 + A_2B_2)$. Even if A_1 and A_2 don't both exist for a single qubit, the average of A_1B_1 exists for the many qubit pairs for which Athena measures A_1 and Bellerophon measures B_1; and similarly, the average of A_2B_1 exists for the many qubit pairs for which Athena measures A_2 and Bellerophon measures B_1. So we obtain the average of each of the four terms from different sets of qubit pairs. We combine these four averages to obtain the average of S, a quantity that may exist only as an average, not a property of any individual qubit pair. But again, if S *does* exist for each qubit pair, then the average of S must not exceed 2. And the assumption that S exists for each qubit pair is much like the assumption that my height and weight both exist even when I'm not measuring them.

We find in the laboratory, again and again, that nature disobeys our common sense. While disobeying our common sense, nature simultaneously adheres to a different set of formulas, those established by quantum mechanics.

This puts physicists in a pickle. Quantum mechanics accurately predicts the outcomes of measurements, but we don't know what to say about particles when we're not looking at them; all we know for sure is that our common sense gets it wrong. So, the interpretation of quantum mechanics remains a topic of speculation, controversy, equivocation, or indifference.

Some physicists argue that unobserved particles are simply none of our business; the business of physics is predicting observations. Let philosophers handle the unobserved particles. Physicists thus divest themselves of awkward questions and focus on what they do best. This viewpoint is admirably humble in its acknowledgment of the limitations of physics. Or is it just lazy?

Some physicists. like Deutsch, argue that every possible measurement outcome occurs simultaneously in parallel universes. Impassioned arguments are made for and against this "many worlds interpretation" (MWI). In its favor, MWI avoids the distinction between abrupt measurements and smooth evolution of probabilities. The laws of quantum mechanics provide probabilities of different outcomes. What, ultimately, determines which outcomes occur and which don't? MWI circumvents this question entirely because *all* outcomes occur. And it explains the flaw in our equation for S like this: A_1 and A_2 exist in separate universes, so in a single universe, we can't define S, which depends on both. Specifically, Athena's qubit has the property A_1 in the universe where she measures A_1, and it has the property A_2 in the universe where she measures A_2. In any one universe, Athena can measure only one of the two properties, so the qubit comes preprogrammed only with the property it needs in the universe it inhabits. If it comes preprogrammed with A_1, A_2 doesn't exist (in this universe), so S, which depends on A_2, doesn't exist either. If S doesn't exist for each pair of qubits, we can't prove that the average of S must be less than 2.

Perhaps, if there is only one universe, the error in our common sense is the belief in free will. Perhaps we are preprogrammed automatons, or the particles under observation diabolically influence our decisions. This possibility, though unpalatable and strange, is duly considered by physicists and philosophers. We could say Athena's qubit is preprogrammed only with the property A_1 (not A_2) because Athena *herself* is preprogrammed to measure A_1. This is called *superdeterminism*, which everyone but physicists just calls *determinism*: There's no free will. Since there's no free will, the qubit doesn't have to be prepared with values for different measurement settings. The one setting that will be chosen was predetermined at the beginning of time, along with the result, A_1. A_2 never existed for this qubit because it was never an option because there's no such thing as options in a superdetermined universe. And if A_2 doesn't exist for each qubit, then S doesn't exist for each qubit pair, so the CHSH inequality doesn't apply.

What if our universe isn't superdetermined, but there's still some mechanism informing the qubit whether A_1 or A_2 will be measured? More specifically, can a signal from the measurement devices reach the qubits before they split and (presumably) lose contact with each other? Experiments of this sort have actually been performed. Nobel laureate Alain Aspect effectively switched measurement settings while the particles were in flight toward the detectors. This last-minute switching should invalidate any signal from the measurement device to the location where the particles originated. There's no known mechanism by which this signal could have any effect, but Aspect got a Nobel Prize for invalidating this speculative theory. He showed that the Bell inequality is disobeyed even when the measurement settings change while the particles are in flight. So if the particle somehow knows, from the moment it originates, what the measurement setting will be, it's not because it receives a signal from the detector.

Aspect's experiment investigated, and rejected, the idea that signals from the detectors influence the qubits at their point of origin. Perhaps, instead, each qubit is influenced by the *final* settings, when the measurements occur, on *both* measurement devices. (We don't know *how* the measurement settings influence both qubits; we're just supposing they do.) Let's look at the first two terms in S: A_1B_1 and A_1B_2. We're assuming that the *same* A_1 appears in both terms. In other words, we're assuming that Athena's result, A_1, is *independent* of whether Bellerophon chooses to measure B_1 or B_2. If Athena's result A_1 actually depends on *Bellerophon's* distant measurement setting, then we would need *two* different mathematical symbols for A_1 (one for each of Bellerophon's measurement settings). Instead of writing A_1 in both cases, we'd have to write something like $A_{1,\text{Bellerophon }1}B_1 - A_{1,\text{Bellerophon }2}B_2$. But Bellerophon's results could also depend on the Athena's measurement settings, so we'd need to write $A_{1,\text{Bellerophon }1}B_{1,\text{Athena }1} - A_{1,\text{Bellerophon }2}B_{2,\text{Athena }1}$.

So S, instead of depending on four variables (A_1, A_2, B_1, and B_2), would depend on eight independent variables. In this case, it *would* be possible for S to exceed 2. This is one possible explanation for our measurement results. It's a bit strange: Athena's results depend on Bellerophon's measurement settings, no matter how far apart Athena and Bellerophon are.

If we thus accept nonlocality, it's possible to preserve realism: Each qubit is preprogrammed with properties that depend on the settings at both measurement devices. This is strange but straightforward if both qubits are measured simultaneously: Each qubit effectively responds to the setting at the detector it's entering, and also to the setting encountered simultaneously by the distant qubit. Ah, but we saw that simultaneity depends on the observer, so Penelope might see Athena and Bellerophon making simultaneous measurements, while Odysseus says Athena's measurement comes first, and Hector thinks Bellerophon's measurement comes first. Now, if Athena's measurement occurs first, then Bellerophon's result depends both on his own setting at the time he measures his qubit and on Athena's setting at the time of her earlier measurement. But the same two measurements occur in the opposite order for a different observer. So does Bellerophon's result depend on Athena's setting at the time of her *future* measurement? Does Bellerphon's qubit know the future?

To avoid this confusion, we might want to scrap both locality and realism: The measurement of one qubit creates a definite state for both qubits. So the qubit measured second is affected by the measurement of the first qubit, and by the setting of its own, later measurement. For all practical purposes, this is exactly what happens, and it's what we've said all along: The state $\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle+|1\rangle|1\rangle\right)$ collapses to either $|0\rangle|0\rangle$ or $|1\rangle|1\rangle$ when either qubit is measured in the computational basis. The only controversy is whether a quantum state describes physical reality, or only our knowledge of it.

Even if different observers disagree on which qubit was measured first, each observer's narrative is self-consistent. If the state collapses from $\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle+|1\rangle|1\rangle\right)$ to $|0\rangle|0\rangle$, it really doesn't matter whose measurement triggered the collapse. But what if Athena and Bellerophon measure in different bases?

Suppose Athena measures her qubit in the computational basis and obtains $|0\rangle$. Bellerophon, however, measures in the x basis and obtains $|+\rangle$. These results are self-consistent regardless of which measurement occurs first. Let's see how.

Odysseus says Athena's measurement occurs first and causes both qubits to collapse to $|0\rangle|0\rangle$. Now Bellerophon's qubit is in state $|0\rangle=\frac{1}{\sqrt{2}}\left(|+\rangle+|-\rangle\right)$, so that when he measures in the x basis, there's a 50% chance of obtaining $|+\rangle$.

Hector, on the other hand, says Bellerophon's measurement occurs first and causes both qubits to collapse to $|+\rangle|+\rangle$. Athena's qubit is in state $|+\rangle=\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)$, so that when she measures in the computational basis, there's a 50% chance of obtaining $|0\rangle$.

So both observers, Odysseus and Hector, have a self-consistent explanation for the results obtained by both Athena and Bellerophon. Odysseus and Hector disagree with each other, unsurprisingly; different observers see a different chronological order, according to special relativity. Odysseus and Hector are both correct, and equally correct, from their own perspectives.

Can we ever prove that the measurement of one qubit physically alters the distant qubit? I believe this claim can be neither proven nor disproven. If we say that the measurement of one particle affects the other, we really mean that the first measurement, of either particle, affects both; subsequent results are determined by this first measurement.

We'd like to look at Athena's particle before and after Bellerophon measures his particle, to see if Bellerophon's measurement affects Athena's particle. But then the initial observation of *Athena's* particle becomes the first measurement that determines subsequent results! Since it's impossible to do a measurement before the first measurement, it's impossible to observe whether the measurement of one particle physically alters the other.

Since that wasn't confusing enough, let's look now at an entangled state of three qubits, $\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle|0\rangle+|1\rangle|1\rangle|1\rangle\right)$. This trio of entangled qubits is named a GHZ state after a trio of physicists, Greenberger, Horne, and Zeilinger (the third recipient of the 2022 Nobel Prize). Suppose we measure all three qubits in the x basis. What results are possible, and with what probabilities? We have to replace each $|0\rangle$ with $|0\rangle=\frac{1}{\sqrt{2}}\left(|+\rangle+|-\rangle\right)$, and each $|1\rangle$ with $|1\rangle=\frac{1}{\sqrt{2}}\left(|+\rangle-|-\rangle\right)$:

$$\begin{aligned}\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle|0\rangle+|1\rangle|1\rangle|1\rangle\right)=\frac{1}{4}\big[&\left(|+\rangle+|-\rangle\right)\left(|+\rangle+|-\rangle\right)\left(|+\rangle+|-\rangle\right)\\&+\left(|+\rangle-|-\rangle\right)\left(|+\rangle-|-\rangle\right)\left(|+\rangle-|-\rangle\right)\big].\end{aligned}\tag{6.12}$$

To compute $\left(|+\rangle+|-\rangle\right)\left(|+\rangle+|-\rangle\right)\left(|+\rangle+|-\rangle\right)$, we first apply the FOIL rule to find that $\left(|+\rangle+|-\rangle\right)\left(|+\rangle+|-\rangle\right)=|+\rangle|+\rangle+|+\rangle|-\rangle+|-\rangle|+\rangle+|-\rangle|-\rangle$. Next, each of these four terms multiplies each ket in the final $\left(|+\rangle+|-\rangle\right)$:

$$\begin{aligned}\left(|+\rangle+|-\rangle\right)\left(|+\rangle+|-\rangle\right)\left(|+\rangle+|-\rangle\right)=&|+\rangle|+\rangle|+\rangle+|+\rangle|-\rangle|+\rangle+|-\rangle|+\rangle|+\rangle\\&+|-\rangle|-\rangle|+\rangle+|+\rangle|+\rangle|-\rangle+|+\rangle|-\rangle|-\rangle\\&+|-\rangle|+\rangle|-\rangle+|-\rangle|-\rangle|-\rangle.\end{aligned}\tag{6.13}$$

Similarly,

$$\begin{aligned}\left(|+\rangle-|-\rangle\right)\left(|+\rangle-|-\rangle\right)\left(|+\rangle-|-\rangle\right)=&|+\rangle|+\rangle|+\rangle-|+\rangle|-\rangle|+\rangle-|-\rangle|+\rangle|+\rangle\\&+|-\rangle|-\rangle|+\rangle-|+\rangle|+\rangle|-\rangle+|+\rangle|-\rangle|-\rangle\\&+|-\rangle|+\rangle|-\rangle-|-\rangle|-\rangle|-\rangle.\end{aligned}\tag{6.14}$$

In Eq. (6.14), we see a minus sign wherever an odd number of qubits are in state $|-\rangle$. These are the terms that cancel out corresponding terms in Eq. (6.13), when Eqs. (6.13) and (6.14) are combined in Eq. (6.12):

$$\begin{aligned}\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle|0\rangle+|1\rangle|1\rangle|1\rangle\right)=\frac{1}{2}\big(&|+\rangle|+\rangle|+\rangle+|-\rangle|-\rangle|+\rangle\\&+|+\rangle|-\rangle|-\rangle+|-\rangle|+\rangle|-\rangle\big).\end{aligned}\tag{6.15}$$

We see that when the three qubits in the GHZ state are measured in the x basis, an odd number of them are found to be $|+\rangle$, and an even number of them are found to be $|-\rangle$. All four outcomes that satisfy this condition are equally likely.

We've seen measurements in the z basis, which yield $|0\rangle$ or $|1\rangle$. We've seen measurements in the x basis, which yield $|+\rangle$ or $|-\rangle$. Now, let's introduce the y basis. A measurement in the y basis yields either $|i\rangle$ or $|-i\rangle$, which are defined in terms of computational basis states as follows:

$$|\mathrm{i}\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + \mathrm{i}|1\rangle\big) \tag{6.16a}$$

and

$$|-\mathrm{i}\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle - \mathrm{i}|1\rangle\big), \tag{6.16b}$$

where i is the square root of −1. The only thing we need to know about i is that $\mathrm{i}^2 = -1$. Otherwise, it behaves like any other algebraic symbol.

We'd like to write $|0\rangle$ and $|1\rangle$ in terms of $|\mathrm{i}\rangle$ and $|-\mathrm{i}\rangle$. Adding Eq. (6.16a) to Eq. (6.16b) gives us

$$|0\rangle = \frac{1}{\sqrt{2}}\big(|\mathrm{i}\rangle + |-\mathrm{i}\rangle\big). \tag{6.17a}$$

Subtracting Eq. (6.16b) from Eq. (6.16a) lets us find

$$|1\rangle = \frac{1}{\mathrm{i}\sqrt{2}}\big(|\mathrm{i}\rangle - |-\mathrm{i}\rangle\big). \tag{6.17b}$$

Now, suppose that the left and middle qubits in the GHZ state are measured in the y basis, and the right qubit is measured in the x basis. So we use Eq. (6.17) to rewrite the left and middle qubits, and we rewrite the right qubit using $|0\rangle = \frac{1}{\sqrt{2}}\big(|+\rangle + |-\rangle\big)$ and $|1\rangle = \frac{1}{\sqrt{2}}\big(|+\rangle - |-\rangle\big)$:

$$\begin{aligned}\frac{1}{\sqrt{2}}\big(|0\rangle|0\rangle|0\rangle + |1\rangle|1\rangle|1\rangle\big) &= \frac{1}{4}\Big[\big(|\mathrm{i}\rangle + |-\mathrm{i}\rangle\big)\big(|\mathrm{i}\rangle + |-\mathrm{i}\rangle\big)\big(|+\rangle + |-\rangle\big)\\ &\quad + \frac{1}{\mathrm{i}^2}\big(|\mathrm{i}\rangle - |-\mathrm{i}\rangle\big)\big(|\mathrm{i}\rangle - |-\mathrm{i}\rangle\big)\big(|+\rangle - |-\rangle\big)\Big]\\ &= \frac{1}{4}\Big[\big(|\mathrm{i}\rangle + |-\mathrm{i}\rangle\big)\big(|\mathrm{i}\rangle + |-\mathrm{i}\rangle\big)\big(|+\rangle + |-\rangle\big)\\ &\quad - \big(|\mathrm{i}\rangle - |-\mathrm{i}\rangle\big)\big(|\mathrm{i}\rangle - |-\mathrm{i}\rangle\big)\big(|+\rangle - |-\rangle\big)\Big]\end{aligned} \tag{6.18}$$

using $\mathrm{i}^2 = -1$. Next, we multiply like we did in Eqs. (6.13) and (6.14) to obtain

$$\begin{aligned}\big(|\mathrm{i}\rangle + |-\mathrm{i}\rangle\big)\big(|\mathrm{i}\rangle + |-\mathrm{i}\rangle\big)\big(|+\rangle + |-\rangle\big) &= |\mathrm{i}\rangle|\mathrm{i}\rangle|+\rangle + |\mathrm{i}\rangle|-\mathrm{i}\rangle|+\rangle + |-\mathrm{i}\rangle|\mathrm{i}\rangle|+\rangle\\ &\quad + |-\mathrm{i}\rangle|-\mathrm{i}\rangle|+\rangle + |\mathrm{i}\rangle|\mathrm{i}\rangle|-\rangle + |\mathrm{i}\rangle|-\mathrm{i}\rangle|-\rangle\\ &\quad + |-\mathrm{i}\rangle|\mathrm{i}\rangle|-\rangle + |-\mathrm{i}\rangle|-\mathrm{i}\rangle|-\rangle\end{aligned} \tag{6.19}$$

and

$$\begin{aligned}-\big(|\mathrm{i}\rangle - |-\mathrm{i}\rangle\big)\big(|\mathrm{i}\rangle - |-\mathrm{i}\rangle\big)\big(|+\rangle - |-\rangle\big) &= -|\mathrm{i}\rangle|\mathrm{i}\rangle|+\rangle + |\mathrm{i}\rangle|-\mathrm{i}\rangle|+\rangle + |-\mathrm{i}\rangle|\mathrm{i}\rangle|+\rangle\\ &\quad - |-\mathrm{i}\rangle|-\mathrm{i}\rangle|+\rangle + |\mathrm{i}\rangle|\mathrm{i}\rangle|-\rangle - |\mathrm{i}\rangle|-\mathrm{i}\rangle|-\rangle\\ &\quad - |-\mathrm{i}\rangle|\mathrm{i}\rangle|-\rangle + |-\mathrm{i}\rangle|-\mathrm{i}\rangle|-\rangle.\end{aligned} \tag{6.20}$$

On the right side of Eq. (6.20), we see a minus sign in front of three kets if there is an even number of minus signs within the ket labels (as either $|-\rangle$ or

$|-\mathrm{i}\rangle$). These are the terms the cancel out terms in Eq. (6.19). So, combining Eqs. (6.18) through (6.20) gives us

$$\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle|0\rangle+|1\rangle|1\rangle|1\rangle\right)=\frac{1}{2}\left(|\mathrm{i}\rangle|-\mathrm{i}\rangle|+\rangle+|-\mathrm{i}\rangle|\mathrm{i}\rangle|+\rangle\right.$$
$$\left.+|\mathrm{i}\rangle|\mathrm{i}\rangle|-\rangle+|-\mathrm{i}\rangle|-\mathrm{i}\rangle|-\rangle\right). \qquad (6.21\mathrm{a})$$

Equation (6.21a) tells us what happens if we have a GHZ state and measure the first two qubits in the y basis and the third one in the x basis: an odd number of qubits will be found in a state whose label includes a minus sign (either $|-\rangle$ or $|-\mathrm{i}\rangle$). What if we want to measure only the middle qubit in the x basis, and the other two in the y basis? We could repeat the derivation of Eq. (6.21a), changing only the order of the three qubits, so that the middle one is either $|+\rangle$ or $|-\rangle$:

$$\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle|0\rangle+|1\rangle|1\rangle|1\rangle\right)=\frac{1}{2}\left(|\mathrm{i}\rangle|+\rangle|-\mathrm{i}\rangle+|-\mathrm{i}\rangle|+\rangle|\mathrm{i}\rangle\right.$$
$$\left.+|\mathrm{i}\rangle|-\rangle|\mathrm{i}\rangle+|-\mathrm{i}\rangle|-\rangle|-\mathrm{i}\rangle\right). \qquad (6.21\mathrm{b})$$

And if we measured the first qubit in the x basis, and the other two in the y basis, we find

$$\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle|0\rangle+|1\rangle|1\rangle|1\rangle\right)=\frac{1}{2}\left(|+\rangle|-\mathrm{i}\rangle|\mathrm{i}\rangle+|+\rangle|\mathrm{i}\rangle|-\mathrm{i}\rangle\right.$$
$$\left.+|-\rangle|\mathrm{i}\rangle|\mathrm{i}\rangle+|-\rangle|-\mathrm{i}\rangle|-\mathrm{i}\rangle\right). \qquad (6.21\mathrm{c})$$

So, in general, if any one of the qubits is measured in the x basis, and the other two are measured in the y basis, an odd number of qubits are found in a state labeled with a minus.

Quantum theory predicts that Eqs. (6.15), (6.21a), (6.21b), and (6.21c) are all correct. We will now show that according to local realism, the four equations cannot all be correct. As we saw with the CHSH inequality, we have two conflicting predictions, one made by quantum theory, and one made by local realism. Again, an experiment is needed to determine which prediction is correct.

According to local realism, measurements reveal properties that qubits already have. So if we measure a qubit in the x basis and obtain $|+\rangle$, the potential for this result must have been hidden within the qubit all along. Let's say the hidden property, which is revealed by a measurement in the x basis, is called X. Let's say that $X=+1$ if the measurement would find the qubit in state $|+\rangle$, and $X=-1$ if the measurement would find the qubit in state $|-\rangle$.

Similarly, if a qubit is measured in the y basis and result $|\mathrm{i}\rangle$ is obtained, this result, too, must have been hidden within the qubit all along. Let's say that the hidden property, revealed by a measurement in the y basis, is called Y. $Y=+1$ if the measurement would yield $|\mathrm{i}\rangle$, and $Y=-1$ if the measurement would yield $|-\mathrm{i}\rangle$.

The qubit cannot know in advance if it will be measured in the x basis, the y basis, or some other basis. So according to local realism, the qubit must hide within itself every property that would be revealed by every possible measurement. For example, each qubit must have a property X that is revealed by a measurement in the x basis, as well as a property Y that is revealed by a measurement in the y basis. If we have three qubits (call them 1, 2, and 3), we can use subscripts (1, 2, 3) on X and Y so we have separate variables for each qubit.

For our GHZ state of three qubits, the first qubit has properties X_1 and Y_1, the second qubit has properties X_2 and Y_2, and the third qubit has properties X_3 and Y_3. Each of these properties is a number that must be +1 or −1.

According to Eq. (6.15), when all three qubits are measured in the x basis, an even number of them are found in the $|-\rangle$ state. This means that if we could look at X_1, X_2, and X_3, an even number of them would be −1. So if we multiply the three numbers together,

$$X_1X_2X_3 = +1 \tag{6.22}$$

because it contains an even number of factors of −1.

Next, let's look at Eq. (6.21a). The measurements on the right side reveal Y_1, Y_2, and X_3. We see, in all four terms, that an odd number of −1 results occur. This means that

$$Y_1Y_2X_3 = -1. \tag{6.23a}$$

Similarly, Eq. (6.21b) implies that

$$Y_1X_2Y_3 = -1, \tag{6.23b}$$

and Eq. (6.21c) implies that

$$X_1Y_2Y_3 = -1. \tag{6.23c}$$

Now, since Y_1 is +1 or −1, in either case, $Y_1Y_1 = +1$. Similarly, $Y_2Y_2 = +1$ and $Y_3Y_3 = +1$. So we can multiply $X_1X_2X_3$ by (+1)(+1)(+1) without changing it:

$$X_1X_2X_3 = X_1X_2X_3(+1)(+1)(+1) = X_1X_2X_3Y_1Y_1Y_2Y_2Y_3Y_3. \tag{6.24}$$

Reordering variables,

$$X_1X_2X_3 = (Y_1Y_2X_3)(Y_1X_2Y_3)(X_1Y_2Y_3). \tag{6.25}$$

We recognize Eq. (6.23) on the right-hand side. Each of the three terms in parentheses is −1, so

$$X_1X_2X_3 = (-1)(-1)(-1) = -1. \tag{6.26}$$

But Eq. (6.26) contradicts Eq. (6.22)! So if local realism is valid, there must be an error in one of our starting points, Eqs. (6.15) and (6.21). But experiment shows that Eqs. (6.15) and (6.21) are both accurate, so local realism must be incorrect.

What can this mean? Do the qubits have no fixed properties before measurement? Do the measurements conjure the properties out of thin air? But then how are the measurements of the three qubits correlated? Do they communicate with one another instantaneously across any distance? Or did the qubits somehow know in advance the bases they would be measured in, through either superdeterminism or some signal containing information about the measurement settings? As stated in an old commercial, in response to the question, “How many licks does it take to get to the center of a Tootsie Roll Pop?”: The world may never know.

Chapter 7

Quantum Adder

Like Regular Addition, but Way More Confusing

A quantum computer is, in fact, a computer. So we'd like a quantum computer to be able to do all the things a regular computer can do, like arithmetic. A quantum computer is not better at arithmetic than a regular computer. But I want to analyze a quantum adder to get used to the idea that measurements of multiple qubits represent ordinary numbers: 0, 1, 2, 3, etc. This will help us in the coming chapters when we study the problems that a quantum computer *can* solve more efficiently than a regular computer can.

First, we need to understand how to represent any whole number using 0's and 1's. This is called the binary number system. If we have a single bit, we can represent only two numbers: 0 and 1. If we have two bits, there are four possible values: 00, 01, 10, and 11. We can convert these numbers from binary (base two) to our ordinary number system (base ten).

To understand binary, we first have to understand base ten. In a base-ten number like 365 (the number of days a year that quantum computing is awesome), the 3 is in the hundred's place, the 6 is in the ten's place, and the 5 is in the one's place: $365 = 3 \times 100 + 6 \times 10 + 5 \times 1$. Each digit is multiplied by a power of 10. In base two, each binary digit (bit) is multiplied by a power of 2. The bit on the far right is in the one's place, but the bit to the left of that is in the two's place: the binary number $10 = 1 \times 2 + 0 \times 1 = 2$, and the binary number $11 = 1 \times 2 + 1 \times 1 = 3$. Sometimes we use subscripts to specify the base: $10_2 = 2_{10}$ and $11_2 = 3_{10}$. Often, though, we omit the subscript and infer from context whether we're using binary or base ten.

If we have a three-bit number, like 100_2, the leftmost bit is in the four's place, so $100_2 = 4_{10}$. And if we have a four-bit number, like 1000_2, the leftmost bit is in the eight's place, so $1000_2 = 8_{10}$. We usually won't need more than four bits, so we might as well list the 16 possible four-bit numbers, in binary and base ten (Table 7.1).

Table 7.1

Binary number	Base-ten equivalent
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

What does this have to do with quantum computing? We've already seen that we can use a single ket to represent two qubits:

$$|0\rangle|0\rangle = |00\rangle$$
$$|0\rangle|1\rangle = |01\rangle$$
$$|1\rangle|0\rangle = |10\rangle$$
$$|1\rangle|1\rangle = |11\rangle$$

It's often convenient to convert the binary number to base ten inside the ket:

$$|0\rangle|0\rangle = |00\rangle = |0\rangle$$
$$|0\rangle|1\rangle = |01\rangle = |1\rangle$$
$$|1\rangle|0\rangle = |10\rangle = |2\rangle$$
$$|1\rangle|1\rangle = |11\rangle = |3\rangle$$

So the general state of two qubits can be written $a_0|0\rangle + a_1|1\rangle + a_2|2\rangle + a_3|3\rangle$. The total probability of obtaining one of the four results is 1, so $|a_0|^2 + |a_1|^2 + |a_2|^2 + |a_3|^2 = 1$. The result obtained can be interpreted as an ordinary number, either 0, 1, 2, or 3. In slightly different words: When we measure multiple qubits at the end of a quantum circuit, the result can be converted to a *single* number in base ten. Quantum computers perform computations; the result of the computations is a number.

Let's be explicit about how this works for three qubits as well:

$$
\begin{aligned}
|0\rangle|0\rangle|0\rangle &= |000\rangle = |0\rangle \\
|0\rangle|0\rangle|1\rangle &= |001\rangle = |1\rangle \\
|0\rangle|1\rangle|0\rangle &= |010\rangle = |2\rangle \\
|0\rangle|1\rangle|1\rangle &= |011\rangle = |3\rangle \\
|1\rangle|0\rangle|0\rangle &= |100\rangle = |4\rangle \\
|1\rangle|0\rangle|1\rangle &= |101\rangle = |5\rangle \\
|1\rangle|1\rangle|0\rangle &= |110\rangle = |6\rangle \\
|1\rangle|1\rangle|1\rangle &= |111\rangle = |7\rangle
\end{aligned}
$$

Prior to measurement, the state of three qubits may be written $a_0|0\rangle + a_1|1\rangle + a_2|2\rangle + a_3|3\rangle + a_4|4\rangle + a_5|5\rangle + a_6|6\rangle + a_7|7\rangle$. The measurement (in the computational basis) effectively collapses the state to a single whole number, ranging from 0 to 7.

Now, we're ready to add binary numbers. How would we add two two-bit numbers, like $10+11$? Well, we could convert to base ten and then add, $2+3=5$, and then convert back to binary: 101. But classical and quantum computers alike need to add one bit at a time. So let's see how that works.

Just as in base-ten addition, we put each number in its own row:

$$
\begin{array}{r}
10 \\
+11 \\
\hline
\end{array}
$$

We add the rightmost column first. $0+1$ is obviously 1:

$$
\begin{array}{r}
10 \\
+11 \\
\hline
1
\end{array}
$$

In the next column, we have $1+1$. In binary, this is a two-bit sum, 10. So we bring down the 0 and carry the 1:

$$
\begin{array}{rrr}
1 & & \\
 & 1 & 0 \\
+ & 1 & 1 \\
\hline
 & 0 & 1
\end{array}
$$

The 1 that we carried is alone in its column, so we bring it straight down, to obtain the final sum of 101:

$$
\begin{array}{rrr}
1 & & \\
 & 1 & 0 \\
+ & 1 & 1 \\
\hline
1 & 0 & 1
\end{array}
$$

That's the whole calculation, but we can be really explicit that there was no carry when we summed the rightmost column. In other words, the carry was 0, which we can write over the second column from the right:

$$\begin{array}{cccc} & 1 & 0 & \\ & & 1 & 0 \\ + & & 1 & 1 \\ \hline & 1 & 0 & 1 \end{array}$$

Now, let's repeat this, using algebraic symbols to represent the sum of any two two-bit numbers, A_1A_0 and B_1B_0. A_1, A_0, B_1, and B_0 are all single bits, 0 or 1. The sum may require three bits, $S_2S_1S_0$:

$$\begin{array}{cccc} & & A_1 & A_0 \\ + & & B_1 & B_0 \\ \hline & S_2 & S_1 & S_0 \end{array}$$

We also need to keep track of the carry bits:

$$\begin{array}{cccc} & C_2 & C_1 & \\ & & A_1 & A_0 \\ + & & B_1 & B_0 \\ \hline & S_2 & S_1 & S_0 \end{array}$$

We immediately see that $S_2 = C_2$ because nothing else is in the leftmost column. Let's make this explicit:

$$\begin{array}{cccc} & C_2 & C_1 & \\ & & A_1 & A_0 \\ + & & B_1 & B_0 \\ \hline & C_2 & S_1 & S_0 \end{array}$$

Our task, then, is to compute C_2, C_1, S_1, and S_0 in terms of A_1, A_0, B_1 and B_0. What a herculean task! What an impossible mission! What a perilous quest! Now that danger has reared its ugly head, have we bravely turned our tail and fled, like Monty Python's Brave Sir Robin?

Yes. We need to fortify ourselves with an easier problem before we return to our quest. Let's add two single bits, A_0 and B_0. The sum may be as large as a two-bit number, S_1S_0. We also need to keep track of the carry, which we'll call C_1:

$$\begin{array}{ccc} & C_1 & \\ & & A_0 \\ + & & B_0 \\ \hline & S_1 & S_0 \end{array}$$

But now, S_1 must be C_1:

$$\begin{array}{ccc} & C_1 & \\ & & A_0 \\ + & & B_0 \\ \hline & C_1 & S_0 \end{array}$$

Table 7.2

A_0	B_0	C_1	S_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

So we need to determine C_1 and S_0 in terms of A_0 and B_0 The circuit that achieves this is called a half adder. The "half" means that there's no carry on the rightmost column.

We can list all four possible combinations of values of A_0 and B_0, and determine the two-bit sum C_1S_0 in all four cases. Table 7.2 shows us that the sum is 00 if $A_0 = B_0 = 0$, the sum is 01 if exactly one of the two addends is 0, and the sum is 10 if $A_0 = B_0 = 1$. But really, we need separate equations for S_0 and C_1. We see that C_1 is simply the product of A_0 and B_0:

$$C_1 = A_0B_0. \tag{7.1}$$

(This product of two bits is called the AND operation in Boolean algebra: $C_1 = 1$ only when both A_0 **and** B_0 are 1.)

In Table 7.2, we see that S_0 is 1 only when A_0 is different from B_0. We recall from Chapter 1 that this is exactly the outcome of the exclusive OR operation, so

$$S_0 = A_0 \oplus B_0. \tag{7.2}$$

That completes our half adder. Now we need to make it quantum. Our input qubits are $|A_0\rangle$ and $|B_0\rangle$, and our circuit needs to generate $|C_1\rangle$ and $|S_0\rangle$. Let's start with $|S_0\rangle$. We recall Eq. (3.2e),

$$\text{CNOT}|\text{control}\rangle|\text{target}\rangle = |\text{control}\rangle|\text{control} \oplus \text{target}\rangle.$$

According to our convention for circuit diagrams, we write the bottom qubit on the left. This implies that $|\text{control}\rangle$ is below $|\text{target}\rangle$, when we write CNOT in the preceding equation. But it's equally possible to put the control above the target, which in fact is the arrangement we will want. So we will reverse the order of the qubits:

$$\text{CNOT}|\text{target}\rangle|\text{control}\rangle = |\text{control} \oplus \text{target}\rangle|\text{control}\rangle.$$

Now, if we make $|A_0\rangle$ the control and $|B_0\rangle$ the target,

$$\text{CNOT}|B_0\rangle|A_0\rangle = |B_0 \oplus A_0\rangle|A_0\rangle = |S_0\rangle|A_0\rangle, \tag{7.3}$$

using Eq. (7.2). So we need a single CNOT gate to compute $|S_0\rangle$, shown in Fig. 7.1.

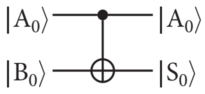


Figure 7.1. A circuit to compute a sum bit, created using the Quantikz LaTeX package.

So we've computed our sum bit. Now we work on the carry bit. The quantum version of Eq. (7.1) is

$$|C_1\rangle = |A_0 B_0\rangle. \tag{7.4}$$

To multiply two bits together, we need a new gate, called the Toffoli, or the controlled-controlled-NOT. This gate, which acts on three qubits, is shown in Fig. 7.2. The Toffoli gate applies a NOT to the target only if both controls are 1. Suppose that the initial state of the target qubit is $|0\rangle$, as in Fig. 7.3. The target will be $|1\rangle$ only when A_0 and B_0 are both 1, which means that the product A_0B_0 is 1. So the target becomes $|1\rangle$ when $A_0B_0 = 1$, and otherwise (when $A_0B_0 = 0$) the target remains $|0\rangle$. In either case, the final state of the target is $|A_0B_0\rangle = |C_1\rangle$, shown in Fig. 7.4. Now we want to combine the circuits that compute $|S_0\rangle$ and $|C_1\rangle$. Since the $|S_0\rangle$ circuit changes $|B_0\rangle$, we'd better compute $|C_1\rangle$ first (Fig. 7.5).

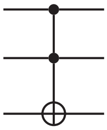


Figure 7.2. The Toffoli gate, created using the Quantikz LaTeX package.

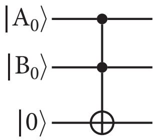


Figure 7.3. The Toffoli gate configured to compute a carry bit, created using the Quantikz LaTeX package.

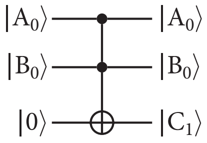


Figure 7.4. A circuit to compute a carry bit, created using the Quantikz LaTeX package.

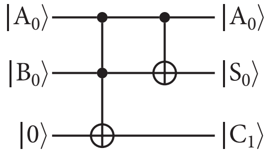


Figure 7.5. A half adder, created using the Quantikz LaTeX package.

Nice! We've now completed our side quest to add two one-bit numbers. Having loaded up on treasure and experience points, we're ready to return to our main quest, the addition of two two-bit numbers:

$$\begin{array}{rccc} & C_2 & C_1 & \\ & & A_1 & A_0 \\ + & & B_1 & B_0 \\ \hline & C_2 & S_1 & S_0 \end{array}$$

We know how to compute S_0 and C_1. But it will be more complicated to compute S_1 and C_2. These two bits each depend on three bits: C_1, A_1, and B_1. The circuit that performs this calculation is called a full adder.

We first need to construct a table to show how C_2 and S_1 depend on C_1, A_1, and B_1. We are summing three individual bits to obtain a two-bit result, C_2S_1. There are eight possible combinations of values of C_1, A_1, and B_1, and we need to consider all of them (Table 7.3).

Let's start with S_1. If we carefully study this column, we observe that S_1 is 1 only when C_2, A_1, and B_1 are all 1, or when exactly one of these is 1. This rule can be written

$$S_1 = C_1 \oplus A_1 \oplus B_1. \tag{7.5}$$

For example, suppose $C_1 = A_1 = B_1 = 1$. Then $S_1 = 1 \oplus 1 \oplus 1$. We recall that $\oplus$ is a difference detector: It compares two bits and outputs 1 when they are different,

Table 7.3

C_1	A_1	B_1	C_2	S_1
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

and 0 when they are the same. We can pick any two of the 1's to combine first: $1 \oplus 1 \oplus 1 = (1 \oplus 1) \oplus 1 = 0 \oplus 1 = 1$. To take one more example, suppose C_1 is 0, but $A_1 = B_1 = 1$. Then $S_1 = 0 \oplus 1 \oplus 1 = (0 \oplus 1) \oplus 1 = 1 \oplus 1 = 0$. We can go through all eight rows of the table and confirm that Eq. (7.5) works every time.

Next, let's look at C_2. We see that it is 1 in four cases:

- A_1 and B_1 and both 1, so $A_1B_1 = 1$
- C_1 and A_1 and both 1, so $C_1A_1 = 1$
- C_1 and B_1 and both 1, so $C_1B_1 = 1$
- C_1, A_1, and B_1 are all 1, so $C_1A_1B_1 = 1$

Let's see how all four cases are satisfied by this equation:

$$C_2 = A_1B_1 \oplus C_1A_1 \oplus C_1B_1. \tag{7.6}$$

If $A_1 = B_1 = 1$ but $C_1 = 0$, then $C_2 = 1 \oplus 0 \oplus 0 = (1 \oplus 0) \oplus 0 = 1 \oplus 0 = 1$. If $C_1 = A_1 = 1$ but $B_1 = 0$, then $C_2 = 0 \oplus 1 \oplus 0 = 1$. Similarly, $C_2 = 1$ if $C_1 = B_1 = 1$ but $A_1 = 0$. Last, if $C_1 = A_1 = B_1 = 1$, then $C_2 = 1 \oplus 1 \oplus 1 = 1$.

Next, we make Eqs. (7.5) and (7.6) quantum:

$$|S_1\rangle = |C_1 \oplus A_1 \oplus B_1\rangle \tag{7.7}$$

and

$$|C_2\rangle = |A_1B_1 \oplus C_1A_1 \oplus C_1B_1\rangle. \tag{7.8}$$

It's helpful to remember that each expression in these kets, no matter how complicated, is a 0 or 1. So Eqs. (7.7) and (7.8) are each either $|0\rangle$ or $|1\rangle$, a computational basis state for a single qubit.

Let's work on building the circuit for Eq. (7.7). We know that a CNOT generates an exclusive OR (Fig. 7.6). To combine with B_1, we just need another CNOT (Fig. 7.7).

Next, we can work on $|C_2\rangle = |A_1B_1 \oplus C_1A_1 \oplus C_1B_1\rangle$. We can start with A_1B_1, which we'll put in an extra qubit that starts as $|0\rangle$, shown in Fig. 7.8. To include the next term, C_1A_1, can we simply add another Toffoli (Fig. 7.9)? Yes, this is correct. The second Toffoli applies a NOT to the target if $C_1A_1 = 1$. We know that one way to apply NOT to a bit is to exclusive-OR it with 1: $\bar{X} = X \oplus 1$. The bottom qubit after the first Toffoli is $|A_1B_1\rangle$, and it has a NOT applied to it if $C_1A_1 = 1$. This transforms the bottom qubit into $|A_1B_1 \oplus C_1A_1\rangle$. A third Toffoli turns the bottom qubit into $|C_2\rangle = |A_1B_1 \oplus C_1A_1 \oplus C_1B_1\rangle$, as in Fig. 7.10.

Next, we combine our circuits for $|S_1\rangle$ and $|C_2\rangle$, calculating C_2 first because the S_1 calculation overrides A_1 and B_1 (Fig. 7.11). At last, we combine this with our half-adder circuit, which computes S_0 and C_1. The half adder has to come first to compute C_1 for the full adder, in Fig. 7.12.

Let's test this circuit on IBM Quantum. In IBM Quantum, I can't label the circuit diagram exactly the way I do in Fig. 7.12, but I can get it close

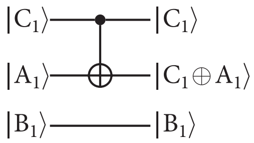


Figure 7.6. Building up to a circuit that computes the sum of three bits, created using the Quantikz LaTeX package.

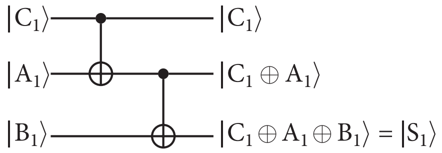


Figure 7.7. A circuit that computes the sum of three bits, created using the Quantikz LaTeX package.

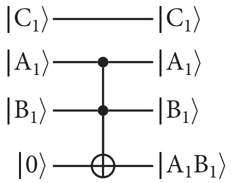


Figure 7.8. Building up to a circuit that computes the carry bit from a sum of three bits, created using the Quantikz LaTeX package.

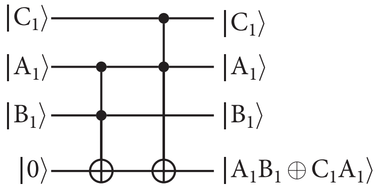


Figure 7.9. Further progress toward a circuit that computes the carry bit from a sum of three bits, created using the Quantikz LaTeX package.

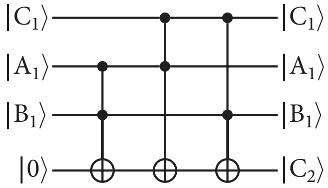


Figure 7.10. A circuit that computes the carry bit from a sum of three bits, created using the Quantikz LaTeX package.

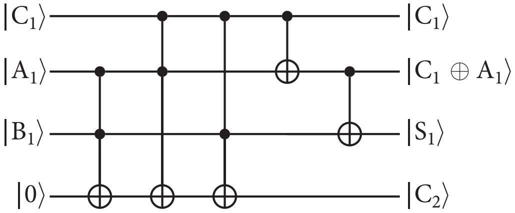


Figure 7.11. A full adder, created using the Quantikz LaTeX package.

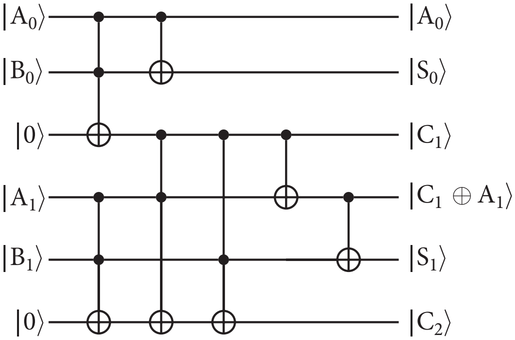


Figure 7.12. A circuit that sums two two-bit numbers, created using the Quantikz LaTeX package.

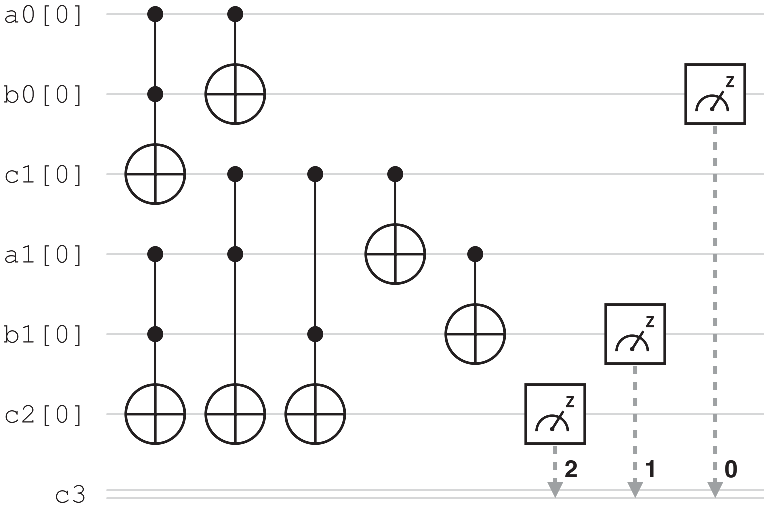


Figure 7.13. A circuit that calculates 0 + 0, created using IBM Quantum.

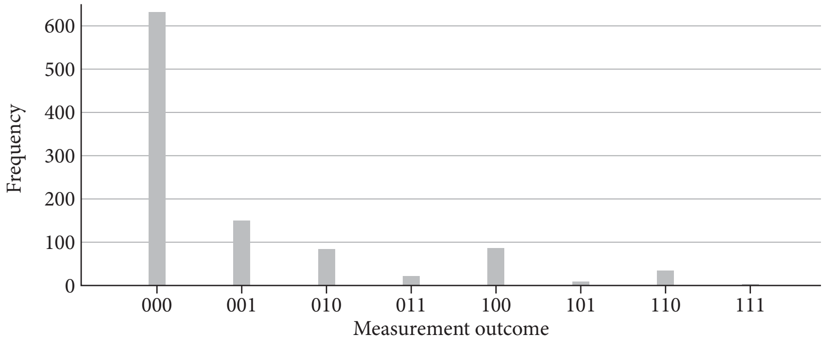


Figure 7.14. Results from the circuit in Fig. 7.13, created using IBM Quantum.

(Fig. 7.13). By default, all qubits in IBM Quantum start out in state $|0\rangle$. When the circuit computes 00 + 00, I expect the sum to be 000. On a simulator, 000 occurs 100% of the time. There are no superpositions in the circuit. However, on a real processor (ibm_oslo), error causes other results to occur some of the time (Fig. 7.14).

Next, how can we compute 1 + 1? This is really 01 + 01, so $A_1A_0 = B_1B_0 = 01$. This means that $A_0 = B_0 = 1$. We need NOT gates on $|A_0\rangle$ and $|B_0\rangle$ to create $|1\rangle$ from the default starting state of $|0\rangle$. $|A_0\rangle$ and $|B_0\rangle$ are the top two qubits, so we place the NOT gates there, before the quantum adder (Fig. 7.15). We

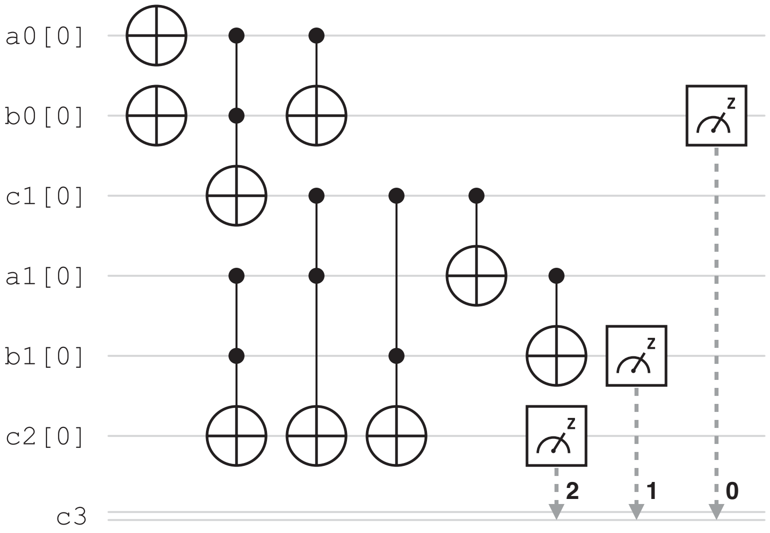


Figure 7.15. A circuit that calculates 1 + 1, created using IBM Quantum.

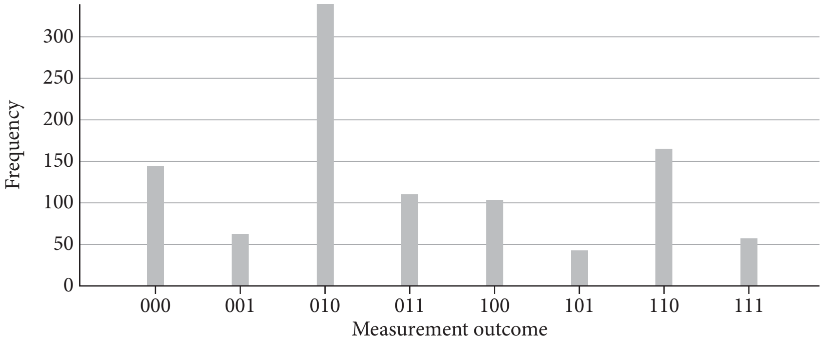


Figure 7.16. Results from the circuit in Fig. 7.15, created using IBM Quantum.

expect the sum to be $01+01=010$, and this is the most probable result on ibm_oslo (Fig. 7.16).

Next, let's compute $1+2=01+10$. Here, $A_0=1$, and $B_1=1$, so we place NOT gates on $|A_0\rangle$ and $|B_1\rangle$, as in Fig. 7.17. We expect the result $01+10=011$, and indeed this is the most likely outcome obtained using ibm_oslo (Fig. 7.18).

Last, let's try $3+3=11+11$. Here, A_1, A_0, B_1, and B_0 are all 1, so we need four NOT gates (Fig. 7.19). We expected the outcome $3+3=6=110$ in binary. On a simulator, we get this result 100% of the time. However, on ibm_oslo,

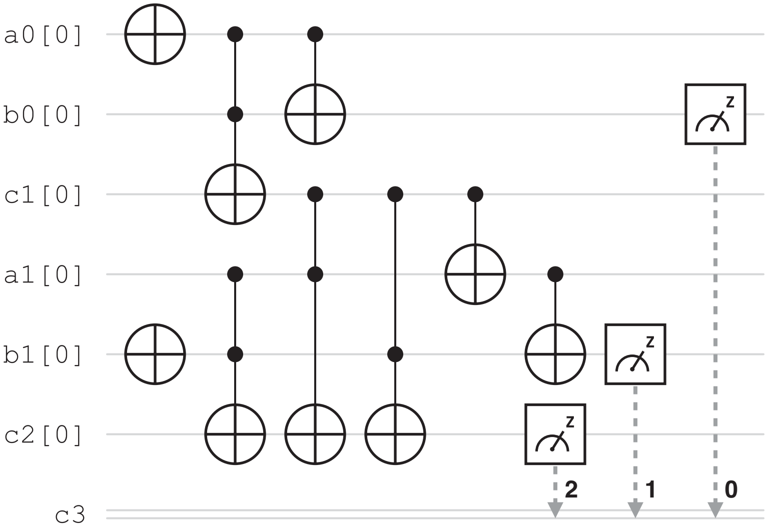


Figure 7.17. A circuit that calculates $1+2$, created using IBM Quantum.

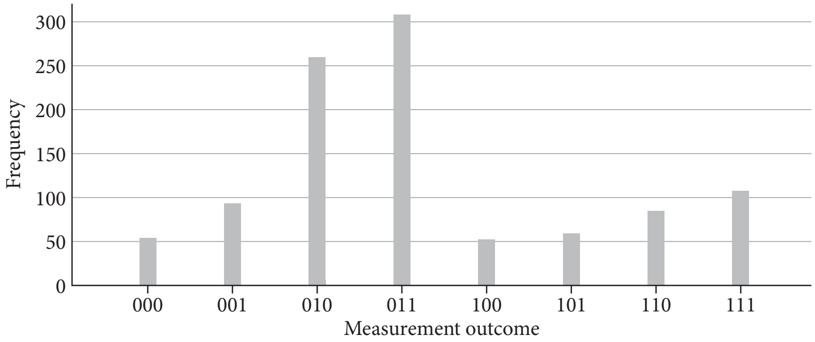


Figure 7.18. Results from the circuit in Fig. 7.17, created using IBM Quantum.

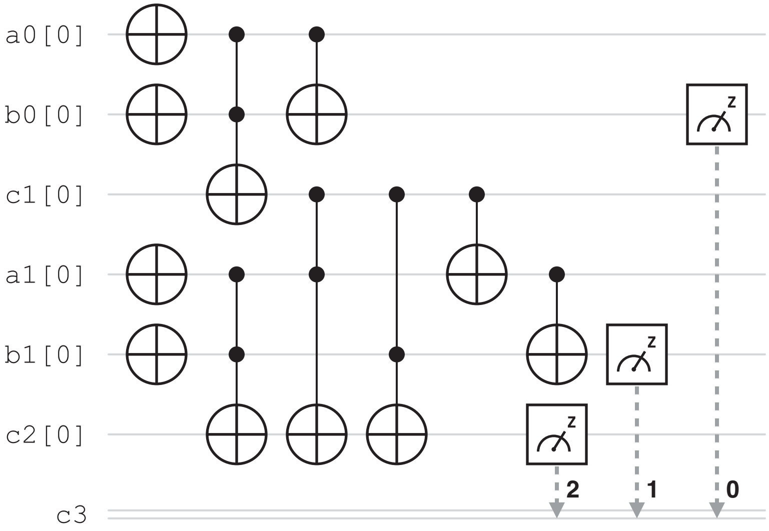


Figure 7.19. A circuit that calculates 3 + 3, created using IBM Quantum.

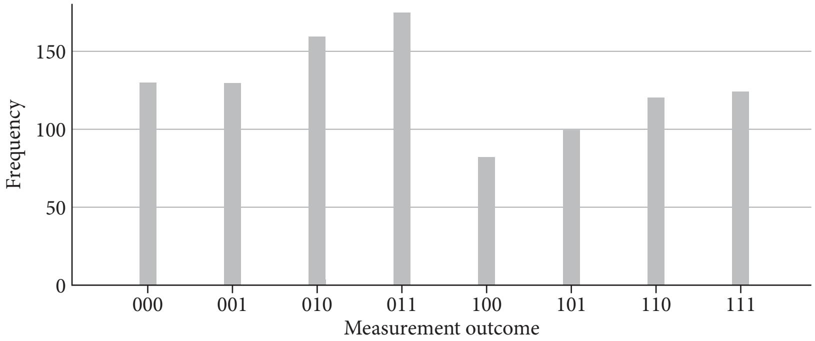


Figure 7.20. Results from the circuit in Fig. 7.19, created using IBM Quantum.

the most likely outcome is 011 (Fig. 7.20). The circuit fails on a real processor due to excessive error. As the number of gates increases, the error accumulates, and false outcomes are more likely. Quantum computing is still in its infancy, and we can't expect infants to do arithmetic correctly every time.

Chapter 8

Grover's Search

One Algorithm to Rule Them All, One Algorithm to Find Them

. . . one algorithm to bring them all and in the darkness bind them. It's a reference to *The Lord of the Rings*. If you didn't get that, I'm revoking your nerd badge. I can do that, you know. I'm a Level 30 Nerd with full privileges.

We arrive, at last, at a practical problem that quantum computers can solve more efficiently than classical computers. At least, in theory. The performance of today's quantum computers is restricted by their limited size and high error rate.

Suppose we have a phone book, in alphabetical order by name. Suppose we know a phone number, like 739-9201, and we want to know whose number it is. Maybe we're private investigators, and we discovered this phone number written in spicy ketchup on the shell of a live turtle swimming in a pond in a desert oasis. We're naturally curious to learn whose number it is. Of course, we could just call the number, but we don't want to disturb a stranger. We perform our investigations at a respectful distance.

So we could search through the phone book until we find the number. That would probably take a long time. Suppose there are a million names in the phone book. Whether we search through the names in alphabetical order, or reverse order, or randomly, the desired number is just as likely to turn up for our millionth name as our first. If we had to repeat this tedious task for many phone numbers, we would expect, on average, to locate each desired number after searching through half the phone book. So our search through a million names likely requires half a million attempts. Even if our phone book is digital, a classical computer might take a noticeable amount of time to perform this search.

On a quantum computer, however, only about a thousand attempts are required to search through a million items. That's an improvement by a factor of 500, compared with a classical computer. The quantum process is called Grover's algorithm. Let's see how it works.

Suppose the first entries in our phone book are as shown in Table 8.1. To configure our phone book for Grover's algorithm, we first need to assign a unique number to each name. The phone numbers aren't necessarily unique; Achilles and Patroclus, for example, might share the same phone number (to save money, those thrifty fellows). I'll call the unique number the *state number*, and it will start at 0 (Table 8.2). In our quantum computer, the state number is a computational basis state, $|0\rangle$, $|1\rangle$, $|2\rangle$, etc. Only one of these states is the Good state, the Great state, the Grail we seek, so let's call it $|G\rangle$.

We now define an oracle, U_f, that multiplies $|G\rangle$ by −1 and has no effect on any of the undesired states. So

$$U_f|G\rangle = -|G\rangle \tag{8.1a}$$

and

$$U_f|j \neq G\rangle = |j \neq G\rangle \tag{8.1b}$$

for all computational basis states $|j\rangle$ other than $|G\rangle$.

We also need an operator called *inversion about the mean*. I'm going to represent this operator with the symbol Inv. To understand this operator, let's consider two qubits. There are four basis states, $|00\rangle = |0\rangle$, $|01\rangle = |1\rangle$, $|10\rangle = |2\rangle$, and $|11\rangle = |3\rangle$. The most general state of two qubits can be written $a_0|0\rangle + a_1|1\rangle + a_2|2\rangle + a_3|3\rangle$. We can calculate the average, or mean, of the probability amplitudes. Let's call the mean m:

$$m = \frac{a_0 + a_1 + a_2 + a_3}{4}. \tag{8.2}$$

Table 8.1

Name	Phone number
Achilles	843-0094
Aeschylus	178-9428
Athena	739-9201
Bellerophon	102-2457

Table 8.2

State number	Name	Phone number
0	Achilles	843-0094
1	Aeschylus	178-9428
2	Athena	739-9201
3	Bellerophon	102-2457

To invert about the mean, we want to take the amplitudes that are greater than the mean, and make them lower than the mean, by the same amount that they were initially higher. So if the mean is 0.5, and one amplitude is 0.7, we want to reduce this to 0.3: Since it starts out 0.2 above the mean, it must end up 0.2 below the mean. This is a little like a communist theory of justice. The amplitudes that are initially below the mean are, in the end, above the mean. If the mean is 0.5, and an amplitude starts out at 0.2, it must become 0.8. The proletariat rises as the bourgeoisie falls. March on, comrades!

Let's come up with an equation for inversion about the mean (possibly for use by the Communist Bureau of Wealth Redistribution). If a_0 is initially above the mean, then $a_0 - m$ is a positive number: the amount by which a_0 is above the mean. We want the amplitude to end up below the mean by this amount, so we want the amplitude to become $m - (a_0 - m) = 2m - a_0$.

We get the same result if a_0 is below the mean. In this case, $m - a_0$ is a positive number, the amount by which a_0 is below the mean. We want the amplitude to end up this much higher than the mean: $m + (m - a_0) = 2m - a_0$. So after inversion about the mean, a_0 changes to $2m - a_0$, a_1 changes to $2m - a_1$, etc. Combining everything into one equation,

$$\begin{aligned}\mathrm{Inv}(a_0|0\rangle + a_1|1\rangle + a_2|2\rangle + a_3|3\rangle) = {} & (2m - a_0)|0\rangle + (2m - a_1)|1\rangle \\ & + (2m - a_2)|2\rangle + (2m - a_3)|3\rangle. \qquad (8.3)\end{aligned}$$

If we have two qubits, we have only four basis states. So we're searching for only one Good state out of four, which isn't very impressive. But Eqs. (8.2) and (8.3) generalize to more qubits in a straightforward way. If we have more qubits, we can search through a larger number of items, and the search takes longer. But the longer the search, the greater the improvement over a classical computer.

Grover's algorithm is simply this:

- Start each qubit in state $H|0\rangle = |+\rangle$. The collective state of these qubits will be called $|S\rangle$, for Sum, as explained later.
- Apply U_f then Inv, and repeat this an optimal number of times. (We'll calculate this later.)
- Measure the qubits. There's an excellent chance that the result is the desired state, $|G\rangle$.

Let's start with a simple example. Suppose we have two qubits, and the Good state $|G\rangle = |3\rangle$. The qubits start in state $|+\rangle|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{2}(|0\rangle|0\rangle + |0\rangle|1\rangle + |1\rangle|0\rangle + |1\rangle|1\rangle) = \frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle)$. If we measure the qubits right now, there's only a 25% chance of obtaining $|G\rangle$.

Next, we apply U_f. According to Eq. (8.1), only $|G\rangle$ is affected. So $|G\rangle$, which in this case is $|3\rangle$, is multiplied by -1. The state becomes $\frac{1}{2}(|0\rangle + |1\rangle + |2\rangle - |3\rangle)$.

Next, we apply Inv. We first have to calculate m, the mean of the four probability amplitudes. The first three amplitudes are 1/2, and the last one is −1/2, so the mean is $m=(1/2+1/2+1/2-1/2)/4=1/4$. After inversion about the mean, the amplitudes that were 1/4 above m will be reduced by 1/2, from 1/2 to 0, and the amplitude that was 3/4 below m will be raised by 3/2, from −1/2 to 1. Or using Eq. (8.3),

$$\mathrm{Inv}\left(\frac{1}{2}|0\rangle+\frac{1}{2}|1\rangle+\frac{1}{2}|2\rangle-\frac{1}{2}|3\rangle\right)=\left(2\times\frac{1}{4}-\frac{1}{2}\right)|0\rangle$$
$$+\left(2\times\frac{1}{4}-\frac{1}{2}\right)|1\rangle+\left(2\times\frac{1}{4}-\frac{1}{2}\right)|2\rangle+\left[2\times\frac{1}{4}-\left(-\frac{1}{2}\right)\right]|3\rangle=|3\rangle.$$

In this example, a single application of U_f and Inv transforms the initial state exactly into $|G\rangle$. This is considered a single query, or a single attempt to find the good state. A classical computer, on the other hand, has only a 25% chance of obtaining the correct result on its first guess.

If we have more qubits, we generally have to apply U_f and Inv more than once. To calculate the optimal number of iterations, we have to do a few pages of math. But it's worth it because we need this math (or at least the results of this math) to make Grover's algorithm work.

Let's first combine all the states that aren't $|G\rangle$. We'll call this combination $|D\rangle$ for the Disappointing states, the Dismal states, the Dreary states, the Decoy states. So in our example, we form the combination $|0\rangle+|1\rangle+|2\rangle$. But this isn't normalized; we want measurements of state $|D\rangle$ to yield each of the three Disappointing states 1/3 of the time. So $|D\rangle=\frac{1}{\sqrt{3}}(|0\rangle+|1\rangle+|2\rangle)$.

Notice that the qubits were initialized (by H gates) to $\frac{1}{2}(|0\rangle+|1\rangle+|2\rangle+|3\rangle)$, an equally weighted superposition of all basis states. We call this $|S\rangle$ for Superposition or Sum. If we have three qubits, the initial state is $|S\rangle=|+\rangle|+\rangle|+\rangle$ $=\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)=\frac{1}{2\sqrt{2}}(|0\rangle+|1\rangle+|2\rangle+|3\rangle+|4\rangle+|5\rangle$ $+|6\rangle+|7\rangle)$. Again, the initial state is an equally weighted superposition of all basis states. If we have n qubits, then putting them each in state $|+\rangle$ creates an equally weighted superposition of all 2^n basis states. This number of basis states is often called $N=2^n$.

Now we want to write $|S\rangle$, the superposition of all basis states, in terms of $|G\rangle$, the Good state, and $|D\rangle$, the normalized superposition of Disappointing states. In our two-qubit example, since

$$|S\rangle=\frac{1}{2}(|0\rangle+|1\rangle+|2\rangle+|3\rangle),$$

$$|G\rangle=|3\rangle,$$

and

$$|D\rangle = \frac{1}{\sqrt{3}}\left(|0\rangle + |1\rangle + |2\rangle\right),$$

we find that

$$|S\rangle = \frac{\sqrt{3}}{2}|D\rangle + \frac{1}{2}|G\rangle. \tag{8.4}$$

In this example, the number of basis states is N=4. Eq. (8.4) is actually a special case of

$$|S\rangle = \sqrt{\frac{N-1}{N}}\,|D\rangle + \frac{1}{\sqrt{N}}\,|G\rangle, \tag{8.5}$$

where, in general,

$$|D\rangle = \frac{1}{\sqrt{N-1}}\left(|0\rangle + |1\rangle + \ldots\right). \tag{8.6}$$

Equation (8.6) includes all the Disappointing states. If $|0\rangle$ or $|1\rangle$ is the Good state, it's omitted from the sum.

In Eq. (8.5), notice that the square of the amplitude of $|D\rangle$, plus the square of the amplitude of $|G\rangle$, is 1. So $|D\rangle$ and $|G\rangle$ form a kind of a basis: If we do a measurement that detects either $|D\rangle$ or $|G\rangle$, we obtain one or the other, with a total probability of 1.

When the sum of two squares is 1, we may be reminded of a trigonometric identity,

$$\cos^2\theta + \sin^2\theta = 1. \tag{8.7}$$

Comparing Eqs. (8.5) and (8.7), we might as well define θ so that

$$\cos\theta = \sqrt{\frac{N-1}{N}} \tag{8.8a}$$

and

$$\sin\theta = \frac{1}{\sqrt{N}}. \tag{8.8b}$$

Why are we bringing in trigonometry? We're actually trying to make the problem easier, not harder. By defining the angle θ, we'll be able to use geometry to simplify our algebra. Here's how.

We will imagine that $|D\rangle$, $|G\rangle$, and $|S\rangle$ are all arrows with a length of 1. Since $|D\rangle$ and $|G\rangle$ are mutually exclusive ($|G\rangle$ does not appear as any term in $|D\rangle$), we'll make $|D\rangle$ and $|G\rangle$ perpendicular; in a sense, no matter how far we travel in the $|D\rangle$ direction, we'll never move an inch in the $|G\rangle$ direction. Let's make $|D\rangle$ horizontal, and $|G\rangle$ vertical, shown in Fig. 8.1.

In Eq. (8.5), we see that $|S\rangle$ has both a $|D\rangle$ part and a $|G\rangle$ part, and the $|D\rangle$ part is larger. This is represented graphically in Fig. 8.2. We see a right

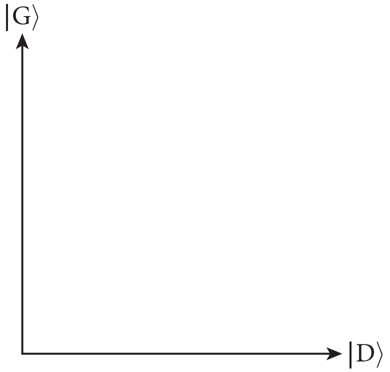


Figure 8.1. A geometric representation of the Good state and the superposition of Disappointing states.

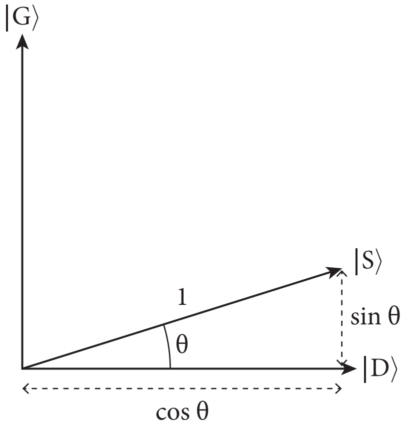


Figure 8.2. The angle θ, defined as the angle between $|S\rangle$ and $|D\rangle$.

triangle with a hypotenuse of length 1. Let's see how the geometry of this triangle is consistent with Eqs. (8.5) and (8.8). Suppose we want to walk from the origin, where the tails of the arrows meet, to the tip of $|S\rangle$. But suppose we can't walk along the hypotenuse; we can only walk horizontally (in the $|D\rangle$ direction) or vertically (in the $|G\rangle$ direction). How far do we have to walk in each direction? We can first walk in the $|D\rangle$ direction, a distance equal to the base of the triangle. This is cosθ. But let's keep track of the direction we've walked, which is $|D\rangle$. We think of $|D\rangle$ as a direction, like East, not a number. Putting together the distance, cosθ, with the direction, $|D\rangle$, we'll say we've walked $\cos\theta|D\rangle$. This is $\sqrt{\frac{N-1}{N}}|D\rangle$, according to Eq. (8.8a). Next, to reach the tip of the $|S\rangle$ arrow, we have to walk in the $|G\rangle$ direction, a distance of sinθ. So we add to our walk sinθ$|G\rangle$, which is $\frac{1}{\sqrt{N}}|G\rangle$, from Eq. (8.8b). So our total walk, from the origin to the tip of $|S\rangle$, is $\sqrt{\frac{N-1}{N}}|D\rangle + \frac{1}{\sqrt{N}}|G\rangle$. This is the geometric interpretation of Eq. (8.5).

What's the point of this geometry? In Grover's algorithm, the initial state is $|S\rangle$. The next state is $U_f|S\rangle$. We want to see where $U_f|S\rangle$ appears geometrically. According to Eq. (8.1), U_f has no effect on $|D\rangle$, but it multiplies $|G\rangle$ by –1. So,

$$U_f|S\rangle = \sqrt{\frac{N-1}{N}}U_f|D\rangle + \frac{1}{\sqrt{N}}U_f|G\rangle = \sqrt{\frac{N-1}{N}}|D\rangle - \frac{1}{\sqrt{N}}|G\rangle. \qquad (8.9)$$

Geometrically, the effect of U_f is to replace our displacement in the $+|G\rangle$ direction with an equal displacement in the $-|G\rangle$ direction (Fig. 8.3).

We see that U_f is effectively a reflection about the $|D\rangle$ line; we obtain the mirror image of what we started with. So far, it appears that U_f is doing more harm than good: We're less aligned with $|G\rangle$ than we were initially. (Our goal to is align with $|G\rangle$. If we align perfectly with $|G\rangle$, our state is simply $|G\rangle$, and we have a 100% chance of measuring the Good state. In Grover's algorithm, typically it's impossible to align perfectly, so we just want to align with $|G\rangle$ as well as we can. Though actually, it would be just as good to align with $-|G\rangle$ because $-|G\rangle$ and $|G\rangle$ are physically indistinguishable.)

After we apply U_f, the next step in Grover's algorithm is to apply Inv, the inversion about the mean. I claim that inversion about the mean is a reflection

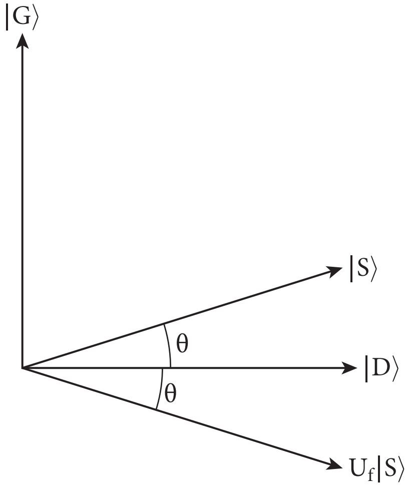


Figure 8.3. U_f acting as a reflection about $|D\rangle$.

about the $|S\rangle$ line. One way to show this is to examine the effect of Inv on $|D\rangle$. Recall that $|D\rangle = \frac{1}{\sqrt{N-1}}(|0\rangle + |1\rangle + \ldots) + 0|G\rangle$. All but one of the N basis states has an amplitude of $\frac{1}{\sqrt{N-1}}$. One of the basis states, $|G\rangle$, has an amplitude of 0. To calculate the mean amplitude, we sum the $N-1$ amplitudes of $\frac{1}{\sqrt{N-1}}$, and divide by N, the total number of basis states. So

$$m = \frac{(N-1)\frac{1}{\sqrt{N-1}}}{N} = \frac{\sqrt{N-1}}{N},$$

using the rule that $\frac{x}{\sqrt{x}} = \frac{x}{\sqrt{x}}\frac{\sqrt{x}}{\sqrt{x}} = \sqrt{x}$.

Next, a generalization of Eq. (8.3), to more than two qubits, is that

$$\begin{aligned} \mathrm{Inv}(a_0|0\rangle + a_1|1\rangle + a_2|2\rangle + a_3|3\rangle + \ldots) &= (2m - a_0)|0\rangle + (2m - a_1)|1\rangle \\ &\quad + (2m - a_2)|2\rangle + (2m - a_3)|3\rangle + \ldots \end{aligned}$$

When Inv acts on $|D\rangle$, we use the fact that 0 is the initial amplitude of $|G\rangle$, so the amplitude of $|G\rangle$ becomes $2m - 0 = 2m$. On the other hand, $\frac{1}{\sqrt{N-1}}$ is the initial amplitude of every Disappointing state, so the amplitude of every Disappointing state becomes $2m - \frac{1}{\sqrt{N-1}}$:

$$\begin{aligned} \mathrm{Inv}|D\rangle &= \left(2m - \frac{1}{\sqrt{N-1}}\right)(|0\rangle + |1\rangle + \ldots) + 2m\,|G\rangle \\ &= \left(\frac{2\sqrt{N-1}}{N} - \frac{1}{\sqrt{N-1}}\right)(|0\rangle + |1\rangle + \ldots) + \frac{2\sqrt{N-1}}{N}\,|G\rangle, \end{aligned}$$

substituting $m = \frac{\sqrt{N-1}}{N}$.

Let's write the first part, $\left(\frac{2\sqrt{N-1}}{N} - \frac{1}{\sqrt{N-1}}\right)(|0\rangle + |1\rangle + \ldots)$, in terms of $|D\rangle = \frac{1}{\sqrt{N-1}}(|0\rangle + |1\rangle + \ldots)$. The amplitude in the first expression is $\frac{2\sqrt{N-1}}{N} - \frac{1}{\sqrt{N-1}}$, and the amplitude in the second expression is $\frac{1}{\sqrt{N-1}}$, so the first expression is $\frac{\left(\frac{2\sqrt{N-1}}{N} - \frac{1}{\sqrt{N-1}}\right)}{\frac{1}{\sqrt{N-1}}} = \left(\frac{2(N-1)}{N} - 1\right) = \frac{N-2}{N}$

times the second expression: $\left(\frac{2\sqrt{N-1}}{N}-\frac{1}{\sqrt{N-1}}\right)(|0\rangle+|1\rangle+\ldots)=\left(\frac{N-2}{N}\right)|D\rangle$.

Substituting this into the expression for $\text{Inv}|D\rangle$,

$$\begin{aligned}\text{Inv}|D\rangle &= \left(\frac{2\sqrt{N-1}}{N}-\frac{1}{\sqrt{N-1}}\right)(|0\rangle+|1\rangle+\ldots)+\frac{2\sqrt{N-1}}{N}|G\rangle \\ &= \left(\frac{N-2}{N}\right)|D\rangle+\frac{2\sqrt{N-1}}{N}|G\rangle. \qquad (8.10)\end{aligned}$$

Let's confirm that Eq. (8.10) is normalized: The total probability of obtaining either $|D\rangle$ or $|G\rangle$ should be 1. The probability of obtaining $|D\rangle$ is $\frac{N^2-4N+4}{N^2}$. The probability of obtaining $|G\rangle$ is $\frac{4(N-1)}{N^2}$. If we add these together, we indeed get 1.

I claim that Eq. (8.10) can be written $\cos(2\theta)|D\rangle+\sin(2\theta)|G\rangle$, with θ defined via Eq. (8.8). Combining $\cos\theta=\sqrt{\frac{N-1}{N}}$ and $\sin\theta=\frac{1}{\sqrt{N}}$ with the rule $\cos(2\theta)=\cos^2\theta-\sin^2\theta$, $\cos(2\theta)=\frac{N-1}{N}-\frac{1}{N}=\frac{N-2}{N}$, precisely the amplitude of $|D\rangle$ in Eq. (8.10). Likewise, using the rule $\sin(2\theta)=2\sin\theta\cos\theta$, $\sin(2\theta)=2\frac{1}{\sqrt{N}}\sqrt{\frac{N-1}{N}}=\frac{2\sqrt{N-1}}{N}$, the amplitude of $|G\rangle$ in Eq. (8.10). So $\text{Inv}|D\rangle=\cos(2\theta)|D\rangle+\sin(2\theta)|G\rangle$ forms an angle of 2θ with $|D\rangle$ in Fig. 8.4, just as $|S\rangle=\cos\theta|D\rangle+\sin\theta|D\rangle$ forms an angle of θ with $|D\rangle$.

This means, as I claimed, that the effect of Inv on $|D\rangle$ is to reflect about the $|S\rangle$ line: $|S\rangle$ is like the mirror between the initial state ($|D\rangle$) and the final

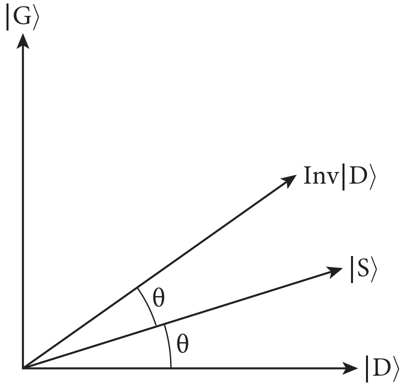


Figure 8.4. Inv acting as a reflection about $|S\rangle$.

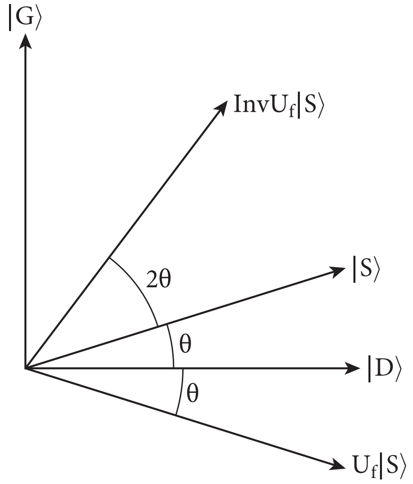


Figure 8.5. The effect of $\mathrm{InvU_f}$, a counterclockwise rotation of 2θ.

state ($\mathrm{Inv}|D\rangle$). In fact, Inv reflects any state about $|S\rangle$. What we really want to know, in Grover's algorithm, is the orientation of $\mathrm{Inv}(U_f|S\rangle)$. Since the angle between $U_f|S\rangle$ and $|S\rangle$ is 2θ, $\mathrm{Inv}(U_f|S\rangle)$ will be reflected about $|S\rangle$, forming an angle of 2θ in the opposite direction (Fig. 8.5).

So the overall effect of applying U_f and then Inv is to rotate 2θ counterclockwise. The goal is to rotate by this 2θ, the optimal number of times, to end up with an arrow as closely aligned with $|G\rangle$ as possible. The smaller θ is, the greater the number of iterations required. We want to iterate the optimal number of times, without going too far: If we rotate too many times, we pass $|G\rangle$, our goal, and get increasingly far from it.

The angle between $|D\rangle$ and $|S\rangle$ is θ. The angle between $|D\rangle$ and $\mathrm{InvU_f}|S\rangle$ is 3θ. Suppose we apply U_f then Inv a second time. Let's call the result $(\mathrm{InvU_f})^2|S\rangle$. The corresponding arrow is rotated 2θ farther away from $|D\rangle$, for a total angle of 5θ. So we see that if we apply both U_f and Inv a total of t times, the angle between $|D\rangle$ and $(\mathrm{InvU_f})^t|S\rangle$ is $(2t+1)\theta$. We want this angle to be as close to 90° as possible, so $(2t+1)\theta = 90°$, or

$$t = \frac{90^\circ}{2\theta} - \frac{1}{2} = \frac{90^\circ}{2\sin^{-1}\frac{1}{\sqrt{N}}} - \frac{1}{2}, \tag{8.11}$$

using Eq. (8.8b) to solve for θ. Knowing that $N = 2^n$, where n is the number of qubits in the circuits, we can just use a calculator or calculator app to solve for t in Eq. (8.11). (Replace 90° with $\pi/2$ radians if your calculator or calculator app is set to radians.) Most likely, t will not be an integer. This means that it's impossible to create a state that aligns perfectly with $|G\rangle$, but we round Eq. (8.11) to the nearest integer to obtain the optimal number of iterations of U_f then Inv.

For example, when we have two qubits, $N=2^2=4$, and Eq. (8.11) gives $t=[90°/(2\times 30°)-1/2]=1$. This means that after a single iteration, we create exactly the state $|G\rangle$ and have a 100% chance of measuring $|G\rangle$. We saw this earlier.

However, when we have three qubits, $N=2^3=8$, and Eq. (8.11) gives $t=1.67$, which we round to 2. So we need to apply U_f, then Inv, and then repeat the sequence a second time: U_f then Inv.

Let's see how this works explicitly. Suppose we have three qubits, so $N=2^3=8$, and

$$|S\rangle=\frac{1}{\sqrt{8}}\left(|0\rangle+|1\rangle+|2\rangle+|3\rangle+|4\rangle+|5\rangle+|6\rangle+|7\rangle\right).$$

The amplitude of each of eight basis states must be $\frac{1}{\sqrt{8}}$ so that the probability of measuring any one of them is 1/8 at this point. Suppose the Good state, which we're searching for, is $|G\rangle=|3\rangle$. Grover's algorithm will take the state from $|S\rangle$ to something close to $|G\rangle$.

Since the optimal number of iterations is $t=2$, we apply U_f then Inv, and then a second time apply U_f then Inv. The first application of U_f, to $|S\rangle$, affects only $|G\rangle$:

$$U_f|S\rangle=\frac{1}{\sqrt{8}}\left(|0\rangle+|1\rangle+|2\rangle-|3\rangle+|4\rangle+|5\rangle+|6\rangle+|7\rangle\right).$$

To apply inversion about the mean, we need to calculate the mean amplitude m. Seven of the eight amplitudes are $\frac{1}{\sqrt{8}}$, and the other one is $-\frac{1}{\sqrt{8}}$. So $m=\frac{7\times\frac{1}{\sqrt{8}}-\frac{1}{\sqrt{8}}}{8}=\frac{3}{4\sqrt{8}}$. Inv converts each amplitude a_j to $2m-a_j$, so

$$\begin{aligned}\text{Inv}U_f|S\rangle&=\left(2\times\frac{3}{4\sqrt{8}}-\frac{1}{\sqrt{8}}\right)\left(|0\rangle+|1\rangle+|2\rangle+|4\rangle+|5\rangle+|6\rangle+|7\rangle\right)\\&\quad+\left(2\times\frac{3}{4\sqrt{8}}+\frac{1}{\sqrt{8}}\right)|3\rangle\\&=\frac{1}{2\sqrt{8}}\left(|0\rangle+|1\rangle+|2\rangle+|4\rangle+|5\rangle+|6\rangle+|7\rangle\right)+\frac{5}{2\sqrt{8}}|3\rangle.\end{aligned}$$

This completes one iteration, but we're not done. The probability of obtaining the Good result so far is $\left(\frac{5}{2\sqrt{8}}\right)^2=\frac{25}{32}$. That's not bad, but it will improve after the second iteration.

Applying U_f the second time gives

$$U_f \mathrm{Inv} U_f|S\rangle = \frac{1}{2\sqrt{8}}(|0\rangle + |1\rangle + |2\rangle + |4\rangle + |5\rangle + |6\rangle + |7\rangle) - \frac{5}{2\sqrt{8}}|3\rangle.$$

Now the final step is the second application of Inv. To find the mean amplitude m, we see that seven of the amplitudes are $\frac{1}{2\sqrt{8}}$, and one is $-\frac{5}{2\sqrt{8}}$. So the mean is $m = \frac{7 \times \frac{1}{2\sqrt{8}} - \frac{5}{2\sqrt{8}}}{8} = \frac{1}{8\sqrt{8}}$. Then changing each amplitude a_j to $2m - a_j$:

$$\begin{aligned} \mathrm{Inv} U_f \mathrm{Inv} U_f|S\rangle &= \left(2 \times \frac{1}{8\sqrt{8}} - \frac{1}{2\sqrt{8}}\right)(|0\rangle + |1\rangle + |2\rangle + |4\rangle \\ &\quad + |5\rangle + |6\rangle + |7\rangle) + \left(2 \times \frac{1}{8\sqrt{8}} + \frac{5}{2\sqrt{8}}\right)|3\rangle \\ &= -\frac{1}{4\sqrt{8}}(|0\rangle + |1\rangle + |2\rangle + |4\rangle + |5\rangle + |6\rangle + |7\rangle) + \frac{11}{4\sqrt{8}}|3\rangle. \end{aligned}$$

And we're done! The probability of obtaining the Good result is 121/128, which is as high as it's going to get. The minus sign in front of the Disappointing states indicates that we've rotated a little too far, counterclockwise of the vertical $|G\rangle$ arrow. If we iterate any more, the probability of $|G\rangle$ will decline.

We're ready to tackle the next question, which is how to build the U_f and Inv gates. Let's start with Inv. Inv is inversion about the mean. Let's first try something a little simpler: inversion about $|111\rangle$. This means that $|111\rangle$ is unaffected, whereas every other computational basis state is multiplied by −1. So:

$$|111\rangle \to |111\rangle$$

and

$$|j\rangle \to -|j\rangle \text{ for j other than 111.}$$

We know that if we multiply every basis state by a global phase factor, like −1, nothing changes physically; the system is unaltered by this mathematical tweak. So, for later convenience, we'll multiply by −1 the definition of inversion about $|111\rangle$:

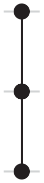
$$|111\rangle \to -|111\rangle$$

and

$$|j\rangle \to |j\rangle \text{ for j other than 111.}$$

This transformation is physically equivalent to our original definition of inversion about $|111\rangle$.

Now, it turns out that there's a quantum gate that multiplies $|111\rangle$ by –1 without affecting any other basis state: the doubly controlled Z gate:



Which two qubits are the controls, and which one is the target? In fact, we can imagine that any one of the three qubits is the target, to which we apply the Z gate. To make this flexibility explicit, we place an identical mark on each qubit.

Recall that Z has no effect on $|0\rangle$, but it multiplies $|1\rangle$ by –1. So in order to get that factor of –1, the target must be $|1\rangle$. But since there are two controls, the controls must be $|1\rangle$ too, or else the Z gate is not applied to the target. Putting this together, we see that all three qubits must be $|1\rangle$ to get the factor of –1. In other words, the doubly controlled Z gate multiplies $|111\rangle$ by –1, without affecting any other basis state. So the doubly controlled Z gate is inversion about $|111\rangle$.

If we want to multiply $|1111\rangle$ by –1, we use a triply controlled Z, which looks like four dots connected by a vertical line. In general, to multiply $|111\ .\ .\ .\ .\rangle$ by –1, we use a "multiply controlled Z" with a dot on every qubit. (The last syllable of "multiply" in "multiply controlled" is pronounced like the last syllable in "doubly" and "triply.")

Our next complication is that the doubly controlled Z is not provided by IBM Quantum, so we have to construct it out of the gates that are available. We can prove that Fig. 8.6 functions as a doubly controlled Z. We need to show that this multiplies $|111\rangle$ by –1, without affecting any other basis state. Clearly, the top two qubits must be $|1\rangle$ to apply the NOT to the bottom qubit. If the top two qubits are not both $|1\rangle$, the NOT is not applied, and the two H gates cancel each other out. But why must the bottom qubit be $|1\rangle$ to get the factor of –1?

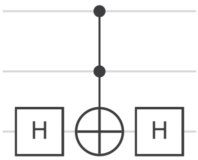


Figure 8.6. Construction of a doubly controlled Z, created using IBM Quantum.

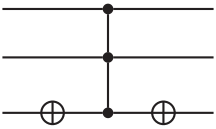


Figure 8.7. A circuit to multiply $|011\rangle$ by –1, created using the Quantikz LaTeX package.

Suppose the bottom qubit is $|0\rangle$. The first H changes it to $\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)=|+\rangle$. If the top qubits are $|11\rangle$ so that the NOT is applied, the bottom qubit changes to $\frac{1}{\sqrt{2}}\left(|1\rangle+|0\rangle\right)$, which actually isn't a change; it's still $|+\rangle$. So the final H changes $|+\rangle$ back to $|0\rangle$: The initial state is restored.

But what if all three qubits are $|1\rangle$? The first H changes the bottom qubit to $\frac{1}{\sqrt{2}}\left(|0\rangle-|1\rangle\right)=|-\rangle$. The NOT changes this to $\frac{1}{\sqrt{2}}\left(|1\rangle-|0\rangle\right)=-|-\rangle$, which is –1 times what it used to be. The final H changes the bottom qubit to $-|1\rangle$: We've acquired a factor of –1, only for the state $|111\rangle$.

Next, suppose we want to multiply some other state by –1. Suppose, for example, we want to multiply only $|011\rangle$ by –1. In fact, this transformation is exactly U_f for our example with $|G\rangle=|3\rangle=|011\rangle$. To multiply $|011\rangle$ by –1, we start with the doubly controlled Z, which multiplies $|111\rangle$ by –1. We simply put a NOT on the qubit with a $|0\rangle$, before and after the doubly controlled Z (Fig. 8.7).

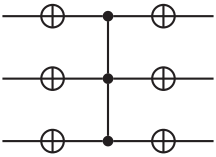


Figure 8.8. A circuit to multiply $|000\rangle$ by –1, created using the Quantikz LaTeX package.

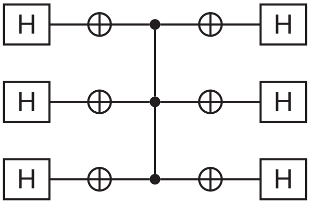


Figure 8.9. Inversion about the mean, created using the Quantikz LaTeX package.

If the state is |011⟩, the first NOT makes this |111⟩, then the doubly controlled Z makes this –|111⟩, and the then final NOT makes this –|011⟩. No other computational basis state is affected (the two NOT gates cancel each other out, without any factor of –1).

If we want to multiply |000⟩ by –1, we need NOTs on all three qubits (Fig. 8.8). This multiplies only |000⟩ by –1. Or, neglecting a global phase factor of –1, this multiplies everything *but* |000⟩ by –1: inversion about |000⟩.

We’ve had some peripheral adventures, but we’re ready to complete our quest, to construct Inv, inversion about the mean. We know that inversion about the mean is a reflection about |S⟩, or, let’s say, an inversion about |S⟩. And we know that we can create |S⟩ by applying H to every qubit, initially in state |0⟩. So an H applied to every qubit takes us from |000⟩, if we have three qubits, to |S⟩. Similarly, an H applied to every qubit takes us from |S⟩ back to |000⟩.

Let me be loose with the math here, but my conclusions can be proven rigorously. To invert about the mean, we start with inversion about |000⟩, which we know how to construct (Fig. 8.8). Then, we apply H to every qubit before and after inversion about |000⟩, shown in Fig. 8.9.

The first column of H’s converts |S⟩, which we want to invert about, to |000⟩. Effectively, we’re mapping |S⟩ onto |000⟩. Then we perform the inversion about |000⟩. Then the final column of H’s maps |000⟩ back to |S⟩. In effect, we’ve inverted about |S⟩, by mapping it in and out of a realm where it’s easier to perform inversions.

The inversion about the mean generalizes to any number of qubits, in a straightforward way. Simply add as many identical lines as necessary to the circuit.

We’ve already seen how to construct U_f: To multiply |G⟩ by –1, without affecting other basis states, start with a multiply controlled Z, and put NOT gates before and after it, on every qubit with |0⟩ in |G⟩. So we’re ready to construct our circuits to implement Grover’s algorithm.

“But wait!” the astute reader objects. “To construct U_f, we need to know what |G⟩ is, and the whole point of Grover’s algorithm is to determine |G⟩. So if we’re able to construct U_f, there’s no point implementing Grover’s search algorithm since we already know what we’re searching for.”

The astute reader is absolutely correct. As a warmup, we will design our circuits assuming that we already know what |G⟩ is. Then we will see how to

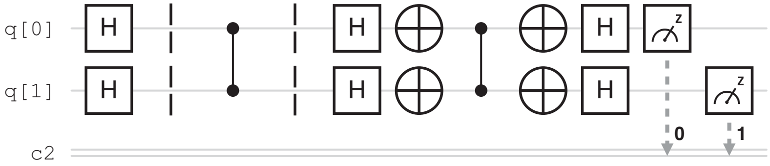


Figure 8.10. Grover’s algorithm for |G⟩ = |3⟩, created using IBM Quantum.

design a circuit that effectively searches a quantum database, without being preprogrammed with the state number we're searching for.

Figure 8.10 is Grover's search algorithm with $|G\rangle = |3\rangle = |11\rangle$. The initial H gates convert $|00\rangle$ to $|{+}{+}\rangle = |S\rangle = \frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle)$, the equally weighted superposition of all basis states for two qubits. The gate between the dashed lines is a controlled Z, which multiplies $|11\rangle = |3\rangle$ by −1 without affecting any other basis states. This is U_f for $|G\rangle = |3\rangle = |11\rangle$. The remaining gates are Inv, inversion about the mean. The measured result is expected to be exactly $|G\rangle$ in this case.

To search instead for $|G\rangle = |2\rangle = |10\rangle$, the only part of the circuit that changes is U_f. We want to multiply $|10\rangle$ by −1, so we need to sandwich the controlled Z between NOT gates on the top qubit, which corresponds with the 0 in $|10\rangle$, shown in Fig. 8.11.

So far, we've been searching for $|G\rangle$ in a space of only four possible values: 0, 1, 2, and 3. If we want to search a larger space, we need a larger circuit. Not only do we need more qubits, but we also need to repeat U_f and Inv according to Eq. (8.11). We've seen that if we have three qubits, we need to repeat U_f and Inv twice. So Fig. 8.12 is Grover's algorithm for $|G\rangle = |7\rangle = |111\rangle$.

The initial H gates create the state $|S\rangle = \frac{1}{\sqrt{8}}(|0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle + |5\rangle + |6\rangle + |7\rangle)$. If the qubits were measured at this point, there would be only a 1/8 chance of obtaining the desired result, $|7\rangle$. Next, there is a Toffoli with H gates on either side of the target. This functions as a doubly controlled Z, which is our U_f that multiplies $|111\rangle$ by −1. Next, we have inversion about the mean. This completes a single iteration within Grover's algorithm. As in the example earlier with $|G\rangle = |011\rangle = |3\rangle$, the probability of measuring $|G\rangle$ at this point is 25/32. We improve this by iterating a second time. We repeat U_f and

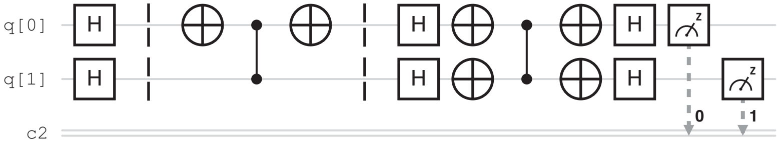


Figure 8.11. Grover's algorithm for $|G\rangle = |2\rangle$, created using IBM Quantum.

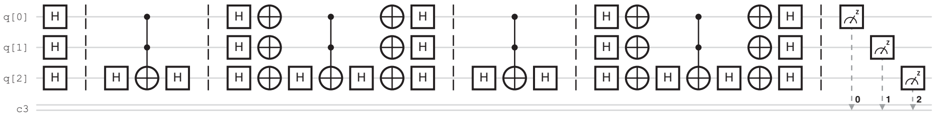


Figure 8.12. Grover's algorithm for $|G\rangle = |7\rangle$, created using IBM Quantum.

Inv. At the end of the circuit, the probability of measuring $|G\rangle$ is 121/128, over 94%.

In every circuit earlier, we constructed U_f using our knowledge of what $|G\rangle$ is. However, the whole purpose of Grover's algorithm is to search for an unknown $|G\rangle$. Let's see how to construct U_f if we don't already know $|G\rangle$.

Suppose we have a simple phone book, with only four names (Table 8.3). Following a short-lived local custom, everyone in this small community is named after a number. And since the community is so small, each phone number is only one digit. The people named 2 and 3 have the same phone number, so apparently they are sharing a phone, to save money. Very thrifty people, 2 and 3.

Now, if we want to know somebody's phone number, it's easy to look up the name, because the names are in numerical order. However, if we know a phone number, and we want to know the corresponding name, the task is harder. The phone numbers are not in any particular order in the phone book. If the phone book had a million entries, we'd have to search through half a million entries, on average, before finding the phone number we sought.

Suppose the desired phone number is 2. According to the phone book, the Good name (which is the state number) is then $|G\rangle=|0\rangle$. But we want to pretend that we don't already know this. So we need a U_f that looks at each name and multiplies the state by −1 only if the corresponding phone number is 2. U_f will incorporate a quantum database that stores all the information in the phone book.

To create the quantum database, we first need to convert the phone book to binary, and let's write the bits as qubits while we're at it (Table 8.4). We see that we need two qubits for the name, and two qubits for the phone number. A group of qubits is sometimes called a *register*, so we'll have a name register, and a phone number register, two qubits each:

name[0]
name[1]
phoneNumber[0]
phoneNumber[1]

If $|\text{name}\rangle=|01\rangle=|0\rangle|1\rangle$, then the qubit on the right (the $|1\rangle$ in this example) is called $|\text{name}[0]\rangle$ and is placed on the top of the circuit. The qubit on

Table 8.3

Name	Phone number
0	2
1	1
2	3
3	3

Table 8.4

$\vert \text{name}\rangle$	$\vert \text{phoneNumber}\rangle$
$\vert 0\rangle\vert 0\rangle$	$\vert 1\rangle\vert 0\rangle$
$\vert 0\rangle\vert 1\rangle$	$\vert 0\rangle\vert 1\rangle$
$\vert 1\rangle\vert 0\rangle$	$\vert 1\rangle\vert 1\rangle$
$\vert 1\rangle\vert 1\rangle$	$\vert 1\rangle\vert 1\rangle$

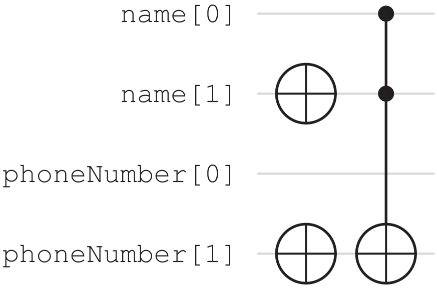


Figure 8.13. Setting the value of phoneNumber[1] for a quantum phone book, created using IBM Quantum.

the left is called $|\text{name}[1]\rangle$. So $|\text{name}\rangle = |\text{name}[1]\rangle|\text{name}[0]\rangle$. Similarly, $|\text{phoneNumber}\rangle = |\text{phoneNumber}[1]\rangle|\text{phoneNumber}[0]\rangle$.

According to the phone book, when $|\text{name}\rangle = |0\rangle|0\rangle$, $|\text{phoneNumber}\rangle = |1\rangle|0\rangle$, and so on. We need to create a quantum database to achieve this. Let's first look at phoneNumber[1]. This is the left qubit of the phone number. According to the phone book, the left qubit is $|1\rangle$ unless $|\text{name}\rangle$ is $|0\rangle|1\rangle$. Let's see how Fig. 8.13 makes $|\text{phoneNumber}[1]\rangle = |1\rangle$ except when $|\text{name}\rangle = |0\rangle|1\rangle$.

phoneNumber[0] and phoneNumber[1] start out in the state $|0\rangle$ by default. The NOT on $|\text{phoneNumber}[1]\rangle$ makes it $|1\rangle$. We need to turn it back to $|0\rangle$ when $|\text{name}\rangle = |0\rangle|1\rangle$. If $|\text{name}\rangle = |0\rangle|1\rangle$, the NOT gate on $|\text{name}[1]\rangle$ makes the name register $|1\rangle|1\rangle$. Then these two $|1\rangle$ qubits are the controls on the Toffoli, which activates the NOT on $|\text{phoneNumber}[1]\rangle$. We'll apply another NOT to $|\text{name}[1]\rangle$ to restore its original state in a moment.

Now we're ready to set the value of $|\text{phoneNumber}[0]\rangle$. According to the phone book, this is $|1\rangle$ except when $|\text{name}\rangle = |0\rangle|0\rangle$. Our complete database is achieved with Fig. 8.14.

The NOT on $|\text{phoneNumber}[0]\rangle$ makes it $|1\rangle$. We have to turn it back to $|0\rangle$ if $|\text{name}\rangle = |0\rangle|0\rangle$. This is achieved by the Toffoli targeting $|\text{phoneNumber}[0]\rangle$. Both qubits of $|\text{name}\rangle$ have had a NOT applied at this point, converting $|0\rangle|0\rangle$

to |1⟩|1⟩, which activates the Toffoli. The final two NOTs restore both qubits of |name⟩ to their original states. The effect of Fig. 8.14 on |phoneNumber⟩|name⟩ is to convert |phoneNumber⟩ from |0⟩|0⟩ to the phone number associated with |name⟩. (|phoneNumber⟩ starts out as |0⟩|0⟩, but |name⟩ can start as any value. In Grover's algorithm, the name register will start as a superposition of all basis states.)

Our circuit so far is a quantum database, the quantum equivalent of the phone book. It's completely independent of which phone number we may choose to search for. If we're searching for the phone number |2⟩ = |1⟩|0⟩, we need U_f to multiply |name⟩ by −1 when |phoneNumber⟩ = |1⟩|0⟩. A schematic diagram of the circuit we want is shown in Fig. 8.15. The open circle on |phoneNumber[0]⟩ indicates a control that requires a |0⟩ instead of a |1⟩. So the factor of −1 is applied to the target only when |phoneNumber[1]⟩ = |1⟩ and |phoneNumber[0]⟩ = |0⟩.

Now, how can we multiply an unspecified state by −1? For example, what gate, or sequence of gates, multiplies α|0⟩ + β|1⟩ by −1? Z multiplies |1⟩ by −1. To multiply |0⟩ by −1, we can first apply a NOT, to make it |1⟩, then apply Z to multiply by −1, and then apply another NOT to recover the original |0⟩. So the complete sequence to multiply any state by −1 is Z (to apply the −1 to |1⟩) and then (to apply the −1 to |0⟩) NOT then Z then NOT.

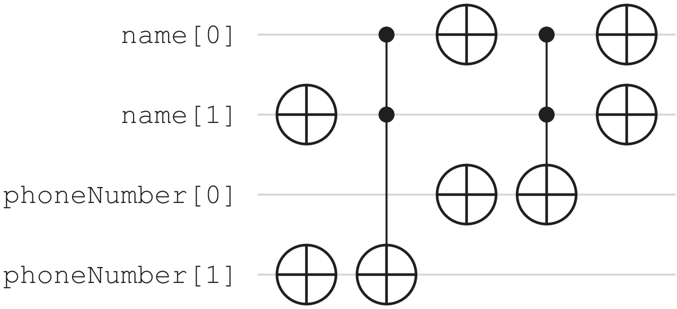


Figure 8.14. A quantum phone book, created using IBM Quantum.

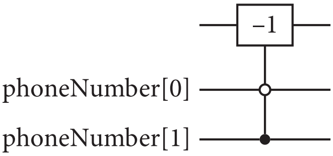


Figure 8.15. The oracle for a desired phone number of 10, created using the Quantikz LaTeX package.

Let's do this explicitly. Start with

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

Apply Z:

$$\alpha|0\rangle - \beta|1\rangle.$$

Apply X:

$$\alpha|1\rangle - \beta|0\rangle.$$

Apply Z:

$$-\alpha|1\rangle - \beta|0\rangle.$$

Apply X:

$$-\alpha|0\rangle - \beta|1\rangle,$$

which is the original state multiplied by –1. So the circuit we need is Fig. 8.16, where the top qubit can be either of the qubits in $|\text{name}\rangle$.

IBM Quantum doesn't allow for an open-circle control that requires $|0\rangle$, so we simply apply a NOT before and after the controls on that qubit (Fig. 8.17). The first NOT turns the required $|0\rangle$ to a $|1\rangle$, which can activate the controls. The final NOT restores the qubit to its original state.

We saw that a doubly controlled Z is created by a Toffoli and two Hadamards, so the achievable circuit to multiply $|\text{name}\rangle$ by –1 when $|\text{phoneNumber}\rangle = |1\rangle|0\rangle$ is the circuit in Fig. 8.18. I put the target on the top qubit of $|\text{name}\rangle$, but the lower qubit would work just as well.

We've worked out two elements of U_f: the quantum database creating $|\text{phoneNumber}\rangle$ states corresponding to $|\text{name}\rangle$ states, and the multiplication

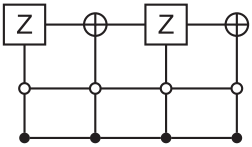


Figure 8.16. Construction of the desired oracle, created using the Quantikz LaTeX package.

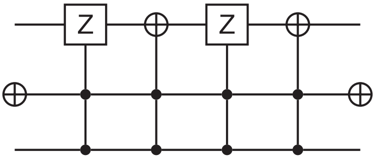


Figure 8.17. The same circuit as Fig. 8.16, without open-circle controls, created using the Quantikz LaTeX package.

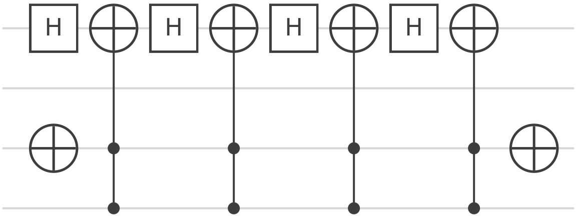


Figure 8.18. The same circuit as Fig. 8.17, constructing the doubly controlled Z gates as shown in Fig. 8.6, created using IBM Quantum.

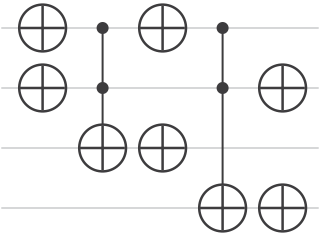


Figure 8.19. The inverse the of the phone book operation shown in Fig. 8.14, created using IBM Quantum.

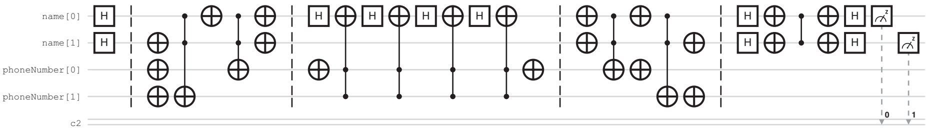


Figure 8.20. Grover's algorithm applied to the quantum phone book, created using IBM Quantum.

by –1 when |phoneNumber⟩ is |2⟩. The final element of U_f is to restore the |phoneNumber⟩ qubits to |0⟩|0⟩ to remove entanglement between the |name⟩ register and the |phoneNumber⟩ register. This is sometimes called uncomputing and is an essential step. The inversion about the mean will not work properly if the two registers are entangled. To restore the |phoneNumber⟩ register to |0⟩|0⟩, the gates that created the quantum database are applied in the opposite order (Fig. 8.19).

Now we're ready for the complete Grover's algorithm circuit to search the quantum phone book for the number |2⟩=|1⟩|0⟩, and return the |name⟩ corresponding to that phone number. The desired |name⟩, according to the phone book, is |0⟩|0⟩. That is the expected output at the end of the circuit. Figure 8.20 shows the complete circuit.

The initial H gates create $|S\rangle$, the superposition of all basis states in the $|\text{name}\rangle$ register. We're searching for a particular $|\text{name}\rangle$, to match the phone number $|10\rangle$. The phone number register is auxiliary and is not included in $|S\rangle$. After the first dashed line, we have the first part of U_f: the quantum database that assigns a $|\text{phoneNumber}\rangle$ state to go with each $|\text{name}\rangle$ state. Now the two registers are entangled: We have a superposition of four terms of $|\text{phoneNumber}\rangle|\text{name}\rangle$, one for each of the four $|\text{name}\rangle$ values. (The NOT on $|\text{phoneNumber}[0]\rangle$ may be placed on either side of the first Toffoli gate because this Toffoli gate does not interact with this qubit.)

After the second dashed line, we have the gates that multiply $|\text{name}\rangle$ by -1 only when $|\text{phoneNumber}\rangle$ is $|1\rangle|0\rangle$. This is the part of the circuit that we would change if we wanted to look up a different phone number.

After the third dashed line, we uncompute the $|\text{phoneNumber}\rangle$ register to disentangle it from the $|\text{name}>$ register. But the -1 multiplying the Good $|\text{name}\rangle$ remains. After the final dashed line, we have inversion about the mean. The final measurement result is theoretically expected to be $|0\rangle|0\rangle$, the $|\text{name}\rangle$ associated with the $|\text{phoneNumber}\rangle = |1\rangle|0\rangle$. We did not assume this result, the Good $|\text{name}\rangle$, anywhere in the circuit.

Let's make a final application of Grover's search algorithm. This algorithm can be applied to any problem that can be solved through trial and error. In many problems, it's easier to test if a solution is correct than it is to come up with the solution yourself. For example, it may be hard to solve the equation $(x+1)^{x-1} = x^2 + 7$. But if I asked you to test whether $x = 3$ was a solution, you'd be able to quickly confirm that it was. We're going to use Grover's algorithm to solve the equation $A + 1 = 3$.

Let's write this in binary: $A + 01 = 11$. Since the sum is two bits, it's clear that A is no more than two bits. We've already seen a circuit that sums two 2-bit numbers, A_1A_0 and B_1B_0 (Fig. 8.21).

Since we want to calculate $A + 01$, let's assign $A = A_1A_0$ and $B_1B_0 = 01$. Since the desired sum is $|S_1S_0\rangle = |11\rangle$, we don't need the final carry, $|C_2\rangle$. We can drop $|C_2\rangle$ and the gates that affect it (Fig. 8.22).

I want $|A_0\rangle$ and $|A_1\rangle$ to be next to each other, to form the $|A\rangle$ register. Grover's algorithm will search for the Good $|A\rangle$ state that solves the equation. I'll put $|B_0\rangle$ and $|B_1\rangle$ next to each other as well. After this rearrangement, the circuit becomes Fig. 8.23.

To complete U_f, we need to multiply $|A\rangle$ by -1 when the sum is $|11\rangle$. Multiplying any state by -1 requires Z, then NOT, then Z, then NOT. We'll put all these gates on $|A_0\rangle$, and control all of these gates with the sum bits (the controls have to both be $|1\rangle$ for a sum of $|11\rangle$). Then we have to uncompute the $|B\rangle$ and $|C\rangle$ registers, as well as $|A_1\rangle$. The complete circuit is shown in Fig. 8.24 (replacing capital A and B with lowercase letters as required by IBM Quantum).

The initial H gates create $|S\rangle$ for the $|A\rangle$ register. The NOT on $|B[0]\rangle$ initializes $|B\rangle$ to $|0\rangle|1\rangle$, the number that we're adding to $|A\rangle$. After the first

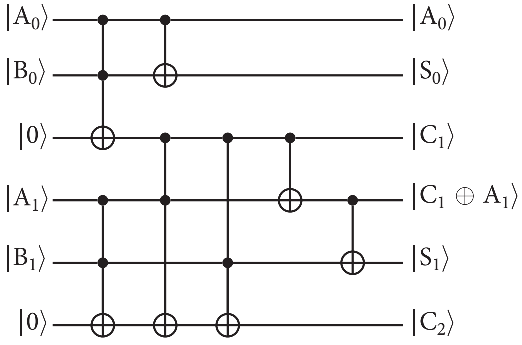


Figure 8.21. A circuit that sums two two-bit numbers, created using the Quantikz LaTeX package.

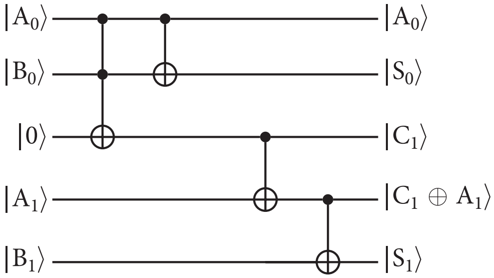


Figure 8.22. A circuit that sums two two-bit numbers without generating a final carry bit, created using the Quantikz LaTeX package.

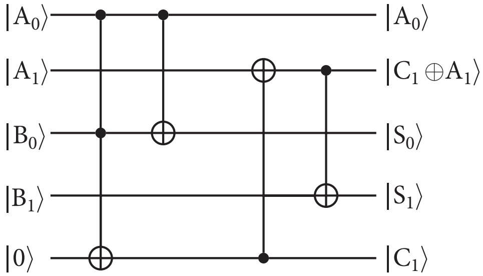


Figure 8.23. A rearrangement of Fig. 8.22, created using the Quantikz LaTeX package.

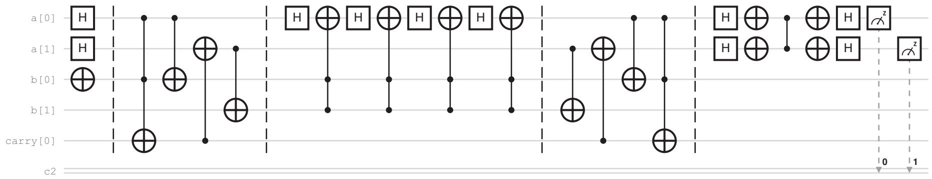


Figure 8.24. Grover's algorithm to solve $A+1=3$, created using IBM Quantum.

dashed line, we have the quantum adder. The sum is stored in the $|B\rangle$ register. After the second dashed line, we multiply $|A[0]\rangle$ by -1 when the sum is $|1\rangle|1\rangle$. Here we're marking the sum that we seek. After the third dashed line, we apply the inverse of the quantum adder to disentangle $|A\rangle$ from the auxiliary qubits ($|B\rangle$ and the carry qubit). Last, we have inversion about the mean.

The measured output is expected to be $A=10$, the solution to $A+01=11$. Nowhere in the circuit did we assume this solution. To instead solve $A+00=11$, or $A+10=11$, or $A+11=11$, we simply change the initialization of the $|B\rangle$ register, to the left of the first dashed line.

Chapter 9

QFT and IQFT … WTF?

In the next three chapters, we're ramping up to Shor's factoring algorithm, a potential threat to classical cryptography and network security. The farthest Shor is almost in sight. We have only to navigate the barrier islands. Watch out for sirens, sorceresses, and cyclopes.

Shor's algorithm makes use of something called the *quantum Fourier transform* (QFT). A Fourier transform provides information about the frequencies that make up a wave. For example, the Fourier transform of the sound wave of a chord indicates the frequencies of the notes in the chord. Frequency is the reciprocal of the period, the amount of time it takes for a full wavelength to pass by a fixed point. More generally, period can be defined as the size of a repeating pattern. For example, the sequence 10001000 has a period of 4 because the repeating pattern, 1000, has a size of 4 digits.

The quantum Fourier transform provides information about the period of amplitudes in a multiqubit state. For example, the two-qubit state $\frac{1}{\sqrt{2}}\left(|00\rangle+|10\rangle\right)=\frac{1}{\sqrt{2}}\left(|0\rangle+|2\rangle\right)=\frac{1}{\sqrt{2}}|0\rangle+0|1\rangle+\frac{1}{\sqrt{2}}|2\rangle+0|3\rangle$ has a period of 2: the periodic sequence of amplitudes is $\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}}, 0$, and the pattern that repeats $\left(\frac{1}{\sqrt{2}},\ 0\right)$ has a size of 2 terms.

The quantum Fourier transform involves some imaginary numbers. We recall that i, the square root of −1, behaves just like any other algebraic symbol. It has the special property $i^2=-1$.

We're going to see i in exponents, such as $e^{i\pi}$. What can this possibly mean? There's an equation called Euler's formula, which states that for some real number θ,

$$e^{i\theta}=\cos\theta+i\sin\theta. \tag{9.1a}$$

We're going to use this formula for just eight special values of θ, so I'll list them all. With θ in radians:

$$e^{i0} = \cos 0 + i\sin 0 = 1 \tag{9.1b}$$

$$e^{i\pi/4} = \cos(\pi/4) + i\sin(\pi/4) = \frac{1+i}{\sqrt{2}} \tag{9.1c}$$

$$e^{i\pi/2} = \cos(\pi/2) + i\sin(\pi/2) = i \tag{9.1d}$$

$$e^{3i\pi/4} = \cos(3\pi/4) + i\sin(3\pi/4) = \frac{-1+i}{\sqrt{2}} \tag{9.1e}$$

$$e^{i\pi} = \cos\pi + i\sin\pi = -1 \tag{9.1f}$$

$$e^{5i\pi/4} = \cos(5\pi/4) + i\sin(5\pi/4) = -\frac{1+i}{\sqrt{2}} \tag{9.1g}$$

$$e^{3i\pi/2} = \cos(3\pi/2) + i\sin(3\pi/2) = -i \tag{9.1h}$$

$$e^{7i\pi/4} = \cos(7\pi/4) + i\sin(7\pi/4) = \frac{1-i}{\sqrt{2}} \tag{9.1i}$$

You may recall that sine and cosine are periodic functions with a period of 2π, which means that $\sin(\theta+2\pi) = \sin\theta$ and $\cos(\theta+2\pi) = \cos\theta$. So it's also true that

$$e^{i(\theta+2\pi)} = e^{i\theta}. \tag{9.1j}$$

We've seen that if we have n qubits, we have $N = 2^n$ basis states. For example, if we have three qubits, we have eight basis states, $|000\rangle$ through $|111\rangle$, or $|0\rangle$ through $|7\rangle$ in base ten. The most general equation for the QFT of a basis state $|j\rangle$ is the definition

$$QFT|j\rangle = \frac{1}{\sqrt{N}}\left(e^{2\pi ij\frac{0}{N}}|0\rangle + e^{2\pi ij\frac{1}{N}}|1\rangle + \ldots + e^{2\pi ij\frac{N-1}{N}}|N-1\rangle\right) \tag{9.2a}$$

if N>2, or just

$$QFT|j\rangle = \frac{1}{\sqrt{2}}\left(e^{2\pi ij\frac{0}{2}}|0\rangle + e^{2\pi ij\frac{1}{2}}|1\rangle\right) \tag{9.2b}$$

if $N = 2$. Let's start with one qubit, and work up to three qubits.

If we have only $n = 1$ qubit, we have $N = 2^1 = 2$ basis states, $|0\rangle$ and $|1\rangle$. So the two possible values of j in Eq. (9.2b) are 0 and 1. Specifically, for $j = 0$,

$$QFT|0\rangle = \frac{1}{\sqrt{2}}\left(e^0|0\rangle + e^0|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \tag{9.3a}$$

because $e^0 = 1$. And for $j = 1$,

$$QFT|1\rangle = \frac{1}{\sqrt{2}}\left(e^{2\pi i\frac{0}{2}}|0\rangle + e^{2\pi i\frac{1}{2}}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(e^0|0\rangle + e^{\pi i}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) \tag{9.3b}$$

using Eq. (9.1f) to simplify $e^{\pi i}$ to -1.

Does Eq. (9.3) remind us of anything? $|0\rangle$ is turned into $\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$, and $|1\rangle$ is turned into $\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$. We know the gate the does this: the

Hadamard! So the single-qubit QFT is simply H. Apparently, we've been doing quantum Fourier transforms all along. The QFT of a single qubit, however, does not clearly show the utility of QFT, so let's move on to two qubits.

For $n=2$ qubits, we have $N=2^2=4$ basis states, $|00\rangle=|0\rangle$, $|01\rangle=|1\rangle$, $|10\rangle=|2\rangle$, and $|11\rangle=|3\rangle$. Equation (9.2a) has a sum of four terms,

$$\mathrm{QFT}|j\rangle=\frac{1}{2}\left(e^{2\pi ij\frac{0}{4}}|0\rangle+e^{2\pi ij\frac{1}{4}}|1\rangle+e^{2\pi ij\frac{2}{4}}|2\rangle+e^{2\pi ij\frac{3}{4}}|3\rangle\right).$$

Now we can explicitly write out the QFT of the four basis states, using Eq. (9.1) as necessary to simplify complex exponentials:

$$\mathrm{QFT}|0\rangle=\frac{1}{2}\left(e^0|0\rangle+e^0|1\rangle+e^0|2\rangle+e^0|3\rangle\right)=\frac{1}{2}\left(|0\rangle+|1\rangle+|2\rangle+|3\rangle\right) \quad (9.4a)$$

$$\begin{aligned}\mathrm{QFT}|1\rangle&=\frac{1}{2}\left(e^{2\pi i\frac{0}{4}}|0\rangle+e^{2\pi i\frac{1}{4}}|1\rangle+e^{2\pi i\frac{2}{4}}|2\rangle+e^{2\pi i\frac{3}{4}}|3\rangle\right)\\&=\frac{1}{2}\left(|0\rangle+i|1\rangle-|2\rangle-i|3\rangle\right) \quad (9.4b)\end{aligned}$$

$$\begin{aligned}\mathrm{QFT}|2\rangle&=\frac{1}{2}\left(e^{2\pi i2\frac{0}{4}}|0\rangle+e^{2\pi i2\frac{1}{4}}|1\rangle+e^{2\pi i2\frac{2}{4}}|2\rangle+e^{2\pi i2\frac{3}{4}}|3\rangle\right)\\&=\frac{1}{2}\left(|0\rangle-|1\rangle+|2\rangle-|3\rangle\right) \quad (9.4c)\end{aligned}$$

$$\begin{aligned}\mathrm{QFT}|3\rangle&=\frac{1}{2}\left(e^{2\pi i3\frac{0}{4}}|0\rangle+e^{2\pi i3\frac{1}{4}}|1\rangle+e^{2\pi i3\frac{2}{4}}|2\rangle+e^{2\pi i3\frac{3}{4}}|3\rangle\right)\\&=\frac{1}{2}\left(|0\rangle-i|1\rangle-|2\rangle+i|3\rangle\right) \quad (9.4c)\end{aligned}$$

What's the point of all this math? Eq. (9.4) gives the QFT of basis states, but we can apply the QFT to superpositions, such as $\frac{1}{\sqrt{2}}\left(|0\rangle+|2\rangle\right)=\frac{1}{\sqrt{2}}|0\rangle+0|1\rangle+\frac{1}{\sqrt{2}}|2\rangle+0|3\rangle$. As noted earlier, this two-qubit state has a period of 2. The QFT of this state effectively indicates the period. It would be nice if the QFT just spat out the number 2. However, the QFT of a two-qubit state is another two-qubit state; it's not a simple number like 2. The process for extracting the period from the QFT is this:

- We have a state of n qubits (so the number of basis states is $N=2^n$).
- We want to know the period r of the amplitudes of the state.
- We determine the QFT of the state.
- In the result, the amplitudes of basis states $|k\rangle$ are nonzero only for **k that are integer multiples of N/r**. Knowing N, r can be extracted.

Let's apply this process to determine the period r of $\frac{1}{\sqrt{2}}(|0\rangle + |2\rangle)$, a two-qubit state. We determine the QFT of this state:

$$\mathrm{QFT}\frac{1}{\sqrt{2}}(|0\rangle + |2\rangle) = \frac{1}{\sqrt{2}}(\mathrm{QFT}|0\rangle + \mathrm{QFT}|2\rangle).$$

Now we just plug in Eqs. (9.4a) and (9.4c):

$$\begin{aligned}\frac{1}{\sqrt{2}}(\mathrm{QFT}|0\rangle + \mathrm{QFT}|2\rangle) &= \frac{1}{\sqrt{2}}\left[\frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle)\right.\\ &\quad \left.+\frac{1}{2}(|0\rangle - |1\rangle + |2\rangle - |3\rangle)\right]\\ &= \frac{1}{\sqrt{2}}(|0\rangle + |2\rangle).\end{aligned}$$

We've just found that $\frac{1}{\sqrt{2}}(|0\rangle + |2\rangle)$ is its own QFT.

Next, we determine r from this result. We see that the result has nonzero amplitudes only for $|k\rangle$ where k is an integer multiple of 2. And since k is an integer multiple of 2, we set this 2 equal to N/r; this is the rule for determining period r. And since $N = 2^2 = 4$ because we have two qubits, $2 = 4/r$, so $r = 4/2 = 2$. This is the correct result.

Now we'll move on to the QFT of three qubits. There are now $N = 2^3 = 8$ basis states, $|000\rangle = |0\rangle$ through $|111\rangle = |7\rangle$. Equation (9.2a) now has eight terms in the sum:

$$\begin{aligned}\mathrm{QFT}|j\rangle = \frac{1}{2\sqrt{2}}\Big(&e^{2\pi i j\frac{0}{8}}|0\rangle + e^{2\pi i j\frac{1}{8}}|1\rangle + e^{2\pi i j\frac{2}{8}}|2\rangle + e^{2\pi i j\frac{3}{8}}|3\rangle + e^{2\pi i j\frac{4}{8}}|4\rangle\\ &+ e^{2\pi i j\frac{5}{8}}|5\rangle + e^{2\pi i j\frac{6}{8}}|6\rangle + e^{2\pi i j\frac{7}{8}}|7\rangle\Big).\end{aligned}$$

And since there are eight basis states $|j\rangle$, we have to work out the details for eight separate values of j. Using Eq. (9.1j) as necessary to keep the θ in $e^{i\theta}$ below 2π,

$$\begin{aligned}\mathrm{QFT}|0\rangle &= \frac{1}{2\sqrt{2}}\left(e^0|0\rangle + e^0|1\rangle + e^0|2\rangle + e^0|3\rangle + e^0|4\rangle + e^0|5\rangle + e^0|6\rangle + e^0|7\rangle\right)\\ &= \frac{1}{2\sqrt{2}}(|0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle + |5\rangle + |6\rangle + |7\rangle) \qquad (9.5a)\end{aligned}$$

$$\begin{aligned}\mathrm{QFT}|1\rangle &= \frac{1}{2\sqrt{2}}\Big(e^{2\pi i\frac{0}{8}}|0\rangle + e^{2\pi i\frac{1}{8}}|1\rangle + e^{2\pi i\frac{2}{8}}|2\rangle + e^{2\pi i\frac{3}{8}}|3\rangle + e^{2\pi i\frac{4}{8}}|4\rangle\\ &\quad + e^{2\pi i\frac{5}{8}}|5\rangle + e^{2\pi i\frac{6}{8}}|6\rangle + e^{2\pi i\frac{7}{8}}|7\rangle\Big)\\ &= \frac{1}{2\sqrt{2}}\big(|0\rangle + e^{\pi i/4}|1\rangle + e^{\pi i/2}|2\rangle + e^{3\pi i/4}|3\rangle + e^{\pi i}|4\rangle\\ &\quad + e^{5\pi i/4}|5\rangle + e^{3\pi i/2}|6\rangle + e^{7\pi i/4}|7\rangle\big) \qquad (9.5b)\end{aligned}$$

$$\begin{aligned}\mathrm{QFT}|2\rangle &= \frac{1}{2\sqrt{2}}\left(e^{2\pi i2\frac{0}{8}}|0\rangle + e^{2\pi i2\frac{1}{8}}|1\rangle + e^{2\pi i2\frac{2}{8}}|2\rangle + e^{2\pi i2\frac{3}{8}}|3\rangle + e^{2\pi i2\frac{4}{8}}|4\rangle\right.\\ &\quad \left.+ e^{2\pi i2\frac{5}{8}}|5\rangle + e^{2\pi i2\frac{6}{8}}|6\rangle + e^{2\pi i2\frac{7}{8}}|7\rangle\right)\\ &= \frac{1}{2\sqrt{2}}\left(|0\rangle + e^{\pi i/2}|1\rangle + e^{\pi i}|2\rangle + e^{3\pi i/2}|3\rangle + |4\rangle\right.\\ &\quad \left.+ e^{\pi i/2}|5\rangle + e^{\pi i}|6\rangle + e^{3\pi i/2}|7\rangle\right)\end{aligned} \tag{9.5c}$$

$$\begin{aligned}\mathrm{QFT}|3\rangle &= \frac{1}{2\sqrt{2}}\left(e^{2\pi i3\frac{0}{8}}|0\rangle + e^{2\pi i3\frac{1}{8}}|1\rangle + e^{2\pi i3\frac{2}{8}}|2\rangle + e^{2\pi i3\frac{3}{8}}|3\rangle + e^{2\pi i3\frac{4}{8}}|4\rangle\right.\\ &\quad \left.+ e^{2\pi i3\frac{5}{8}}|5\rangle + e^{2\pi i3\frac{6}{8}}|6\rangle + e^{2\pi i3\frac{7}{8}}|7\rangle\right)\\ &= \frac{1}{2\sqrt{2}}\left(|0\rangle + e^{3\pi i/4}|1\rangle + e^{3\pi i/2}|2\rangle + e^{\pi i/4}|3\rangle + e^{\pi i}|4\rangle\right.\\ &\quad \left.+ e^{7\pi i/4}|5\rangle + e^{\pi i/2}|6\rangle + e^{5\pi i/4}|7\rangle\right)\end{aligned} \tag{9.5d}$$

$$\begin{aligned}\mathrm{QFT}|4\rangle &= \frac{1}{2\sqrt{2}}\left(e^{2\pi i4\frac{0}{8}}|0\rangle + e^{2\pi i4\frac{1}{8}}|1\rangle + e^{2\pi i4\frac{2}{8}}|2\rangle + e^{2\pi i4\frac{3}{8}}|3\rangle + e^{2\pi i4\frac{4}{8}}|4\rangle\right.\\ &\quad \left.+ e^{2\pi i4\frac{5}{8}}|5\rangle + e^{2\pi i4\frac{6}{8}}|6\rangle + e^{2\pi i4\frac{7}{8}}|7\rangle\right)\\ &= \frac{1}{2\sqrt{2}}\left(|0\rangle + e^{\pi i}|1\rangle + |2\rangle + e^{\pi i}|3\rangle + |4\rangle + e^{\pi i}|5\rangle + |6\rangle + e^{\pi i}|7\rangle\right)\end{aligned} \tag{9.5e}$$

$$\begin{aligned}\mathrm{QFT}|5\rangle &= \frac{1}{2\sqrt{2}}\left(e^{2\pi i5\frac{0}{8}}|0\rangle + e^{2\pi i5\frac{1}{8}}|1\rangle + e^{2\pi i5\frac{2}{8}}|2\rangle + e^{2\pi i5\frac{3}{8}}|3\rangle + e^{2\pi i5\frac{4}{8}}|4\rangle\right.\\ &\quad \left.+ e^{2\pi i5\frac{5}{8}}|5\rangle + e^{2\pi i5\frac{6}{8}}|6\rangle + e^{2\pi i5\frac{7}{8}}|7\rangle\right)\\ &= \frac{1}{2\sqrt{2}}\left(|0\rangle + e^{5\pi i/4}|1\rangle + e^{\pi i/2}|2\rangle + e^{7\pi i/4}|3\rangle + e^{\pi i}|4\rangle\right.\\ &\quad \left.+ e^{\pi i/4}|5\rangle + e^{3\pi i/2}|6\rangle + e^{3\pi i/4}|7\rangle\right)\end{aligned} \tag{9.5f}$$

$$\begin{aligned}\mathrm{QFT}|6\rangle &= \frac{1}{2\sqrt{2}}\left(e^{2\pi i6\frac{0}{8}}|0\rangle + e^{2\pi i6\frac{1}{8}}|1\rangle + e^{2\pi i6\frac{2}{8}}|2\rangle + e^{2\pi i6\frac{3}{8}}|3\rangle + e^{2\pi i6\frac{4}{8}}|4\rangle\right.\\ &\quad \left.+ e^{2\pi i6\frac{5}{8}}|5\rangle + e^{2\pi i6\frac{6}{8}}|6\rangle + e^{2\pi i6\frac{7}{8}}|7\rangle\right)\\ &= \frac{1}{2\sqrt{2}}\left(|0\rangle + e^{3\pi i/2}|1\rangle + e^{\pi i}|2\rangle + e^{\pi i/2}|3\rangle + |4\rangle\right.\\ &\quad \left.+ e^{3\pi i/2}|5\rangle + e^{\pi i}|6\rangle + e^{\pi i/2}|7\rangle\right)\end{aligned} \tag{9.5g}$$

$$\begin{aligned}\mathrm{QFT}|7\rangle &= \frac{1}{2\sqrt{2}}\left(e^{2\pi i7\frac{0}{8}}|0\rangle + e^{2\pi i7\frac{1}{8}}|1\rangle + e^{2\pi i7\frac{2}{8}}|2\rangle + e^{2\pi i7\frac{3}{8}}|3\rangle + e^{2\pi i7\frac{4}{8}}|4\rangle\right.\\ &\quad \left.+ e^{2\pi i7\frac{5}{8}}|5\rangle + e^{2\pi i7\frac{6}{8}}|6\rangle + e^{2\pi i7\frac{7}{8}}|7\rangle\right)\\ &= \frac{1}{2\sqrt{2}}\left(|0\rangle + e^{7\pi i/4}|1\rangle + e^{3\pi i/2}|2\rangle + e^{5\pi i/4}|3\rangle + e^{\pi i}|4\rangle\right.\\ &\quad \left.+ e^{3\pi i/4}|5\rangle + e^{\pi i/2}|6\rangle + e^{\pi i/4}|7\rangle\right)\end{aligned} \tag{9.5h}$$

Wow, that was the most boring thing I've ever done. Possibly the most boring thing *anyone's* ever done. I now have scientific evidence that it's not actually possible to die of boredom.

Now let's use the QFT to determine the period of the three-qubit state $\frac{1}{\sqrt{2}}\left(|0\rangle+|4\rangle\right)$.

$$\mathrm{QFT}\frac{1}{\sqrt{2}}\left(|0\rangle+|4\rangle\right)=\frac{1}{\sqrt{2}}\left(\mathrm{QFT}|0\rangle+\mathrm{QFT}|4\rangle\right).$$

And now we just plug in Eqs. (9.5a) and (9.5e), and use $e^{\pi i}=-1$:

$$\begin{aligned}\frac{1}{\sqrt{2}}\left(\mathrm{QFT}|0\rangle+\mathrm{QFT}|4\rangle\right)&=\frac{1}{\sqrt{2}}\left[\frac{1}{2\sqrt{2}}\left(|0\rangle+|1\rangle+|2\rangle+|3\rangle+|4\rangle+|5\rangle+|6\rangle+|7\rangle\right)\right.\\&\quad\left.+\frac{1}{2\sqrt{2}}\left(|0\rangle-|1\rangle+|2\rangle-|3\rangle+|4\rangle-|5\rangle+|6\rangle-|7\rangle\right)\right]\\&=\frac{1}{2}\left(|0\rangle+|2\rangle+|4\rangle+|6\rangle\right).\end{aligned}$$

Now we examine the result of the QFT: $\frac{1}{2}\left(|0\rangle+|2\rangle+|4\rangle+|6\rangle\right)$. There are nonzero amplitudes only for $|k\rangle$ where k is a multiple of 2. This means that $2=N/r$, or $r=N/2$, where $N=2^3=8$, so $r=4$. Indeed, the period of $\frac{1}{\sqrt{2}}\left(|0\rangle+|4\rangle\right)$ is $4:\frac{1}{\sqrt{2}}\left(|0\rangle+|4\rangle\right)=\frac{1}{\sqrt{2}}|0\rangle+0|1\rangle+0|2\rangle+0|3\rangle+\frac{1}{\sqrt{2}}|4\rangle+0|5\rangle+0|6\rangle+0|7\rangle$. The periodic sequence of amplitudes is $\frac{1}{\sqrt{2}}, 0, 0, 0, \frac{1}{\sqrt{2}}, 0, 0, 0$. The repeating pattern is $\frac{1}{\sqrt{2}}, 0, 0, 0$, which has a size of 4.

Our next task is to figure out how to construct the QFT out of standard gates. We already saw that for a single qubit, the QFT is just H.

For two qubits, we return to Eq. (9.4), but we write the numbers in binary:

$$\mathrm{QFT}|00\rangle=\frac{1}{2}\left(|00\rangle+|01\rangle+|10\rangle+|11\rangle\right) \tag{9.6a}$$

$$\mathrm{QFT}|01\rangle=\frac{1}{2}\left(|00\rangle+i|01\rangle-|10\rangle-i|11\rangle\right) \tag{9.6b}$$

$$\mathrm{QFT}|10\rangle=\frac{1}{2}\left(|00\rangle-|01\rangle+|10\rangle-|11\rangle\right) \tag{9.6c}$$

$$\mathrm{QFT}|11\rangle=\frac{1}{2}\left(|00\rangle-i|01\rangle-|10\rangle+i|11\rangle\right) \tag{9.6d}$$

Next, we want to factor each expression on the right side into a product of two single-qubit states. Equation (9.6a) is easy. We've seen this before:

$$\mathrm{QFT}|00\rangle=\frac{1}{2}\left(|00\rangle+|01\rangle+|10\rangle+|11\rangle\right)=\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right). \tag{9.7a}$$

To factor Eq. (9.6b), we recognize that an i appears where the qubit on the right is $|1\rangle$, and a minus sign appears where the qubit on the left is $|1\rangle$. This suggests factoring as follows:

$$\text{QFT}|01\rangle = \frac{1}{2}\left(|00\rangle + \text{i}|01\rangle - |10\rangle - \text{i}|11\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle + \text{i}|1\rangle\right). \qquad (9.7\text{b})$$

We can confirm that the factored result is correct by using FOIL multiplication.

To factor Eq. (9.6c), we notice that a minus sign appears where the qubit on the right is $|1\rangle$, so

$$\text{QFT}|10\rangle = \frac{1}{2}|00\rangle - |01\rangle + |10\rangle - |11\rangle) = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right). \qquad (9.7\text{c})$$

And last, to factor Eq. (9.6d), we see that an i appears where the qubit on the right is $|1\rangle$, which suggests that the factored expression for the qubit on the right will include $\text{i}|1\rangle$. We see that a minus sign appears when one qubit is $|0\rangle$ and the other is $|1\rangle$. This means that the factored expression for each qubit should have a minus sign before the $|1\rangle$. Ultimately, this means

$$\begin{aligned}\text{QFT}|11\rangle &= \frac{1}{2}\left(|00\rangle - \text{i}|01\rangle - |10\rangle + \text{i}|11\rangle\right)\\ &= \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle - \text{i}|1\rangle\right). \qquad (9.7\text{d})\end{aligned}$$

The next step is to concoct a single equation that works for all four cases of Eq. (9.7). To do this, we'll write $\text{QFT}|\text{j}_1\text{j}_0\rangle$, where j_1 and j_0 are each either 0 or 1. Now let's look at the factored expressions in Eq. (9.7), and focus on the first qubit (the qubit on the left). This qubit is either $\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$ or $\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$. In fact, the qubit is $\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$ when $\text{j}_0 = 0$, and the qubit is $\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$ when $\text{j}_0 = 1$. In other words, the qubit on the left, in the result, is $\text{H}|\text{j}_0\rangle$.

Next, we focus on the qubit on the right in the result in Eq. (9.7). We get a minus sign when $\text{j}_1 = 1$, which suggests that we start with $\text{H}|\text{j}_1\rangle$. And the $|1\rangle$ gets a factor of i when $\text{j}_0 = 1$. Notice that $\text{i} = e^{\text{i}\pi/2}$. Notice, further, that $e^{\text{i}\pi\text{j}_0/2}$ is i

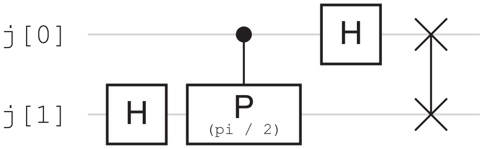


Figure 9.1. The two-qubit quantum Fourier transform, created using IBM Quantum.

when $j_0 = 1$, and it's 1 when $j_0 = 0$. So instead of saying that the $|1\rangle$ gets a factor of i when $j_0 = 1$, we can say that it *always* gets of factor of $e^{i\pi j_0/2}$. The gate that multiplies $|1\rangle$ by $e^{i\pi j_0/2}$ is called $P(\pi j_0/2)$. In general, the $P(\theta)$ gate, or phase gate, has no effect on $|0\rangle$, but it multiplies $|1\rangle$ by the relative phase factor $e^{i\theta}$:

$$P(\theta)|0\rangle = |0\rangle$$
$$P(\theta)|1\rangle = e^{i\theta}|1\rangle$$

So the general equation for the QFT for two qubits is

$$\mathrm{QFT}|j_1 j_0\rangle = H|j_0\rangle P(\pi j_0/2) H|j_1\rangle, \tag{9.8}$$

where the phase gate acts on what comes after it.

At last, we can construct the QFT for two qubits. Let's look at the circuit in Fig. 9.1 and show that it's equivalent to Eq. (9.8). The initial state is $|j_1 j_0\rangle$. After the H gate on the bottom qubit, the state becomes $H|j_1\rangle|j_0\rangle$. Next, we have a $P(\pi/2)$ that acts only when $j_0 = 1$. Equivalently, we can say that $P(\pi j_0/2)$ *always* acts (on the bottom qubit) because when $j_0 = 0$, we get $P(0)$, which doesn't do anything. So after the P gate, the state is $P(\pi j_0/2)H|j_1\rangle|j_0\rangle$. After the H gate on the top qubit, the state is is $P(\pi j_0/2)H|j_1\rangle H|j_0\rangle$. Comparing this to Eq. (9.8), we see that the two qubits are in the wrong order. So we apply a SWAP gate, which simply reverses the order. The SWAP is shown by the vertical line with X's at each end.

Now we move on to the QFT circuit for three qubits. We need the equivalent of Eq. (9.7) for three qubits. In Eq. (9.5a), we see that $\mathrm{QFT}|0\rangle = \mathrm{QFT}|000\rangle$ is the equally weighted superposition of all eight basis states for three qubits. An equally weighted superposition, like Eq. (9.7a), can be factored into $\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$ for each qubit:

$$\begin{aligned}\mathrm{QFT}|000\rangle &= \frac{1}{2\sqrt{2}}\big(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle \\ &\quad + |101\rangle + |110\rangle + |111\rangle\big) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\end{aligned} \tag{9.9a}$$

If we want to confirm the final factored form, we simply multiply two factors together to obtain $\frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right)$, and then we multiply each of the four terms by the final factor of $\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$.

Now let's rewrite Eq. (9.5b) in binary, to prepare to factor it:

$$\begin{aligned}\mathrm{QFT}|001\rangle = \frac{1}{2\sqrt{2}}\big(&|000\rangle + e^{\pi i/4}|001\rangle + e^{\pi i/2}|010\rangle + e^{3\pi i/4}|011\rangle + e^{\pi i}|100\rangle \\ &+ e^{5\pi i/4}|101\rangle + e^{3\pi i/2}|110\rangle + e^{7\pi i/4}|111\rangle\big)\end{aligned}$$

The trick here is to look at the terms that each have one $|1\rangle$: $|001\rangle$, $|010\rangle$, and $|100\rangle$. $|001\rangle$ is multiplied by $e^{\pi i/4}$, which means that the qubit on the right can be factored into $\frac{1}{\sqrt{2}}\left(|0\rangle + e^{\pi i/4}|1\rangle\right)$. Similarly, $|010\rangle$ is multiplied by $e^{\pi i/2}$, so the middle qubit can be factored into $\frac{1}{\sqrt{2}}\left(|0\rangle + e^{\pi i/2}|1\rangle\right)$. Last, $|100\rangle$ is multiplied by $e^{\pi i}$, so the qubit that's left (and on the left) can be factored into $\frac{1}{\sqrt{2}}\left(|0\rangle + e^{\pi i}|1\rangle\right)$:

$$\begin{aligned}\text{QFT}|001\rangle &= \frac{1}{2\sqrt{2}}\Big(|000\rangle + e^{\pi i/4}|001\rangle + e^{\pi i/2}|010\rangle + e^{3\pi i/4}|011\rangle + e^{\pi i}|100\rangle \\ &\quad + e^{5\pi i/4}|101\rangle + e^{3\pi i/2}|110\rangle + e^{7\pi i/4}|111\rangle\Big) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{\pi i}|1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle + e^{\pi i/2}|1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle + e^{\pi i/4}|1\rangle\right)\end{aligned} \tag{9.9b}$$

We can confirm that the factoring is correct by multiplying out the factors completely. Alternatively, we can pick a term with more than one $|1\rangle$, like $e^{3\pi i/4}|011\rangle$. This term is obtained by multiplying the $|0\rangle$ of the left qubit by the $e^{\pi i/2}|1\rangle$ of the middle qubit and the $e^{\pi i/4}|1\rangle$ of the right qubit. $e^{\pi i/2}$ multiplied by $e^{\pi i/4}$ indeed equals the $e^{3\pi i/4}$ in front of $|011\rangle$.

We apply the same process to obtain the QFT of the remaining computational basis states. Again, the trick is to look at the factors multiplying $|100\rangle$, $|010\rangle$, and $|001\rangle$. If $|100\rangle$ is multiplied by C, then the first qubit (on the left) can be factored into $\frac{1}{\sqrt{2}}\left(|0\rangle + C|1\rangle\right)$. A similar rule applies to the second and third qubits (from the left). These are the results:

$$\begin{aligned}\text{QFT}|010\rangle &= \frac{1}{2\sqrt{2}}\Big(|000\rangle + e^{\pi i/2}|001\rangle + e^{\pi i}|010\rangle + e^{3\pi i/2}|011\rangle + |100\rangle \\ &\quad + e^{\pi i/2}|101\rangle + e^{\pi i}|110\rangle + e^{3\pi i/2}|111\rangle\Big) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle + e^{\pi i}|1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle + e^{\pi i/2}|1\rangle\right)\end{aligned} \tag{9.9c}$$

$$\begin{aligned}\text{QFT}|011\rangle &= \frac{1}{2\sqrt{2}}\Big(|000\rangle + e^{3\pi i/4}|001\rangle + e^{3\pi i/2}|010\rangle + e^{\pi i/4}|011\rangle + e^{\pi i}|100\rangle \\ &\quad + e^{7\pi i/4}|101\rangle + e^{\pi i/2}|110\rangle + e^{5\pi i/4}|111\rangle\Big) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{\pi i}|1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle + e^{3\pi i/2}|1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle + e^{3\pi i/4}|1\rangle\right)\end{aligned} \tag{9.9d}$$

$$\begin{aligned}\text{QFT}|100\rangle &= \frac{1}{2\sqrt{2}}\Big(|000\rangle + e^{\pi i}|001\rangle + |010\rangle + e^{\pi i}|011\rangle + |100\rangle \\ &\quad + e^{\pi i}|101\rangle + |110\rangle + e^{\pi i}|111\rangle\Big) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle + e^{\pi i}|1\rangle\right)\end{aligned} \tag{9.9e}$$

$$\begin{aligned}\text{QFT}|101\rangle &= \frac{1}{2\sqrt{2}}\big(|000\rangle + e^{5\pi i/4}|001\rangle + e^{\pi i/2}|010\rangle + e^{7\pi i/4}|011\rangle + e^{\pi i}|100\rangle \\ &\quad + e^{\pi i/4}|101\rangle + e^{3\pi i/2}|110\rangle + e^{3\pi i/4}|111\rangle\big) \\ &= \frac{1}{\sqrt{2}}\big(|0\rangle + e^{\pi i}|1\rangle\big)\frac{1}{\sqrt{2}}\big(|0\rangle + e^{\pi i/2}|1\rangle\big)\frac{1}{\sqrt{2}}\big(|0\rangle + e^{5\pi i/4}|1\rangle\big) \qquad (9.9\text{f})\end{aligned}$$

$$\begin{aligned}\text{QFT}|110\rangle &= \frac{1}{2\sqrt{2}}\big(|000\rangle + e^{3\pi i/2}|001\rangle + e^{\pi i}|010\rangle + e^{\pi i/2}|011\rangle + |100\rangle \\ &\quad + e^{3\pi i/2}|101\rangle + e^{\pi i}|110\rangle + e^{\pi i/2}|111\rangle\big) \\ &= \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big)\frac{1}{\sqrt{2}}\big(|0\rangle + e^{\pi i}|1\rangle\big)\frac{1}{\sqrt{2}}\big(|0\rangle + e^{3\pi i/2}|1\rangle\big) \qquad (9.9\text{g})\end{aligned}$$

$$\begin{aligned}\text{QFT}|111\rangle &= \frac{1}{2\sqrt{2}}\big(|000\rangle + e^{7\pi i/4}|001\rangle + e^{3\pi i/2}|010\rangle + e^{5\pi i/4}|011\rangle \\ &\quad + e^{\pi i}|100\rangle + e^{3\pi i/4}|101\rangle + e^{\pi i/2}|110\rangle + e^{\pi i/4}|111\rangle\big) \\ &= \frac{1}{\sqrt{2}}\big(|0\rangle + e^{\pi i}|1\rangle\big)\frac{1}{\sqrt{2}}\big(|0\rangle + e^{3\pi i/2}|1\rangle\big)\frac{1}{\sqrt{2}}\big(|0\rangle + e^{7\pi i/4}|1\rangle\big) \qquad (9.9\text{h})\end{aligned}$$

Now we're ready to come up with a single equation equivalent to the eight versions of Eq. (9.9). We want a single equation for QFT$|j_2j_1j_0\rangle$, where j_2, j_1, and j_0 are each either 0 or 1. Examining the factored results in Eq. (9.9), we see that the qubit on the left always gets factored into either $\frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big)$ or $\frac{1}{\sqrt{2}}\big(|0\rangle + e^{\pi i}|1\rangle\big)$. In fact, we see a pattern, looking from Eq. (9.9a) to (9.9b) to (9.9c), etc.: In the QFT, the qubit on the left alternates between $\frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big)$ and $\frac{1}{\sqrt{2}}\big(|0\rangle + e^{\pi i}|1\rangle\big) = \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big)$. So when j_0 is 0, the qubit on the left becomes $\frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big) = H|0\rangle$, and when j_0 is 1, the qubit on the left becomes $\frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) = H|1\rangle$. In all cases, the qubit on the left becomes $H|j_0\rangle$, just as in Eq. (9.8), for two qubits.

The middle qubit in the QFT in Eq. (9.9) cycles through four expressions: $\frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big)$, $\frac{1}{\sqrt{2}}\big(|0\rangle + e^{\pi i/2}|1\rangle\big) = \frac{1}{\sqrt{2}}\big(|0\rangle + i|1\rangle\big)$, $\frac{1}{\sqrt{2}}\big(|0\rangle + e^{\pi i}|1\rangle\big) = \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big)$, and $\frac{1}{\sqrt{2}}\big(|0\rangle + e^{3\pi i/2}|1\rangle\big) = \frac{1}{\sqrt{2}}\big(|0\rangle - i|1\rangle\big)$. These are the same four expressions for the right qubit in Eq. (9.7). All four expressions are equivalent to $P(\pi j_0/2)H|j_1\rangle$, as in Eq. (9.8).

So far, we've seen that the first qubit in the QFT of three qubits is $H|j_0\rangle$, and the second qubit is $P(\pi j_0/2)H|j_1\rangle$. We can show that the third qubit is $P(\pi j_0/4)P(\pi j_1/2)H|j_2\rangle$. We could test this by going through all eight versions of Eq. (9.9). For example, in Eq. (9.9h), $j_0 = j_1 = j_2 = 1$, and $P(\pi j_0/4)P(\pi j_1/2)H|j_2\rangle =$

$\mathrm{P}(\pi/4)\,\mathrm{P}(\pi/2)\,\mathrm{H}|1\rangle = \mathrm{P}\left(\frac{\pi}{4}\right)\mathrm{P}\left(\frac{\pi}{2}\right)\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) = \mathrm{P}\left(\frac{\pi}{4}\right)\frac{1}{\sqrt{2}}\left(|0\rangle - e^{i\pi/2}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle - e^{3i\pi/4}|1\rangle\right)$. This doesn't quite look like the third qubit in Eq. (9.9h), $\frac{1}{\sqrt{2}}\left(|0\rangle + e^{7\pi i/4}|1\rangle\right)$. These two expressions actually are equal because $-1 = e^{i\pi}$, so $-e^{3i\pi/4} = e^{i\pi}e^{3i\pi/4} = e^{7\pi i/4}$. So the complete expression for the QFT for three qubits is

$$\mathrm{QFT}|j_2 j_1 j_0\rangle = \mathrm{H}|j_0\rangle \mathrm{P}(\pi j_0/2)\mathrm{H}|j_1\rangle \mathrm{P}(\pi j_0/4)\mathrm{P}(\pi j_1/2)\mathrm{H}|j_2\rangle. \tag{9.10}$$

Let's see how Fig. 9.2 implements Eq. (9.10). The initial state is $|j_2\rangle|j_1\rangle|j_0\rangle$. After the H on the bottom qubit, the state is $\mathrm{H}|j_2\rangle|j_1\rangle|j_0\rangle$. Next, P(π/2) is applied to the bottom qubit, but only if $j_1 = 1$. Equivalently, $\mathrm{P}(\pi j_1/2)$ is always applied to the bottom qubit, so the state becomes $\mathrm{P}(\pi j_1/2)\mathrm{H}|j_2\rangle|j_1\rangle|j_0\rangle$. Next, P(π/4) is applied to the bottom qubit if $j_0 = 1$, or, equivalently, $\mathrm{P}(\pi j_0/4)$ is always applied to the bottom qubit. The state becomes $\mathrm{P}(\pi j_0/4)\mathrm{P}(\pi j_1/2)\mathrm{H}|j_2\rangle|j_1\rangle|j_0\rangle$.

After the H on the middle qubit, the state is $\mathrm{P}(\pi j_0/4)\mathrm{P}(\pi j_1/2)\mathrm{H}|j_2\rangle\mathrm{H}|j_1\rangle|j_0\rangle$. The final controlled P gate is effectively $\mathrm{P}(\pi j_0/2)$, which makes the state $\mathrm{P}(\pi j_0/4)\mathrm{P}(\pi j_1/2)\mathrm{H}|j_2\rangle\mathrm{P}(\pi j_0/2)\mathrm{H}|j_1\rangle|j_0\rangle$. The H on the top qubit makes the state $\mathrm{P}(\pi j_0/4)\mathrm{P}(\pi j_1/2)\mathrm{H}|j_2\rangle\mathrm{P}(\pi j_0/2)\mathrm{H}|j_1\rangle\mathrm{H}|j_0\rangle$. Comparing this to Eq. (9.10), we see that the first and last qubits are reversed, so we correct this with a SWAP.

Let's again apply the three-qubit QFT to states with periodic amplitudes, so that we can extract the period from the QFT. As before, let's try the state $\frac{1}{\sqrt{2}}|0\rangle + 0|1\rangle + 0|2\rangle + 0|3\rangle + \frac{1}{\sqrt{2}}|4\rangle + 0|5\rangle + 0|6\rangle + 0|7\rangle$. The sequence of amplitudes is $\frac{1}{\sqrt{2}}, 0, 0, 0, \frac{1}{\sqrt{2}}, 0, 0, 0$. The pattern is a sequence of four items that repeat, so the period is $r = 4$. Let's pretend that we don't know this result, and we'll use the QFT to determine r.

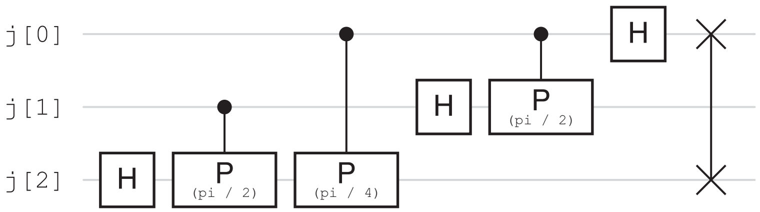


Figure 9.2. The three-qubit quantum Fourier transform, created using IBM Quantum. Originally published in Jed Brody and Kristen Gram, "Factoring 15 with a Remote Quantum Computer: A Complete Guide for Beginners," *European Journal of Physics*, April 2024, https://iopscience.iop.org/article/10.1088/1361-6404/ad32dc/pdf, under open license CC BY 4.0.

We have two choices. We can repeat what we did previously and apply Eq. (9.5) to our state to manually calculate the QFT. Alternatively, we can construct the state and the QFT in IBM Quantum, and then run the circuit to determine the QFT. This is less work! The quantum processor (or simulator) does the math for us.

Before the QFT, we first need to create the state $\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|4\rangle=\frac{1}{\sqrt{2}}(|000\rangle+|100\rangle)=\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|00\rangle=H|0\rangle|00\rangle$. Since the default initial state is $|0\rangle$ for all qubits, we simply need to apply an H to the bottom qubit to create the desired state, $\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|4\rangle$. The complete circuit is shown in Fig. 9.3. The H gate before the dashed line creates the state $\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|4\rangle$, and the rest of the circuit is the QFT.

The simulated results are shown in Fig. 9.4. The possible results are 000, 010, 100, and 110. Converting these binary numbers to base ten, they are 0, 2, 4, and 6, which are multiplies of 2. This tells us that 2 = N/r, where r is the period we seek, and $N=2^3=8$ since we have three qubits. So r = N/2 = 8/2 = 4, which is correct.

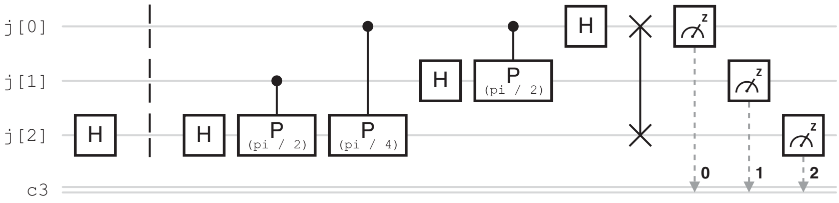


Figure 9.3. The circuit to generate the QFT of $\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|4\rangle$, created using IBM Quantum.

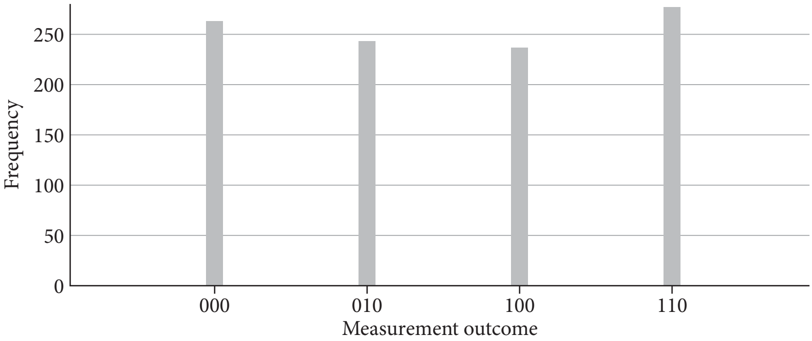


Figure 9.4. Results from the circuit in Fig. 9.3, created using IBM Quantum.

Let's try one more example. Let's find the QFT of $\frac{1}{2}|0\rangle + 0|1\rangle + \frac{1}{2}|2\rangle + 0|3\rangle + \frac{1}{2}|4\rangle + 0|5\rangle + \frac{1}{2}|6\rangle + 0|7\rangle$. The periodic sequence of amplitudes is 1/2, 0, 1/2, 0, 1/2, 0, 1/2, 0. The repeating pattern consists of two items, so the period r = 2.

To first create the state of $\frac{1}{2}|0\rangle + \frac{1}{2}|2\rangle + \frac{1}{2}|4\rangle + \frac{1}{2}|6\rangle = \frac{1}{2}\big(|000\rangle + |010\rangle + |100\rangle + |110\rangle\big) = \frac{1}{2}\big(|00\rangle + |01\rangle + |10\rangle + |11\rangle\big)|0\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big)\frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big)|0\rangle$, we need to apply an H gate to both the bottom qubit and the middle qubit. The QFT is unaffected, so the total circuit is what's shown in Fig. 9.5.

The simulated results are in Fig. 9.6. The possible results are 0 and 4, in base ten. These are multiples of 4, so 4 = N/r, so r = N/4 = 8/4 = 2. This is the correct result.

Later, we will need a circuit that reverses the effect of the QFT. In other words, we want the inverse quantum Fourier transform (IQFT). If the IQFT is applied immediately after the QFT, all the individual gates must cancel each other out so that nothing happens. So the IQFT consists of the inverses of the individual gates of the QFT, in the opposite order. The individual gates are

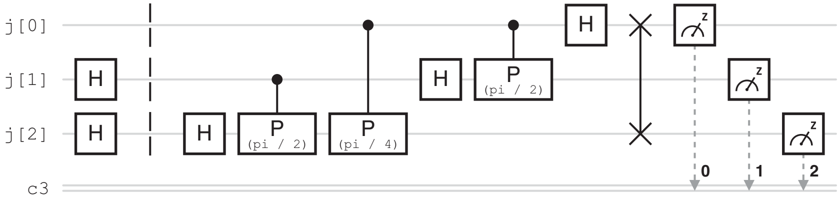


Figure 9.5. The circuit to generate the QFT of $\frac{1}{2}|0\rangle + \frac{1}{2}|2\rangle + \frac{1}{2}|4\rangle + \frac{1}{2}|6\rangle$, created using IBM Quantum.

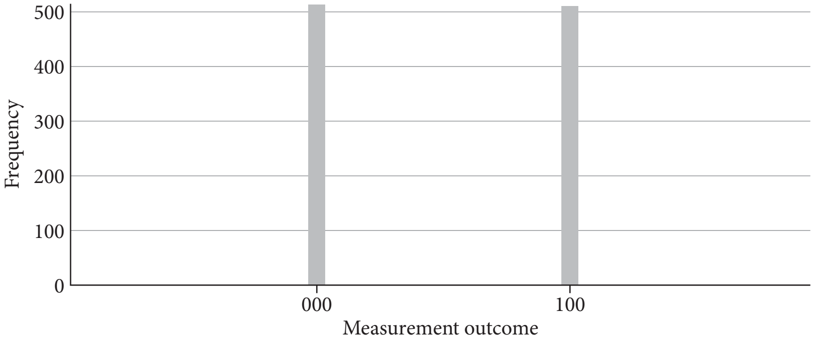


Figure 9.6. Results from the circuit in Fig. 9.5, created using IBM Quantum.

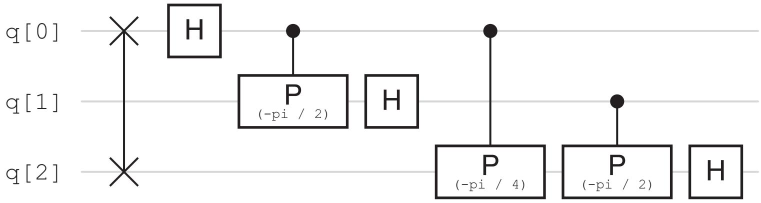


Figure 9.7. The three-qubit inverse quantum Fourier transform, created using IBM Quantum.

H gates, SWAP gates, and controlled P gates. We know that H is its own inverse, and it's pretty clear that SWAP is its own inverse. What's the inverse of a P gate? Since P(θ) multiplies $|1\rangle$ by $e^{i\theta}$, the inverse must multiply $|1\rangle$ by $e^{-i\theta}$. P(−θ) does this. The IQFT for three qubits therefore looks like Fig. 9.7.

Before leaving this chapter, let's find an equivalent expression for Eq. (9.10),

$$\mathrm{QFT}|j_2j_1j_0\rangle = \mathrm{H}|j_0\rangle \mathrm{P}(\pi j_0/2)\mathrm{H}|j_1\rangle \mathrm{P}(\pi j_0/4)\mathrm{P}(\pi j_1/2)\mathrm{H}|j_2\rangle.$$

The leftmost qubit on the right side of the equation is $\mathrm{H}|j_0\rangle$. We can write $\mathrm{H}|j_0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\pi j_0}|1\rangle\right)$ because $e^{i\pi j_0} = +1$ when $j_0 = 0$ and $e^{i\pi j_0} = -1$ when $j_0 = 1$, using Eqs. (9.1b) and (9.1f). Let's use $\mathrm{H}|j_0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\pi j_0}|1\rangle\right)$ and the equivalent expressions for $\mathrm{H}|j_1\rangle$ and $\mathrm{H}|j_2\rangle$:

$$\begin{aligned}\mathrm{QFT}|j_2j_1j_0\rangle = {} & \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\pi j_0}|1\rangle\right)\mathrm{P}\left(\pi j_0/2\right)\frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\pi j_1}|1\rangle\right) \\ & \mathrm{P}\left(\pi j_0/4\right)\mathrm{P}\left(\pi j_1/2\right)\frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\pi j_2}|1\rangle\right).\end{aligned}$$

The P(θ) gates simply multiply the $|1\rangle$ of the targeted qubit by $e^{i\theta}$, so

$$\begin{aligned}\mathrm{QFT}|j_2j_1j_0\rangle = {} & \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\pi j_0}|1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\pi(j_1 + j_0/2)}|1\rangle\right) \\ & \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\pi(j_2 + j_1/2 + j_0/4)}|1\rangle\right).\end{aligned} \tag{9.11}$$

This equation will come in handy in the next chapter.

Chapter 10

Quantum Phase Estimation

I Can Value Eigenvalues

We are closer than ever to the farthest Shor. We can smell the cooking fires and hear the barking of the dogs of the people encamped there. We just have to push through a little more math, and then we can understand how Shor's algorithm has the potential to menace internet security.

Shor's algorithm makes use of a circuit that performs *quantum phase estimation*. This is a circuit that determines something called the *eigenvalue* of an operator. To understand eigenvalues, let's look at the Z gate.

We recall that $Z|0\rangle = |0\rangle$ and $Z|1\rangle = -|1\rangle$. We see that when Z acts on $|1\rangle$, the result is $|1\rangle$, times -1. $|1\rangle$ is called an *eigenstate* of Z, which means that when Z acts on $|1\rangle$, we get back $|1\rangle$ times a number, -1; this number is called an *eigenvalue*. When Z acts on $|0\rangle$, we get back $|0\rangle$ times +1, so $|0\rangle$ is another eigenstate of Z, and +1 is the corresponding eigenvalue.

The eigenstates of Z are the computational basis states, $|0\rangle$ and $|1\rangle$. But other gates have eigenstates that are more complicated. Consider the X gate. We recall that $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$. $|0\rangle$ and $|1\rangle$ are not eigenstates of X: When X acts on $|0\rangle$, we do not get back $|0\rangle$ times a number, and when X acts on $|1\rangle$, we do not get back $|1\rangle$ times a number. So the eigenstates of X must be some superposition of $|0\rangle$ and $|1\rangle$: $\alpha|0\rangle + \beta|1\rangle$.

With just a little algebra, we can determine α and β to identify the eigenstates of X. If an eigenstate of X is $\alpha|0\rangle + \beta|1\rangle$, then $X(\alpha|0\rangle + \beta|1\rangle)$ must equal $\alpha|0\rangle + \beta|1\rangle$ times the eigenvalue, which is often called λ:

$$X(\alpha|0\rangle + \beta|1\rangle) = \lambda(\alpha|0\rangle + \beta|1\rangle).$$

The left side becomes $\alpha|1\rangle + \beta|0\rangle$, and the right side is simply $\lambda\alpha|0\rangle + \lambda\beta|1\rangle$, so

$$\alpha|1\rangle + \beta|0\rangle = \lambda\alpha|0\rangle + \lambda\beta|1\rangle.$$

The amplitude of $|0\rangle$ must be the same on both sides, so $\beta = \lambda\alpha$. Similarly, the amplitude of $|1\rangle$ must be the same on both sides, so $\alpha = \lambda\beta$. If we plug $\alpha = \lambda\beta$

into $\beta=\lambda\alpha$, we get $\beta=\lambda^2\beta$, so $\lambda^2=1$. This equation has two solutions for λ, +1, and −1. So the two eigenvalues of X are +1 and −1.

To find the eigenstates, we take one eigenvalue at a time and plug it into $\beta=\lambda\alpha$. When the eigenvalue $\lambda=1$, $\beta=\alpha$. To get a normalized state $\alpha|0\rangle+\beta|1\rangle$, we can choose $\alpha=\beta=\frac{1}{\sqrt{2}}$, so the eigenstate is $\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle$. For the other eigenvalue, $\lambda=-1$, $\beta=\lambda\alpha=-\alpha$. So we can choose $\beta=-\alpha=-\frac{1}{\sqrt{2}}$, giving us the other eigenstate, $\frac{1}{\sqrt{2}}|0\rangle-\frac{1}{\sqrt{2}}|1\rangle$.

Gates that act on two or more qubits also have eigenvalues and eigenstates. For example, the controlled Z, CZ, multiplies $|11\rangle$ by −1 and has no effect on $|00\rangle$, $|01\rangle$, and $|10\rangle$. So $|11\rangle$ is an eigenstate with eigenvalue −1, and $|00\rangle$, $|01\rangle$, and $|10\rangle$ are all eigenstates that share the eigenvalue +1. (Eigenstates that share the same eigenvalue are called *degenerate*, which is not intended as a moral judgment.)

So far, all the eigenvalues have been +1 or −1. It can be shown that the eigenvalues λ of all quantum operators satisfy $|\lambda|^2=1$. So +1 and −1 are possible eigenvalues, but so is $e^{i\theta}$ for any real number θ. (Recall that $|\lambda|^2=\lambda\lambda^*$, where λ^* is the complex conjugate of λ. To find the complex conjugate of λ, simply replace every i with −i. So if $\lambda=e^{i\theta}$, $\lambda^*=e^{-i\theta}$, and $|\lambda|^2=\lambda\lambda^*=e^{i\theta}e^{-i\theta}=e^0=1$.)

Since the eigenvalues of every quantum operator can be written $\lambda=e^{i\theta}$, our goal is to determine θ. The determination, or estimation, of θ is called *quantum phase estimation*. Now, suppose that $\lambda=-1$, which means that $\theta=\pi$, according to Eq. (9.1f). How can our measurement of qubits give us the value π? When we measure qubits, we get 0's and 1's. It would take a lot of 0's and 1's to accurately estimate π.

So, we define $j=\frac{\theta}{2\pi}$ so that the eigenvalue we seek is

$$\lambda=e^{i\theta}=e^{2\pi ij}. \tag{10.1}$$

Our circuit will give us j, from which we calculate the eigenvalue λ. Now, if $\lambda=-1=e^{i\pi}=e^{2\pi j}$, then $j=1/2$. $j=1/2$ is a much simpler number than $\theta=\pi$. But still, how can our measurement of 0's and 1's give us 1/2? In fact, the j we seek is always a fraction. How do we represent a fraction with 0's and 1's?

In ordinary base ten, 0.1 is one-tenth, 0.01 is one-hundredth, etc. Every position after the decimal point gets smaller by a factor of 10. In base two, every place after the decimal point gets smaller by a factor of 2. So 0.1 is one-half, 0.01 is one-quarter, and 0.001 is one-eighth. Let's look at the fractions we can represent with three bits after the decimal point:

$$0.000=0$$
$$0.001=1/8$$
$$0.010=1/4$$
$$0.011=1/4+1/8=3/8$$
$$0.100=1/2$$

$$0.101 = 1/2 + 1/8 = 5/8$$
$$0.110 = 1/2 + 1/4 = 3/4$$
$$0.111 = 1/2 + 1/4 + 1/8 = 7/8$$

So if we have three fractional bits, we can represent 0/8, 1/8, 2/8, 3/8, 4/8, 5/8, 6/8, and 7/8.

When we do quantum phase estimation, we need to decide how many qubits to use to estimate j. I will use three qubits, so our measurement will yield one of eight values: 000, 001, 010, 011, 100, 101, 110, and 111. We choose to use the three qubits to represent the fractional bits of j. So if the measured result is 010, for example, this means that $j = 0.010 = 1/4$, and $\theta = 2\pi j = \pi/2$, and finally the eigenvalue $\lambda = e^{i\theta} = e^{i\pi/2} = i$, according to Eq. (9.1d). The eight possible eigenvalues that can be estimated with three qubits are shown in Table 10.1, where the final column makes use of Eq. (9.1).

If we allocated four qubits to estimate j, there would be 16 possible results, and 16 possible estimates of the eigenvalue. Each additional qubit doubles the number of possible results. For our purposes, three qubits are sufficient.

To prepare to study the circuit that implements quantum phase estimation, consider Fig. 10.1, where $|v\rangle$ is an eigenstate of U so that $U|v\rangle = e^{i\theta}|v\rangle$. Since the control is $|0\rangle$, the U gate is not applied, and nothing happens.

Table 10.1

Measured result	j	$\theta = 2\pi j$	$\lambda = e^{i\theta}$
000	$0.000 = 0$	0	1
001	$0.001 = 1/8$	$\pi/4$	$\frac{1+i}{\sqrt{2}}$
010	$0.010 = 1/4$	$\pi/2$	i
011	$0.011 = 3/8$	$3\pi/4$	$\frac{-1+i}{\sqrt{2}}$
100	$0.100 = 1/2$	π	-1
101	$0.101 = 5/8$	$5\pi/4$	$-\frac{1+i}{\sqrt{2}}$
110	$0.110 = 3/4$	$3\pi/2$	$-i$
111	$0.111 = 7/8$	$7\pi/4$	$\frac{1-i}{\sqrt{2}}$

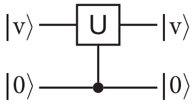


Figure 10.1. A controlled operator that does not act on its eigenvalue because the control is $|0\rangle$, created using the Quantikz LaTeX package.

What if the control is $|1\rangle$, as in Fig. 10.2? Since the control is $|1\rangle$, U acts on $|v\rangle$. Since $|v\rangle$ is an eigenstate, $U|v\rangle$ is $|v\rangle$ times the eigenvalue, $e^{i\theta}$.

Now, the circuit we really want to understand is in Fig. 10.3. The initial state is $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|v\rangle=\frac{1}{\sqrt{2}}(|0\rangle|v\rangle+|1\rangle|v\rangle)$. The controlled U has no effect on $|0\rangle|v\rangle$ because the control is $|0\rangle$. However, when the controlled U acts on the $|1\rangle|v\rangle$ term, the U acts on $|v\rangle$ because the control is $|1\rangle$, so this term is multiplied by $e^{i\theta}$. The final state is $\frac{1}{\sqrt{2}}(|0\rangle|v\rangle+|1\rangle e^{i\theta}|v\rangle)=\frac{1}{\sqrt{2}}(|0\rangle+e^{i\theta}|1\rangle)|v\rangle$. So even though the *upper* qubit is the target, the controlled U effectively attaches a factor of $e^{i\theta}$ to the $|1\rangle$ in the *lower* qubit. (This is called *phase kickback*.)

Given that $|v\rangle$ is an eigenstate of U, we want to determine the associated eigenvalue, $\lambda=e^{i\theta}=e^{2\pi ij}$. If we allocate three qubits to determine j, our measurement will yield three fractional bits of j. If $j=0.j_2j_1j_0$ is a binary fraction, the measurement will yield $j_2j_1j_0$. Let's see how Fig. 10.4 performs the quantum phase estimation.

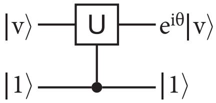


Figure 10.2. A controlled operator that acts on its eigenvalue because the control is $|1\rangle$, created using the Quantikz LaTeX package.

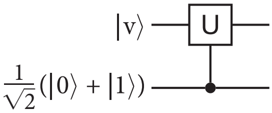


Figure 10.3. A controlled operator with a superposition on the control, created using the Quantikz LaTeX package.

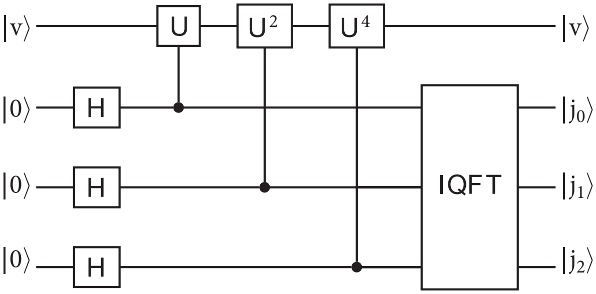


Figure 10.4. Quantum phase estimation, created using the Quantikz LaTeX package.

The initial state is $|0\rangle|0\rangle|0\rangle|v\rangle$, where the top qubit, $|v\rangle$, is an eigenstate of U. After the H gates, the overall state is $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|v\rangle$. The controlled U, as we've just seen, effectively attaches a factor of $e^{i\theta}$ to the $|1\rangle$ in the control qubit. The state after the controlled U is thus $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle+e^{i\theta}|1\rangle)|v\rangle$.

Since $U^2|v\rangle=UU|v\rangle=Ue^{i\theta}|v\rangle=e^{i\theta}U|v\rangle=e^{2i\theta}|v\rangle$, the controlled U^2 effectively attaches a factor of $e^{2i\theta}$ to the $|1\rangle$ in the control qubit. So the state after the controlled U^2 is $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle+e^{2i\theta}|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle+e^{i\theta}|1\rangle)|v\rangle$. By the same logic, the controlled U^4 attaches a factor of $e^{4i\theta}$ to the $|1\rangle$ in the control qubit, producing the state $\frac{1}{\sqrt{2}}(|0\rangle+e^{4i\theta}|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle+e^{2i\theta}|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle+e^{i\theta}|1\rangle)|v\rangle$.

Next, we will substitute $\theta=2\pi j=2\pi(0.j_2j_1j_0)$. $0.j_2j_1j_0$ is a binary fraction, so it's equal to $j_2/2+j_1/4+j_0/8$, so $\theta=\pi(j_2+j_1/2+j_0/4)$. Substituting this into our expression for the state after the controlled U^4, we obtain

$$\frac{1}{\sqrt{2}}\left(|0\rangle+e^{i\pi(4j_2+2j_1+j_0)}|1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle+e^{i\pi(2j_2+j_1+j_0/2)}|1\rangle\right)$$
$$\frac{1}{\sqrt{2}}\left(|0\rangle+e^{i\pi(j_2+j_1/2+j_0/4)}|1\rangle\right)|v\rangle. \tag{10.2}$$

This can be simplified. Consider the factor $e^{i\pi(4j_2+2j_1+j_0)}$. Using the algebraic rule $e^{a+b}=e^ae^b$, $e^{i\pi(4j_2+2j_1+j_0)}=e^{4j_2\pi i}e^{2j_1\pi i}e^{j_0\pi i}$. Let's start with the middle factor, $e^{2j_1\pi i}$. j_1 is either 0 or 1. In either case, $e^{2j_1\pi i}$ is 1: If j_1 is 0, $e^{2j_1\pi i}=e^0=1$ because anything to the 0 power is 1, and if j_1 is 1, $e^{2j_1\pi i}=e^{2\pi i}=e^0$ from Eq. (9.1j). Since $e^{2j_1\pi i}=1$ in any case, and a factor of 1 doesn't do anything, the factor of $e^{2j_1\pi i}$ can be dropped. Similarly, the factor of $e^{4j_2\pi i}$ is $e^0=1$ if j_2 is 0, and if j_2 is 1, then $e^{4j_2\pi i}=e^{4\pi i}=e^{2\pi i}=e^0$ from repeated use of Eq. (9.1j). So the factor of $e^{4j_2\pi i}$ is always 1 and can be dropped.

The middle term in Eq. (10.2) is $e^{i\pi(2j_2+j_1+j_0/2)}=e^{2j_2\pi i}e^{j_1\pi i}e^{j_0\pi i/2}$. By the reasoning in the previous paragraph, $e^{2j_2\pi i}$ is always 1 and can be dropped. Dropping from Eq. (10.2) all factors of 1, we obtain $\frac{1}{\sqrt{2}}(|0\rangle+e^{i\pi j_0}|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle+e^{i\pi(j_1+j_0/2)}|1\rangle)\frac{1}{\sqrt{2}}(|0\rangle+e^{i\pi(j_2+j_1/2+j_0/4)}|1\rangle)|v\rangle$. And what to our wondering eyes does appear but the right-hand side of Eq. (9.11), followed by $|v\rangle$ for the top qubit. So the state of the qubits just before the IQFT is exactly $QFT|j_2j_1j_0\rangle|v\rangle$. The IQFT cancels the QFT, and the final state of the circuit is simply $|j_2j_1j_0\rangle|v\rangle$. So a measurement of the bottom three qubits yields $j_2j_1j_0$, from which we determine $j=0.j_2j_1j_0$, from which we determine the eigenvalue $\lambda=e^{i\theta}=e^{2\pi ij}$. Are we having fun yet? (Answer: Yes, so very much!) Let's see how this works with a specific example.

Suppose we know that $|1\rangle$ is an eigenstate of Z, but we don't know the eigenvalue. So we decide to apply quantum phase estimation. The circuit is in Fig. 10.5.

The top qubit needs to be initialized to $|1\rangle$, the eigenstate we're interested in. Since $|0\rangle$ is the default starting state, we use a NOT to create the $|1\rangle$. Next, we apply an H gate to the three qubits that will be used to estimate the eigenvalue. Next we have a controlled Z, which shows up as a line with a dot at each end in IBM Quantum. Next we need a controlled Z^2, or equivalently, two controlled Z gates in a row. We could actually omit these gates because $Z^2 = I$. Similarly, we could omit the four controlled Z gates that act as a controlled Z^4. Following this, we have the IQFT from Fig. 9.7, and then we measure the three bottom qubits to determine $j = 0.j_2 j_1 j_0$. Figure 10.6 gives the results obtained on a real quantum processor called ibmq_lima.

Ideally, 100% of the results would be 100. Due to error, other results occur occasionally, but 100 is clearly the dominant result. This means that j is the binary fraction 0.100, which is 1/2. So the eigenvalue is $e^{i\theta} = e^{2\pi ij} = e^{2\pi i/2} = e^{\pi i} = -1$, which is exactly the correct eigenvalue! $Z|1\rangle = -|1\rangle$, so Z acting on its eigenstate $|1\rangle$ returns the eigenstate multiplied by −1.

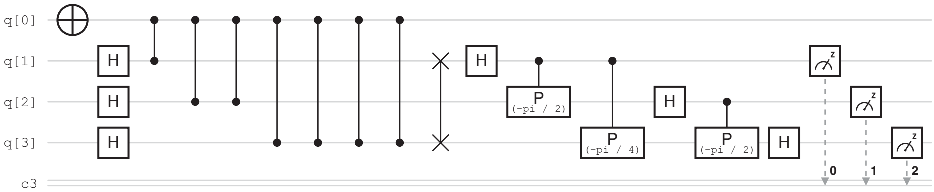


Figure 10.5. Quantum phase estimation to determine the eigenvalue associated with the $|1\rangle$ eigenvector of Z, created using IBM Quantum.

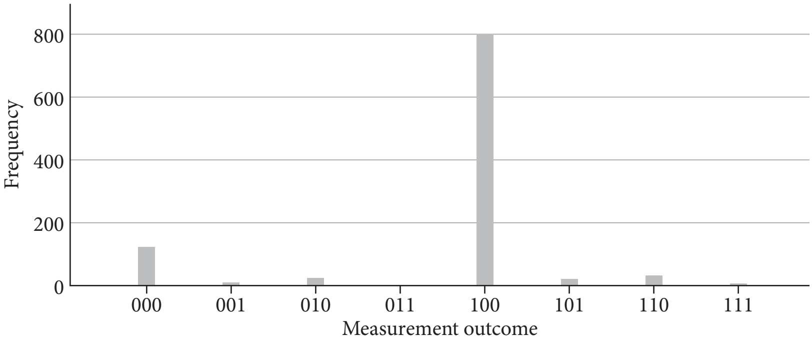


Figure 10.6. Results from the circuit in Fig. 10.5, created using IBM Quantum.

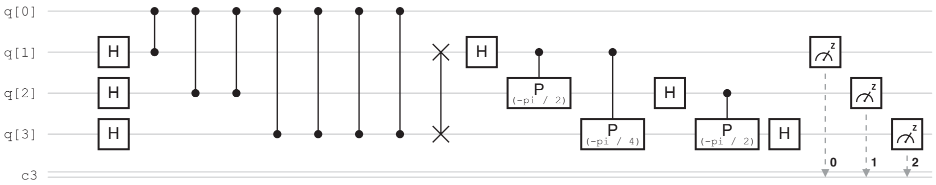


Figure 10.7. Quantum phase estimation to determine the eigenvalue associated with the $|0\rangle$ eigenvector of Z, created using IBM Quantum.

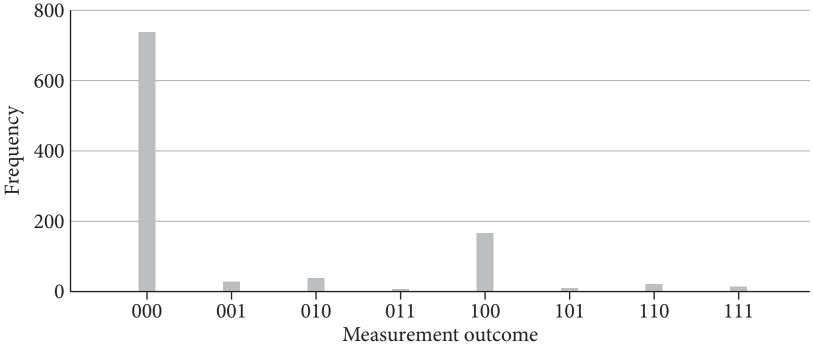


Figure 10.8. Results from the circuit in Fig. 10.7, created using IBM Quantum.

If we want to determine the eigenvalue associated with the other eigenstate, $|0\rangle$, we simply remove the NOT gate so that the top qubit remains initialized to $|0\rangle$, as in Fig. 10.7. The measured results on ibmq_lima are given in Fig. 10.8. The dominant result in this case is 000, so $j = 0.000 = 0$, so the eigenvalue is $e^{2\pi ij} = e^0 = +1$. Again, this is correct because $Z|0\rangle = |0\rangle$: when Z acts on $|0\rangle$, it returns the eigenstate multiplied by +1.

Let's try one more example. Suppose we know that $|1\rangle$ is an eigenstate of $P(\pi/4)$, and we want to know the corresponding eigenvalue. We construct the circuit in Fig. 10.9. The NOT gate initializes the top qubit to $|1\rangle$, the eigenstate we're interested in. We have the H gates, as always, on the three qubits that will estimate three fractional bits of j. Then we have a controlled $P(\pi/4)$, and then a controlled $[P(\pi/4)]^2$. We could implement this with two controlled $P(\pi/4)$ gates in a row. However, it's simpler to recognize that since $[P(\pi/4)]^2|1\rangle = P(\pi/4)P(\pi/4)|1\rangle = e^{i\pi/4}P(\pi/4)|1\rangle = e^{i\pi/4}e^{i\pi/4}|1\rangle = e^{i\pi/2}|1\rangle$, $[P(\pi/4)]^2 = P(\pi/2)$. Similarly, the controlled $[P(\pi/4)]^4$ is equivalent to a controlled $P(\pi)$. Last, we have the IQFT and the measurements.

The results from ibmq_lima are shown in Fig. 10.10. The dominant result is 001, which would be the only result, in the absence of error. So

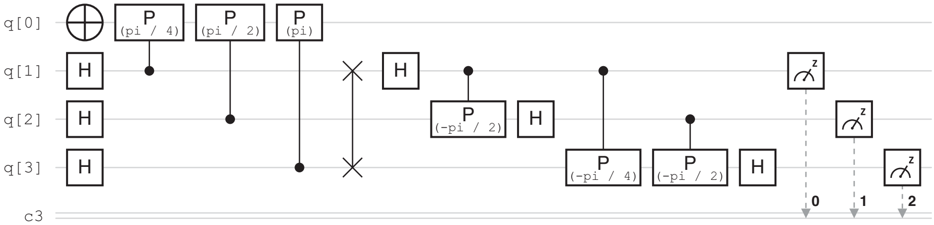


Figure 10.9. Quantum phase estimation to determine the eigenvalue association with the $|1\rangle$ eigenvector of P(π/4), created using IBM Quantum. Originally published in Jed Brody and Kristen Gram, "Factoring 15 with a Remote Quantum Computer: A Complete Guide for Beginners," *European Journal of Physics*, April 2024, https://iopscience.iop.org/article/10.1088/1361-6404/ad32dc/pdf, under open license CC BY 4.0.

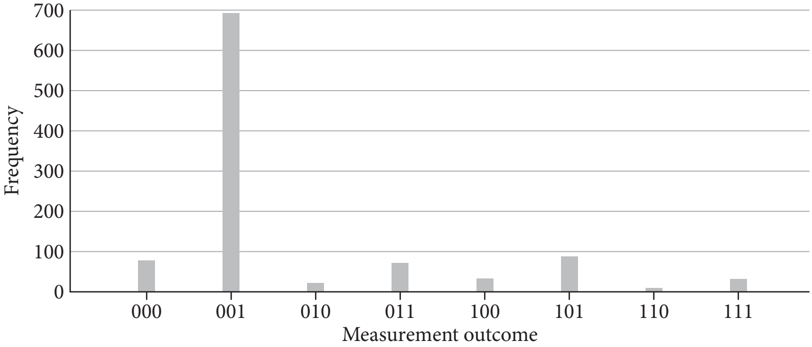


Figure 10.10. Results from the circuit in Fig. 10.9, created using IBM Quantum.

$j = 0.001 = 1/8$ and the eigenvalue is $e^{2\pi i/8} = e^{\pi i/4}$. This is exactly correct since $P(\pi/4)|1\rangle = e^{i\pi/4}|1\rangle$.

Next, suppose the top qubit is not initialized to either of the eigenstates of P(π/4), $|0\rangle$ and $|1\rangle$. Let's start the top qubit in an equally weighted superposition of these two eigenstates, $\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$. We achieve this by applying an H gate to the initial $|0\rangle$ in the top qubit (Fig. 10.11).

What results do we expect when the top qubit starts out in a superposition of the two eigenstates of P(π/4)? Theoretically, half of the results should give the eigenvalue associated with $|0\rangle$, and the other half should give the eigenvalue associated with $|1\rangle$. The actual results on ibmq_lima are dominated by the two expected values (000 and 001), though error causes other values to appear at lower frequencies (Fig. 10.12).

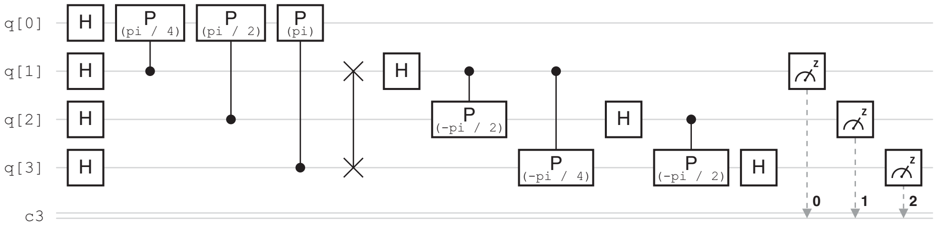


Figure 10.11. Quantum phase estimation to determine both eigenvalues association of P(π/4), created using IBM Quantum.

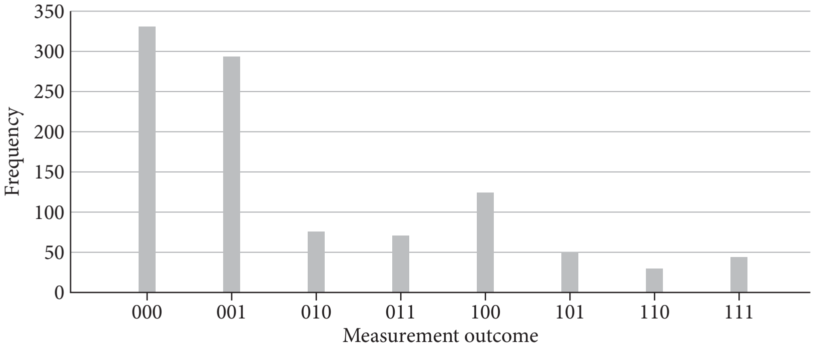


Figure 10.12. Results from the circuit in Fig. 10.11, created using IBM Quantum.

We're inching toward Shor's algorithm. In our implementation of Shor's algorithm, we will want to determine the eigenvalues of an operator U that acts on four qubits. So our phase estimation circuit will look like Fig. 10.13. We'll call the top four qubits the *eigenstate register*, since we want U to act on its eigenstates. The bottom three qubits are called the *eigenvalue register*, since they will estimate the eigenvalues of U.

The U in Shor's algorithm performs a kind of multiplication that turns one basis state into another basis state. For example, if U performs multiplication by 2, it turns $|1\rangle$ into $|2\rangle$: $U|1\rangle = |2\rangle$. It also turns $|2\rangle$ into $|4\rangle$: $U|2\rangle = |4\rangle$. We could just as well write the numbers in binary: $U|0001\rangle = |0010\rangle$ and $U|0010\rangle = U|0100\rangle$, using four qubits because we've decided that our U will act on four qubits.

Now, the biggest number that four qubits can represent is 15 = 1111. So what happens when U tries to multiply 8 by 2? We will use something called *modular arithmetic*, the arithmetic of remainders. AmodB is defined as the remainder when A is divided by B. So 15mod12 = 3 because 15 divided by 12 is 1 remainder 3. 37mod2 = 1 because 37 divided by 2 is 18 remainder 1.

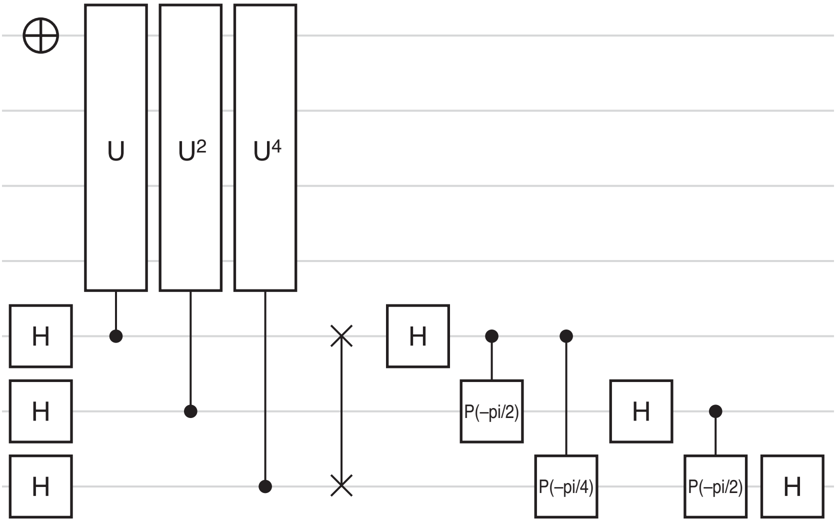


Figure 10.13. Quantum phase estimation to determine the eigenvalues of an operator that acts on four qubits, created using algassert.com/quirk. Originally published in Jed Brody and Kristen Gram, "Factoring 15 with a Remote Quantum Computer: A Complete Guide for Beginners," *European Journal of Physics*, April 2024, https://iopscience.iop.org/article/10.1088/1361-6404/ad32dc/pdf, under open license CC BY 4.0.

Suppose some integer J is multiplied by another integer A. Then we want to know the remainder when this product is divided by yet another integer N: AJmodN. (From now on, N is just the number we're dividing by to find a remainder. N is no longer 2^n, the number of basis states. It's standard in quantum computing books to use N for these two unrelated purposes, but it's *not* standard to give the reader a heads-up.) So if A is 2 and J is 8 and N is 15, $\text{AJmodN} = (2 \times 8)\text{mod}15 = 16\text{mod}15 = 1$. So our modular multiplication operator, U, acts on a basis state $|J\rangle$ as follows:

$$U|J\rangle = |\text{AJmodN}\rangle. \tag{10.3}$$

Suppose U multiplies by 2mod15: $U|J\rangle = |2\text{Jmod}15\rangle$. For J less than 8, $U|J\rangle$ is just $|2J\rangle$. For example, $U|7\rangle = |(2\times 7)\text{mod}15\rangle = |14\rangle$ because 14 divided by 15 is 0 remainder 14. Even if J is greater than 8, we're still certain that 2Jmod15 will be less than 15 because the remainder is less than 15 when any number is divided by 15. So four qubits, which can represent a number as big as 15, are sufficient to represent 2Jmod15 for any J.

Next, let's look at modular exponentiation: A^0modN, A^1modN, A^2modN, etc. Choosing A = 4 and N = 15, we find

$$4^0\text{mod}15 = 1$$
$$4^1\text{mod}15 = 4$$

$$\begin{aligned}
&4^2 \text{mod} 15 = 1\\
&4^3 \text{mod} 15 = 4\\
&4^4 \text{mod} 15 = 1\\
&4^5 \text{mod} 15 = 4
\end{aligned}$$

Apparently, there's a repeating pattern. The number of terms in the repeating pattern is called the *order* r of 4mod15. In this case, the order $r = 2$ because there are two terms in the repeating pattern, 1 and 4. Equivalently, the order of 4mod15 is the smallest positive r such that $4^r\text{mod}15 = 1$. We see that $4^2\text{mod}15 = 1$, so $r = 2$. $4^4\text{mod}15$ also is 1, but 2 is smaller than 4, which is why the order is 2 instead of 4.

Here's another example. Let's find the order of 2mod15:

$$\begin{aligned}
&2^0 \text{mod} 15 = 1\\
&2^1 \text{mod} 15 = 2\\
&2^2 \text{mod} 15 = 4\\
&2^3 \text{mod} 15 = 8\\
&2^4 \text{mod} 15 = 1\\
&2^5 \text{mod} 15 = 2\\
&2^6 \text{mod} 15 = 4\\
&2^7 \text{mod} 15 = 8
\end{aligned}$$

In this case, the repeating pattern has four terms, 1, 2, 4, and 8, so the order of 2mod15 is $r = 4$. Equivalently, we see that $2^4\text{mod}15$ is 1, and there is no smaller positive exponent such that $2^r\text{mod}15 = 1$, so the order $r = 4$.

It can be shown that the order r of AmodN is the number of eigenstates of the operator U defined in Eq. (10.3). For example, if $U|J\rangle = |4J\text{mod}15\rangle$, the order of 4mod15 is 2, so there are 2 eigenstates of U. It can also be shown that the eigenstates of this U are

$$\frac{1}{\sqrt{2}}\left(|1\rangle + |4\rangle\right), \text{ with eigenvalue } 1,$$

and

$$\frac{1}{\sqrt{2}}\left(|1\rangle - |4\rangle\right), \text{ with eigenvalue } -1.$$

We won't derive this from scratch, but let's prove that it's true. Let's make U act on its first eigenstate, $\frac{1}{\sqrt{2}}\left(|1\rangle + |4\rangle\right)$:

$$\begin{aligned}
U\frac{1}{\sqrt{2}}\left(|1\rangle + |4\rangle\right) &= \frac{1}{\sqrt{2}}\left(U|1\rangle + U|4\rangle\right)\\
&= \frac{1}{\sqrt{2}}\left(|4\text{mod}15\rangle + |16\text{mod}15\rangle\right)\\
&= \frac{1}{\sqrt{2}}\left(|4\rangle + |1\rangle\right),
\end{aligned}$$

which is exactly what we started with, so the eigenvalue is 1.

Now let's have U act on its second eigenstate, $\frac{1}{\sqrt{2}}\left(|1\rangle - |4\rangle\right)$:

$$\begin{aligned} U\frac{1}{\sqrt{2}}\left(|1\rangle - |4\rangle\right) &= \frac{1}{\sqrt{2}}\left(U|1\rangle - U|4\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|4\mathrm{mod}15\rangle - |16\mathrm{mod}15\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|4\rangle - |1\rangle\right). \end{aligned}$$

This is exactly −1 times what we started with, so the eigenvalue is −1.

More generally, it can be shown (though mercifully we won't show it) that the eigenvalues of U are $e^{2\pi is/r}$ for all whole numbers s less than r. So when $r=2$, as in the preceding example, s can be either 0 or 1, and the eigenvalues are $e^0=1$ and $e^{2\pi i/2}=e^{\pi i}=-1$, exactly as we just saw.

There's one more useful property of the eigenstates of U (which is still the modular multiplication operator defined in Eq. [10.3]): The sum of eigenstates of U, divided by $\sqrt{r}$, is exactly $|1\rangle$. Let's see how this works for the two eigenstates of U given by $U|J\rangle = |4J\mathrm{mod}15\rangle$. We showed that the eigenstates are $\frac{1}{\sqrt{2}}\left(|1\rangle + |4\rangle\right)$ and $\frac{1}{\sqrt{2}}\left(|1\rangle - |4\rangle\right)$. I now claim that if we sum these and divide by $\sqrt{r} = \sqrt{2}$, we'll get $|1\rangle$:

$$\left[\frac{1}{\sqrt{2}}\left(|1\rangle + |4\rangle\right) + \frac{1}{\sqrt{2}}\left(|1\rangle - |4\rangle\right)\right]\Big/\sqrt{2} = \frac{2}{\sqrt{2}}|1\rangle/\sqrt{2} = |1\rangle.$$

This is useful because $|1\rangle$ is an easy state to create. If we start the eigenstate register in the state $|1\rangle = |0001\rangle$, we're actually starting it in a superposition of all the eigenstates of U. So when we perform quantum phase estimation, the final measurement is equally likely to yield any of the eigenvalues of U.

Recall that the eigenvalues of U are $e^{2\pi is/r}$ for all whole numbers s less than r. Suppose the order r is something we don't know. We can perform quantum phase estimation to determine r. In this context, the quantum phase estimation circuit is called the *order-finding circuit*. This is how it works.

Quantum phase estimation lets us determine $j=0.j_2j_1j_0$, which gives us the eigenvalue $e^{2\pi ij}$. Since the eigenvalues of U are $e^{2\pi is/r}$, $j=s/r$ for all whole numbers s less than r. So the possible values of j are 0/r, 1/r, 2/r, and so on, up to $(r-1)/r$. The denominator in all cases is r, which is the number we want to determine. So if we use the order-finding circuit and find that j is 0 or 1/2, we conclude that $r=2$: When $r=2$, s is either 0 or 1, and the possible values of j are 0/2 and 1/2. Since we have three qubits in the eigenvalue register, the actual measured results would be either 000 such that $j=0.000=0$, or 100 such that $j=0.100=1/2$, again understanding 0.100 as a binary fraction.

If we use the order-finding circuit and find that j is 0, 1/4, 1/2, or 3/4, we conclude that r = 4: When r = 4, s is 0, 1, 2, or 3, and the possible values of j are 0/4, 1/4, 2/4, and 3/4. If we looked only at 2/4 = 1/2, we might conclude that r = 2 since the simplified denominator is 2. The 1/4 and 3/4 results are what convince us that r must be 4. The actual measured results would be 000 such that j = 0.000 = 0, 010 such that j = 0.010 = 1/4, 100 such that j = 0.100 = 1/2, or 0.110 such that j = 0.110 = 3/4.

The final challenge is to construct U out of standard gates. There are six U's that we will use in Shor's algorithm, and we will construct each of these in some detail: $U|J\rangle = |2J\text{mod}15\rangle$, $U|J\rangle = |4J\text{mod}15\rangle$, $U|J\rangle = |7J\text{mod}15\rangle$, $U|J\rangle = |8J\text{mod}15\rangle$, $U|J\rangle = |11J\text{mod}15\rangle$, and $U|J\rangle = |13J\text{mod}15\rangle$. We recall that our circuit, Fig. 10.13, contains a controlled U, and controlled U^2, and a controlled U^4.

Let's start with $U|J\rangle = |2J\text{mod}15\rangle$. This operator changes $|1\rangle$ to $|2\rangle$, $|2\rangle$ to $|4\rangle$, $|3\rangle$ to $|6\rangle$, and so on. But actually, we don't need it to work for every possible $|J\rangle$. We know that the eigenstate register starts in the state $|1\rangle$, which we saw is a superposition of the eigenstates of U. So actually, we just need U to change $|1\rangle$ to $|2\rangle$. In binary, we're changing $|0001\rangle$ to $|0010\rangle$. Two NOT gates accomplish this. In the order-finding circuit, U is controlled, so the NOTs are controlled NOTs. Figure 10.14 shows the beginning of the order-finding circuit.

The top four qubits are the eigenstate register. The first NOT gate initializes the eigenstate register to $|0001\rangle$. The bottom three qubits are the eigenvalue register, and the order-finding circuit includes an H gate on each of these.

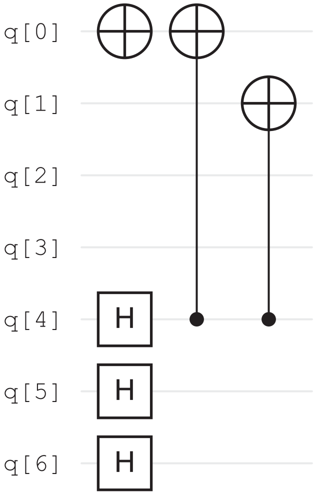


Figure 10.14. The first part of the circuit to determine the order of 2mod15, created using IBM Quantum.

The two controlled NOT gates form the controlled U. If the U is applied, it changes $|0001\rangle$ to $|0010\rangle$.

The next gate that we'll need is a controlled U^2. Since U multiplies by 2, U^2 multiplies by 4. So U^2 changes $|1\rangle$ to $|4\rangle$, $|2\rangle$ to $|8\rangle$, $|3\rangle$ to $|12\rangle$, and so on. Just as we didn't need U itself to operate on every possible basis state, we don't need U^2 to operate on every possible basis state. After the controlled U, the eigenstate register is in a superposition of $|1\rangle$ (if U didn't act) and $|2\rangle$ (if U acted on $|1\rangle$). So we need our controlled U^2 to operate properly on $|1\rangle$ and $|2\rangle$. As stated earlier, U^2 changes $|1\rangle$ to $|4\rangle$ and $|2\rangle$ to $|8\rangle$. In binary, $U^2|0001\rangle = |0100\rangle$, and $U^2|0010\rangle = |1000\rangle$. So we need gates that transform $|0001\rangle$ to $|0100\rangle$, and $|0010\rangle$ to $|1000\rangle$. Changing $|0001\rangle$ to $|0100\rangle$ is achieved by a SWAP, on the top qubit and the third qubit from the top. Similarly, changing $|0010\rangle$ to $|1000\rangle$ is achieved by a SWAP on the second and fourth qubits from the top. The SWAPs are controlled since U^2 is controlled, by the middle qubit of the eigenvalue register. The circuit so far is shown in Fig. 10.15.

The first SWAP changes $|0001\rangle$ without changing $|0010\rangle$, and the second SWAP does the reverse, so we don't get any undesired changes. We cannot replace a controlled SWAP with two CNOTs. For example, if we replaced the first controlled SWAP with two CNOTs, these would correctly change $|0001\rangle$ to $|0100\rangle$, but they would incorrectly change $|0010\rangle$ to $|0111\rangle$.

After the controlled U^2, we have the controlled U^4. But we can show that U^4 doesn't do anything and can be neglected. Since U multiplies by 2mod15, U^4 multiplies by $2^4 = 16$mod15. Multiplication by 16mod15 doesn't do anything. For example, multiplying 1 by 16mod15 produces 16mod15 = 1, which

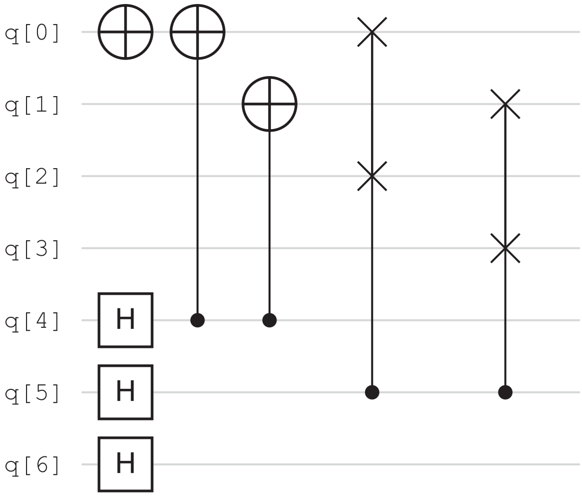


Figure 10.15. The circuit to determine the order of 2mod15, excluding the IQFT, created using IBM Quantum.

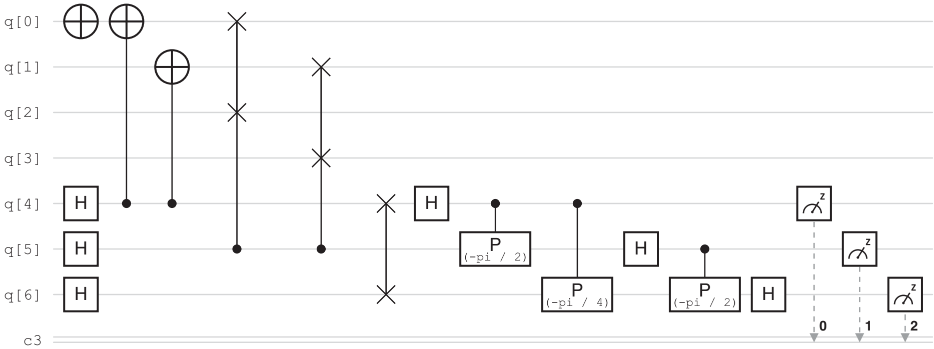


Figure 10.16. The circuit to determine the order of 2mod15, created using IBM Quantum. Originally published in Jed Brody and Kristen Gram, "Factoring 15 with a Remote Quantum Computer: A Complete Guide for Beginners," *European Journal of Physics*, April 2024, https://iopscience.iop.org/article/10.1088/1361-6404/ad32dc/pdf, under open license CC BY 4.0.

is what we started with. Similarly, multiplying 2 by 16mod15 produces 32mod15 = 2 (because 32 divided by 15 is 2 remainder 2), which again is what we started with. No matter what number we multiply by 16mod15, we get back the number we started with (because we're actually multiplying by 1 = 16mod15). So the circuit is completed with the IQFT and measurement of the eigenvalue register (Fig. 10.16).

The CNOTs, which form the controlled U, and the controlled SWAPs, which form the controlled U^2, are the only elements of the circuit that are specific to U. The rest of the circuit is the same for modular multiplication by 4, 7, 8, 11, and 13.

So let's figure out the controlled U and controlled U^2 for modular multiplication by 4, $U|J\rangle = |4J\text{mod}15\rangle$. The complete circuit is shown in Fig. 10.17 (this circuit, and the remaining circuits in this chapter, were constructed and run on IBM Quantum by my student, Kristen Gram).

Again, the controlled U consists of two CNOT gates, which now turn $|1\rangle = |0001\rangle$ into $|4\rangle = |0100\rangle$. For this U, U^2 doesn't do anything. U is multiplication by 4mod15, so U^2 is multiplication by $4^2 = 16$mod15. We already saw the multiplication by 16mod15 doesn't do anything.

Next, the circuit for $U|J\rangle = |7J\text{mod}15\rangle$ is shown in Fig. 10.18. The CNOTs form the controlled U, which changes $|1\rangle = |0001\rangle$ into $|7\rangle = |0111\rangle$. The controlled U^2 needs to multiply either $|1\rangle$ or $|7\rangle$ by 7^2mod15. Since 7^2mod15 = 49mod15 = 4, multiplication by 7^2mod15 is the same as multiplication by 4mod15. So $|1\rangle$ needs to change to $|4\rangle$, and $|7\rangle$ needs to change to $|4\times 7\text{mod}15\rangle = |28\text{mod}15\rangle = |13\rangle$. The first controlled SWAP changes $|1\rangle = |0001\rangle$ to $|4\rangle = |0100\rangle$, and the second controlled SWAP changes $|7\rangle = |0111\rangle$ to $|13\rangle = |1101\rangle$.

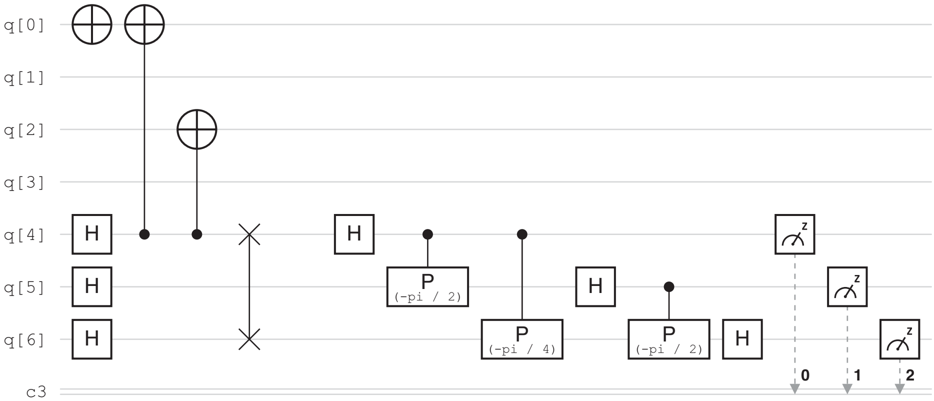


Figure 10.17. The circuit to determine the order of 4mod15, created using IBM Quantum. Originally published in Jed Brody and Kristen Gram, "Factoring 15 with a Remote Quantum Computer: A Complete Guide for Beginners," *European Journal of Physics*, April 2024, https://iopscience.iop.org/article/10.1088/1361-6404/ad32dc/pdf, under open license CC BY 4.0.

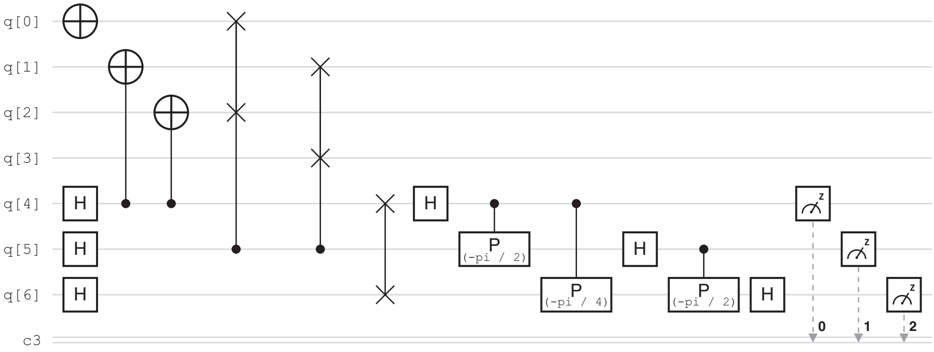


Figure 10.18. The circuit to determine the order of 7mod15, created using IBM Quantum. Originally published in Jed Brody and Kristen Gram, "Factoring 15 with a Remote Quantum Computer: A Complete Guide for Beginners," *European Journal of Physics*, April 2024, https://iopscience.iop.org/article/10.1088/1361-6404/ad32dc/pdf, under open license CC BY 4.0.

Next is the circuit for $U|J\rangle = |8J\text{mod}15\rangle$, shown in Fig. 10.19. The CNOTs form the controlled U, which changes $|1\rangle = |0001\rangle$ into $|8\rangle = |1000\rangle$. The controlled U^2 multiplies by $8^2\text{mod}15 = 64\text{mod}15 = 4\text{mod}15$. The controlled U^2 needs to be able to act on either $|1\rangle$ or $|8\rangle$. If it acts on $|1\rangle$, it changes $|1\rangle = |0001\rangle$ to $|4\rangle = |0100\rangle$. The first controlled SWAP implements this. If U^2 acts on $|8\rangle$,

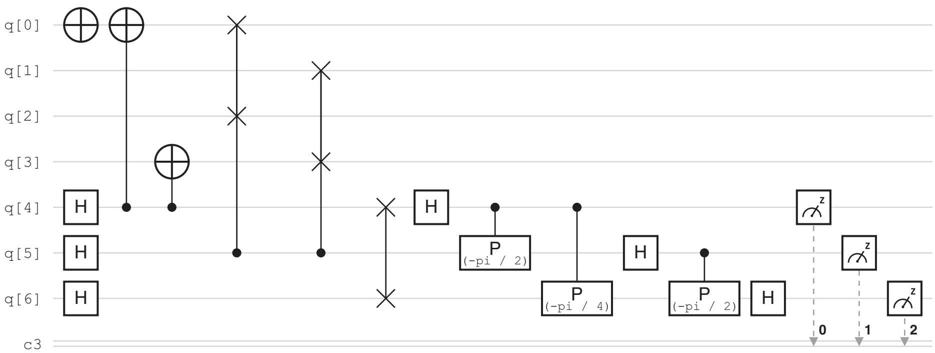


Figure 10.19. The circuit to determine the order of 8mod15, created using IBM Quantum. Originally published in Jed Brody and Kristen Gram, “Factoring 15 with a Remote Quantum Computer: A Complete Guide for Beginners,” *European Journal of Physics*, April 2024, https://iopscience.iop.org/article/10.1088/1361-6404/ad32dc/pdf, under open license CC BY 4.0.

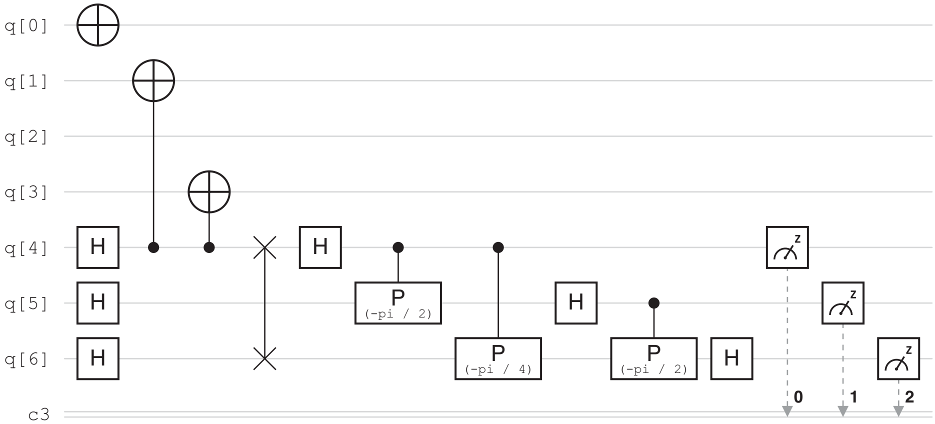


Figure 10.20. The circuit to determine the order of 11mod15, created using IBM Quantum. Originally published in Jed Brody and Kristen Gram, “Factoring 15 with a Remote Quantum Computer: A Complete Guide for Beginners,” *European Journal of Physics*, April 2024, https://iopscience.iop.org/article/10.1088/1361-6404/ad32dc/pdf, under open license CC BY 4.0.

it changes $|8\rangle = |1000\rangle$ to $|32\text{mod}15\rangle = |2\rangle = |0010\rangle$. The second controlled SWAP implements this.

Next up is $U|J\rangle = |11J\text{mod}15\rangle$, in Fig. 10.20. The CNOTs are the controlled U, which is able to change $|1\rangle = |0001\rangle$ to $|11\rangle = |1011\rangle$. Since $11^2\text{mod}15 = 121\text{mod}15 = 1$, U^2 effectively is a multiplication by 1, which can be ignored.

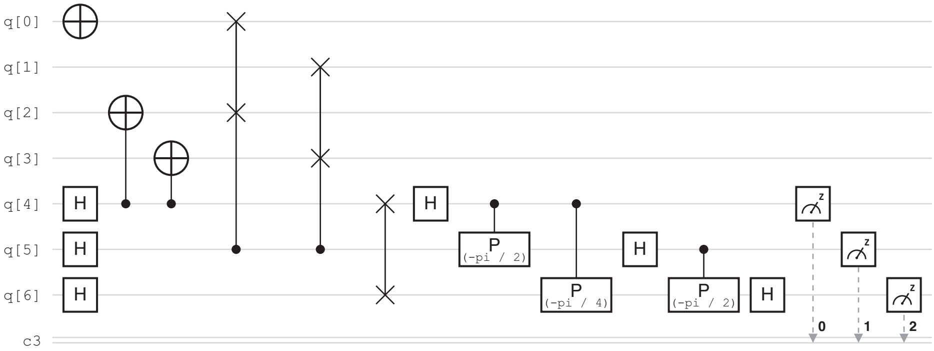


Figure 10.21. The circuit to determine the order of 13mod15, created using IBM Quantum. Originally published in Jed Brody and Kristen Gram, "Factoring 15 with a Remote Quantum Computer: A Complete Guide for Beginners," *European Journal of Physics*, April 2024, https://iopscience.iop.org/article/10.1088/1361-6404/ad32dc/pdf, under open license CC BY 4.0.

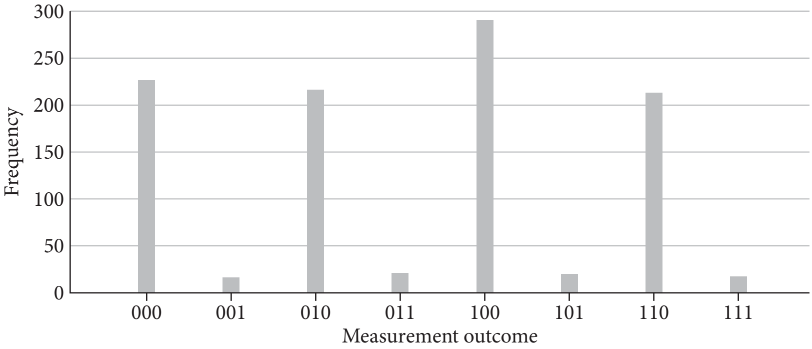


Figure 10.22. Results from the circuit in Fig. 10.16, created using IBM Quantum.

Last, we have $U|J\rangle = |13J\text{mod}15\rangle$ in Fig. 10.21. The controlled U is implemented by the CNOTs, which may change $|1\rangle = |0001\rangle$ to $|13\rangle = |1101\rangle$. The controlled U^2 needs to be able to multiply either 1 or 13 by 13^2mod15 = 169mod15 = 4mod15. So U^2 changes $|1\rangle = |0001\rangle$ to $|4\rangle = |0100\rangle$, which is achieved by the first controlled SWAP. U^2 changes $|13\rangle = |1101\rangle$ to $|52\text{mod}15\rangle = |7\rangle = |0111\rangle$, which is achieved by the second controlled SWAP.

We've just seen six order-finding circuits, designed to find the order r of 2mod15, 4mod15, 7mod15, 8mod15, 11mod15, and 13mod15. All six circuits were run on the IBM Quantum processor called ibm_perth. Figure 10.22 gives the results for 2mod15.

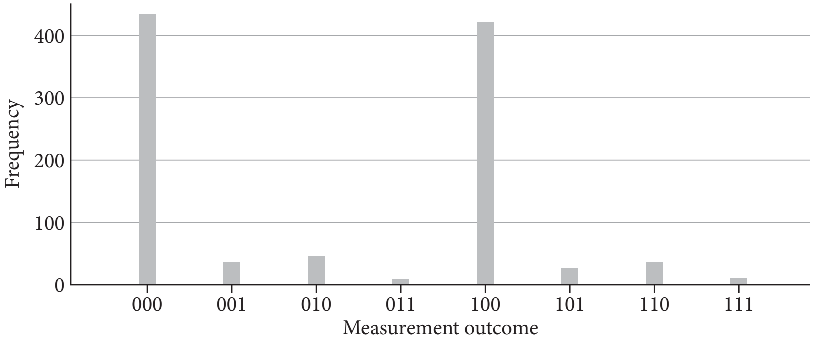


Figure 10.23. Results from the circuit in Fig. 10.17, created using IBM Quantum.

The dominant results are 000, 010, 100, and 110. (The small frequencies of the other results are due to experimental error, and do not occur in simulations.) So the possible values of j = s/r, where s is a whole number less than r, are 0.000 = 0/4, 0.010 = 1/4, 0.100 = 2/4, and 0.110 = 3/4. We conclude that r = 4 is the order of 2mod15. 4 indeed is the smallest r that gives 2^rmod15 = 1, so the circuit works! This seems like a lot of trouble over a little bit of arithmetic, but we've actually just implemented Shor's algorithm (the quantum part, at least)! In the next chapter, we'll see how the order of 2mod15 is related to factoring 15.

The measured results for 7mod15, 8mod15, and 13mod15 look much like the results for 2mod15. I won't show all the results because they all look the same: Neglecting error, the results are 000, 010, 100, and 110. So s/r again is 0/4, 1/4, 2/4, or 3/4, and r = 4. We can confirm that the order of 7mod15, 8mod15, and 13mod15 is 4: 7^4mod15 = 8^4mod15 = 13^4mod15 = 1.

The results are different for 4mod15 (Fig. 10.23). Now, the results are 000 and 100, neglecting error. So the possible values of s/r are 0.000 and 0.100 = 1/2. This means that r = 2. Indeed, 4^2mod15 = 1, so 2 is the order of 4mod15. Similar results are obtained for 11mod15, indicating that the order of 11mod15 is also 2.

Chapter 11

The Farthest Shor

Breaking the Internet

At long last, beneath moonlit clouds, we splash into the shallows and stagger to the Shor. The farthest Shor. Shor's factoring algorithm. We drop to our knees and kiss the damp sand, remembering the roaring perils of our voyage. We would like nothing more than to unstrap our dented armor and rest, but what can heroes do? We pluck the arrows out of our wooden shields and continue our adventure.

Shor's factoring algorithm is a quantum menace to an internet security protocol called *Rivest-Shamir-Adleman (RSA) encryption*. RSA encryption is based on the inability of classical computers to factor extremely large numbers. Rather, classical computers can factor extremely large numbers, but it takes years, and by then you have a new credit card number, so it doesn't matter if a hacker decrypts your old one.

RSA encryption is based on number theory, a branch of mathematics that seems absolutely useless, but it keeps all our data safe. Recall that AmodN is the remainder when A is divided by N. Here are some facts proven in number theory:

$$(A+B)\mathrm{mod}N=(A\mathrm{mod}N+B\mathrm{mod}N)\mathrm{mod}N. \tag{11.1}$$

For example, if we want to know the remainder when $37+64$ is divided by 15, it's a little easier to first find the remainders when 37 and 64 are separately divided by 15: $(37+64)\mathrm{mod}15=(37\mathrm{mod}15+64\mathrm{mod}15)\mathrm{mod}15=(7+4)\mathrm{mod}15=11$.

What about the remainder when 37×64 is divided by 15? We definitely don't want to do this in our head. Unless we use this rule:

$$(A\times B)\mathrm{mod}N=(A\mathrm{mod}N\times B\mathrm{mod}N)\mathrm{mod}N. \tag{11.2}$$

So $(37\times 64)\mathrm{mod}15=(37\mathrm{mod}15\times 64\mathrm{mod}15)\mathrm{mod}15=(7\times 4)\mathrm{mod}15=28\mathrm{mod}15=13$.

Here's another fact: For any prime number P and any number X that is not divisible by P,

$$X^{P-1}\mathrm{mod}P = 1. \qquad (11.3)$$

This is called *Fermat's little theorem*, and RSA encryption depends on it. Let's confirm that Eq. (11.3) works for P = 3 and X values of 2, 4, 5 and 7:

$$\begin{gathered} 2^2\mathrm{mod}3 = 1 \\ 4^2\mathrm{mod}3 = 1 \\ 5^2\mathrm{mod}3 = 1 \\ 7^2\mathrm{mod}3 = 1 \end{gathered}$$

All these equations are true.

Now, we're ready for RSA encryption. Suppose that Clytemnestra wants to receive information from Agamemnon, and she wants it to be encrypted so that an eavesdropper can't pry into their (rather lurid) personal life. Clytemnestra chooses two extremely large prime numbers, P and Q. Then she multiplies them together and calculates

$$N = PQ. \qquad (11.4)$$

Then she makes N public! She makes no attempt to hide it! Classical computers can't factor N (within a reasonable time frame), so only Clytemnestra knows P and Q, even after she broadcasts the number N.

Next, Clytemnestra chooses a number E that is less than (P – 1)(Q – 1), and that has no common factors with (P – 1)(Q – 1). She makes E public too. N and E are called the *public key*.

Then, Clytemnestra secretly calculates the D that satisfies

$$DE\mathrm{mod}[(P-1)(Q-1)] = 1. \qquad (11.5)$$

There is an efficient process to determine D, called the *extended Euclidean algorithm*. (Perhaps Clytemnestra learned it from Euclid himself.) But only Clytemnestra can determine D because only she knows P and Q. D is the private key.

Equation (11.5) says that when the product DE is divided by (P – 1)(Q – 1), the remainder is 1. In other words, DE is 1 plus some integer times (P – 1)(Q – 1). Let's call the integer L, so

$$DE = 1 + L(P-1)(Q-1). \qquad (11.6)$$

Meanwhile, Agamemnon composes his message to his wife, and he converts the message to some integer A (which must be less than N). Perhaps Agamemnon first represents each letter with five bits, so a = 00001, b = 00010, c = 00011, etc. So if Agamemnon's message is "cab," A = 000110000100010 = 2048 + 1024 + 32 + 2 = 3106. Although it takes some work to convert 3106 back to cab, A is not yet encrypted. Agamemnon's method to convert letters to numbers is perhaps too obvious. So Agamemnon computes the cipher

$$C = A^{E} \bmod N, \tag{11.7}$$

using the public key, N and E. Then he sends this to Clytemnestra.

Only Clytemnestra knows D, so only Clytemnestra can compute $C^{D} \bmod N$, which she decides to do. Using Eq. (11.7),

$$C^{D} \bmod N = A^{DE} \bmod N. \tag{11.8}$$

How can we simplify this? We need half a page of algebra. Here goes. We start by writing Fermat's little theorem, Eq. (11.3), choosing $X = A^{Q-1}$:

$$(A^{Q-1})^{P-1} \bmod P = A^{(Q-1)(P-1)} \bmod P = 1. \tag{11.9}$$

Eq. (11.9) says that the remainder is 1 when $A^{(Q-1)(P-1)}$ is divided by P. In other words, $A^{(Q-1)(P-1)}$ is 1 plus some integer times P. Let's call this integer U, so

$$A^{(Q-1)(P-1)} = 1 + UP. \tag{11.10}$$

Now let's write Fermat's little theorem again, substituting Q for P, and choosing $X = A^{P-1}$:

$$(A^{P-1})^{Q-1} \bmod Q = A^{(P-1)(Q-1)} \bmod Q = 1. \tag{11.11}$$

Equation (11.11) says that the remainder is 1 when $A^{(P-1)(Q-1)}$ is divided by Q, so $A^{(P-1)(Q-1)}$ is 1 plus some integer times Q. Choosing V for the integer,

$$A^{(P-1)(Q-1)} = 1 + VQ. \tag{11.12}$$

Combining Eqs. (11.10) and (11.12),

$$UP = VQ. \tag{11.13}$$

A multiple of P is a multiple of Q. Since P and Q are both prime numbers, neither one is a multiple of the other. The only way a multiple of P can be a multiple of Q is if both sides of Eq. (11.13) are in fact multiples of the product PQ. So

$$UP = VQ = WPQ, \tag{11.14}$$

where W is yet another integer. Substituting Eq. (11.14) into Eq. (11.12),

$$A^{(P-1)(Q-1)} = 1 + WPQ = 1 + WN \tag{11.15}$$

because N is PQ, the product of the two prime numbers.

Now, let's go all the way back to Eq. (11.8): $C^{D} \bmod N = A^{DE} \bmod N$. This is what we wanted to simplify. This is the value that only Clytemnestra can calculate because only she knows the private key D. Replacing DE with Eq. (11.6),

$$C^{D} \bmod N = A^{1 + L(P-1)(Q-1)} \bmod N. \tag{11.16}$$

On the right side, we have one factor of A, and L factors of $A^{(P-1)(Q-1)}$:

$$C^{D} \bmod N = A \times [A^{(P-1)(Q-1)}]^{L} \bmod N. \tag{11.17}$$

Next, we use Eq. (11.15):

$$C^{D}\text{mod}N = A \times [1+WN]^{L}\text{mod}N. \tag{11.18}$$

Then using Eq. (11.2), we can copy modN onto each factor:

$$C^{D}\text{mod}N = \{A\text{mod}N \times [(1+WN)\text{mod}N]^{L}\}\text{mod}N. \tag{11.19}$$

When $1+WN$ is divided by N, the result is W remainder 1. So $(1+WN)\text{mod}N = 1$, and Eq. (11.19) simplifies all the way to

$$C^{D}\text{mod}N = A\text{mod}N = A. \tag{11.20}$$

$A\text{mod}N = A$ because A is less than N. So when Clytemnestra uses the private key D to compute $C^{D}\text{mod}N$, she obtains Agamemnon's secret message A.

Let's take an example. Suppose Clytemnestra chooses $P=59$ and $Q=61$. If she wanted the encryption to be secure, P and Q would have to be much larger, but let's use small numbers in this example. Clytemnestra computes the product $N=3599$ and makes this public. We'll pretend that nobody knows how to factor N to determine P and Q. Clytemnestra then chooses $E=7$ and makes this public too.

Clytemnestra then uses the extended Euclidean algorithm to solve for D in $DE\text{mod}(P-1)(Q-1)=1$, so $7D\text{mod}3480=1$. She finds $D=2983$. (You can compute this at wolframalpha.com by typing "inverse of 7mod3480.")

Agamemnon has composed his message, $A=3106$. Then he computes the cipher $C=A^{E}\text{mod}N=3106^{7}\text{mod}3599=565$. (You can do this and all similar computations at wolframalpha.com too.) He transmits the cipher to Clytemnestra. If an eavesdropper obtained the cipher, they'd be unable to make heads or tails of it because they don't know the private key, D. Only Clytemnestra can use D to compute $C^{D}\text{mod}N=565^{2983}\text{mod}3599=3106$. She has successfully obtained Agamemnon's secret message.

If an eavesdropper is able to factor N into P and Q, then they are able to calculate D just as Clytemnestra does, and thereby decrypt the cipher just as Clytemnestra does. So the security of RSA encryption depends on the impossibility of factoring N.

This is how Shor's algorithm factors N:

1. Randomly choose some integer X that's less than N, and more than 1. Test whether X contains a factor of N. (There's an efficient way to do this, called *Euclid's algorithm*, related to the extended Euclidean algorithm. We'll learn Euclid's algorithm later.) If X contains a factor of N, congratulations! You've factored N! Otherwise, if X contains no factors of N, proceed with Shor's algorithm.
2. Use the order-finding circuit to find the order r of XmodN. This is the only step that requires a quantum computer. Recall that r is the smallest positive integer such that $X^{r}\text{mod}N=1$.

3. If the order r is odd, or if $X^{r/2}\text{mod}N = N - 1$, return to step 1 to pick a different X.
4. Otherwise, find the greatest common divisor of $X^{r/2} - 1$ and N, and of $X^{r/2} + 1$ and N. These are the factors of N.

Let's see why this process does indeed reveal the factors of N. It's breathtaking that order-finding has anything to do with factoring, but it does! Order-finding is actually just what we need to factor a product of two prime numbers.

Since r is the order of XmodN, $X^r\text{mod}N = 1$. Since X^r divided by N has a remainder of 1, $X^r - 1$ must be evenly divisible by N:

$$(X^r - 1)\text{mod}N = 0. \tag{11.21}$$

Recall the "difference of two squares formula," $(a + b)(a - b) = a^2 - b^2$. Using $a = X^{r/2}$ and $b = 1$, Eq. (11.21) becomes

$$(X^{r/2} + 1)(X^{r/2} - 1)\text{mod}N = 0. \tag{11.22}$$

This says that the product $(X^{r/2} + 1)(X^{r/2} - 1)$ is evenly divisible by N. There are three ways this can happen:

- $X^{r/2} + 1$ is divisible by N.
- $X^{r/2} - 1$ is divisible by N.
- Neither $X^{r/2} + 1$ nor $X^{r/2} - 1$ is divisible by N. Instead, $X^{r/2} + 1$ is a multiple of one prime factor of N, and $X^{r/2} - 1$ is a multiple of the other prime factor of N. For example, if N is 15, $X^{r/2}$ could be 169, as we'll see. Then $X^{r/2} + 1$ is 170, which is a multiple of 5, and $X^{r/2} - 1$ is 168, which is a multiple of 3. Thus $(X^{r/2} + 1)(X^{r/2} - 1)$ is a multiple of 15 because 5 is factor of the first term in parentheses, and 3 is a factor of the second.

We'll now show that the first two bullet points are impossible, so the third bullet point is the only possibility.

According to the first bullet point, $X^{r/2} + 1$ is divisible by N, or in other words, $X^{r/2} + 1$ is a multiple of N. This means that $X^{r/2}$ is 1 less than a multiple of N: $X^{r/2}$ is $N - 1$ or $2N - 1$ or $3N - 1$, etc. Notice that if any of these is divided by N, the remainder is $N - 1$. For example, if $3N - 1$ is divided by N, the result is 2N, remainder $N - 1$. So if $X^{r/2}$ is divided by N, the remainder is $N - 1$, so $X^{r/2}\text{mod}N = N - 1$. But this is specifically rejected by step 3 in Shor's algorithm. So $X^{r/2} + 1$ must not be divisible by N.

According to the second bullet point, $X^{r/2} - 1$ is a multiple of N, so $X^{r/2}$ is 1 more than a multiple of N. So $X^{r/2}$ is $N + 1$ or $2N + 1$ or $3N + 1$, etc. So when $X^{r/2}$ is divided by N, the remainder is 1: $X^{r/2}\text{mod}N = 1$. But this would mean that r/2, not r, is the order of XmodN: The order is the smallest exponent that gives a remainder of 1. Since r is the order of XmodN, $X^{r/2}\text{mod}N$ cannot be 1, so $X^{r/2} - 1$ cannot be a multiple by N.

We're left with only the third bullet point, which guarantees that $X^{r/2}+1$ and $X^{r/2}-1$ each contain a factor of N. So our goal is achieved: We've obtained the factors of N.

Now let's see how our data from the previous chapter factors 15. We used the order-finding circuit to find the order of 2mod15, 4mod15, 7mod15, 8mod15, 11mod15, and 13mod15. In fact, these are all six choices of Xmod15 that can be used to factor 15 with a quantum computer. X can't be 3, 5, 6, 9, or 10 because these numbers contain factors of N. (If we picked one of these values of X, Euclid's algorithm would immediately factor 15 without needing to use Shor's algorithm.) X can't be 14 because we would find that the order of 14mod15 is 2, and $14^{2/2}$mod15 = 14 = 15 − 1, which is specifically excluded by the third step in Shor's algorithm. If we tried to proceed with X = 14 and r = 2, the final step would give 14 + 1 = 15 and 14 − 1 = 13, which do not separately contain the two factors of 15.

Let's choose X = 2. We already have the necessary data. In the previous chapter, we used a quantum processor to find that the order or 2mod15 is r = 4. So the final step of Shor's algorithm gives $2^{4/2}+1=5$ and $2^{4/2}-1=3$, the two factors of 15. Immediate success!

Next, we choose X = 4. In the previous chapter, our data from a real quantum processor indicated that the order of 4mod15 is r = 2. So the final step of Shor's algorithm gives $4^{2/2}+1=5$ and $4^{2/2}-1=3$. Another immediate success!

Next, X = 7. Our order-finding circuit indicated that the order of 7mod15 is r = 4. So the final step of Shor's algorithm gives $7^{4/2}+1=50$ and $7^{4/2}-1=48$. We might notice that 50 contains a factor of 5, and 48 contains a factor of 3. But suppose we didn't spot this right away. In this case, we use Euclid's algorithm to extract the factors of 15 from 50 and 48.

First, we'll use Euclid's algorithm to find the greatest common divisor of 50 and 15. Euclid's algorithm is simply this: Replace the larger of the two numbers with the difference between them. Repeat this process until the two numbers are the same. So the steps of the process are the following:

50	15	
35	15	replacing 50 with 50 − 15
20	15	replacing 35 with 35 − 15
5	15	replacing 20 with 20 − 15
5	10	replacing 15 with 15 − 5
5	5	replacing 10 with 10 − 5

So we obtain 5, one of the factors of 15. We then apply Euclid's algorithm to extract the other factor of 15 from 48:

48	15	
33	15	replacing 48 with 48 − 15
18	15	replacing 33 with 33 − 15
3	15	replacing 18 with 18 − 15

3	12	replacing 15 with 15 – 3
3	9	replacing 12 with 12 – 3
3	6	replacing 9 with 9 – 3
3	3	replacing 6 with 6 – 3

Shor's algorithm again succeeds, though it requires some classical calculations after obtaining the order of 7mod15 from the quantum circuit.

Next, we'll try 8mod15. Our order-finding circuit indicated that the order of 8mod15 is r = 4. The final step of Shor's algorithm gives $8^{4/2} + 1 = 65$ and $8^{4/2} - 1 = 63$. Euclid's algorithm extracts the factors of 15 from these two numbers.

For 11mod15, our order-finding circuit gave us r = 2. The final step of Shor's algorithm gives $11^{2/2} + 1 = 12$ and $11^{2/2} - 1 = 10$, which separately contain the two factors of 15.

Last, our order-finding circuit determined that the order of 13mod15 is r = 4. Shor's algorithm gives $13^{4/2} + 1 = 170$ and $13^{4/2} - 1 = 168$. From these two numbers, Euclid's algorithm extracts the prime factors of 15.

When using Shor's algorithm to factor 15, we determine the order r of Xmod15, where r turns out to be either 2 or 4. Recall that the raw data from the order-finding circuits gives us s/r as binary fractions. So if s/r = 1/2, the binary fraction is 0.1, and if s/r = 1/4, the binary fraction is 0.01.

There are two kinds of fractions, those that can be represented by terminating decimals, and those that must be represented by repeating decimals. In base ten, 1/4 = 0.25 is a terminating decimal because 0.25 is exactly equal to 1/4. On the other hand, 1/3 = 0.333333 . . . is a repeating decimal because we need an infinite number of digits to represent 1/3 exactly. The general rule in base ten is that the decimal is terminating if the denominator contains factors of only 2 and 5, the factors of 10. (I learned this from my eleventh-grade math teacher. It blew my mind that I'd been unacquainted with such a basic fact of arithmetic.) So, for example, 50 = 2 × 5 × 5 contains factors of only 2 and 5, so 1/50 is a terminating decimal, 0.02. On the other hand, 30 contains a factor of 3, so 1/30 is a repeating decimal, 0.0333333 . . . .

In base two, fractions are terminating only if the denominator is a power of 2: 2, 4, 8, 16, etc. When we factored 15, we were lucky that the order r was always a power of 2. But what if r is any other number, like 6, which cannot be represented exactly as a binary fraction?

Suppose we're using Shor's algorithm to factor 21. Since 21 = 10101, we need five qubits to represent 21 in the eigenstate register. Let's also have five qubits in the eigenvalue register. So our measurements will yield five-bit numbers. Suppose we run the order-finding circuit to determine the order of 2mod21. When the eigenvalue register is measured, about 1/6 of the time we get 00000, about 1/6 of the time we get 00101, about 1/6 of the time we get 01011, about 1/6 of the time we get 10000, about 1/6 of the time we get 10101, and about 1/6 of the time we get 11011. (These are the results that would be expected in theory; I didn't actually run the circuits. Though you may if you like.) How do we extract the order of 2mod21 from these six results?

First, we recall that the results represent binary fractions, so we put 0 and a decimal point in front of each:

00000 indicates 0.00000 = 0
00101 indicates 0.00101 = 1/8 + 1/32 = 5/32
01011 indicates 0.01011 = 1/4 + 1/16 + 1/32 = 11/32
10000 indicates 0.10000 = 1/2
10101 indicates 0.10101 = 1/2 + 1/8 + 1/32 = 21/32
11011 indicates 0.11011 = 1/2 + 1/4 + 1/16 + 1/32 = 27/32

So our measured results are effectively 0, 5/32, 11/32, 1/2, 21/32, and 27/32. The biggest denominator is 32, but that is not the order of 2mod21. The biggest denominator is 32 simply because we have five fractional bits, and 0.00001 is 1/32. In fact, 5/32 and the other fractions may be only *approximations* to s/r, where r is what we want, the order of 2mod21 (and s is a whole number less than r).

So how do we determine what fraction is approximated by 5/32? The first step is to "flip and split" until all numerators are 1. Specifically, we first *flip* 5/32 by writing 1 over its reciprocal:

$$\frac{5}{32}=\frac{1}{\frac{32}{5}}$$

Then we *split* the improper fraction into an integer plus a proper fraction:

$$\frac{5}{32}=\frac{1}{\frac{32}{5}}=\frac{1}{6+\frac{2}{5}}$$

Then we repeat the process with the new proper fraction, 2/5. We flip:

$$\frac{5}{32}=\frac{1}{\frac{32}{5}}=\frac{1}{6+\frac{2}{5}}=\frac{1}{6+\frac{1}{\frac{5}{2}}}$$

Last, we split the improper fraction 5/2 into an integer plus a proper fraction:

$$\frac{5}{32}=\frac{1}{\frac{32}{5}}=\frac{1}{6+\frac{2}{5}}=\frac{1}{6+\frac{1}{\frac{5}{2}}}=\frac{1}{6+\frac{1}{2+\frac{1}{2}}}$$

We've completed the first step because all numerators are now 1. Fun, right? So far, we haven't done any approximations. Everything is exact so far.

To find what 5/32 might approximate, we drop what comes after the last +, which is 1/2:

$$\frac{5}{32} \approx \frac{1}{6+\frac{1}{2}} = \frac{1}{\frac{13}{2}} = \frac{2}{13}$$

So, possibly, 5/32 is an approximation to 2/13. So possibly, 2/13 = s/r. Does r = 13? We simply test whether 2^{13}mod21 is 1. It is not, so 2/13 is not the correct value. We proceed to further simplify the fraction by again dropping what comes after +:

$$\frac{5}{32} \approx \frac{1}{6+\frac{1}{2}} \approx \frac{1}{6}$$

Does 1/6 = s/r? Let's test if the order of 2mod21 is 6: 2^6mod21 = 1, so r = 6 is correct. At this point we can complete the final step of Shor's algorithm: $2^{6/2}+1=9$ and $2^{6/2}-1=7$, where 7 is one of factor 21, and 9 contains the other factor.

For more practice, let's see if we can factor 21 with the other expected results (11/32, 1/2, 21/32, and 27/32). For 11/32, we flip:

$$\frac{11}{32} = \frac{1}{\frac{32}{11}}$$

Then split:

$$\frac{11}{32} = \frac{1}{\frac{32}{11}} = \frac{1}{2+\frac{10}{11}}$$

Then flip 10/11:

$$\frac{11}{32} = \frac{1}{\frac{32}{11}} = \frac{1}{2+\frac{10}{11}} = \frac{1}{2+\frac{1}{\frac{11}{10}}}$$

Then split 11/10:

$$\frac{11}{32} = \frac{1}{\frac{32}{11}} = \frac{1}{2+\frac{10}{11}} = \frac{1}{2+\frac{1}{\frac{11}{10}}} = \frac{1}{2+\frac{1}{1+\frac{1}{10}}}$$

All numerators are 1, so the first process is complete.

Then, we drop the fraction after the final +:

$$\frac{11}{32} \approx \frac{1}{2+\frac{1}{1}} = \frac{1}{3}$$

Could s/r = 1/3? If we test r = 3, we find that 2^3mod21 is not 1, so the order of 2mod21 is not 3. But perhaps s/r is indeed 1/3, and s is not 1. The next simplest guess for s is s = 2, so r = 6, which is the correct order of 2mod21.

The next expected result is 1/2. We don't have to simplify this; it's already simple. Does s/r = 1/2? We first try r = 2, which doesn't work. If s = 2, r = 4, which still doesn't work. Next, we try s = 3, so r = 6, which is correct.

Next, 21/32. If we flip and split until all numerators are 1, we find

$$\frac{21}{32} = \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{10}}}}$$

Then we drop the fraction, 1/10, after the last +, to see what 21/32 approximates:

$$\frac{21}{32} \approx \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1}}} = \cfrac{1}{1 + \cfrac{1}{2}} = \cfrac{1}{\cfrac{3}{2}} = \frac{2}{3}$$

Does s/r = 2/3? If we try r = 3, that is not correct. But if we then try s = 2 so r = 6, that is correct.

Last, 27/32:

$$\frac{27}{32} = \cfrac{1}{1 + \cfrac{1}{5 + \cfrac{1}{2 + \cfrac{1}{2}}}}$$

We drop the 1/2 after the last +:

$$\frac{27}{32} \approx \cfrac{1}{1 + \cfrac{1}{5 + \cfrac{1}{2}}} = \cfrac{1}{1 + \cfrac{1}{\cfrac{11}{2}}} = \cfrac{1}{\cfrac{13}{11}} = \frac{11}{13}$$

Could s/r = 11/13? r = 13 is not correct. Setting s = 2, so r = 26, just makes it worse: The order of 2mod21 cannot be larger than 21 because there are only 21 possible remainders when a number is divided by 21. So we further simplify the fraction by dropping the 1/2 after the last plus sign in $\cfrac{1}{1 + \cfrac{1}{5 + \cfrac{1}{2}}}$:

$$\frac{27}{32} \approx \cfrac{1}{1 + \cfrac{1}{5}} = \cfrac{1}{\cfrac{6}{5}} = \frac{5}{6}$$

Could s/r = 5/6? Yes, r = 6 is correct.

The larger the number N that we're factoring, the more qubits we need. The eigenstate register needs enough qubits to represent the number N. If N is a huge number, hundreds of digits long, as in practical RSA encryption, we need a lot of qubits. The eigenvalue register needs even more qubits. To ensure that the eigenvalue is estimated with sufficient precision, the eigenvalue register should have enough qubits to count up to N^2. (In the preceding example, I used the same number of qubits in both registers, just to keep the arithmetic from getting too unwieldy.)

So that's Shor's algorithm. Why hasn't it broken the internet yet? So far, quantum computers are too small (too few qubits) and noisy (too much error). Potentially these shortcomings can be overcome, but by then we may use classical cryptosystems that are invulnerable to quantum attacks. If all else fails, quantum key distribution, from Chapter 1, is invulnerable to quantum attacks. So we've come full circle and have now examined both edges of this double-edged sword: Quantum information science menaces classical cryptography while also furnishing a viable alternative to it.

Experts debate whether quantum computers will revolutionize technology or remain an academic curiosity. On this question, I don't have much of an opinion, or even a preference. I do think it's likely that quantum computers will help chemists simulate molecules to engineer new drugs and nanomaterials. In any case, the joy of learning quantum computing is its own reward, even if there's little practical benefit beyond leveling up our nerdiness.

Chapter 12

How to Correct Those Flipping Errors

We've seen that real quantum processors do not quite live up to expectations. The results deviate from theory. As quantum circuits grow in size, the errors accumulate and snowball, until the results are no better than random numbers. And yet quantum circuits need to be large to confer any quantum advantage, because small quantum circuits can be efficiently simulated by classical computers. So the beast of quantum error must be vanquished. To that end, we have *quantum error correction*, a deep subject of active research. We will dip only one toe in these choppy waters.

The error in quantum results can arise in a variety of ways. There is readout error, which means that sometimes a $|0\rangle$ is measured as a 1 instead of 0, and vice versa. There is error in the quantum gates, which means that the X gate, for example, does not reliably convert $|0\rangle$ to $|1\rangle$. And sometimes the qubits just mutate, changing from $|0\rangle$, for example, to something else. The cause of this error is called *quantum decoherence*, which we will return to in our final chapter. We will see that quantum decoherence sheds light not only on naughty qubits but also on Schrödinger's famously unfortunate cat.

One kind of quantum error is called a *bit flip*, which means that $|0\rangle$ flips to $|1\rangle$, and $|1\rangle$ flips to $|0\rangle$. So if a qubit is in a general state $\alpha|0\rangle+\beta|1\rangle$, and it undergoes a bit flip, it becomes $\alpha|1\rangle+\beta|0\rangle$. Let's look at a circuit that both detects and corrects the bit flip (Fig. 12.1).

We want to protect the top qubit from a possible bit-flip error. The first step is to entangle it with two additional qubits. After the first two CNOT gates, the state of the three qubits is $\alpha|000\rangle+\beta|111\rangle$. This means that $|000\rangle$ represents $|0\rangle$, and $|111\rangle$ represents $|1\rangle$. We haven't cloned our qubit into the state $(\alpha|0\rangle+\beta|1\rangle)(\alpha|0\rangle+\beta|1\rangle)(\alpha|0\rangle+\beta|1\rangle)$; the no-cloning theorem forbids it. Still, we've effectively backed up our data twice. One of the three qubits may be corrupted by error, but the other two are likely to remain uncorrupted, as long as the error rate is reasonably low. The error symbol in the circuit diagram indicates that one of the three qubits may undergo a bit-flip error, but

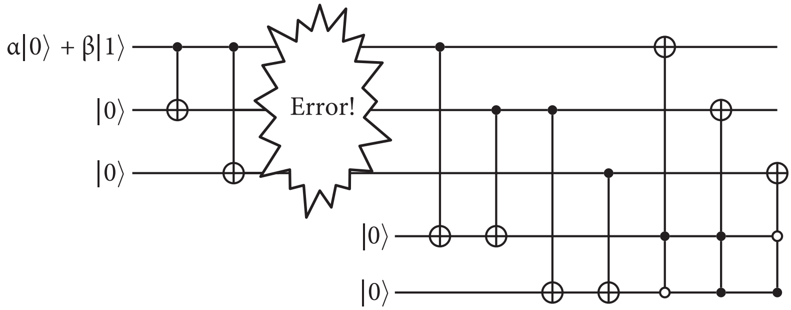


Figure 12.1. Correcting a bit-flip error, created using the Quantikz LaTeX package, including some of the code in the Quantikz tutorial by Alastair Kay (https://arxiv.org/pdf/1809.03842).

we don't know which one. The rest of the circuit detects the corrupted qubit and restores it to its rightful state.

The bottom two qubits determine which qubit, if any, is corrupted. Let's see how. If the top qubit undergoes a bit-flip error, the state of the top three qubits changes from $\alpha|000\rangle+\beta|111\rangle$ to $\alpha|001\rangle+\beta|110\rangle$. Adding in the bottom two qubits, the state is $\alpha|00001\rangle+\beta|00110\rangle$. The first CNOT (after "Error!") makes this $\alpha|01001\rangle+\beta|00110\rangle$, which the second CNOT turns into $\alpha|01001\rangle+\beta|01110\rangle$, which the third CNOT turns into $\alpha|01001\rangle+\beta|11110\rangle$, which the fourth CNOT turns into $\alpha|01001\rangle+\beta|01110\rangle=|01\rangle(\alpha|001\rangle+\beta|110\rangle)$. Evidently, when the top qubit undergoes a bit-flip error, the bottom two qubits go into the state $|01\rangle$. Similarly, we can show that a bit-flip error in the second-from-top qubit makes the bottom two qubits $|11\rangle$, and a bit-flip error in the third-from-top qubit makes the bottom two qubits $|10\rangle$. If there is no error, then the bottom two qubits remain $|00\rangle$. So the location of any bit-flip error is specified by the bottom two qubits.

The last three gates correct the error. These are doubly controlled NOT gates, controlled by the bottom two qubits. The open circle indicates that the control must be $|0\rangle$ to cause the NOT to act on the target. So when the target is on the top qubit, the NOT acts only when the bottom two qubits are $|01\rangle$. This is exactly the condition for an error on the top qubit, so $|01\rangle(\alpha|001\rangle+\beta|110\rangle)$ is corrected to $|01\rangle(\alpha|000\rangle+\beta|111\rangle)$. A similar process corrects errors on the next two qubits from the top.

A nearly identical circuit (Fig. 12.2) corrects a different error, called the *phase flip*, which changes $|+\rangle$ to $|-\rangle$. We begin with a qubit in state $\alpha|+\rangle+\beta|-\rangle$. We want to entangle it with two other qubits, to create the state $\alpha|{+}{+}{+}\rangle+\beta|{-}{-}{-}\rangle$. This is achieved by the gates before the Error: The first H transforms $\alpha|+\rangle+\beta|-\rangle$ to $\alpha|0\rangle+\beta|1\rangle$. Just as in the previous circuit, the two CNOT gates create the state $\alpha|000\rangle+\beta|111\rangle$. Then the three H gates just before the Error transform the state to $\alpha|{+}{+}{+}\rangle+\beta|{-}{-}{-}\rangle$.

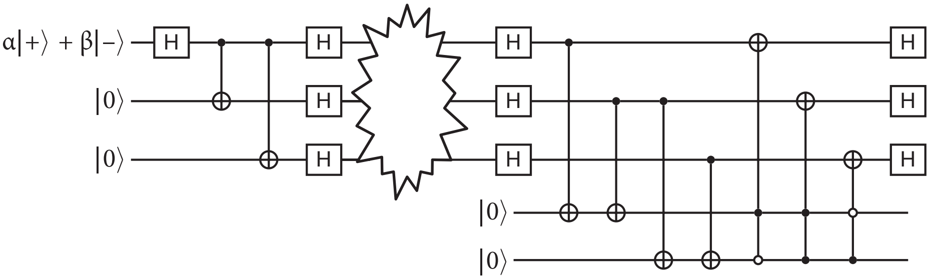


Figure 12.2. Correcting a phase-flip error, created using the Quantikz LaTeX package, including some of the code in the Quantikz tutorial by Alastair Kay (https://arxiv.org/pdf/1809.03842).

The Error symbol in this case represents a phase-flip error on one of the three qubits, at most. A phase-flip error on the top qubit, for example, makes the state $\alpha|{+}{+}{-}\rangle + \beta|{-}{-}{+}\rangle$. The subsequent H gates change $|+\rangle$ to $|0\rangle$ and $|-\rangle$ to $|1\rangle$, so the state becomes $\alpha|001\rangle + \beta|110\rangle$. The error correction circuit then functions exactly as before, locating the corrupted qubit and correcting it. The final three H gates turn $|0\rangle$ back into $|+\rangle$ and $|1\rangle$ back into $|-\rangle$.

We might wonder why quantum computers are so error-prone. Qubits are very fragile little things, susceptible to unwanted influences from their surroundings, in the process called *quantum decoherence*. To understand decoherence, we need the math that I've been putting off as long as possible. But now, it's time to descend to the deepest level of the dungeon, where the most valuable treasure is fiercely guarded. It's time to enter the matrix.

Some calculations will become easier once you add matrices to your toolbox (or spell book, to stick with the fantasy metaphor). Some calculations can't be done any other way.

Chapter 13

Enter the Matrix

A matrix is a rectangular grid of numbers. For example,

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

This is a square matrix: two rows and two columns. The four numbers in the matrix are called *matrix elements*. In some books, a matrix is enclosed by curved brackets, (), instead of square brackets, []. It's just an aesthetic choice.

If a matrix has only a single column, it's usually called a *column vector*. For example:

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

We can add two matrices of identical size and shape. We simply sum the corresponding elements. For example,

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 0+2 \\ 1+3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}.$$

If a matrix (or vector) is multiplied by a single number (often called a *scalar*), we simply multiply each element by that number. For example,

$$4\begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 4\times 2 \\ 4\times 3 \end{bmatrix} = \begin{bmatrix} 8 \\ 12 \end{bmatrix}.$$

We can combine scalar multiplication with matrix (or vector) addition. For example,

$$4\begin{bmatrix} 2 \\ 3 \end{bmatrix} + 5\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 4\times 2 \\ 4\times 3 \end{bmatrix} + \begin{bmatrix} 5\times 0 \\ 5\times 1 \end{bmatrix} = \begin{bmatrix} 8 \\ 12 \end{bmatrix} + \begin{bmatrix} 0 \\ 5 \end{bmatrix} = \begin{bmatrix} 8 \\ 17 \end{bmatrix}.$$

We often want to multiply two matrices. This is more complicated than addition; we do not simply multiply the corresponding elements. When we multiply two matrices, the order matters: AB does not necessarily equal BA. Mathematicians call this *noncommutativity*, but it's like putting on shoes and socks: If you put on your shoes first, and *then* your socks, the outcome is not the same as what you get by doing things in the usual order. In dressing oneself as well as in matrix multiplication, the order matters.

To perform the matrix multiplication AB, the numbers of *columns* of A must equal the number of *rows* of B. The product AB will have the number of *rows* of A and the number of *columns* of B. For example, if A and B are both two-by-two square matrices, the product AB is also a two-by-two square matrix. Taking another example, if A is a two-by-two matrix, and C is a column vector with two elements, then the product AC will be the same size and shape as C (a column vector with two elements).

Let's suppose

$$\mathrm{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

and

$$\mathrm{B} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}.$$

The product AB has four elements, which we need to determine:

$$\mathrm{AB} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}.$$

The **top left** element of AB is determined by the **top** row of A and the **left** column of B. The **first** element of the **top** row of A is multiplied by the **first** element of the **left** column of B: $1 \times 5 = 5$.

$$\mathrm{AB} = \begin{bmatrix} \mathbf{1} & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} \mathbf{5} & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} \mathbf{1} \times \mathbf{5} + ? & ? \\ ? & ? \end{bmatrix}.$$

Then, the **second** element of the **top** row of A is multiplied by the **second** element of the **left** column of B: $2 \times 7 = 14$.

$$\mathrm{AB} = \begin{bmatrix} 1 & \mathbf{2} \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 5 & 6 \\ \mathbf{7} & 8 \end{bmatrix} = \begin{bmatrix} 1 \times 5 + \mathbf{2} \times \mathbf{7} & ? \\ ? & ? \end{bmatrix}.$$

Then these two products are summed: $5 + 14 = 19$. This is the first element of the product AB:

$$\mathrm{AB} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 19 & ? \\ ? & ? \end{bmatrix}.$$

The **top right** element of AB is determined by the **top** row of A and the **right** column of B, following similar rules:

$$\mathrm{AB} = \begin{bmatrix} \mathbf{1} & \mathbf{2} \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 5 & \mathbf{6} \\ 7 & \mathbf{8} \end{bmatrix} = \begin{bmatrix} 19 & \mathbf{1} \times \mathbf{6} + \mathbf{2} \times \mathbf{8} = 22 \\ ? & ? \end{bmatrix}.$$

The **bottom left** element of AB is determined by the **bottom** row of A and the **left** column of B:

$$\mathrm{AB} = \begin{bmatrix} 1 & 2 \\ \mathbf{3} & \mathbf{4} \end{bmatrix} \begin{bmatrix} \mathbf{5} & 6 \\ \mathbf{7} & 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ \mathbf{3} \times \mathbf{5} + \mathbf{4} \times \mathbf{7} = 43 & ? \end{bmatrix}.$$

Last, the **bottom right** element of AB is determined by the **bottom** row of A and the **right** column of B:

$$\mathrm{AB} = \begin{bmatrix} 1 & 2 \\ \mathbf{3} & \mathbf{4} \end{bmatrix} \begin{bmatrix} 5 & \mathbf{6} \\ 7 & \mathbf{8} \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & \mathbf{3} \times \mathbf{6} + \mathbf{4} \times \mathbf{8} = 50 \end{bmatrix}.$$

Let's apply exactly the same rules to determine the product BA:

$$\mathrm{BA} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 5 \times 1 + 6 \times 3 = 23 & 5 \times 2 + 6 \times 4 = 34 \\ 7 \times 1 + 8 \times 3 = 31 & 7 \times 2 + 8 \times 4 = 46 \end{bmatrix}.$$

We see that AB does not equal BA.

Now let's choose

$$\mathrm{C} = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

and compute the product AC:

$$\mathrm{AC} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \end{bmatrix} = \begin{bmatrix} ? \\ ? \end{bmatrix}.$$

The top element of AC is determined by the top row of A and the one column of C. The rule is similar to what we used earlier. We work our way across the top row of A, while moving down the one column of C:

$$\mathrm{AC} = \begin{bmatrix} \mathbf{1} & \mathbf{2} \\ 3 & 4 \end{bmatrix} \begin{bmatrix} \mathbf{5} \\ \mathbf{6} \end{bmatrix} = \begin{bmatrix} \mathbf{1} \times \mathbf{5} + \mathbf{2} \times \mathbf{6} = 17 \\ ? \end{bmatrix}.$$

And the bottom element of AC is determined by the bottom row of A and the one column of C:

$$\mathrm{AC} = \begin{bmatrix} 1 & 2 \\ \mathbf{3} & \mathbf{4} \end{bmatrix} \begin{bmatrix} \mathbf{5} \\ \mathbf{6} \end{bmatrix} = \begin{bmatrix} 17 \\ \mathbf{3} \times \mathbf{5} + \mathbf{4} \times \mathbf{6} = 39 \end{bmatrix}.$$

Now we know how to multiply matrices. But so what? What does this have to do with quantum computing? Feast your eyes on this: We will define

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

and

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

This means that a generic qubit can be written

$$\alpha|0\rangle + \beta|1\rangle = \alpha\begin{bmatrix}1\\0\end{bmatrix} + \beta\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}\alpha\\0\end{bmatrix} + \begin{bmatrix}0\\\beta\end{bmatrix} = \begin{bmatrix}\alpha\\\beta\end{bmatrix}.$$

Every single-qubit gate can be written as a two-by-two matrix. For example,

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

Now the action of a gate on a qubit is simply matrix multiplication! We recall that $Z|0\rangle = |0\rangle$. Writing this as a matrix multiplication,

$$Z|0\rangle = \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\times 1 + 0\times 0\\0\times 1 + (-1)\times 0\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix} = |0\rangle.$$

It works! How about $Z|1\rangle = -|1\rangle$?

$$Z|1\rangle = \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}1\times 0 + 0\times 1\\0\times 0 + (-1)\times 1\end{bmatrix} = \begin{bmatrix}0\\-1\end{bmatrix} = -\begin{bmatrix}0\\1\end{bmatrix} = -|1\rangle.$$

Again, it works!

We notice that the first column of Z is $Z|0\rangle = |0\rangle$, and the second column of Z is $Z|1\rangle = -|1\rangle$. This is a general rule: The columns of any single-qubit gate U are $U|0\rangle$ and $U|1\rangle$. We can use this fact to figure out how to write the X gate as a (two-by-two square) matrix. The first column of X is

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

And the second column of X is

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

Putting these together, we find

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

As another example, let's figure out the matrix representation of H. The first column of H is

$$H|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix},$$

and the second column of H is

$$H|1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}.$$

Putting these together, we get

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

If we're given a gate as a matrix, we can reverse what we just did, to write the gate in terms of its actions on $|0\rangle$ and $|1\rangle$. For example, the "Square Root of NOT" gate is

$$\sqrt{X} = \frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix}.$$

It's called $\sqrt{X}$ because if you multiply it by itself (using the rules of matrix multiplication, and $i^2=-1$), you get X. Try it!

The first column of $\sqrt{X}$ is $\sqrt{X}|0\rangle$, so we can determine that

$$\sqrt{X}|0\rangle = \frac{1}{2}\begin{bmatrix} 1+i \\ 1-i \end{bmatrix} = \frac{1}{2}\left[(1+i)|0\rangle + (1-i)|1\rangle\right].$$

Similarly,

$$\sqrt{X}|1\rangle = \frac{1}{2}\begin{bmatrix} 1-i \\ 1+i \end{bmatrix} = \frac{1}{2}\left[(1-i)|0\rangle + (1+i)|1\rangle\right].$$

As a final example, let's write the identity gate I as a matrix. Since the first column of I is

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

and the second column of I is

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

we find the identity matrix

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Simply by using matrix multiplication, we can prove $Z^2=I$, $X^2=I$, and $H^2=I$. Matrix multiplication is a bit less laborious than proving the same thing with kets, as we did for H^2 in Chapter 2.

A column vector represents a ket. How about a row vector, like [1 0]? This row vector is represented by the symbol $\langle 0|$. Recall that *ket* is the second syllable in $\langle$brac|ket$\rangle$. Since $|0\rangle$ is a ket, $\langle 0|$ logically is a *brac*, though almost everybody calls it a *bra*. I call it a brac, which is extremely unusual but not completely unheard of. My last quantum mechanics course was taught by a professor who said *brac*. He didn't even give the disclaimer that almost no one else on earth says that. But he was the chair of the physics department and could get away with it. He could've called it a jock strap if he wanted to.

Generally, if $|\Psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$, then we define $\langle\Psi| = [\alpha^* \, \beta^*]$, where the asterisk, as usual, represents the complex conjugate. There's a good reason to use complex conjugates, which we will see soon.

Now that we have row vectors, we can multiply them with column vectors. For example, if we multiply $\langle\psi|$ by some column vector like $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, we have $\langle\psi||0\rangle$, where the double bar is usually abbreviated to a single bar: $\langle\psi|0\rangle$. This is called an *inner product*. Since

$$\langle\Psi|0\rangle = [\alpha^* \, \beta^*]\begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

we compute this product by first multiplying the left element of the row vector by the top element of the column vector: $(\alpha^*)(1)$. Then we add the product of the right element of the row vector and the bottom element of the column vector: $(\beta^*)(0)$. So the inner product is a scalar: $(\alpha^*)(1) + (\beta^*)(0) = \alpha^*$.

Notice that the inner product of a computational basis state with itself (with its brac equivalent, more precisely) is 1:

$$\langle 0|0\rangle = [1 \;\; 0]\begin{bmatrix} 1 \\ 0 \end{bmatrix} = 1 \times 1 + 0 \times 0 = 1,$$

and

$$\langle 1|1\rangle = [0 \;\; 1]\begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0 \times 0 + 1 \times 1 = 1.$$

But, the inner product of one basis state with another is 0:

$$\langle 0|1\rangle = [1 \;\; 0]\begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1 \times 0 + 0 \times 1 = 0.$$

The computational basis states are called *normalized* because the inner product of each state with itself is 1. They are called *orthogonal* (a generalization of perpendicular) because the inner product of one state with the other is 0. Because they are both normalized and orthogonal, they are called *orthonormal*.

Notice that if $|\Psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ is normalized, then

$$1 = \langle\Psi|\Psi\rangle = [\alpha^* \, \beta^*]\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \alpha^* \, \alpha + \beta^* \, \beta = |\alpha|^2 + |\beta|^2,$$

which is the condition that the probabilities of both measurement outcomes must sum to 1. Why are the probability amplitudes in the column vector made into complex conjugates in the row vector? So that normalization (the inner product is 1) is the same as having probabilities sum to 1.

We can form something called an *outer product* if we reverse the order of the brac and ket: $|0\rangle\langle\psi|$, for example. In this case, the product is a two-by-two matrix. The top element of the column vector multiplies the left element of the row vector to form the upper left element of the matrix. And so on for the other three elements of the matrix:

$$|0\rangle\langle\Psi| = \begin{bmatrix} 1 \\ 0 \end{bmatrix}[\alpha^* \; \beta^*] = \begin{bmatrix} 1\times\alpha^* & 1\times\beta^* \\ 0\times\alpha^* & 0\times\beta^* \end{bmatrix} = \begin{bmatrix} \alpha^* & \beta^* \\ 0 & 0 \end{bmatrix}$$

Any gate can be written as a sum of outer products. For example, since

$$|0\rangle\langle 1| = \begin{bmatrix} 1 \\ 0 \end{bmatrix}[0 \;\; 1] = \begin{bmatrix} 1\times 0 & 1\times 1 \\ 0\times 0 & 0\times 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

and

$$|1\rangle\langle 0| = \begin{bmatrix} 0 \\ 1 \end{bmatrix}[1 \;\; 0] = \begin{bmatrix} 0\times 1 & 0\times 0 \\ 1\times 1 & 1\times 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix},$$

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

Now we can see how outer and inner products combine in a remarkably convenient way, which is really why physicists use brac-ket notation. We know that $X|0\rangle = |1\rangle$. Let's see how to recover this, using outer and inner products:

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

Abbreviating the double bars to single bars, we recognize the inner products $\langle 1|0\rangle$ and $\langle 0|0\rangle$:

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

It works!

How do we represent a two-qubit state as a column vector? $|0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, where ⊗ indicates a *Kronecker product*. To compute a Kronecker product, each element of the first matrix (or vector) multiplies the entire second matrix (or vector). So:

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1\begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ 0\begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

Similar calculations show

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix},$$

and

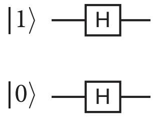
$$|11\rangle = |1\rangle \otimes |1\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

We can compute the Kronecker product of two matrices like $\mathrm{H} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.

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

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

Now we are able to analyze the following circuit three ways:



Method 1: the familiar method

The initial state is $|0\rangle \otimes |1\rangle$. We can compute the final state if we apply H to each qubit: $\mathrm{H}|0\rangle \otimes \mathrm{H}|1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) = \tfrac{1}{2}\left(|0\rangle + |1\rangle\right)\left(|0\rangle - |1\rangle\right)$ $= \tfrac{1}{2}\left(|00\rangle - |01\rangle + |10\rangle - |11\rangle\right)$. For comparison with the other two methods, let's write this result as a column vector:

$$\frac{1}{2}\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}.$$

Method 2: a separate matrix for each qubit

Alternatively, we can use matrices to compute the final state of each qubit. $H|0\rangle \otimes H|1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, by matrix multiplication on each side of the ⊗. Now we can take the Kronecker product of the two column vectors:

$$\frac{1}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1\begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ 1\begin{bmatrix} 1 \\ -1 \end{bmatrix} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix},$$

which agrees with Method 1.

Method 3: a single matrix for the transformation of both qubits

The initial state is $|0\rangle \otimes |1\rangle$. We can use the ⊗ symbol with gates too, so the final state is $(H \otimes H)(|0\rangle \otimes |1\rangle)$. We've already computed these two Kronecker products. So $(H \otimes H)(|0\rangle \otimes |1\rangle) = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$.

How do we do this matrix multiplication? The result will be a column vector with four elements. To find the top element of the result, we use the top row of the four-by-four matrix, and the entire column vector that it multiplies. To find the second element of the result, we used the second row of the four-by-four matrix and the entire column vector. The pattern continues:

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

Our next miscellaneous matrix topic is called *unitarity*. If a qubit is in state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, we know that $|\alpha|^2 + |\beta|^2$ must equal 1. If a quantum gate acts on $|\psi\rangle$, the qubit may change to $\gamma|0\rangle + \delta|1\rangle$, but the probabilities must still sum to 1: $|\gamma|^2 + |\delta|^2 = 1$. All quantum gates must obey this rule, a kind of conservation of probability. Matrices that do this are called *unitary*.

It can be shown that every unitary matrix has another special property: its *inverse* equals its *conjugate transpose*. Recall that the inverse of a matrix U is written U^{-1}, and the product of a matrix and its inverse is the identity matrix: $UU^{-1}=I$. The conjugate transpose of U is written $U^\dagger$ and is defined like this:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^\dagger = \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix}.$$

To find the conjugate transpose of a matrix, you take the complex conjugate of every element, and then form the transpose: swap the upper right element with the lower left element. If the matrix is larger, the transpose is found by turning the top row into the left column, the second-from-top row into the second-from-left column, etc. This is a reflection about a diagonal line from upper left to lower right.

We can confirm that H, for example, is unitary. We recall that H is its own inverse, so

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

We see also that $H^\dagger = H$ because taking the complex conjugate of the (real) elements doesn't change anything, and swapping upper right with lower left doesn't change anything. Since $H^\dagger = H^{-1}$, H is unitary.

If you stop reading this chapter out of boredom, you will disappear without a trace. Yes, that's right. The final topic in this chapter is the *trace* of a square matrix. This is simply the sum of the diagonal elements, where diagonal is defined as the line from the upper left to the lower right. So, for example, the trace of $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is $\mathrm{Tr}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 2$, but $\mathrm{Tr}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = 0$. Incidentally, $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is called a *diagonal matrix*, and $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ is called an *anti-diagonal matrix*.

To find the trace of something like $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$, with a scalar factor out front, you can either pull the scalar outside of the trace and multiply by the scalar at the end:

$$\mathrm{Tr}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \frac{1}{\sqrt{2}}\mathrm{Tr}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \frac{1}{\sqrt{2}}2,$$

or you can put the scalar inside the matrix and then take the trace:

$$\mathrm{Tr}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \mathrm{Tr}\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{2}{\sqrt{2}}.$$

Chapter 14

Quantum Decoherence and the Infinitely More Than Nine Lives of Schrödinger's Cat

At an incredible discount, you've purchased a mixed bag of a hundred qubits: 50 are $|0\rangle$, and 50 are $|1\rangle$. You don't know which are which; that's why they were on sale. When you randomly select a qubit and measure it in the computational basis, you have a 50% chance of getting $|0\rangle$ and a 50% chance of getting $|1\rangle$.

Your bag of qubits superficially resembles a (more expensive) pure bag of 100 qubits that are all in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. In both cases, a measurement of one qubit is just as likely to yield $|0\rangle$ as $|1\rangle$. However, we can do an experiment to distinguish between the mixed bag and the pure bag. If we pull a qubit from the pure bag and apply the H gate, the state transforms from $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ to $|0\rangle$, and then a measurement is certain to yield $|0\rangle$. On the other hand, if we pull a qubit from the mixed bag and apply the H gate, we get either $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ or $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, and a measurement in either case is 50% likely to yield $|1\rangle$. We may have to pull several qubits, apply H to each, and measure them, before we're confident that we have the pure bag or the mixed bag: If we get $|0\rangle$ every time, we have the pure bag. If we get $|0\rangle$ about half the time, we have the mixed bag.

A qubit from the pure bag is said to be in a *pure state*. A qubit from the mixed bag is said to be in a *mixed state*. All the qubits in the first thirteen chapters have been in pure states. Only a pure state can be represented as a ket or the corresponding column vector. To represent a mixed state, we need something just a little more complicated: the *density matrix*.

If a qubit in a mixed state has probability p_0 of being $|0\rangle$ and a probability p_1 of being $|1\rangle$, the density matrix, ρ, is defined as

$$\rho = p_0|0\rangle\langle 0| + p_1|1\rangle\langle 1| = p_0\begin{bmatrix} 1 \\ 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \end{bmatrix} + p_1\begin{bmatrix} 0 \\ 1 \end{bmatrix}\begin{bmatrix} 0 & 1 \end{bmatrix}$$
$$= p_0\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + p_1\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} p_0 & 0 \\ 0 & p_1 \end{bmatrix}.$$

In our example, $p_0 = p_1 = ½$, so

$$\rho = \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The density matrix of this mixed state is diagonal.

A density matrix is versatile enough to represent a pure state too. The density matrix of a pure state $|\psi\rangle$ is

$$\rho = |\psi\rangle\langle\psi|.$$

In our example, $|\Psi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$, so

$$\rho = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

We see that the density matrix of the pure state is not diagonal.

The *purity* of a state is defined as $\mathrm{Tr}(\rho^2)$. The purity of a pure state is 1, and the purity of a mixed state is less than 1. The purity of the pure state is our example is

$$\mathrm{Tr}\left(\frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\right)^2 = \frac{1}{4}\mathrm{Tr}\left(\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\right) = \frac{1}{4}\mathrm{Tr}\left(\begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right) = 1$$

as promised. The purity of the mixed state in our example is

$$\mathrm{Tr}\left(\frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)^2 = \frac{1}{4}\mathrm{Tr}\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = \frac{1}{4}\mathrm{Tr}\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = \frac{1}{2}.$$

This mixed state has a purity of ½, which is the smallest possible purity for a single qubit. A state with the smallest possible purity is said to be a *completely mixed state*, and the probability of measuring a result in *any* basis is ½. We already saw that when measuring the mixed state in the computational basis, the probability of either $|0\rangle$ or $|1\rangle$ is ½. If we measure instead in the x basis, the probability of either $|+\rangle$ or $|-\rangle$ is still ½, regardless of whether the qubit was $|0\rangle$ or $|1\rangle$ when initially pulled out of the mixed bag. (This implies that pulling a qubit from the mixed bag is effectively a measurement in the computational basis. I think it's okay to think of it this way, though really, before an actual measurement, the proper way to describe the mixed-state qubit is with its density matrix.)

Next, let's look at the density matrix for an entangled two-qubit state: $|\Psi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|1\rangle\right)$. I will add L and R subscripts to emphasize the Left and Right qubits: $|\Psi\rangle = \frac{1}{\sqrt{2}}\left(|0_L\rangle|0_R\rangle + |1_L\rangle|1_R\rangle\right)$. The density matrix is

$$\rho = |\Psi\rangle\langle\Psi| = \frac{1}{2}\left(|0_L\rangle|0_R\rangle + |1_L\rangle|1_R\rangle\right)\left(\langle 0_L|\langle 0_R| + \langle 1_L|\langle 1_R|\right).$$

If you like, you can write this as a four-by-four matrix and confirm that $\mathrm{Tr}(\rho^2) = 1$; this is a pure state that we studied in earlier chapters.

But what if we measure only one of the two qubits and ignore the other? What if we don't even have access to the second qubit? Maybe we misplaced it, but we know it's entangled with the qubit we've retained. Is there a way to represent the density matrix of the one remaining qubit?

Yes! We can define a *reduced density matrix* for one of the two entangled qubits. The reduced density matrix for the qubit on the left is defined as

$$\rho_L = \langle 0_R|\rho|0_R\rangle + \langle 1_R|\rho|1_R\rangle.$$

A brac with an R subscript forms an inner product with a ket with an R subscript. The bracs and kets with R subscripts do not interact with bracs and kets with L subscripts; the symbols with different subscripts pass through each other like ghosts (or like scalars).

So the reduced density matrix for the qubit on the left is

$$\begin{aligned}\rho_L &= \frac{1}{2}\langle 0_R|\left(|0_L\rangle|0_R\rangle + |1_L\rangle|1_R\rangle\right)\left(\langle 0_L|\langle 0_R| + \langle 1_L|\langle 1_R|\right)|0_R\rangle \\ &\quad + \frac{1}{2}\langle 1_R|\left(|0_L\rangle|0_R\rangle + |1_L\rangle|1_R\rangle\right)\left(\langle 0_L|\langle 0_R| + \langle 1_L|\langle 1_R|\right)|1_R\rangle.\end{aligned}$$

In the top line, the $\langle 0_R|$ on the left gets distributed to form inner products with both $|0_R\rangle$ and $|1_R\rangle$. The $|0_R\rangle$ on the far right of the top line forms inner products with both $\langle 0_R|$ and $\langle 1_R|$. Something very similar happens on the bottom line. So

$$\begin{aligned}\rho_L &= \frac{1}{2}\left(|0_L\rangle\langle 0_R|0_R\rangle + |1_L\rangle\langle 0_R|1_R\rangle\right)\left(\langle 0_L|\langle 0_R|0_R\rangle + \langle 1_L|\langle 1_R|0_R\rangle\right) \\ &\quad + \frac{1}{2}\left(|0_L\rangle\langle 1_R|0_R\rangle + |1_L\rangle\langle 1_R|1_R\rangle\right)\left(\langle 0_L|\langle 0_R|1_R\rangle + \langle 1_L|\langle 1_R|1_R\rangle\right).\end{aligned}$$

Using $\langle 0_R|0_R\rangle = \langle 1_R|1_R\rangle = 1$ and $\langle 0_R|1_R\rangle = \langle 1_R|0_R\rangle = 0$, the usual single-qubit orthonormality,

$$\rho_L = \frac{1}{2}\left(|0_L\rangle\langle 0_L| + |1_L\rangle\langle 1_L|\right) = \frac{1}{2}\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\begin{bmatrix} 0 & 1 \end{bmatrix}\right) = \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

We see that this is exactly the completely mixed state of a single qubit. The state of *both* qubits, combined, must be pure: $|\Psi\rangle = \frac{1}{\sqrt{2}}\left(|0_L\rangle|0_R\rangle + |1_L\rangle|1_R\rangle\right)$ is a pure state, as is any state that can be written in terms of kets. But one of these two qubits, measured alone, behaves as a completely mixed state: When measured in *any* basis, the probability of either result is ½.

For example, if only the left qubit of $\frac{1}{\sqrt{2}}\left(|0_L\rangle|0_R\rangle + |1_L\rangle|1_R\rangle\right)$ is measured in the computational basis, it's clear that there's a 50% chance of obtaining either $|0_L\rangle$ or $|1_L\rangle$. What if we measure in the x basis? We showed in Chapter 5 that $\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|+\rangle|+\rangle + |-\rangle|-\rangle\right)$. We can add L and R subscripts if we like: $\frac{1}{\sqrt{2}}\left(|0_L\rangle|0_R\rangle + |1_L\rangle|1_R\rangle\right) = \frac{1}{\sqrt{2}}\left(|+_L\rangle|+_R\rangle + |-_L\rangle|-_R\rangle\right)$. So there's a 50% chance of either $|+_L\rangle$ or $|-_L\rangle$ when the qubit on the left is measured in the x basis. No matter what basis we choose, there's a 50% chance of either result when measuring just one qubit in this entangled pair.

Now we can introduce *quantum decoherence*, the gale that topples our fragile quantum states. We can carefully prepare a qubit the pure state $\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$ for use in a quantum algorithm. But the qubit soon interacts uncontrollably with its surrounding environment: air molecules and photons, for example. The qubit becomes entangled with its environment. For simplicity, let's pretend there are two possible states of the qubit's environment, $|0_E\rangle$ and $|1_E\rangle$. Through the interaction of the qubit and its environment, if the qubit is measured in state $|0\rangle$, the environment is in state $|0_E\rangle$; if the qubit is found in state $|1\rangle$, the environment is in state $|1_E\rangle$. So the entangled state of the qubit and its environment is $\frac{1}{\sqrt{2}}\left(|0\rangle|0_E\rangle + |1\rangle|1_E\rangle\right)$.

This is exactly like the entangled two-qubit state we just looked at. The reduced density matrix of the original qubit is thus $\rho = \frac{1}{2}\left(|0\rangle\langle 0| + |1\rangle\langle 1|\right)$, and the original qubit is effectively in a completely mixed state. It no longer has the desired pure state $|+\rangle$. It is no longer in a well-defined superposition at all. Instead, a measurement in any basis gives a random result. A qubit in a completely mixed state is worthless for quantum measurements, unless our goal is random number generation. (If the qubit were entangled with another qubit, we could apply gates to disentangle them and achieve a pure state before measurement. However, a qubit entangled with the environment is mired in a mixed state because we can't reliably manipulate the environment.)

So everything we've studied in the whole entire book is undermined by quantum decoherence, which researchers are assiduously working to mitigate. On the other hand, quantum decoherence sheds some light on Schrödinger's cat, the most famous thought-experiment in the history of physics. If you search the internet, you can find nerds debating whether quantum decoherence solves the mystery of Schrödinger's cat. It's adorable.

The tale has oft been told. An innocent cat (what other kind is there?) is locked in a box with some radioactive material. But that's actually not the harmful part. The radioactive material is only 50% likely to undergo a single nuclear decay that will be detected by a Geiger counter. If radioactive decay is detected, the Geiger counter electronically triggers the release of poison gas. *That's* the harmful part.

Radioactive decay is a quantum process, so the state of the radioactive material can be written as a quantum superposition: $\frac{1}{\sqrt{2}}\left(|\text{decay}\rangle + |\text{no decay}\rangle\right)$. This seems to imply that the cat is also in a quantum superposition, $\frac{1}{\sqrt{2}}\left(|\text{dead}\rangle + |\text{alive}\rangle\right)$. Schrödinger is pointing out the absurdity of a cat that is in a superposition of life and death. Something must be wrong with our mathematical expression.

What's wrong, perhaps, is that we didn't account for entanglement with the environment. Applying our simplified model, the combined state of the cat and its environment is $\frac{1}{\sqrt{2}}\left(|\text{dead}\rangle|0_E\rangle + |\text{alive}\rangle|1_E\rangle\right)$. The cat is thus in a completely mixed state, $\rho = \frac{1}{2}\left(|\text{dead}\rangle\langle\text{dead}| + |\text{alive}\rangle\langle\text{alive}|\right)$, not a superposition at all. There's never any possibility of the dead cat interacting with the live version of itself.

And yet. The combined state of the cat and its environment remains a pure state, a superposition. So some people say that quantum decoherence must be combined with the many worlds interpretation. The combined state of the cat and its environment forever remains a pure state, though the $|\text{dead}\rangle$ and $|\text{alive}\rangle$ terms in the superposition can't interact with each other. Instead, they specify two noninteracting branches of the multiverse. In this view, the total quantum state of the whole entire multiverse remains always a pure state, a superposition of everything that is physically possible, separated into noninteracting branches.

But how is a branching multiverse consistent with *probabilities* of specific measurements? And what determines which branch we end up in? And if a single universal state vector is the ultimate reality, am I the totality of every possible version of myself, rather than the relatively tiny version in this branch? Have we rediscovered the ancient wisdom that the small self is an illusion? In one branch of the multiverse, *you* wrote this book, and you're wondering why you couldn't come up with a more satisfying finale.

Acknowledgments

Thanks to Jermey Matthews for the initial encouragement to write this book.

Thanks to Hallie Stebbins, Chloe Coy, Wanda España, William Pagdatoon, Elizabeth Blazejewski, Mark Bellis, Jennifer Harris, and the rest of the team at Princeton University Press. Behind the scenes, dedicated specialists expertly perform myriad tasks that even the author is scarcely aware of.

Thanks to the reviewers for conscientious and really helpful feedback. Thanks to my student Kristen Gram for reviewing a draft and for contributing to the Shor chapter. I alone am responsible for all uncorrected errors and dreadful puns.

The Bell inequality experiment was my initiation into quantum information science. So thanks to Kiko Galvez for training me how to set up the experiment. Thanks to the Reichert Foundation for providing funding to set up the experiment in my own teaching lab. Thanks to the Advanced Laboratory Physics Association, which coordinated both the training and funding.

Thanks to Stefan Boettcher for assigning me to teach quantum computing, which opened a quantum gate to new adventures.

Thanks to IBM Quantum for an amazing, free resource.

I also need to acknowledge the forty literary agents who rejected my science-fiction novel. Out of frustration over my unwanted fiction, I wrote a nonfiction book about quantum computing. The energy had to go somewhere. It was therapeutic. So thanks to you, faithful reader, for being part of it.

Appendix A

Further Reading

There are lots of books about quantum computing and quantum information science. Here are some of my favorites.

Thomas Wong, *Introduction to Classical and Quantum Computing*.

This is easily my favorite quantum computing textbook. Most other quantum computing textbooks contain sentences or whole chapters that I don't really understand. But if I concentrate, I can understand every one of Wong's statements. He is clear, concise, and logical, and he tells you if he decides to skip steps in mathematical derivations.

Wong says the only prerequisite is precalculus, and he teaches you linear algebra. But if you want a more detailed introduction to matrix arithmetic, you might want to first study *Quantum Computing for Everyone*.

And what's more, Wong's ebook is free: https://www.thomaswong.net/introduction-to-classical-and-quantum-computing-1e4p.pdf. Thank you, Thomas Wong. You are a quantum superhero. I'd send you a fruit basket, but I don't know if you like fruit. Or baskets.

Franklin de Lima Marquezino, Renato Portugal, and Carlile Lavor, *A Primer on Quantum Computing*.

This is my second favorite quantum computing textbook, and the only reason I know about it is that I almost co-taught quantum computing with the lead author. Franklin is a friend and collaborator of my former department chair, and he was planning to visit Emory in spring 2020. The pandemic scuttled those plans, and I ended up teaching the course alone.

A Primer on Quantum Computing is very short and doesn't cover a few fun topics like quantum teleportation, quantum key distribution, and Bell inequalities. However, it includes a lot of details about specific examples of Grover's algorithm and Shor's algorithm. I relied heavily on these detailed examples to learn and teach the material, and you don't find this level of detail in many textbooks, including Wong's.

> Bernard Zygelman, *A First Introduction to Quantum Computing and Information.*

This is another obscure textbook that I like better than some of the more standard ones. There are a few chapters about quantum device physics: how to engineer qubits and quantum gates. Few books include this topic, which requires some serious background in quantum mechanics.

> Chris Bernhardt, *Quantum Computing for Everyone.*

Notice how authors are tripping over themselves to make their titles as unintimidating as possible. Bernhardt's book requires only precalculus as a prerequisite, and he teaches matrix algebra in depth, but doesn't really cover Shor's algorithm at all. So his book is complementary to mine. In case you skipped my whole book to get to "Further Reading": I don't have any matrices until the last two chapters, but I have three full chapters on Shor's algorithm and its scaffolding (quantum Fourier transforms and quantum phase estimation).

> Michael Nielsen and Isaac Chuang, *Quantum Computing and Quantum Information.*

The standard text. Best approached if you already have a degree in physics, math, computer science, or preferably all three.

The following books are about the meaning of quantum physics. The meaning or philosophical interpretation of quantum physics is sometimes called "foundations of physics," which, along with quantum computing, is a branch of quantum information science.

> David Deutsch, *The Fabric of Reality: The Science of Parallel Universes—and Its Implications.*

A pioneer of quantum computing discourses on his worldview, or rather, worlds view: He is a fearless champion of the many worlds interpretation of quantum physics. I find his arguments unpersuasive and his conclusions unsupported, but beware, they may be correct regardless. Maybe he saves his rigorous arguments for his technical papers that contain math. *The Fabric of Reality* is interesting in the way that science fiction is interesting, but I'm not sure what it has to do with reality. Or with fabric.

David Kaiser, *How the Hippies Saved Physics: Science, Counterculture, and the Quantum Revival.*

For decades, most physicists disdained serious inquiry into the philosophy of quantum physics. At a time when no one else cared about Bell inequalities, hippie physicists in Berkeley speculated about plausible and implausible consequences of quantum physics. Did they set the stage for recent advances in quantum information science? I love this book, though I'm neither a hippie (too uptight) nor a physicist (my PhD is in electrical and computer engineering).

Adam Becker, *What Is Real? The Unfinished Quest for the Meaning of Quantum Physics*.

So rigorously logical, you could use the narrative throughline as a straightedge. Becker embraces Einstein's position that there *is* an objective reality independent of observation.

Philip Ball, *Beyond Weird: Why Everything You Thought You Knew about Quantum Physics Is Different*.

This is a great book to read as a counterpoint to Becker's book. Becker and Ball have opposite opinions about many topics: Whether the many worlds interpretation is a leading contender or an untestable absurdity, whether it makes sense to speak of objective reality independent of observation, and whether Niels Bohr's legacy is institutionalized incoherence or piercing insight.

Tanya Bub and Jeffrey Bub, *Totally Random: Why Nobody Understands Quantum Mechanics (A Serious Comic on Entanglement)*.

This is literally a comic book! There are some inside jokes and thought-provoking insights if you already know a little about philosophical interpretations of quantum theory.

Jed Brody, *Quantum Entanglement*.

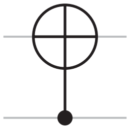
If, against all odds, against all logic and reason, against common sense and expert opinion, you enjoy my writing, then check out my earlier book. I needed to bulk it up and wanted to add a chapter about quantum computing, but I didn't know anything about quantum computing at the time. So instead I added a chapter about special relativity, which has almost nothing to do with the rest of the book. Much like this section of further reading. And that, my friends, is how it's done.

Appendix B

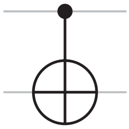
Table of Quantum Gates

Table B.1 lists the single-qubit gates used in this book (please ignore the matrix column if you haven't gotten to that chapter yet).

Any of these gates can be *controlled* such that it acts on a *target* qubit only when the *control* qubit is $|1\rangle$. For example, the controlled NOT,



applies a NOT to the target (the top qubit shown above) when the control (indicated by the dot) is $|1\rangle$. If the control is on the top and the target is on the bottom, the controlled NOT looks like this:



If there are two controls, the gate acts only when both controls are $|1\rangle$. For example, the Toffoli gate, the doubly controlled NOT,

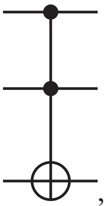
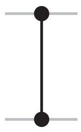

,

Table B.1

Gate	Symbol	Action on Basis States	Matrix
I, identity gate	$\boxed{I}$	$I\vert 0\rangle=\vert 0\rangle$ $I\vert 1\rangle=\vert 1\rangle$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
X, NOT gate	$\oplus$ or $\boxed{X}$	$X\vert 0\rangle=\vert 1\rangle$ $X\vert 1\rangle=\vert 0\rangle$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Y	$\boxed{Y}$	$Y\vert 0\rangle=i\vert 1\rangle$ $Y\vert 1\rangle=-i\vert 0\rangle$	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Z	$\boxed{Z}$	$Z\vert 0\rangle=\vert 0\rangle$ $Z\vert 1\rangle=-\vert 1\rangle$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
H, Hadamard gate	$\boxed{H}$	$H\vert 0\rangle=\frac{1}{\sqrt{2}}(\vert 0\rangle+\vert 1\rangle)$ $H\vert 1\rangle=\frac{1}{\sqrt{2}}(\vert 0\rangle-\vert 1\rangle)$	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
P(θ), phase gate	$\boxed{P(\theta)}$ or $\boxed{\begin{matrix} P \\ (\theta) \end{matrix}}$	$P(\theta)\vert 0\rangle=\vert 0\rangle$ $P(\theta)\vert 1\rangle=e^{i\theta}\vert 1\rangle$	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}$
$R_y(\theta)$, y-axis rotation	$\boxed{RY(\theta)}$ or $\boxed{\begin{matrix} RY \\ (\theta) \end{matrix}}$	$R_y(\theta)\vert 0\rangle=\cos(\theta/2)\vert 0\rangle$ $+\sin(\theta/2)\vert 1\rangle$ $R_y(\theta)\vert 1\rangle=-\sin(\theta/2)\vert 0\rangle$ $+\cos(\theta/2)\vert 1$	$\begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$

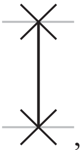
applies a NOT to the target when both controls are $|1\rangle$. Here, the target is the bottom qubit, but in general, any of the qubits may be the target.

The controlled-Z is sometimes represented by



We can imagine that either qubit is the control, and either is the target, because this gate affects the qubits only when both are $|1\rangle$.

A gate that always acts on two qubits is the SWAP gate,

,

which simply swaps the states of two qubits.

Appendix C

Exercises

1.1. If our secret message is 11001 and our key is 10101, what is the cipher?

1.2. If the cipher is 11001 and the key is 10101, what is the secret message?

1.3. Write $\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$ in terms of $|+\rangle$ and $|-\rangle$.

1.4. Write $\frac{1}{2}|+\rangle + \frac{\sqrt{3}}{2}|-\rangle$ in terms of $|0\rangle$ and $|1\rangle$.

1.5. The state of a qubit is $\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$. When it is measured in the computational basis, what are the probabilities of obtaining 0 and 1?

1.6. The state of a qubit is $\frac{1}{2}|0\rangle + x|1\rangle$. What is a possible value of x?

2.1. Determine $Z(\alpha|0\rangle + \beta|1\rangle)$.

2.2. Determine $H(\alpha|0\rangle + \beta|1\rangle)$.

2.3. Write the circuit equivalent to $ZH|1\rangle$ and determine the state at the end of the circuit.

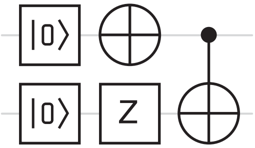
2.4. Write the circuit equivalent to $ZHZX|1\rangle$ and determine the state at the end of the circuit.

2.5. Write the circuit that transforms $|0\rangle$ into $|-\rangle$.

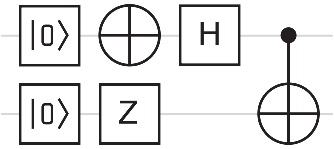
2.6. Write the circuit that transforms $|+\rangle$ into $|1\rangle$.

3.1. Use FOIL multiplication to write $\left[\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\right]\left[\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle\right]$ in terms of the four two-qubit computational basis states. What is the probability of measuring each possible result?

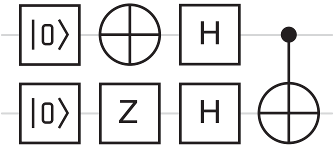
3.2. Determine the final state of the qubits in this circuit:



3.3. Determine the final state of the qubits in this circuit:



3.4. Determine the final state of the qubits in this circuit:



4.1. Suppose the bottom qubit in Fig. 4.1 is measured, but Odysseus never receives the measurement result, so the Z gate is never applied. How does this affect quantum teleportation? What is the probability that Odysseus's qubit attains the desired state, and what is his qubit's state if the teleportation fails?

4.2. Suppose the middle qubit in Fig. 4.1 is measured, but Odysseus never receives the measurement result, so the final NOT gate is never applied. How does this affect quantum teleportation? What is the probability that Odysseus's qubit attains the desired state, and what is his qubit's state if the teleportation fails?

4.3. If Penelope wants to send 11 through quantum dense coding, she replaces the question mark in Fig. 4.2 with a NOT gate and a Z gate. Does the order of the two gates matter? Why or why not?

6.1. Using Eqs. (6.3) through (6.5), show that $R_y(-\theta)|\theta\rangle = |0\rangle$.

6.2. In a CHSH experiment, what is <S> if the four measurement angles (α_1, α_2, β_1, and β_2) are all the same?

6.3. Write $|i\rangle$ and $|-i\rangle$ in terms of $|+\rangle$ and $|-\rangle$.

7.1. Convert 23_{10} to binary.

7.2. Convert 10101_2 to base ten.

7.3. Write out the quantum circuit to compute $2+3=5$. Determine and explain the final state of each qubit.

8.1. Modify Fig. 8.11 for $|G\rangle = |1\rangle = |01\rangle$.

8.2. Modify Fig. 8.12 for $|G\rangle = |5\rangle = |101\rangle$.

8.3. Modify Fig. 8.18 to multiply by −1 the name associated with phone number $|0\rangle|1\rangle$.

8.4. Figure 8.24 uses Grover's algorithm to solve $x+1=3$. Modify the circuit to solve $x+2=3$.

8.5. Modify Fig. 8.24 to solve $x+1=2$.

9.1. Go through Fig. 9.3 one gate a time and determine the state of the qubits after each gate. Assume the initial state is $|000\rangle$.

9.2. Determine the QFT of the two-qubit state $\frac{1}{\sqrt{2}}\left(|1\rangle+|3\rangle\right)$.

9.3. Determine the QFT of the three-qubit state $\frac{1}{\sqrt{2}}\left(|1\rangle+|5\rangle\right)$.

9.4. Determine the QFT of the three-qubit state $\frac{1}{\sqrt{2}}\left(|2\rangle+|6\rangle\right)$.

10.1. We found the eigenstates and eigenvalues of X by solving $X(\alpha|0\rangle+\beta|1\rangle) = \lambda(\alpha|0\rangle+\beta|1\rangle)$. Solve the equivalent problem for Y.

10.2. Modify Fig. 10.5 to find the eigenvalues of X. You need to change two things: You need to initialize the top qubit to an eigenstate of X, and you need to replace every controlled-Z with a controlled-X. What results do you expect?

10.3. Modify Fig. 10.5 to find the eigenvalues of Y.

10.4. Determine 13mod4, 16mod5, and 21mod8.

10.5. Determine the order of 2mod7 and 4mod7.

10.6. Redesign Figs. 10.16 through 10.21, with only two qubits in the eigenvalue register. What results are expected in each case?

11.1. If $N=3599$ and $E=7$, determine the cipher if the secret message is $A=2025$. Then decipher it using the factors of 3599.

11.2. Use Euclid's algorithm to find the greatest common divisor of 39 and 65.

12.1. Show how Fig. 12.1 corrects a bit-flip error in the second qubit from the top.

12.2. Show how Fig. 12.2 corrects a phase-flip error in the second qubit from the top.

13.1. Compute the products $\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} -2 & 2 \\ 2 & -2 \end{bmatrix}$, $\begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 0 & -1 \\ -3 & 0 \end{bmatrix}$, and $\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} -5 & 6 \\ 7 & -5 \end{bmatrix}$.

13.2. Compute the products $\begin{bmatrix} 2 & 2 \\ 3 & 3 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$, $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 2 \\ 1 \end{bmatrix}$, and $\begin{bmatrix} 3 & 3 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} -5 \\ 0 \end{bmatrix}$.

13.3. Using matrix multiplication, determine $HZ|+\rangle$, $ZH|0\rangle$, and $XHZ|-\rangle$.

13.4. Determine $\langle 0|+\rangle$, $\langle +|-\rangle$, and $\langle -|1\rangle$.

13.5. Determine the matrices $|0\rangle\langle 1|$, $|+\rangle\langle +|$, and $|-\rangle\langle 0|$.

13.6. Write H, Z, and I as sums of outer products, and show that the correct results are obtained when these expressions act on the computational basis states.

13.7. Determine the Kronecker products $|0\rangle \otimes |+\rangle$, $|+\rangle \otimes |+\rangle$, and $|+\rangle \otimes |0\rangle$.

13.8. Determine the Kronecker products X⊗X, H⊗I, and X⊗H. Let each four-by-four matrix multiply the column vector $|0\rangle \otimes |0\rangle$, and show that the same result is obtained when letting each gate act on a single qubit.

Index