Introduction to Quantum Information and Qiskit

Conor Ryan
Theoretical Physics Student Association

Objectives

- Understand what quantum mechanical behaviour is important in quantum information theory
- Understand the mathematical background used in quantum information theory(linear algebra, Dirac notation,...)
- Study the implementation of one- and two-qubit quantum gates in quantum circuits
- Study differences between classical and quantum computation
- Look at various applications of quantum information theory
- Complete Qiskit tutorial
  - Quantum Teleportation algorithm
  - Superdense Coding algorithm

Definition (Quantum Information Theory)

Def: The exploitation of quantum mechanical properties to analyse, process and communicate information.

Definition (Quantum Information Theory)

Def: The exploitation of quantum mechanical properties to analyse, process and communicate information.

Definition (Qubit)

Def: A two level quantum system/the quantum mechanical equivalent of a bit.

Definition (Quantum Information Theory)

Def: The exploitation of quantum mechanical properties to analyse, process and communicate information.

Definition (Qubit)

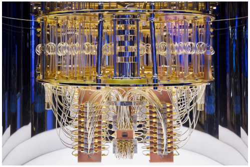
Def: A two level quantum system/the quantum mechanical equivalent of a bit.

Key Quantum Properties:

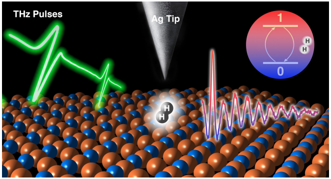
Quantum Superposition: Quantum objects(such as qubits) can exist in multiple states until they are measured.

Quantum Entanglement: Quantum objects can be linked together over apparently infinite distances where the behaviour of one of the objects depends on the other.

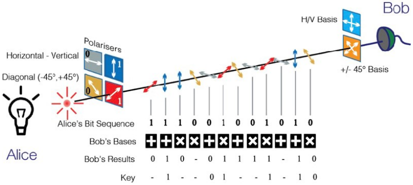
Quantum Technology



(A) Quantum Computer



(B) Quantum Sensor



Quantum Cryptography

Mathematical Toolbox

Wavefunctions take the form $\Psi(\vec{x}, t)$ usually. An alternative notation or way of expressing quantum states commonly used particularly in quantum information is Dirac notation.

Mathematical Toolbox

Wavefunctions take the form $\Psi(\vec{x}, t)$ usually. An alternative notation or way of expressing quantum states commonly used particularly in quantum information is Dirac notation.

$$|\alpha\rangle = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, \langle\alpha| = |\alpha\rangle^\dagger = (a_1^*, a_2^*, \ldots, a_n^*) \tag{1}$$

In this notation we can still perform the usual vector operations.

$$\langle\alpha|\beta\rangle = a_1^* b_1 + \cdots + a_n^* b_n = \sum_{i=1}^{n} a_i^* b_i \tag{2}$$

$$\langle\alpha|\beta\rangle = \langle\beta|\alpha\rangle^* \tag{3}$$

$$|\alpha\rangle \otimes |\beta\rangle = \begin{pmatrix} a_1 \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \\ \vdots \\ a_n \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \end{pmatrix} = \begin{pmatrix} a_1 b_1 \\ \vdots \\ a_1 b_m \\ \vdots \\ a_n b_1 \\ \vdots \\ a_n b_m \end{pmatrix} \tag{4}$$

Note: The dimension of our new vector is $n \times m$.

All quantum states $|\psi\rangle$ exist in a linear vector space over the field $\mathbb{C}$ possessing an inner product which induces a norm and is complete with respect to this norm. This vector space is called a Hilbert space, $\mathcal{H}$, and has dimension $d = 2^n$ with n being the number of qubits.

$\mathcal{H}$ satisfies the usual vector space properties of being **closed under vector addition and scalar multiplication**.

Properties satisfied by $|u\rangle, |v\rangle, |w\rangle \in \mathcal{H}$ are

$$\langle u|(c_1|v\rangle + c_2|w\rangle) = c_1\langle u|v\rangle + c_2\langle u|w\rangle \tag{5}$$

$$\langle v|v\rangle \geq 0 \tag{6}$$

$$\langle v|v\rangle = 0 \iff |v\rangle = 0 \tag{7}$$

A set of orthonormal vectors $|e_k\rangle$ is **complete** if

$$\sum_k |e_k\rangle\langle e_k| = \mathbb{I} \iff |v\rangle = \sum_k \langle e_k|v\rangle|e_k\rangle \tag{8}$$

In quantum mechanics there are also particular types of matrices which are important.

In quantum mechanics there are also particular types of matrices which are important.

These matrices are called **hermitian** and **unitary** matrices.

Hermitian matrices satisfy $H^\dagger = H$.

Unitary matrices satisfy $U^\dagger = U^{-1} \iff U^\dagger U = UU^\dagger = \mathbb{I}$.

Note: Matrices can be both hermitian and unitary.

Properties of the hermitian conjugate,

$$(AB)^\dagger = B^\dagger A^\dagger \tag{9}$$

$$(A|v\rangle)^\dagger = \langle v|A^\dagger \tag{10}$$

($\dagger$ is the hermitian conjugate = transpose and complex conjugate)

Recall eigenvalues and eigenvectors of a matrix.

$$A|v_j\rangle = \lambda_j|v_j\rangle \tag{11}$$

λ_j form the set of eigenvalues of A which is called the **spectrum**. $|v_j\rangle$ form the set of eigenvectors of A.

We can then represent A as $A = \sum_j \lambda_j|v_j\rangle\langle v_j|$.

Note: $\sum_j |v_j\rangle\langle v_j|$ can be considered as the projection operator P which projects a quantum system onto the subspace created by the set of eigenvectors.

Properties of Hermitian Matrices:

1. Hermitian matrices have real eigenvalues.

$A|v\rangle = \lambda|v\rangle \implies \langle v|A|v\rangle = \lambda$

$\lambda^* = \langle v|A|v\rangle^* = \langle v|A^\dagger|v\rangle = \langle v|A|v\rangle = \lambda$

Properties of Hermitian Matrices:

1. Hermitian matrices have real eigenvalues.

$A|v\rangle = \lambda|v\rangle \implies \langle v|A|v\rangle = \lambda$

$\lambda^* = \langle v|A|v\rangle^* = \langle v|A^\dagger|v\rangle = \langle v|A|v\rangle = \lambda$

2. Two eigenvectors of a hermitian matrix are orthogonal if their eigenvalues are unequal.

$A|v\rangle = \lambda|v\rangle,\ A|w\rangle = \mu|w\rangle$

$0 = \langle v|A|w\rangle - \langle v|A^\dagger|w\rangle = \mu\langle v|w\rangle - \lambda\langle v|w\rangle = (\mu - \lambda)\langle v|w\rangle \iff \langle v|w\rangle = 0$

Fundamentals of Quantum Mechanics

We will now relate these ideas to more exact processes in quantum physics.

Fundamentals of Quantum Mechanics

We will now relate these ideas to more exact processes in quantum physics.

Given a quantum state $|\psi\rangle$ which is normalised($\langle\psi|\psi\rangle = 1$), we can perform a measurement on it to obtain resultant state $|\alpha\rangle$ with probability $|\langle\alpha|\psi\rangle|^2$.

A measurement can be considered as the projection operator $P = |\alpha\rangle\langle\alpha|$.

Observables are typically described by hermitian operators(although some arguments have been made that this isn't always the case).

When performing a measurement we end up in some eigenstate $|\alpha\rangle$ but actually measure the eigenvalue of this state, a.

Observables are typically described by hermitian operators(although some arguments have been made that this isn't always the case).

When performing a measurement we end up in some eigenstate $|\alpha\rangle$ but actually measure the eigenvalue of this state, a.

Given the probabilistic nature of quantum mechanics we are not guaranteed to end up in the same eigenstate after every measurement of the same quantum state.

For N copies of a physical system allowing us to perform N measurements we obtain the eigenvalue a an amount of times $N|\langle\alpha|\psi\rangle|^2$ which gives the exact amount of times we should measure it in the limit $N \to \infty$.

Observables are typically described by hermitian operators(although some arguments have been made that this isn't always the case).

When performing a measurement we end up in some eigenstate $|\alpha\rangle$ but actually measure the eigenvalue of this state, a.

Given the probabilistic nature of quantum mechanics we are not guaranteed to end up in the same eigenstate after every measurement of the same quantum state.

For N copies of a physical system allowing us to perform N measurements we obtain the eigenvalue a an amount of times $N|\langle\alpha|\psi\rangle|^2$ which gives the exact amount of times we should measure it in the limit $N \to \infty$.

Another important quantity in measurement is the **expectation value**.
$\langle\psi|A|\psi\rangle = \sum_k a_k \langle\psi|\alpha_k\rangle\langle\alpha_k|\psi\rangle = \sum_k a_k p_k$
Note: If $|\psi\rangle$ is already an eigenstate then the expectation value will just be a because that is what we constantly measure when observing our physical system.

Into the World of Quantum Information

We will now take what we have covered so far in general quantum mechanics and change it to the framework of quantum information.

Into the World of Quantum Information

We will now take what we have covered so far in general quantum mechanics and change it to the framework of quantum information.

In quantum information we work in the 2-dimensional computational basis $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.
We describe our single qubit quantum state in the form $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

To obtain the correct probabilities when measuring our qubit we have the condition $|\alpha|^2 + |\beta|^2 = 1$. $\alpha, \beta \in \mathbb{C}$

Into the World of Quantum Information

We will now take what we have covered so far in general quantum mechanics and change it to the framework of quantum information.

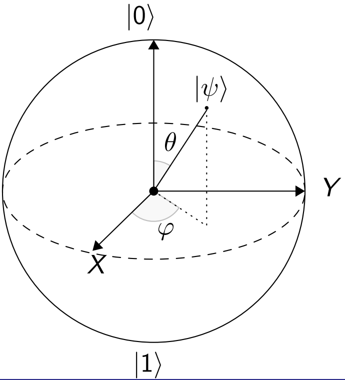
In quantum information we work in the 2-dimensional computational basis $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.
We describe our single qubit quantum state in the form $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

To obtain the correct probabilities when measuring our qubit we have the condition $|\alpha|^2 + |\beta|^2 = 1$. $\alpha, \beta \in \mathbb{C}$

We typically measure in the computational basis(Z-basis). There are alternate bases in which we can perform measurements such as the X-basis($|+\rangle, |-\rangle$) or Y-basis($|+i\rangle, |-i\rangle$).

Visualizing Quantum States

Quantum states are pictured on the **Bloch Sphere**.



Note: $|0\rangle, |1\rangle$ are orthogonal because they are basis vectors. However, on the Bloch Sphere any angle between vectors is doubled. In terms of Bloch Sphere coordinates we can write the quantum state of a qubit as $|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle$ The surface of the Bloch Sphere is a valid Hilbert Space. Any unitary transformation causes rotations of the quantum state around the Bloch Sphere.

Introduction to Quantum Circuits

We will now study how to perform operations on quantum states through the use of quantum circuits.
A quantum circuit with no operations takes the form

$$\begin{array}{l} |0\rangle \; \text{——} \\ |0\rangle \; \text{——} \\ \quad\vdots \\ |0\rangle \; \text{——} \end{array}$$

All qubits are initialised in the state $|0\rangle$ and we denote the state of the composite system as $|0\rangle^{\otimes n} = |0\rangle \otimes |0\rangle \otimes \cdots \otimes |0\rangle = |00\ldots0\rangle$
This composite state exists inside the Hilbert space
$\mathcal{H} = \mathcal{H}^0 \otimes \mathcal{H}^1 \otimes \cdots \otimes \mathcal{H}^{n-1}$

Single Qubit Quantum Gates

We will now look at some operations we can perform on qubits. The simplest of these gates are the X, Y and Z gates.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = |0\rangle\langle 1| + |1\rangle\langle 0| \tag{12}$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = -i|0\rangle\langle 1| + i|1\rangle\langle 0| \tag{13}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1| \tag{14}$$

These matrices are also the Pauli matrices σ_X, σ_Y and σ_Z which are hermitian and unitary.

Single Qubit Gates

Let's look at how these gates affect our basis vectors.

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

The X-gate is a bit flip gate.

$$Y|0\rangle = -i|0\rangle\langle 1|0\rangle + i|1\rangle\langle 0|0\rangle = i|1\rangle$$
$$Y|1\rangle = -i|0\rangle\langle 1|1\rangle + i|1\rangle\langle 0|1\rangle = -i|0\rangle$$

The Y-gate results in a bit flip along with a change in phase.

$$Z|0\rangle = |0\rangle\langle 0|0\rangle - |1\rangle\langle 1|0\rangle = |0\rangle$$
$$Z|1\rangle = |0\rangle\langle 0|1\rangle - |1\rangle\langle 1|1\rangle = -|1\rangle$$

The Z-gate results in a phase flip for $|1\rangle$.

Single Qubit Gates

There are other more complex single qubit gates.

Let's look at the Hadamard gate.

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|) \tag{15}$$

$$H|0\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle\langle 0|0\rangle + |0\rangle\langle 1|0\rangle + |1\rangle\langle 0|0\rangle - |1\rangle\langle 1|0\rangle) = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$$
$$H|1\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle\langle 0|1\rangle + |0\rangle\langle 1|1\rangle + |1\rangle\langle 0|1\rangle - |1\rangle\langle 1|1\rangle) = \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$$

Single Qubit Gates

There are other more complex single qubit gates.

Let's look at the Hadamard gate.

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|) \tag{15}$$

$$H|0\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle\langle 0|0\rangle + |0\rangle\langle 1|0\rangle + |1\rangle\langle 0|0\rangle - |1\rangle\langle 1|0\rangle) = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$$
$$H|1\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle\langle 0|1\rangle + |0\rangle\langle 1|1\rangle + |1\rangle\langle 0|1\rangle - |1\rangle\langle 1|1\rangle) = \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$$

Other generalised single qubit gates such as $R_X(\phi), R_Y(\phi), R_Z(\phi)$ are also possible which correspond to rotations of an angle ϕ around the corresponding axis on the Bloch Sphere.
There other gates which are less common such the P, S and T gates, but they are just special versions of the R_Z-gate up to a phase factor.

Two Qubit Gates

We now turn our attention to two-qubit gates which are mainly used to perform entangling operations between two qubits.

The most common of these gates is the CNOT gate.

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{16}$$

This is a conditional operation where the resulting state of the second(target) qubit depends on the state of the first(control) qubit. If the control qubit is in the state $|0\rangle$ then nothing happens to the target qubit. If the control qubit is in the state $|1\rangle$ then a bit-flip(X-gate) is performed on the target qubit.

Two Qubit Gates

The CNOT-gate could also be called the CX-gate because X is the conditional operation which is performed.

This means we can also have the less common CY- and CZ-gates which operate in the same way as CX but instead perform the Y and Z operations.

In general, in a quantum circuit we can construct a controlled gate which will perform some arbitrary unitary gate and not just the Pauli gates.

Two Qubit Gates

Other two-qubit operations not related specifically to entangling operations can also be performed.

One such gate is the SWAP-gate which swaps the state of two qubits.

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{17}$$

This gate is particularly useful when looking at real quantum chips. If we want to entangle two qubits which are not very closely connected on the quantum chip several SWAP gates can be performed which will improve the connectivity.

Two Qubit Gates

The SWAP gate provides a good example of how quantum hardware works.

Two Qubit Gates

The SWAP gate provides a good example of how quantum hardware works.

In quantum hardware systems there is a set of basis gates.
These basis gates typically consist of gates such as the I, X, R_Z, S_X and CNOT gates.

Two Qubit Gates

The SWAP gate provides a good example of how quantum hardware works.

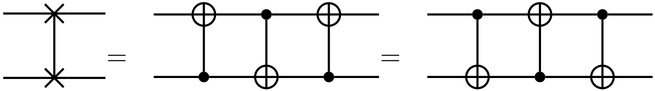
In quantum hardware systems there is a set of basis gates.
These basis gates typically consist of gates such as the I, X, R_Z, S_X and CNOT gates.

As you can see, gates such as the SWAP gate are not in this set of basis gates.

Gates such as the SWAP gate which are not a part of the basis gates are implemented using a composition of basis gates.

Two Qubit Gates

Let's look at this composition for the SWAP gate.



We see can that the particular two qubit gate we want to implement can be composed of CNOT gates in two different ways.

They are composed in this way in order to remove the conditional behaviour of the CNOT-gates.

It is done in this way so that any initial state goes through a sequence of bit-flips in such a way that the states of the two qubits are swapped.

Classical vs Quantum Computation

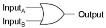
Aside from working from the principles of classical information vs quantum information and various results in algorithm runtimes and so on, there are some fundamental differences between classical and quantum computation which are interesting to look at.

We have seen that quantum gates are unitary. This implies they are **reversible**.

This is unlike classical computation where some gates are actually **irreversible** and lead to a loss of information.

An example is the classical XOR-gate vs the quantum CNOT-gate.

2-input OR gate



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Classical vs Quantum Computation

This is important because any arbitrary unitary gate can be approximated by elementary unitary gates. The only problem is that it can't be done efficiently.

Classical vs Quantum Computation

This is important because any arbitrary unitary gate can be approximated by elementary unitary gates. The only problem is that it can't be done efficiently.

Definition (Universality)

Def: The capability to approximate any unitary operator on an arbitrary number of qubits.

Definition (Universal Gate Set)

Def: A combination of single qubit gates and a universal two-qubit gate.

Classical vs Quantum Computation

This is important because any arbitrary unitary gate can be approximated by elementary unitary gates. The only problem is that it can't be done efficiently.

Definition (Universality)

Def: The capability to approximate any unitary operator on an arbitrary number of qubits.

Definition (Universal Gate Set)

Def: A combination of single qubit gates and a universal two-qubit gate.

Classical computers do have their own universality property but it involves Boolean functions and non-reversable gates.
To see a proof of universality on quantum computers see "Quantum Computation and Quantum Information - Chuang & Nielsen" Section 4.5.

Applications of Quantum Information

We have seen a lot about quantum information, but what can it actually do?
Here are some applications of quantum information mainly associated with quantum computing.

- Quantum Simulation
  - The nearest-term application with uses in quantum chemistry, drug development and material design.
- Quantum Cryptography
  - Using quantum mechanical ideas to encrypt data so it cannot be hacked or studying how quantum decryption algorithms can be used to hack classical encryption methods.
- Quantum Machine Learning
  - Utilising the advantages of quantum information combined with machine learning algorithms.
- Quantum Finance
  - Finding difficult to see patterns in complex financial data, studying the probabilistic fluctuations in financial markets.

Introduction to Qiskit

Qiskit is a Python based quantum software developed by IBM.

It is capable of creating and executing quantum circuits and algorithms on both the local classical device and IBM's quantum hardware.

Like any software Qiskit contains various modules and packages.

Qiskit contains individual application modules which are Qiskit Nature, Qiskit Finance, Qiskit Optimization and Qiskit Machine Learning. This makes performing the applications easier and also allows each module to be updated without affecting the others.

Other quantum softwares are available. Some options are OpenQasm(IBM), Cirq(Google), Q#(Microsoft) and Braket(Amazon).

Introduction to Qiskit

Let's now look at some important commands to construct quantum circuits and algorithms in Qiskit.

1. qc = QuantumCircuit(no. of qubits, no. of bits)
2. qc.*quantumgate*(qubit) (e.g qc.x(), qc.h(), qc.rx(angle,))
3. qc.cx(control qubit, target qubit)
4. qc.measure(qubit, bit)
5. qc.draw('mpl')
6. backend = Aer.get_backend('qasm_simulator')
7. job_sim = execute(qc, backend, shots=).result().get_counts()
8. plot_histogram(job_sim)

Quantum Teleportation

The first quantum algorithm we will look at is Quantum Teleportation.

To properly construct any quantum algorithm we need to know the physical process we are trying to simulate.

Quantum Teleportation

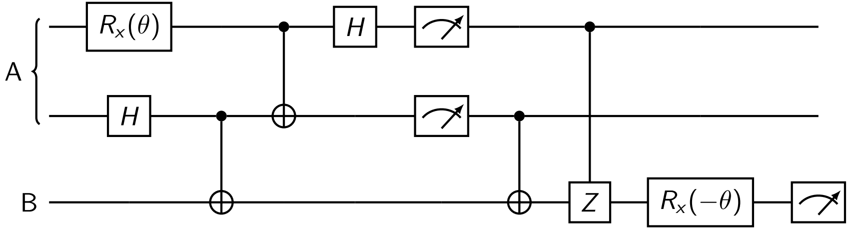
The first quantum algorithm we will look at is Quantum Teleportation.

To properly construct any quantum algorithm we need to know the physical process we are trying to simulate.

- We have two people(Person A & Person B).
- Person A wants to send the arbitrary quantum state of their qubit to Person B.
- This is done through quantum teleportation, which transmits the state of Person A's qubit to Person B through entanglement while collapsing the state of the original qubit so the no-cloning theorem isn't violated.

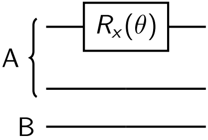
Quantum Teleportation

To perform quantum teleportation we will need 2 qubits for Person A, 1 qubit for Person B and 3 bits.



Quantum Teleportation

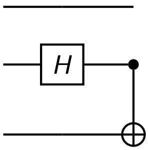
Let's take a more detailed look at what exactly is happening in each stage of this quantum circuit.



In the first stage of the circuit Person A is creating the arbitrary quantum state they want to send to Person B.

The state we want to send is now $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. The total state of the system is now $|\psi\rangle_{tot} = \alpha|000\rangle + \beta|100\rangle$.
Note: The ordering of states in the kets is in Qiskit notation(right to left) instead of how we would read it normally(left to right).

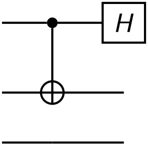
Quantum Teleportation



In this section of the circuit Person A uses their 2nd qubit to entangle with Person B's qubit to create a Bell state which is maximally entangled.
$|\psi\rangle_{tot} \rightarrow |0\rangle \otimes (H|0\rangle) \otimes (\alpha|0\rangle + \beta|1\rangle) \rightarrow |0\rangle \otimes \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right) \otimes (\alpha|0\rangle + \beta|1\rangle)$

After applying the CNOT gate the state becomes
$|\psi\rangle_{tot} = \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|110\rangle + \beta|001\rangle + \beta|111\rangle)$

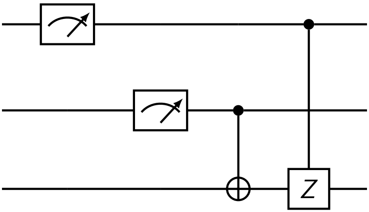
Quantum Teleportation



We now want to create a quantum state with a superposition of qubit 0 and qubit 1 being in the states $|00\rangle, |10\rangle, |01\rangle$ and $|11\rangle$, each of the possible measurement combinations. Applying the quantum gates shown on the left we get the new total state, $|\psi\rangle_{tot} = \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|110\rangle + \beta|011\rangle + \beta|101\rangle)$ after the CNOT gate. Then, after the Hadamard gate $|\psi\rangle_{tot} =$

$$\frac{1}{\sqrt{2}}\left[\alpha|00\rangle\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right) + \alpha|11\rangle\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right) + \beta|01\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) + \beta|10\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\right]$$

Quantum Teleportation



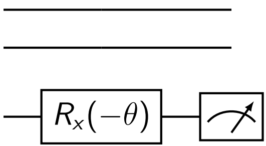
Factoring
our quantum state we see
$|\psi\rangle_{tot} = \frac{1}{2}[(\alpha|0\rangle + \beta|1\rangle)|00\rangle + (\alpha|1\rangle + \beta|0\rangle)|10\rangle + (\alpha|0\rangle - \beta|1\rangle)|01\rangle + (\alpha|1\rangle - \beta|0\rangle)|11\rangle]$
For successful
quantum teleportation we
want the state of the 3rd qubit to always be $\alpha|0\rangle + \beta|1\rangle$. What we measure qubit 0 and qubit 1 to be determines the operations we have to perform on qubit 3 to achieve this.

Quantum Teleportation

$$\begin{aligned} \text{Measure } |00\rangle &\implies \text{apply } I\text{-gate.} \\ \text{Measure } |10\rangle &\implies \text{apply } X\text{-gate.} \\ \text{Measure } |01\rangle &\implies \text{apply } Z\text{-gate.} \\ \text{Measure } |11\rangle &\implies \text{apply } ZX\text{-gates.} \end{aligned}$$

Quantum Teleportation

$$\text{Measure } |00\rangle \implies \text{apply } I\text{-gate.}$$
$$\text{Measure } |10\rangle \implies \text{apply } X\text{-gate.}$$
$$\text{Measure } |01\rangle \implies \text{apply } Z\text{-gate.}$$
$$\text{Measure } |11\rangle \implies \text{apply } ZX\text{-gates.}$$



This section of the circuit is just to check that our quantum teleportation algorithm was correct. The teleportation itself was completed after the previous section of the circuit.

Quantum Teleportation

Now it's up to you to code this circuit in Qiskit it and execute it on the noiseless qasm_simulator.
Begin by importing the following packages and commands.

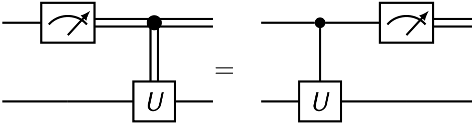
In [1]:

```
import numpy as np
from qiskit import *
from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister
from qiskit import IBMQ, Aer, transpile
from qiskit.visualization import plot_histogram, plot_bloch_multivector, array_to_latex, plot_state_city
from qiskit.extensions import Initialize
from qiskit.result import marginal_counts
```

After this all you will need is the commands on the other slide to create, execute and plot the results of this algorithm. There's a Quantum Teleportation section in the Qiskit textbook which you can use for reference.

Quantum Teleportation

On quantum hardware measurements can only be performed at the end of a circuit, not in the middle, so we'll need to use the deferred measurement principle.



Instead of having our controlled gates after the measurement we can move them to before the measurement.
Measuring early can have its advantages when running quantum algorithms on classical hardware or using quantum-classical hybrid algorithms on NISQ devices.

Quantum Teleportation

To run the algorithm on the hardware you will need to use this code.

- from qiskit import IBMQ
  provider = IBMQ.enable_account('API Token')
  IBMQ.providers()
- provider.backends()
- from qiskit.providers.ibmq import least_busy
  import qiskit.tools.jupyter
  HW = least_busy(provider.backends(filters=lambda b:
  b.configuration().n_qubits >= 3 and not b.configuration().simulator
  and b.status().operational==True))
  print(HW)
- bk_real = provider.get_backend('HW')
  counts_noisy_real=execute(qc,shots=8192,backend =
  bk_real).result().get_counts()
- plot_histogram(counts_noisy_real)

Superdense Coding

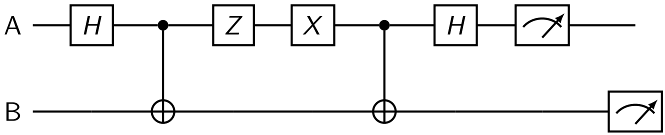
Superdense Coding is another quantum algorithm where it works in a way which is inverse to Quantum Teleportation. In Quantum Teleportation we teleported the state of 1 qubit with the help of 2 bits. In Superdense Coding we will tramsit 2 bits of information using 1 qubit.

Superdense Coding

Superdense Coding is another quantum algorithm where it works in a way which is inverse to Quantum Teleportation. In Quantum Teleportation we teleported the state of 1 qubit with the help of 2 bits. In Superdense Coding we will tramsit 2 bits of information using 1 qubit.

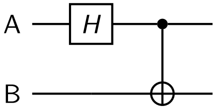
- We once again have Person A and Person B and they have one qubit each.
- Person A wants to send a message to Person B. This message is one of 00, 10, 01, or 11.
- This is achieved through entangling the 2 qubits, performing certain gates depending on the message to be encoded, and then performing measurements at the end.

Superdense Coding



The Z- and X-gates in this circuit are optional. They will have to be coded in an *if* loop depending on the message you are choosing to send.

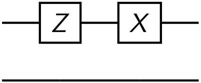
Superdense Coding



In this section of the circuit we create a Bell state.

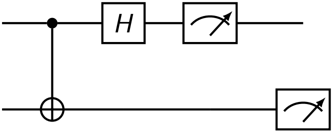
$$\begin{aligned} |\psi\rangle_{tot} &= \text{CNOT}[|0\rangle \otimes (H|0\rangle)] \\ &= \text{CNOT}\left[|0\rangle \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\right] \\ &= \frac{|00\rangle + |11\rangle}{\sqrt{2}} \end{aligned} \tag{18}$$

Superdense Coding



Message	Gate	Output$(\frac{1}{\sqrt{2}})$
00	I	$\|00\rangle + \|11\rangle$
10	X	$\|01\rangle + \|10\rangle$
01	Z	$\|00\rangle - \|11\rangle$
11	ZX	$-\|01\rangle + \|10\rangle$

Superdense Coding



Output($\frac{1}{\sqrt{2}}$)	CNOT	H
$\lvert 00\rangle + \lvert 11\rangle$	$\lvert 00\rangle + \lvert 01\rangle$	$\lvert 00\rangle$
$\lvert 01\rangle + \lvert 10\rangle$	$\lvert 11\rangle + \lvert 10\rangle$	$\lvert 10\rangle$
$\lvert 00\rangle - \lvert 11\rangle$	$\lvert 00\rangle - \lvert 01\rangle$	$\lvert 01\rangle$
$-\lvert 01\rangle + \lvert 10\rangle$	$-\lvert 11\rangle + \lvert 10\rangle$	$\lvert 11\rangle$

We apply the CNOT and H-gates to change our entangled state back into a pure state which is the message we want to send in quantum state form, which we then measure, resulting in the message in classical form.

Resources

Quantum Mechanics

Introduction to Quantum Mechanics - David J. Griffiths

Modern Quantum Mechanics - J. J. Sakurai

Quantum Information

Quantum Computation and Quantum Information - I. Chuang and M. Nielsen

Simulating Physics With Computers - Feynman(1982)

Universal Quantum Simulators - Lloyd(1996)

Qiskit

Qiskit textbook - `https://qiskit.org/learn`

Qiskit Youtube Channel - `https://www.youtube.com/c/qiskit`

Qiskit Slack - `https://join.slack.com/t/qiskit/shared_invite/zt-1dancl0jq-iR3zqEmEQa7pRsMN_Bmz1g`

Qiskit Github - `https://github.com/Qiskit`