Lecture slides can be found online at:

`https://github.com/glassnotes/Intro-QC-TRIUMF/summer-students-2019`

On your phone/laptop, go to

`menti.com`

and input the code

`87 83 08.`

It's not an actual quiz, and you don't have to put a real name!

Intro to Quantum Computing

TRIUMF summer student seminar #1

Olivia Di Matteo
Quantum Information Science Associate, TRIUMF

14 August 2019

Overview

Learning goals

Quantum computers will be a very important computational tool in the future. Now is the time to learn how to use them!

At a conceptual level, you'll be able to...

- Explain the motivation behind building quantum computers
- Describe the principles that give quantum computers their "source of power"
- Explain the idea of quantum advantage
- List the major technological players, the main physical implementations, and the successes and challenges of current-generation machines

Learning goals

Using a mixture of theory and hands-on activities, you'll...

- Perform computations and measurements on a single qubit
- Express quantum computations as quantum circuits
- Perform computations on multiple qubits
- Implement a circuit that *teleports* a single-qubit state

Motivation

Physical limitations

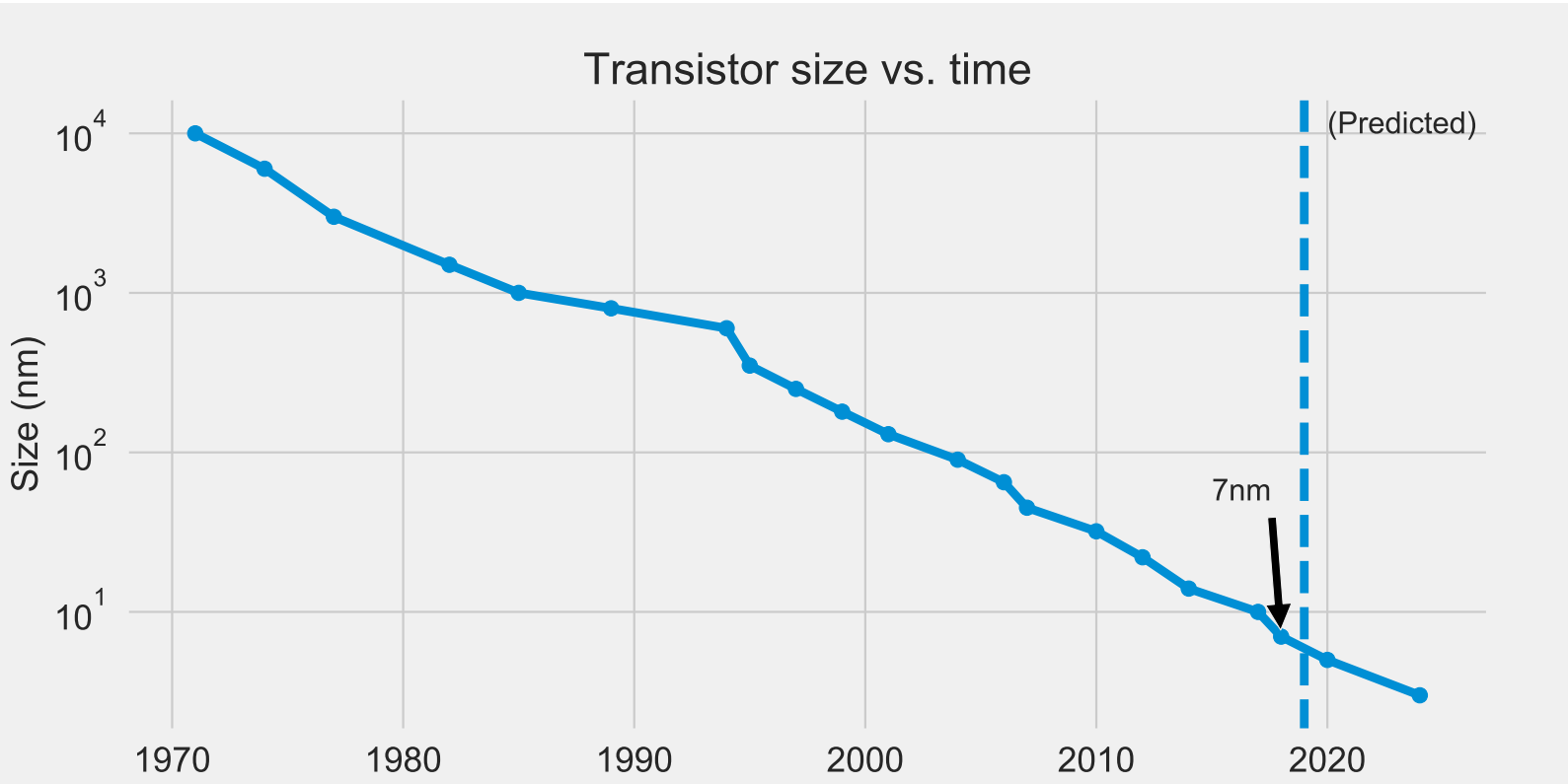
'Classical' computers are made more powerful by

- making smaller transistors
- putting more transistors onto a single chip
- using multiple processors in parallel

Moore's law suggests that every 18 months - 2 years the number of transistors on a chip will double, while the cost is halved.

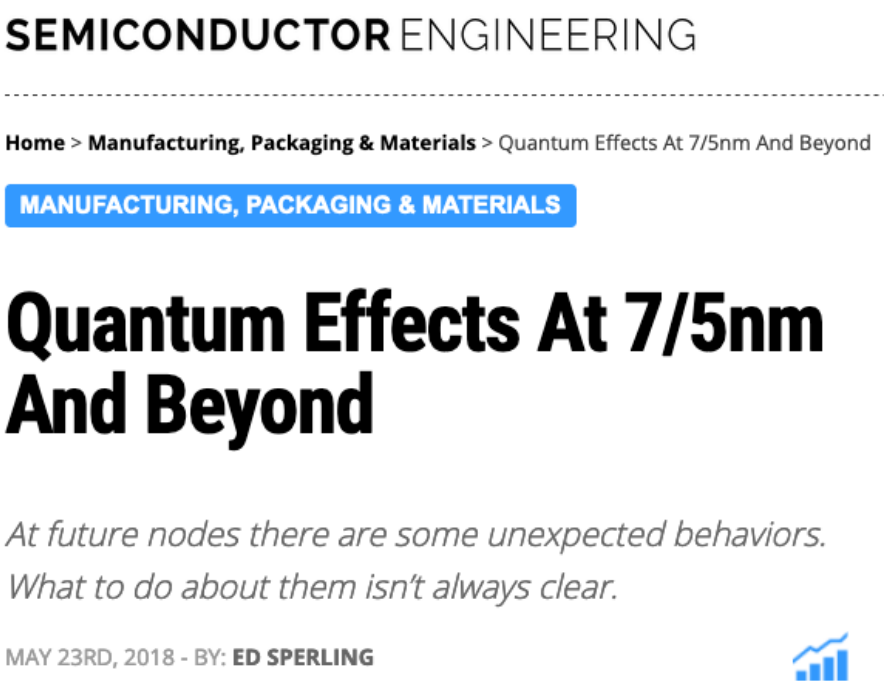
Why quantum computing?

But we are approaching the physical limits of how small a transistor we can put on a chip before quantum effects (tunneling) will become a problem.



Data source: `https://en.wikipedia.org/wiki/Semiconductor_device_fabrication`

Why quantum computing?



SEMICONDUCTOR ENGINEERING

Home > Manufacturing, Packaging & Materials > Quantum Effects At 7/5nm And Beyond

MANUFACTURING, PACKAGING & MATERIALS

Quantum Effects At 7/5nm And Beyond

At future nodes there are some unexpected behaviors. What to do about them isn't always clear.

MAY 23RD, 2018 - BY: **ED SPERLING**

Manufacturing of devices with 7nm chips began late in 2018.

Plans for 5nm to be released 2019-2020; design specs available as recently as April 2019.

Chips with transistors smaller than 7nm have required new and costly fabrication techniques; unclear whether anything smaller than 3nm is viable.

Computational limitations

Some problems will take an intractable amount of time to run on classical computers.

Parallelization can help, but we still cannot fully counteract the exponential complexity of some problems.

Sometimes that's a good thing:

- cryptographic infrastructure is built on such mathematically hard problems

Why quantum computing?

But usually it just prevents us from doing interesting things:

- solving complex optimization problems
- simulation of molecules and quantum systems
- searching large spaces
- machine learning with large amounts of data

What is quantum computing?

Quantum computation is the manipulation of the state of a physical quantum system in order to solve a problem.

Many quantum algorithms use 2-level quantum systems called *qubits* to solve problems more efficiently than the best-known classical algorithms:

- Integer factorization *(Shor's algorithm)*
- Searching large configuration spaces *(Grover's algorithm)*
- Simulating quantum systems *(Hamiltonian simulation)*
- Linear algebra *(e.g. HHL algorithm)*
- Machine learning

What's so good about qubits?

- The size of the mathematical space grows *exponentially* in the number of qubits
- We can make linear combinations of all possible qubit states, i.e. we can put them in *superposition*
- We can *entangle* multiple qubits, and use these states as a resource in many algorithms

Single-qubit systems

From bits to qubit I: bits

All computation in our computers today is done with bits.

The state* of a bit is *either* 0 or 1.

A set of n bits is represented by n real numbers (0s and 1s).

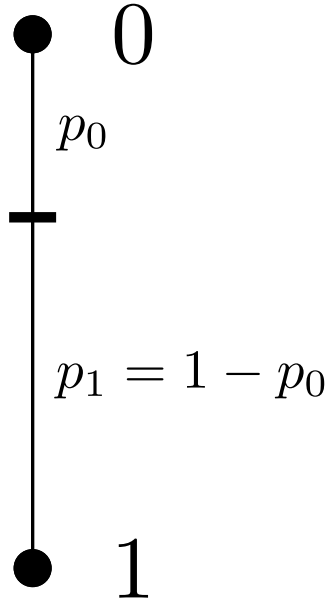
- 0
- 1

*Physically this is represented by some voltage and its state depends on whether or not that voltage is above/below a threshold value

From bits to qubit II: probabilistic bits

The value of a bit of information can be governed by a probability distribution.

For example, the outcome of a coin flip is a single bit of information, but before it is flipped it is probabilistic bit with both outcomes equally likely.



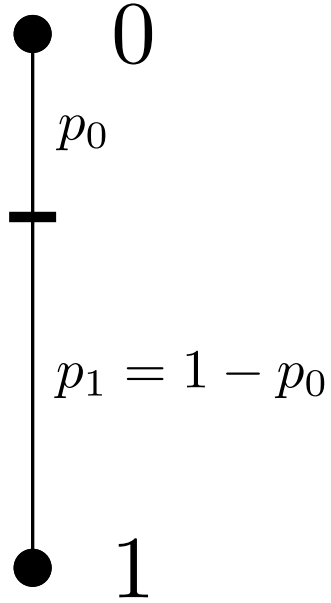
From bits to qubit II: probabilistic bits

We can assign a probabilistic bit a state based on its probability distribution:

$$\psi = \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} \qquad (1)$$

where p_0 and p_1 are real numbers and we must have

$$p_0 + p_1 = 1. \qquad (2)$$



From bits to qubit II: probabilistic bits

We can transform one probability distribution to another using a stochastic matrix,

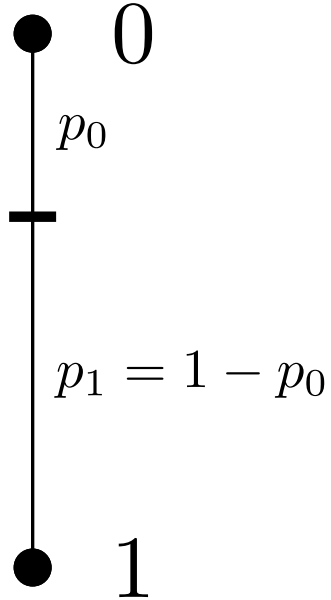
$$A = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \qquad (3)$$

where all $a_{ij} \in [0, 1]$, and the sum of every row is 1.

Then

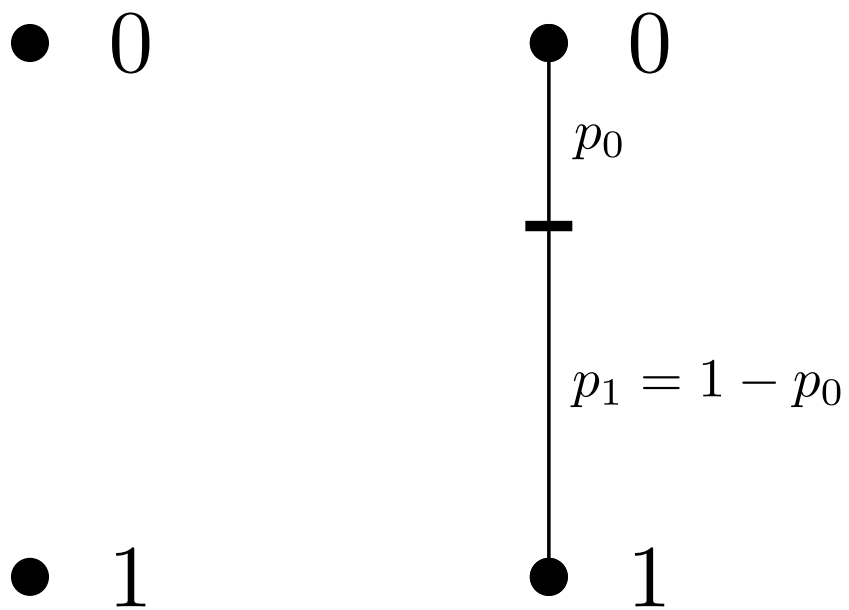
$$\begin{pmatrix} q_0 \\ q_1 \end{pmatrix} = A \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} \qquad (4)$$

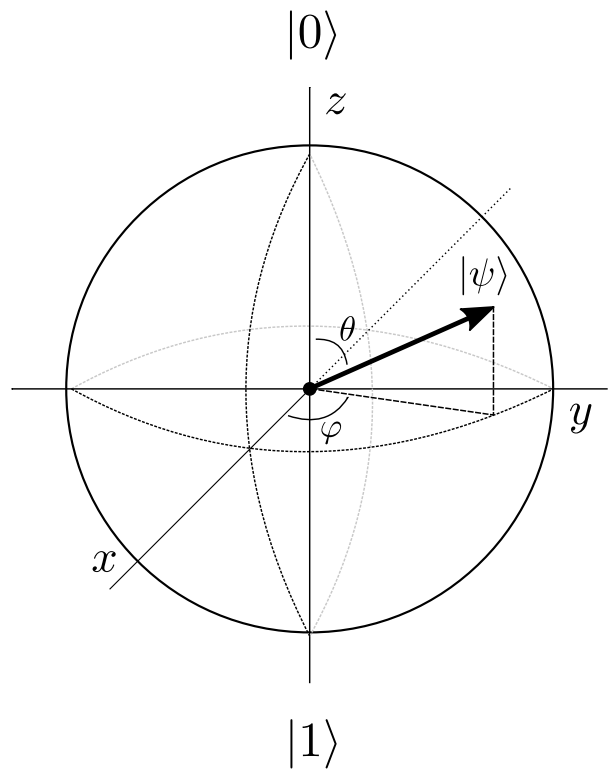
and $q_0 + q_1 = 1$.



From bits to qubit III: qubits

Extension of probabilistic bits to 2-level quantum systems:





Mathematical representation of qubits

Extension of probabilistic bits to 2-level quantum systems.

Instead of

$$\psi = \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} \tag{5}$$

where $p_0, p_1 \in \mathbb{R}$ and $p_0 + p_1 = 1$, we have

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{6}$$

where $\alpha, \beta \in \mathbb{C}$, and $|\alpha|^2 + |\beta|^2 = 1$.

Mathematical representation of qubits

A qubit state is a unit vector that lives in a 2-dimensional complex vector space called *Hilbert space*.

Qubit states are written as a linear combination, or *superposition*, of an *orthonormal basis* of the Hilbert space. The most commonly used one is the **computational basis**:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{7}$$

Then,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad \alpha, \beta \in \mathbb{C}, \tag{8}$$

where unit length is ensured by having $|\alpha|^2 + |\beta|^2 = 1$.

Mathematical representation of qubits

The complex parameters α and β are called *amplitudes*.

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

If we measure this qubit, we will find it in state $|0\rangle$ with probability $|\alpha|^2$, and state $|1\rangle$ with probability $|\beta|^2$. (Hence the need for the restriction $|\alpha|^2 + |\beta|^2 = 1$.)

Such a measurement is *destructive* - afterwards, the qubit stays in the state in which we observed it.

We will talk more about measurements later today.

Using qubits for computation

The amplitudes are the key here - when we measure the qubits at the end of our computation, they are what determine the outcome frequencies. We should make them meaningful to the problem.

But for this we need a way of:

1. putting qubits into superposition
2. manipulating the qubit to get the amplitudes that we want
3. combining multiple qubits
4. measuring the qubits to get the answer

How do we (mathematically) perform operations on qubits?

Unitary operations

Single-qubit states are manipulated by 2×2 unitary matrices, i.e. elements of $U(2)$. U is a unitary matrix if

$$U^\dagger U = U U^\dagger = \mathbb{1}$$

Unitary operations *the normalization of qubit states*. If

$$U(\alpha|0\rangle + \beta|1\rangle) = \alpha'|0\rangle + \beta'|1\rangle,$$

then

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

The principle of superposition

Unitary operations act *linearly* on superpositions:

$$U(\alpha|0\rangle + \beta|1\rangle) = \alpha U|0\rangle + \beta U|1\rangle \tag{9}$$

This is a key contributor to the power of quantum computing!

Creating a superposition: the Hadamard gate

Perhaps the most important unitary in quantum computing is the Hadamard gate:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{10}$$

This gate creates a *uniform superposition* of computational basis states...:

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

...and also undoes it:

$$\begin{aligned} H|+\rangle &= |0\rangle \\ H|-\rangle &= |1\rangle \end{aligned}$$

Single-qubit gates: Paulis

You are all familiar with the Pauli operators, X, Y, and Z:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{11}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{12}$$

$$Y = iZX = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{13}$$

These are unitary, and can do some very useful things to qubit states.

Single-qubit gates: Paulis

X is called the **bit flip** operation:

$$X|0\rangle = |1\rangle, \tag{14}$$
$$X|1\rangle = |0\rangle \tag{15}$$

Z is called the **phase flip** operation:

$$Z|0\rangle = |0\rangle, \tag{16}$$
$$Z|1\rangle = -|1\rangle \tag{17}$$

Single-qubit gates: applying sequences of gates

We apply *products* of single-qubit operations to represent performing gates in sequence. For example,

$$\begin{aligned} XZH|0\rangle &= XZ\left(\frac{|0\rangle}{\sqrt{2}} + \frac{|1\rangle}{\sqrt{2}}\right) \\ &= X\left(\frac{|0\rangle}{\sqrt{2}} - \frac{|1\rangle}{\sqrt{2}}\right) \\ &= \frac{|1\rangle}{\sqrt{2}} - \frac{|0\rangle}{\sqrt{2}} \end{aligned}$$

Products are applied from right to left.

Single-qubit gates: rotations

X, Y, and Z are special cases of more general qubit rotations:

$$R_x(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix} \tag{18}$$

$$R_y(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix} \tag{19}$$

$$R_z(\theta) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \tag{20}$$

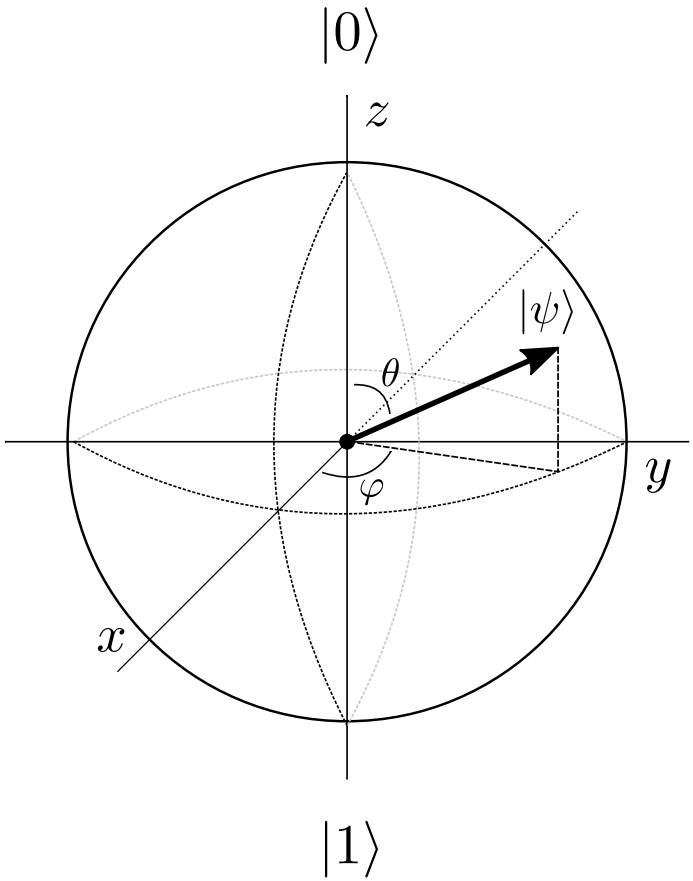
$X = R_x(\pi), \quad Y = R_y(\pi), \quad Z = R_z(\pi).$

R_x and R_y can put qubits into *non-uniform* superpositions.

... wait. Rotations? Rotations around *what*?

Visualizing a qubit: the Bloch sphere

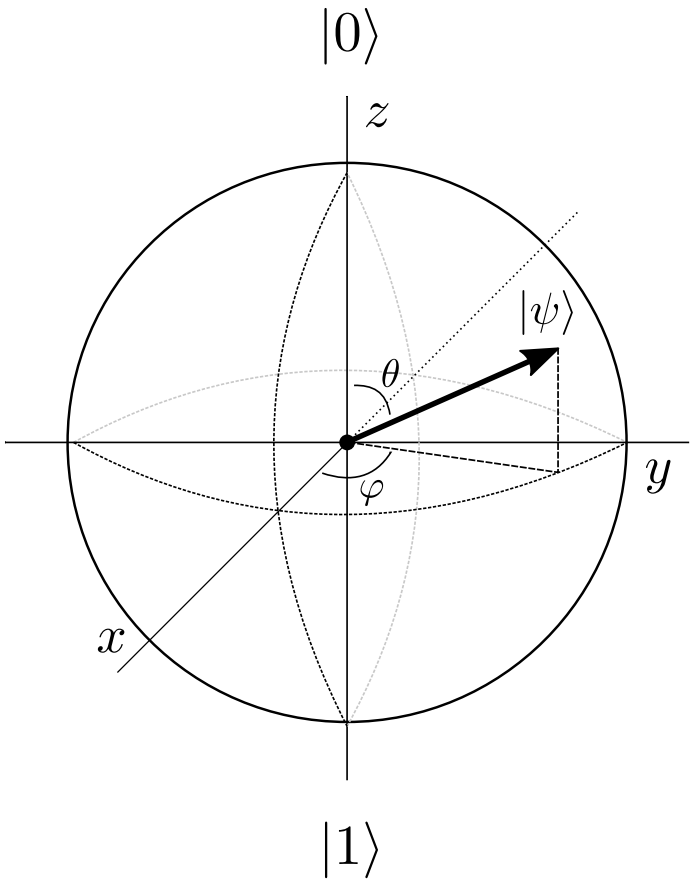
Single-qubit ket states can be represented on the surface of a sphere of radius 1 called the *Bloch sphere*.



Visualizing a qubit: the Bloch sphere

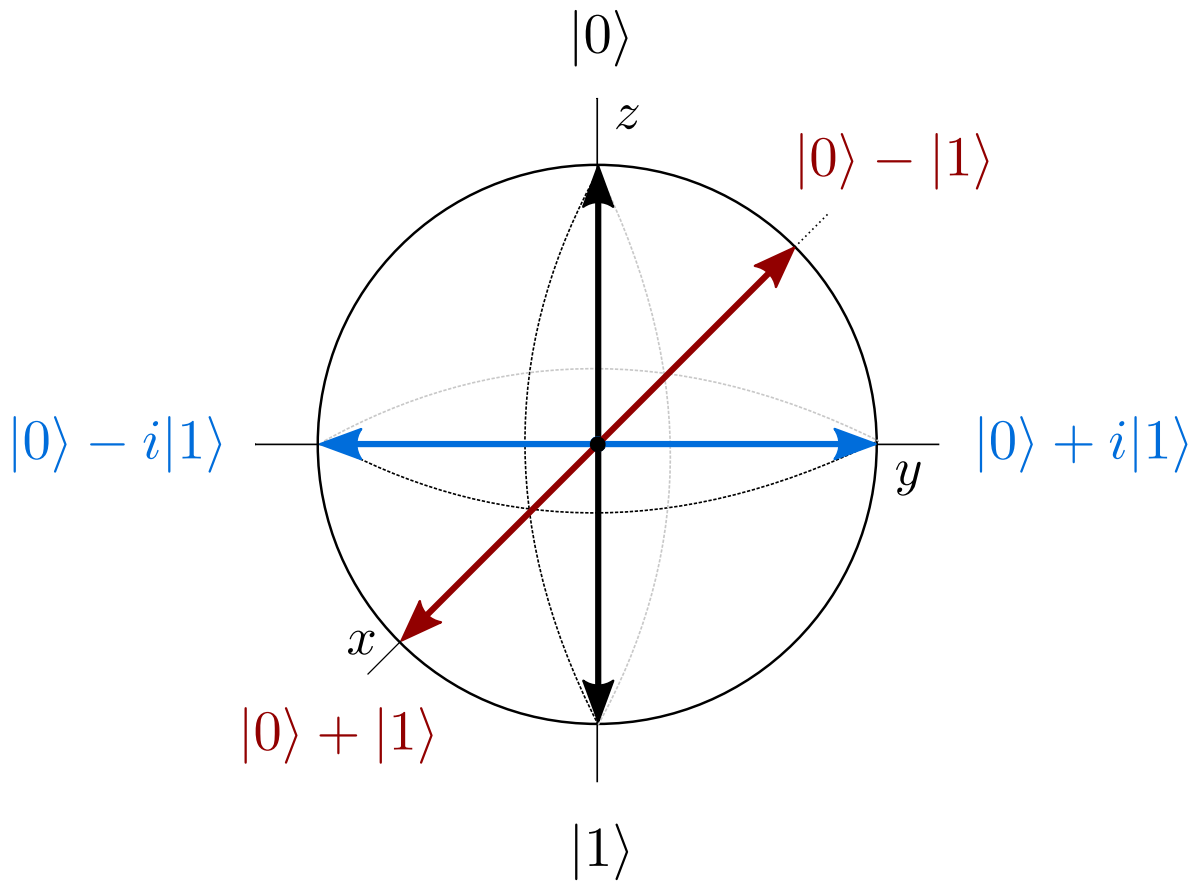
States can be plotted on the sphere using a parameterized form:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi}\sin\left(\frac{\theta}{2}\right)|1\rangle \tag{21}$$



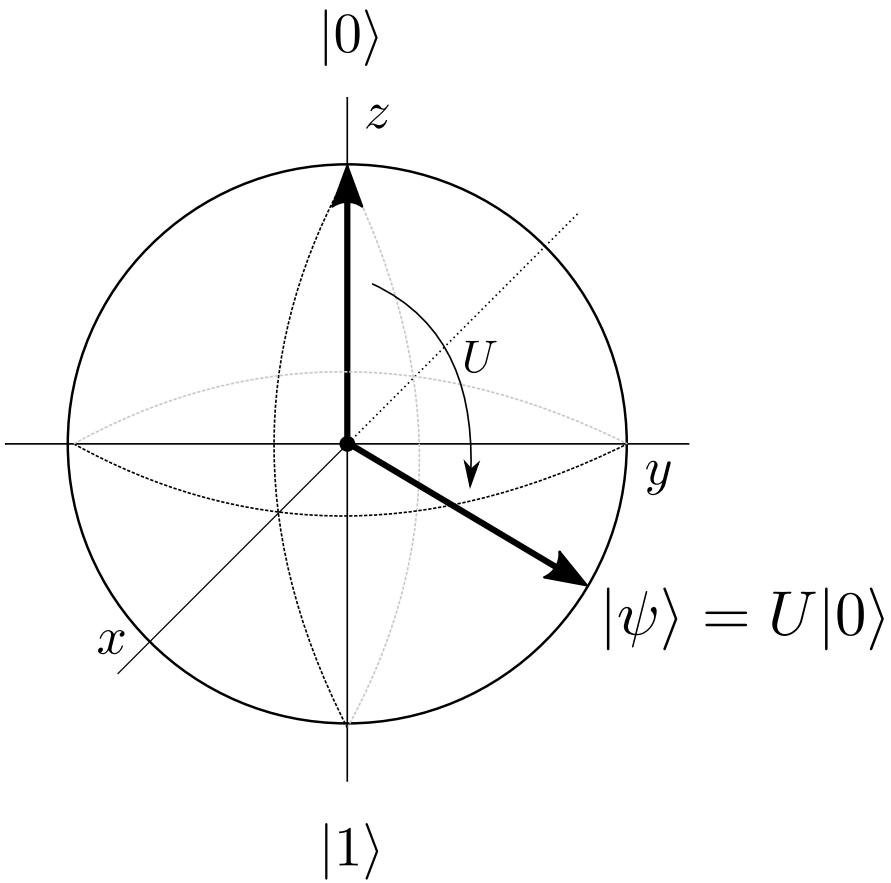
Visualizing a qubit: the Bloch sphere

The axis points correspond to the *eigenstates* of σ_x, σ_y, and σ_z.

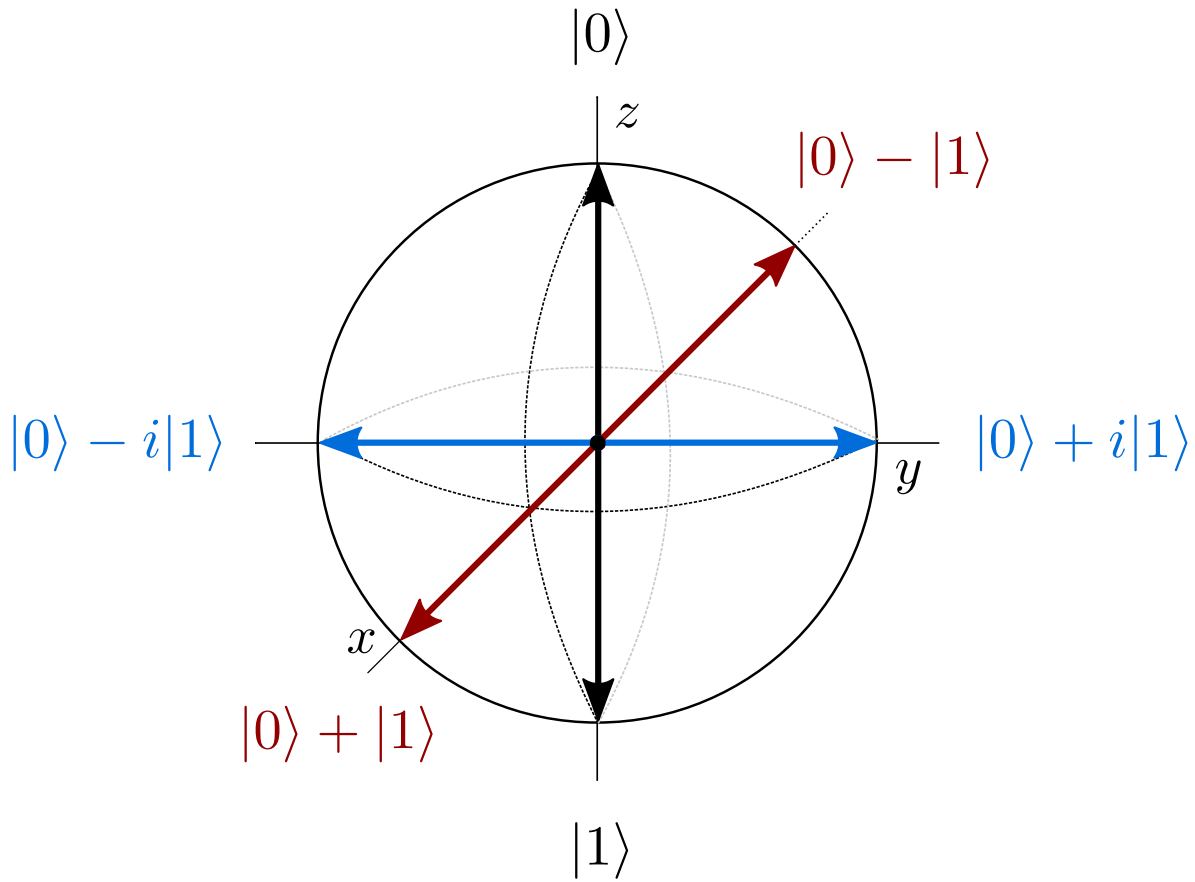


Unitary operations

Unitary matrices rotate state vectors around the Bloch sphere



Operations on the Bloch sphere



Single-qubit gates: complex phase gates

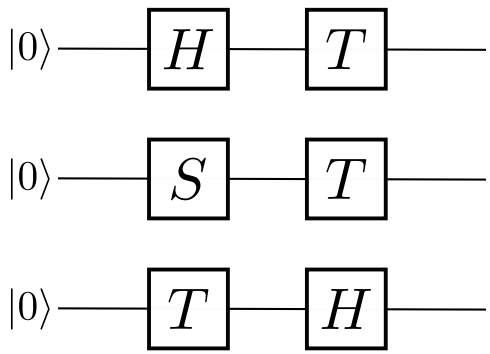
There are two more very important single-qubit gates:

Gate	Unitary	Action
Phase gate $S = \sqrt{Z}$	$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$	$1/4$ turn around z-axis
T gate, $T = \sqrt{S}$	$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$	$1/8$ turn around z-axis

These gates change the *relative phase* between $|0\rangle$ and $|1\rangle$. They don't affect the magnitudes of the amplitudes, but we will see that they *do* affect the measurement statistics!

Quantum circuits

That is a lot of matrices and vectors - it is going to get even more tedious when we bring in more qubits.
Thankfully we have *quantum circuits*.



Gates are applied from left to right, e.g. apply H on qubit 1, then apply T.

Unitaries are applied from right to left, e.g. qubit 1 gets hit with $U = TH$.

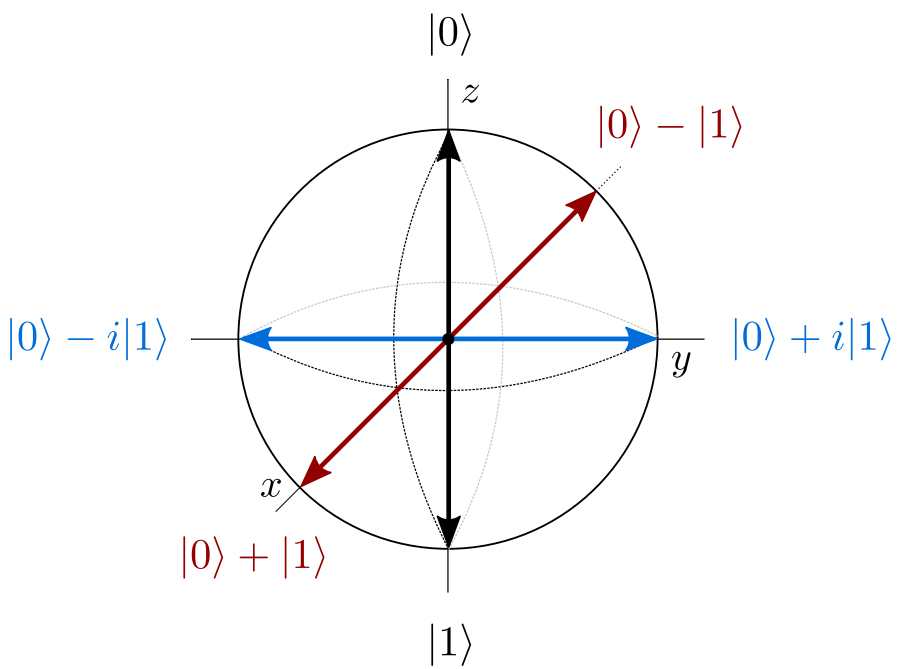
Single-qubit operations: hands-on with Quirk

Navigate to:

`https://algassert.com/quirk`

Single-qubit operations: hands-on with Quirk

Task 1: Using only H and Z gates, implement an X gate

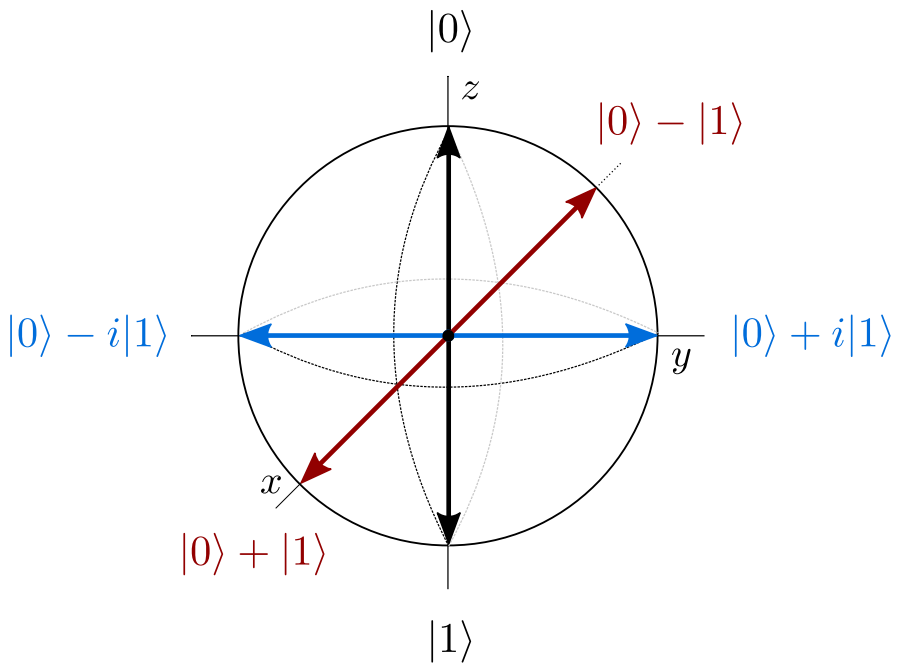


$$
\begin{aligned}
X|0\rangle &= |1\rangle \\
X|1\rangle &= |0\rangle \\
Z|0\rangle &= |0\rangle \\
Z|1\rangle &= -|1\rangle \\
H|0\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \\
H|1\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)
\end{aligned}
$$

Note: any gates you add get added to the URL, so to save a circuit, you can just bookmark the webpage.

Task 2: Starting from $|0\rangle$, prepare the state

$$|\psi\rangle = \frac{1}{2}|0\rangle + \frac{\sqrt{3}i}{2}|1\rangle$$



$$R_x(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

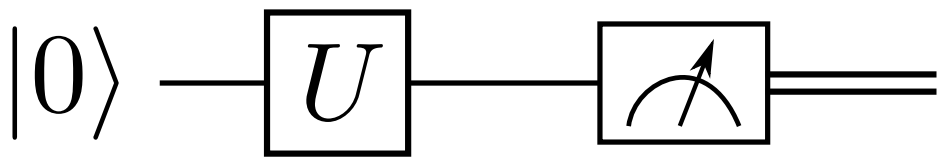
$$R_y(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

$$R_z(\theta) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}$$

Measurement

We know now that *unitary operations* can be used to manipulate our qubits to perform a computation. But after we're done computing, how do we get the answer? We need to measure our system.

A measurement in a circuit is represented by a box with a dial:



The two wires coming out of it indicate a *classical bit* - the outcome of the measurement is not a qubit, it's either 0 or 1!

Measurement

We need a mathematical formalism for measuring qubits, to see what the state system is in after the computation.

This could be a whole lecture series on its own, so I will keep it simple and cover only projective measurements, which you may be familiar with if you've taken QM.

Projective measurements

A Hermitian matrix Π is a *projector* if $\Pi^2 = \Pi$.

A *projective measurement* is a set of projectors $\mathcal{B} = \{\Pi_i\}_{i=0}^{M}$ where

$$\sum_{i=0}^{M} \Pi_i = \mathbb{1} \tag{22}$$

For every single-qubit ket state $|\psi\rangle$, the 2×2 matrix $|\psi\rangle\langle\psi|$ is a projector. If we take a set of orthonormal kets $\{|\psi_i\rangle\}$ (i.e. a basis), together they form a projective measurement.

For a single qubit,

$$|\psi_0\rangle\langle\psi_0| + |\psi_1\rangle\langle\psi_1| = \mathbb{1} \tag{23}$$

Projective measurements

When we make a projective measurement on a state $|\varphi\rangle$ in basis $\{|\psi_i\rangle\}$, the probability of obtaining outcome i is

$$\begin{aligned} \Pr(\text{outcome i}) &= \mathrm{Tr}(\Pi_i |\varphi\rangle\langle\varphi|) \\ &= |\langle\psi_i|\varphi\rangle|^2 \end{aligned}$$

If we observe outcome i, following the measurement the system will be left in state $|\psi_i\rangle$[1].

[1] Actually things are a bit more subtle than this; for a good overview of projective measurements, see `https://www.people.vcu.edu/~sgharibian/courses/CMSC491/notes/Lecture%203%20-%20Measurement.pdf`

Projective measurements

Example: measuring in the computational basis

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

The computational basis $\{|0\rangle, |1\rangle\}$ gives a projective measurement:

$$|0\rangle\langle 0| + |1\rangle\langle 1| = \mathbb{1} \tag{24}$$

Then if we measure $|\psi\rangle$,

$$\begin{aligned} \Pr(0) &= |\langle 0|\psi\rangle|^2 = |\alpha|^2 \\ \Pr(1) &= |\langle 1|\psi\rangle|^2 = |\beta|^2 \end{aligned}$$

Basis changes

We can measure in any orthonormal basis by applying a suitable unitary transformation to the computational basis vectors.

Example: measuring in the *Hadamard basis*:

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

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

You can check that $|+\rangle\langle+| + |-\rangle\langle-| = \mathbb{1}$ is indeed a valid projective measurement.

Basis changes

Then, for $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$,

$$\begin{aligned}
\Pr(+) &= |\langle +|\psi\rangle|^2 \\
&= \frac{1}{2}|\alpha\langle 0|0\rangle + \alpha\langle 0|1\rangle + \beta\langle 1|0\rangle + \beta\langle 1|1\rangle|^2 \\
&= \frac{1}{2}|\alpha + \beta|^2 \\
\Pr(-) &= \frac{1}{2}|\alpha - \beta|^2
\end{aligned}$$

Measuring in the Hadamard basis

Why would we want to measure in different bases?

Example

Consider $|+\rangle$ and $|-\rangle$.

If we measure in the computational basis, for both states we will get 0 with probability $1/2$ and also 1 with probability $1/2$. It's impossible to know which state we have!

But if we measure in the Hadamard basis, we will get either *only* $+$ or *only* $-$.

The measurement statistics change depending on which basis we measure in! Tomorrow, we will see an example (quantum teleporation) of how this is useful.

Exercise: Phase factors

Let's consider another set of orthonormal states:

$$|y_+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \tag{25}$$

$$|y_-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \tag{26}$$

If we measure $|y_+\rangle$ in the computational basis, what are the outcome probabilities?

$$\Pr(0) = \Pr(1) = \frac{1}{2} \tag{27}$$

If we measure $|y_+\rangle$ in the Hadamard basis, what are the outcome probabilities?

$$\Pr(+) = \Pr(-) = \frac{1}{2} \tag{28}$$

Phase factors

These states are indistinguishable in both bases! This might seem strange, since

$$|y_+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \tag{29}$$

$$|y_-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \tag{30}$$

look so similar to

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{31}$$

$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{32}$$

The *relative* phase between $|0\rangle$ and $|1\rangle$ clearly has a large effect on the state even though the magnitudes of the amplitudes in other bases don't change.

Global vs. relative phase

Changing the **global phase**

$$|\psi\rangle \to e^{i\gamma}|\psi\rangle \tag{33}$$

does not change the measurement statistics.

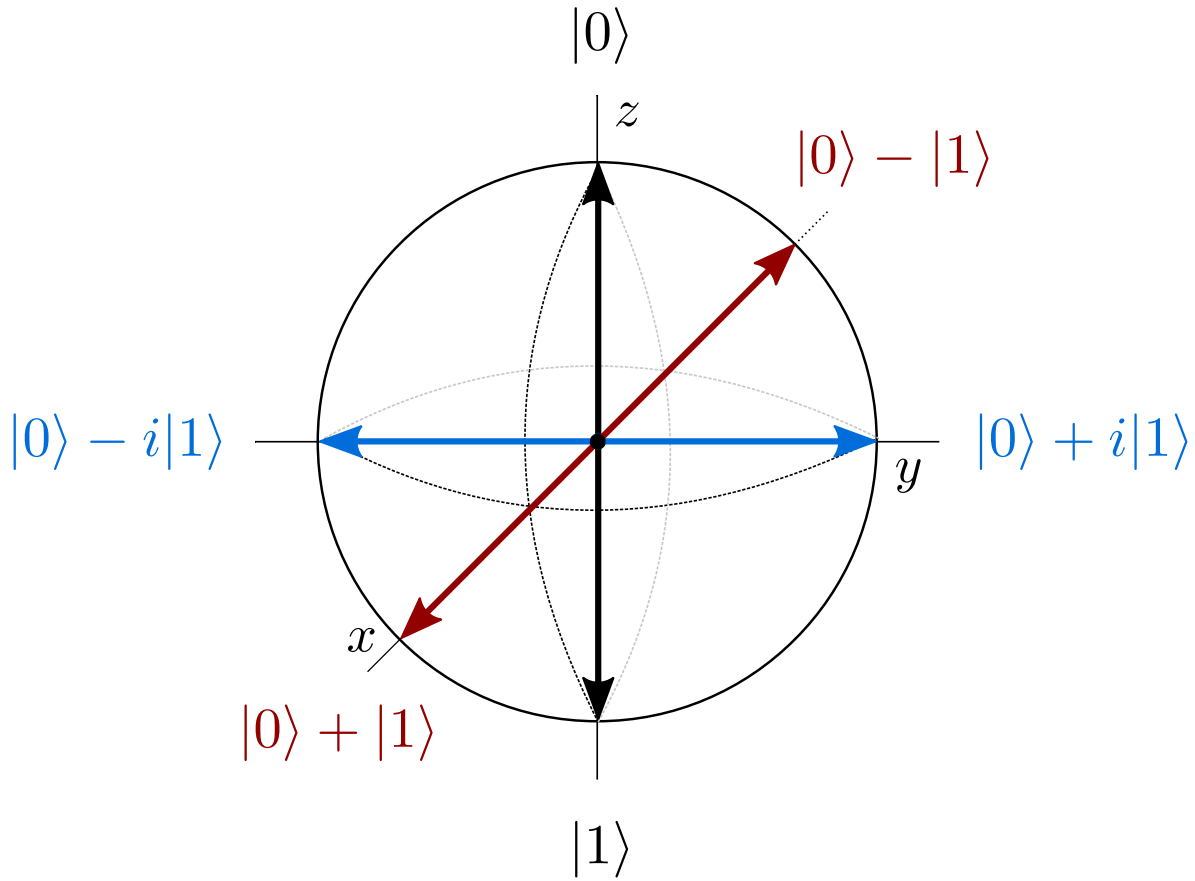
But changing the *relative phase*, i.e.

$$|\psi\rangle \to \alpha|0\rangle + e^{i\gamma}\beta|1\rangle \tag{34}$$

does change the measurement statistics, depending on which basis you are measuring in.

Global vs. relative phase

It is easier to see visually how these states are different - relative phases rotate us around the z axis of the Bloch sphere.



Review

We talked about:

- Why quantum computing will be needed in the future
- What qubits are, and how they are represented mathematically
- Common single-qubit gates
- Measurement of single-qubit systems

Go back to

`menti.com`

and input the code

`87 83 08.`

Next time

- Multi-qubit systems and entanglement
- Quantum teleportation
- Quantum advantage
- Overview of current-generation quantum hardware

For more information, advanced topics, and references, check out my introductory QC lecture notes on Github:

`https://github.com/glassnotes/Intro-QC-TRIUMF`

I will post the slides from today there as well.

Multi-qubit systems

Tensor products

Hilbert spaces compose under the *tensor product*.

Example

Let

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad B = \begin{pmatrix} e & f \\ g & h \end{pmatrix}. \tag{35}$$

The tensor product of A and B, $A \otimes B$ is

$$A \otimes B = \begin{pmatrix} a \begin{pmatrix} e & f \\ g & h \end{pmatrix} & b \begin{pmatrix} e & f \\ g & h \end{pmatrix} \\ c \begin{pmatrix} e & f \\ g & h \end{pmatrix} & d \begin{pmatrix} e & f \\ g & h \end{pmatrix} \end{pmatrix} = \begin{pmatrix} ae & af & be & bf \\ ag & ah & bg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{pmatrix} \tag{36}$$

Multi-qubit systems

Qubit state vectors are also combined using the *tensor product*:

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \tag{37}$$

An n-qubit state is therefore a vector of length 2^n.

Multi-qubit systems

The tensor product is linear and distributive, so if we have

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad |\varphi\rangle = \gamma|0\rangle + \delta|1\rangle, \tag{38}$$

then they tensor together to form

$$\begin{aligned} |\psi\rangle \otimes |\varphi\rangle &= (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle) \\ &= \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma\ |10\rangle + \beta\delta|11\rangle \end{aligned}$$

The states $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ are the computational basis vectors for 2 qubits; we can create arbitrary linear combinations of them as long as the normalization on the coefficients holds.

Multi-qubit systems

Single-qubit unitary operations also compose under tensor product.

For example, apply U_1 to qubit $|\psi\rangle$ and U_2 to qubit $|\varphi\rangle$:

$$(U_1 \otimes U_2)(|\psi\rangle \otimes |\varphi\rangle) = (U_1|\psi\rangle) \otimes (U_2|\varphi\rangle) \tag{39}$$

If an n-qubit ket is a vector with length 2^n, then a unitary acting on n qubits has dimension $2^n \times 2^n$.

Multi-qubit systems

Exercise: Suppose we have two qubits in the state $|01\rangle$. What state do we get when we apply H to the first qubit and S to the second qubit?

Solution: We know:

$$H|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right), \quad S|1\rangle = i|1\rangle \tag{40}$$

So then

$$\begin{aligned}(H \otimes S)|01\rangle &= (H|0\rangle) \otimes (S|1\rangle) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \otimes i|1\rangle \\ &= \frac{i}{\sqrt{2}}\left(|01\rangle + |11\rangle\right)\end{aligned}$$

Exercise: Consider the 2-qubit state

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{41}$$

Find

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad |\varphi\rangle = \gamma|0\rangle + \delta|1\rangle \tag{42}$$

such that

$$|\Psi\rangle = |\psi\rangle \otimes |\varphi\rangle \tag{43}$$

Solution: This is impossible (sorry!)

Entanglement

The state

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{44}$$

is **entangled**.

We cannot describe the two qubits individually, we can only described their combined state.

Paraphrasing from John Preskill: *it's like you're reading a book, but instead of reading the pages sequentially, you have to read it all at the same time in order to understand it.*

Entanglement

Furthermore, the measurement outcomes of

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{45}$$

are *perfectly correlated*.

For example, if I measure the first qubit and get 0, I'll get 0 for the second qubit as well!

Entanglement is not limited to two qubits. In principle we can entangle as many as we like:

$$|\Psi\rangle = |00\cdots0\rangle + |11\cdots1\rangle \tag{46}$$

A measurement outcome of 0 on qubit 1 means we'll get 0 on *all other qubits* too.

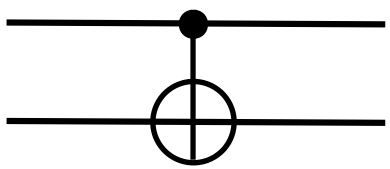
Entangling gates

How do we make an entangled state (in theory)? Previous 2-qubit operations we saw were expressed as tensor products of single-qubit ones.

There exist *entangling gates* that will turn a non-entangled, or separable, state into an entangled one. The most commonly used one is the controlled-NOT, or CNOT:

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

$$\begin{aligned} \text{CNOT}|00\rangle &= |00\rangle \\ \text{CNOT}|01\rangle &= |01\rangle \\ \text{CNOT}|10\rangle &= |11\rangle \\ \text{CNOT}|11\rangle &= |10\rangle \end{aligned}$$



The first qubit is the *control* qubit - it controls whether or not an X (NOT) gate is applied to the second qubit.

Entangling gates: CNOT

Exercise: What happens when we apply a CNOT to qubits in state $|+\rangle \otimes |0\rangle$?

$$\begin{aligned}
\text{CNOT}\left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle\right] &= \text{CNOT}\left[\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)\right] \\
&= \frac{1}{\sqrt{2}}(\text{CNOT}|00\rangle + \text{CNOT}|10\rangle) \\
&= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)
\end{aligned}$$

We've gone from a separable state to an entangled one!

Universal gate sets

Now that we've seen the CNOT gate, in principle *I don't need to introduce to you any additional single- or multi-qubit gates*.

That's a good thing - there are an infinite number of unitaries, and there's no way we can individually program each one into our quantum computing hardware.

Thankfully, the CNOT and a few single-qubit gates are all we need for universal quantum computation!

How many is "a few"?

Universal gate sets

Single-qubit universal gate set

If you have a quantum computer that can perform

$$\{H,\ \ T\} \tag{47}$$

then you can implement *any* other single-qubit unitary up to an arbitrary precision.

Multi-qubit universal gate set

If you have a quantum computer that can perform

$$\{H,\ \ T,\ \ \mathrm{CNOT}\} \tag{48}$$

then you can implement *any* other multi-qubit unitary up to an arbitrary precision.