Quantum Computing - Notes - v0.3.1

260236

April 2025

Preface

Every theory section in these notes has been taken from the sources:

- Course slides. [1]

About:

GitHub repository

These notes are an unofficial resource and shouldn't replace the course material or any other book on quantum computing. It is not made for commercial purposes. I've made the following notes to help me improve my knowledge and maybe it can be helpful for everyone.

As I have highlighted, a student should choose the teacher's material or a book on the topic. These notes can only be a helpful material.

Contents

1 Introduction

1.1 Complex Numbers recap

Complex Numbers play a fundamental role in quantum mechanics and quantum computing. For this reason, here is a brief summary of the most important concepts:

- **Definition of a Complex Number**. A complex number z is written as:

$$z = x + iy$$

  Where:

  - x is the **real part** ($\mathrm{Re}(z) = x$).
  - y is the **imaginary part** ($\mathrm{Im}(z) = y$).
  - i is the **imaginary unit**, satisfying $i^2 = -1$.

  A complex number can also be expressed in **polar form**:

$$z = re^{i\varphi}$$

  Where:

  - $r = |z| = \sqrt{x^2 + y^2}$ is the **modulus** (also called **magnitude**).
  - $\varphi = \arg(z) = \tan^{-1}\left(\frac{y}{x}\right)$ is the **argument** (also called **phase angle**).

  Using Euler's formula:

$$e^{i\varphi} = \cos\varphi + i\sin\varphi$$

  We can rewrite z as:

$$z = r\left(\cos\varphi + i\sin\varphi\right)$$

- **Complex Conjugate**. The **Complex Conjugate** of z is:

$$\bar{z} = x - iy = re^{-i\varphi}$$

  Properties:

  - $z \cdot \bar{z} = |z|^2 = \left(\sqrt{x^2 + y^2}\right)^2 = x^2 + y^2$
  - The **conjugate reverses the sign of the imaginary part**.

- **Operations on Complex Numbers**

  - **Addition** and **Subtraction**:

$$\begin{aligned}(a + ib) + (c + id) &= (a + c) + i\,(b + d)\\ (a + ib) - (c + id) &= (a - c) + i\,(b - d)\end{aligned}$$

  - **Multiplication**. Using the distributive property:

$$(a + ib)\,(c + id) = ac + iad + ibc + i^2bd$$

    Since $i^2 = -1$, we get:

$$(ac - bd) + i\,(ad + bc)$$

– **Division**. To divide $\frac{z_1}{z_2}$, multiply by the conjugate of the denominator:

$$\frac{a+ib}{c+id} = \frac{(a+ib)(c-id)}{c^2+d^2}$$

Expanding:

$$\frac{(ac+bd)+i(bc-ad)}{c^2+d^2}$$

- **Complex Exponentiation**. Using Euler's formula:

$$e^{i\theta} = \cos\theta + i\sin\theta$$

For integer powers:

$$(e^{i\theta})^n = e^{i\cdot n\cdot\theta}$$

For fractional exponents (roots):

$$z^{\frac{1}{n}} = r^{\frac{1}{n}} e^{\frac{i(\varphi+2\pi k)}{n}}, \quad k = 0, 1, \ldots, n-1$$

- **Rotation Using Complex Numbers**. Multiplying by $e^{i\psi}$ rotates a complex number by an angle ψ:

$$z' = ze^{i\psi}$$

Since:

$$e^{i\psi} = \cos\psi + i\sin\psi$$

This means:

$$\underbrace{(x+iy)}_{z}\underbrace{(\cos\psi + i\sin\psi)}_{e^{i\psi}}$$

Expanding:

$$(x\cos\psi - y\sin\psi) + i(x\sin\psi + y\cos\psi)$$

Thus, the new coordinates are:

$$x' = x\cos\psi - y\sin\psi, \quad y' = x\sin\psi + y\cos\psi$$

Which is a standard **2D rotation matrix**:

$$\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix} = \begin{bmatrix}\cos\psi & -\sin\psi\\ \sin\psi & \cos\psi\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}$$

And it rotates a point counterclockwise by an angle ψ in the 2D plane. This is important because rotations in the Bloch sphere (which represents qubits) are described by operations similar to this matrix.

- **Hermitian (Conjugate Transpose) of a Vector**. For a vector of complex numbers:

$$\mathbf{z} = \begin{bmatrix}a\\ b\end{bmatrix}$$

The **Hermitian conjugate** (denoted $\mathbf{z}^\dagger$ or $\mathbf{z}^H$) is:

$$\mathbf{z}^H = \begin{bmatrix}\bar{a} & \bar{b}\end{bmatrix}$$

Where $\bar{a}$ and $\bar{b}$ are the complex conjugates.

These concepts are fundamental because complex numbers describe quantum states. Also, Euler's formula provides a powerful tool for representing phase shifts. Finally, rotation and multiplication are key to quantum operations, and the hermitian conjugate is crucial in quantum mechanics.

1.2 Dirac's Notation

Dirac Notation, also called **bra-ket notation**, is a powerful mathematical framework used in quantum mechanics to **describe quantum states and their transformations**.

What is a Ket?

A **Ket** $|v\rangle$ (is equal to the linear algebra annotation $\overrightarrow{v}$) is a **column vector** in a Hilbert space, that **represents a quantum state**.

$$|v\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$$

Where a and b are complex numbers (amplitudes of the quantum state). In other words, a ket $|v\rangle$ is a **state vector**, and what we usually require is that it has **unit norm**, meaning:

$$\langle v|v\rangle = 1$$

This ensures that the total probability of measurement outcomes is 1. So, a ket is a normalized column vector in a Hilbert space, meaning it has unit norm.

What is a Superposition and why is it related to the Ket?

Superposition is a fundamental principle of quantum mechanics that **applies to all quantum systems**, not just qubits.

Definition 1: Superposition

Superposition means that a **system can exist in several possible states at the same time until a measurement is made**.

More in general, a quantum state $|\psi\rangle$ in a system with multiple possible states can exist in a **linear combination** of these states:

$$|\psi\rangle = c_1 |\psi_1\rangle + c_2 |\psi_2\rangle + \cdots + c_n |\psi_n\rangle$$

Where:

- $|\psi_1\rangle, |\psi_2\rangle, \ldots, |\psi_n\rangle$ are **basis states** of the system.
- $c_1, c_2, \ldots, c_n$ are **complex probability amplitudes**.
- The system is in **all states at the same time**, with the probability of measuring each state given by $|c_i|^2$.
- The state is **normalized**, meaning:

$$|c_1|^2 + |c_2|^2 + \cdots + |c_n|^2 = 1$$

Example 1: Single-Particle Systems (Quantum Mechanics)

In general quantum mechanics, a particle can exist in **multiple positions simultaneously** as a wave function $\Psi(x)$:

$$|\Psi\rangle = \int \Psi(x)\ |x\rangle\ \mathrm{d}x$$

- The particle is in a **superposition of all possible positions** $|x\rangle$.
- Measurement collapses the wave function to a single position.

How Can a Particle Exist in Multiple States at Once? This question touches the heart of quantum mechanics, where our everyday intuition breaks down. The **key idea** is that a **quantum particle** is not just a tiny ball, it **is a wave function that spreads across multiple possibilities at once**.

1. **Quantum Particles Are Waves, Not Just Points.** In classical physics, we think of particles as tiny, solid objects, like a small ball that always has a precise position and velocity.

   In quantum mechanics, however, particles behave more like waves. These waves are described by a wave function $\Psi(x)$, which represents the **probability of finding the particle at different locations**.

   The key idea is:

   - The **wave function spreads across space**, meaning the **particle does not have a single location** before measurement.
   - Instead, it exists in a superposition of all possible locations.

2. **The Double-Slit Experiment: Proof That a Particle Can Be in Two Places at Once.** The double-slit experiment demonstrates that light and matter can exhibit behavior of both classical particles and classical waves. In 1927, Davisson and Germer and, independently, George Paget Thomson and his research student Alexander Reid demonstrated that electrons show the same behavior, which was later extended to atoms and molecules.

   Double Slit Experiment explained! by Jim Al-Khalili

   What Happens in Classical Physics? If we throw tiny balls at a screen with two slits, each ball will pass through one slit or the other. After many throws, **we get two lines** behind the slits, corresponding to the **two possible paths**.

   What Happens in Quantum Mechanics? If we **send a single electron** (or photon) towards two slits, it **behaves like a wave**.

It **passes through both slits at the same time** and interferes with itself, creating an interference pattern. This means the electron was in a **superposition of passing through both slits at once**.

If we try to measure which slit the electron goes through, the **superposition collapses**, and it behaves like a classical particle!

3. **Quantum Superposition: More Than Just Probability**. A common misconception is that a particle in superposition is just **an unknown state**, like a coin that is either heads or tails, but we just don't know which. This is wrong, because quantum superposition is much deeper.

   A quantum state is a **combination of all possibilities**. *Until measurement*, the system is in **all possible states at once**.

   Mathematically, for an electron in **two locations** x_1 and x_2:

   $$|\psi\rangle = a\,|x_1\rangle + b\,|x_2\rangle$$

   - The electron is **literally in both places simultaneously**.
   - The coefficients a and b are **complex numbers representing the probability amplitudes**.
   - **Interference** between these amplitudes **creates quantum effects that cannot be explained by classical probability**.

4. **Superposition in Quantum Computing**. Quantum computing **directly uses** the fact that a particle can be in multiple states at once.

   - A **classical bit** can only be **0 or 1**.
   - A **qubit (quantum bit)** (we will explain this later) can be in a superposition:
     $$|\psi\rangle = a\,|0\rangle + b\,|1\rangle$$

   This means a quantum computer can **perform many calculations simultaneously**. Therefore, **superposition allows quantum computers to process information exponentially faster than classical computers for certain tasks**.

5. **Why Don't We See Superposition in Everyday Life?** In our daily experience, objects are **not in multiple states at once** because of a process called **quantum decoherence**.

   Quantum superposition is **fragile**. When a quantum system interacts with the environment (air, light, etc.), the superposition **collapses into one definite state**. This is why large objects (like humans or cars) do not appear in multiple places at once.

   However, experiments (like the Double-Slit experiment) confirm that superposition is real at microscopic scales (electrons, photons, atoms, and even molecules).

The key **properties of Superposition** are:

1. **Linearity**: Any combination of valid quantum states is also a valid quantum state.
2. **Interference**: Quantum states in superposition can interfere, leading to constructive or destructive interference.
3. **Measurement Collapse**: When measured, the superposition collapses into a single outcome.
4. **Phase Information**: Unlike classical probabilities, quantum superpositions include complex phases that affect interference patterns.

What is a Bra?

A **Bra** $\langle v|$ (is equal to the linear algebra annotation $\vec{v}^H$) is the **conjugate transpose (Hermitian conjugate) of the ket** $|v\rangle$.

Mathematically, if we start with the ket $|v\rangle$:

$$|v\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$$

The bra is obtained by:

1. **Transposing the ket** (switching it from column to a row).
2. **Taking the complex conjugate of each element**.

So the bra $\langle v|$ is:

$$\langle v| = (\langle v|)^\dagger = \begin{bmatrix} \bar{a} & \bar{b} \end{bmatrix}$$

A bra is a **mathematical object that allows us to compute inner products and measure probabilities**.

But why do we need another topic called "Bra"?

A **bra** $\langle v|$ does not represent a physical system by itself. Instead, it is a **mathematical tool** used to:

1. **Extract** information from a quantum state.
2. **Compute** inner products (which determine probabilities).
3. **Define** quantum operators and measurements.

The key idea of bras and kets working together is:

- A ket $|v\rangle$ represents a quantum system.
- A bra $\langle v|$ is like a *test function* that helps us extract measurable information from a quantum system.

When they are combined as $\langle v|v\rangle = 1$, we obtain a **probability amplitude**.

Dirac's Notation: Multiplications

The following list shows how Dirac's notation is used to describe how kets and bras interact through inner products, matrix-vector multiplications, and operator applications:

- **Inner (Scalar) Product:** $\langle x|y\rangle$

  The **inner product** (or scalar product) between two quantum states $|x\rangle$ and $|x\rangle$ is written as:

  $$\langle x|y\rangle$$

  Where:

  - The **bra** $\langle x|$ is the **conjugate transpose** (row vector) of the ket $|x\rangle$.
  - The **ket** $|y\rangle$ is a **column vector**.
  - The inner product is the **dot product of these two vectors**, resulting in a **scalar (complex number)**.

  The inner product tells us **how much two quantum states "overlap"**.

  - If $\langle x|y\rangle = 0$, the states are **orthogonal** (completely different).
  - If $\langle x|y\rangle = 1$, the states are **identical**.

  Example 2: Inner (Scalar) Product

  Suppose we have two quantum states:

  $$|x\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$$
  $$|y\rangle = \begin{bmatrix} c \\ d \end{bmatrix}$$

  Then:

  $$\langle x|y\rangle = \begin{bmatrix} \bar{a} & \bar{b} \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} = \bar{a}c + \bar{b}d$$

- **Matrix-Ket Multiplication:** $M|v\rangle$

  A quantum system evolves by **applying a matrix** (operator) M **to a quantum state** (ket):

  $$M|v\rangle$$

  Where:

  - $|v\rangle$ is a **column vector** (a quantum state).
  - M is a **matrix** (a quantum operator).

  The result is a **new quantum state** (a transformed column vector).

  Matrix-Ket Multiplication shows how **quantum gates (unitary matrices) transform quantum states**.

Example 3: Matrix-Ket Multiplication

Let's take:

$$M = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad |v\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$$

Then:

$$M\,|v\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} b \\ a \end{bmatrix}$$

- **Concatenated Multiplications:** $\langle x|M|y\rangle$

  A general quantum mechanical expression is:

  $$\langle x|M|y\rangle$$

  This is the **expected value or transition amplitude**, which means:

  1. Apply the operator M to $|y\rangle$ first:

     $$M\,|y\rangle$$

     Which gives a **new quantum state**.
  2. Take the inner product with $\langle x|$:

     $$\langle x|\,(M\,|y\rangle)$$

     Which results in a scalar (complex number).

  The result is a **scalar** that tells us the **probability amplitude of transitioning from $|y\rangle$ to $|x\rangle$ via M.**

Example 4

Let's compute:

$$\langle x|M|y\rangle$$

Using:

$$|x\rangle = \begin{bmatrix} a \\ b \end{bmatrix} \qquad |y\rangle = \begin{bmatrix} c \\ d \end{bmatrix} \qquad M = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

1. Compute $M\,|y\rangle$:

   $$M\,|y\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} d \\ c \end{bmatrix}$$

2. Compute: $\langle x|\,(M\,|y\rangle)$:

   $$\langle x|\,(M\,|y\rangle) = \begin{bmatrix} \bar{a} & \bar{b} \end{bmatrix} \begin{bmatrix} d \\ c \end{bmatrix} = \bar{a}d + \bar{b}c$$

1.3 Single Qubits

A **Qubit** is a **two-level quantum system**, meaning it **has only two basis states**:

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

Any quantum state of a single qubit can be written as a **linear combination (*superposition*)** of these two basis states:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \qquad \alpha \wedge \beta \neq 0$$

Where:

- α, β are **complex numbers** called **Probability Amplitudes**. If either $\alpha = 0$ or $\beta = 0$, the **state is NOT in superposition** (it is a pure basis state).
- The **normalization condition** holds:

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

  To ensure total probability is 1.

Therefore, a single qubit is described as a **2D complex vector** in a Hilbert space.

Matrix Representation of Qubit States

Quantum states can be expressed in **matrix form** as **column vectors**:

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

And, a general qubit state is:

$$|\psi\rangle = \begin{bmatrix}\alpha\\\beta\end{bmatrix}$$

Where α and β are complex numbers.

What is a Basis?

A **Basis** is a **set of vectors that define a coordinate system in which we describe quantum states**. Furthermore, basis should **always be orthonormal** (orthogonal and norm equal to one) because it ensures that quantum states are independent, complete, and allow meaningful probability calculations. For a **single qubit**, we typically use **two orthonormal basis states** $|0\rangle$ and $|1\rangle$, forming the **computational basis**:

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

Any qubit state can be written as:

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

Where α and β are complex numbers satisfying $|\alpha|^2 + |\beta|^2 = 1$.

In other words, a **basis allows us to describe quantum states as linear combinations of simpler states**.

Why do we have a choice of different bases and why should we choose them?

While the **computational basis** $\{|0\rangle, |1\rangle\}$ is the *standard*, we are **not forced** to use it! We can choose **other bases** depending on the situation, and they help in different computations. This is because a **different basis simply provides a new way to describe the same quantum state**.

Another common basis is the **Hadamard basis**, defined as:

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

The basis is also important because the **choice of basis affects the measurement results**. For example, let's take the qubit state:

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

- If measured in the **computational basis** $\{|0\rangle, |1\rangle\}$, it has a 50% chance of collapsing to $|0\rangle$ and 50% to $|1\rangle$.
- If measured in the **Hadamard basis** $\{|+\rangle, |-\rangle\}$, it **always collapses to** $|+\rangle$. Then, the probability of collapsing into $|+\rangle$ is 100%, and the probability of collapsing into $|-\rangle$ is 0%.

  Proof. We measure $|\psi\rangle$ in the Hadamard basis, therefore we must express it using $|+\rangle$ and $|-\rangle$.

  Since:

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

  If we manipulate $|\psi\rangle$ a bit, we can see this:

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

  And there is no component of $|-\rangle$. QED

What happens when we measure a Qubit?

In classical computing, a bit is either 0 or 1. In quantum computing, a qubit exists in a **superposition** of $|0\rangle$ and $|1\rangle$, but **when measured, it collapses into one of these basis states**. This is because the measurement is probabilistic and destroys the superposition.

If a qubit is in state:

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

Measurement **forces the qubit to collapse** into either $|0\rangle$ or $|1\rangle$. The **probability of each outcome** is given by the **squared magnitudes** of the coefficients:

$$P\,(0) = |\alpha|^2 \qquad P\,(1) = |\beta|^2$$

After measurement, the qubit **loses superposition** and remains in the measured state.

Example 5: Qubit Measurement

Consider the qubit:

$$|\psi\rangle = \frac{3}{5}\,|0\rangle + \frac{4}{5}\,|1\rangle$$

- The probability of measuring $|0\rangle$ is:

$$P\,(0) = \left(\frac{3}{5}\right)^2 = 0.36$$

- The probability of measuring $|1\rangle$ is:

$$P\,(1) = \left(\frac{4}{5}\right)^2 = 0.64$$

If we **measure** the qubit:

- With 36% probability, it collapses to $|0\rangle$.
- With 64% probability, it collapses to $|1\rangle$.

After measurement, the qubit **remains in that state** until modified by another operation.

In the previous example, we can observe that the measurement **collapses** the quantum state **into one of the basis** states with a **probability determined by its amplitude**.

What happens if we measure twice?

If the qubit collapses to $|0\rangle$ in the first measurement, a second measurement in the same basis will return $|0\rangle$ with probability 1. This is because the qubit is already in $|0\rangle$ and has no component of $|1\rangle$ left. Therefore, **repeating a measurement in the same basis always gives the same result**.

Measurement as a Fundamental Axiom

The behavior of quantum measurement is **not derived from other principles**, it is an **axiom of quantum mechanics**:

1. Measurement collapses the quantum state.
2. The probability of each outcome is given by the squared amplitude.
3. A second measurement (in the same basis) gives the same result with probability 1.

Therefore, the measurement is a **fundamental rule** of quantum mechanics.

Fundamental limitation of Quantum Computing

Unlike a classical bit, which can be only 0 or 1, a qubit can exist in any superposition:

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle$$

Where a and b are complex numbers that satisfy $|a|^2 + |b|^2 = 1$. Since a and b can take infinitely many values, **a single qubit *theoretically* has an infinite number of possible states**.

Can a Qubit Store More than One Classical Bit? One might hope that because a qubit has infinitely many states , it could **store and transmit more than one classical bit of information**. However, **this is not possible** because:

- ✗ **A single measurement only gives one classical bit**. Measuring $|\psi\rangle$ forces it to collapse into $|0\rangle$ or $|1\rangle$. The outcome follows probabilities $P(0) = |a|^2$ and $P(1) = |b|^2$. Since the result is **just one binary outcome**, it **cannot reveal both a and b at the same time**.
- ✗ **Measurement Destroys the Quantum State**. Once we measure a qubit, its **original state is lost**. This means we cannot measure a and b separately, even if we repeat the measurement.

Unfortunately, this is a limitation, because a single qubit contains **infinite information theoretically**, but in practice, we can **only extract one classical bit per measurement**.

Furthermore, another problem is that we cannot copy a quantum state. The no-cloning theorem (explained later) states that it is **impossible to perfectly copy an arbitrary quantum state**. This has major consequences:

- **We cannot measure a qubit's twice**. In classical computing, we can copy and measure a bit multiple times. In quantum computing, copying is not possible. Once a qubit is measured, the original superposition is destroyed.
- *Why can't we just copy and measure?* Suppose we want to copy a qubit $|\psi\rangle$ and measure both copies. **Quantum mechanics forbids perfect duplication of unknown quantum states**. This prevents duplicating quantum information and extracting more than one bit of classical information per qubit. A proof will be provided later with a better explanation.

The state space of a Single Qubit

A **single qubit exists in a two-dimensional complex Hilbert space**:

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

Where:

- α and β are complex numbers (probability amplitudes).
- The normalization condition ensures:

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

Since α and β can take complex values, a **qubit is more than just a point** in 2D, it has **four real parameters** (two from each complex number). However, due to normalization and global phase invariance, **only two real parameters are needed to describe a qubit**.

Therefore, the **space of all possible qubit states is a continuous space, not just discrete values like classical bits**.

Block Sphere representation of the qubit (and why)

The **bloch sphere** is a geometric representation of a qubit's state that helps visualize its properties. Since a general qubit state is:

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

We can rewrite it using two angles θ and ϕ (as **spherical coordinates**):

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

Where:

- θ is called **polar angle** (latitude). It determines how much of $|0\rangle$ and $|1\rangle$ are mixed (the qubit is not just in one state, but in both simultaneously).
  - When $\theta = 0 \to |\psi\rangle = |0\rangle$
    It is a pure state.
  - When $\theta = \pi \to |\psi\rangle = |1\rangle$
    It is a pure state.
  - When $\theta = \frac{\pi}{2} \to |\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + e^{i\phi}\frac{1}{\sqrt{2}}|1\rangle$
    It is called equal superposition.
- ϕ is called **relative phase** (longitude). It **controls the phase relationship** between $|0\rangle$ and $|1\rangle$. Changing ϕ **does not affect measurement probabilities**, but it affects interference when qubits interacts with other qubits.

In general, on the block sphere:

- The **north pole** $(0, 0, 1)$ is $|0\rangle$.
- The **south pole** $(0, 0, -1)$ is $|1\rangle$.
- Any other point represents a **superposition state** (eq. 1, page 17).

So, the Bloch sphere shows **how a qubit evolves** under quantum operations, making it easier to understand **rotations**, **phase shifts**, and **measurements**.

Why is the bloch sphere important? The bloch sphere **helps us visualize** superposition, phase, and quantum operations intuitively.

1. It gives a visual representation of qubit states. Classical bits are just points (0 or 1), whereas a qubit exists everywhere on the sphere.
2. It shows quantum gates as rotations. As we will see in the following pages, quantum gates rotate the qubit around the sphere. For example, the Hadamard gate rotates $|0\rangle$ to $|+\rangle$, moving from the north pole to the equator.
3. It helps understand measurement. Measuring a qubit collapses it to either $|0\rangle$ or $|1\rangle$, removing phase information.

Unfortunately, the $e^{i\phi}$ factor does not affect the global sphere?

Wrong! Two quantum state vectors are considered **equivalent if they differ only by a global phase factor**. This means that if we have two quantum states:

$$\begin{aligned} |\psi\rangle &= \alpha\,|0\rangle + \beta\,|1\rangle \\ |\psi'\rangle &= e^{i\gamma}\,(\alpha\,|0\rangle + \beta\,|1\rangle) \end{aligned}$$

Where $e^{i\gamma}$ is a **global phase factor** (a complex number with magnitude 1), then **these two states are physically identical**.

A **global phase** is a complex factor of the form:

$$e^{i\gamma} = \cos\gamma + i\sin\gamma$$

Which multiplies the entire quantum state but has no physical impact on measurement probabilities.

Therefore, a qubit state is **not changed** by multiplying it by a global phase factor $e^{i\gamma}$, meaning that:

$$|\psi\rangle \sim e^{i\gamma}\,|\psi\rangle$$

This means that the **block sphere represents only unique qubit states**, since the global phase doesn't affect the measurement.

Example 6: Identical Qubit States

Suppose we have two quantum states:

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

And:

$$|\psi'\rangle = e^{1\pi}\left(\frac{1}{\sqrt{2}}\,|0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle\right)$$

Since $e^{i\pi} = -1$, we can simplify:

$$|\psi'\rangle = -\frac{1}{\sqrt{2}}\,|0\rangle - \frac{1}{\sqrt{2}}\,|1\rangle$$

Even though $|\psi\rangle$ and $|\psi'\rangle$ look different mathematically, they are physically the same because they only differ by a global phase factor $e^{i\pi}$.
The global phase does not affect the measurement results. In fact, the probabilities remain for both ψ and ψ':

$$P(0) = \left|-\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2} \qquad P(1) = \left|-\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}$$

Since measurement gives the same results for both states, we consider them physically identical.

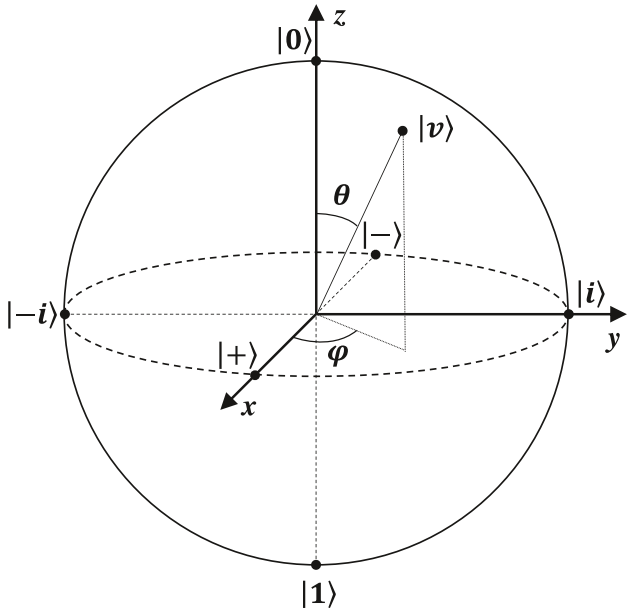


Figure 1: Bloch sphere representation of qubit.

2 Single Qubit Gates

2.1 Operations on Qubits

In quantum computing, **operations on qubits** are fundamental to information processing and computation. These operations can be broadly **classified** into:

- **Logic Gates**
- **Measurements**
- **Initialization Procedures**

Each of these plays a distinct role in the manipulation and evolution of quantum states, which are represented as **superpositions** of basis states $|0\rangle$ and $|1\rangle$.

Logic Gates: Reversible Transformations

Quantum **Logic Gates** serve as the **building blocks of quantum circuits**. Unlike classical logic gates, quantum gates operate under the principles of quantum mechanics, notably **unitarity** and **reversibility**. These gates can act on a single qubit or on multiple qubits simultaneously, and their **primary function is to transform the state of the qubit(s)** involved.

Key properties of quantum logic gates include:

- They are **reversible**: this implies that any operation performed by a gate can be undone by applying its inverse. In quantum mechanics, reversibility is linked to the **requirement that gates be unitary matrices**.
- They **preserve the norm** of the quantum state, a direct consequence of unitarity.
- Gates **change the qubit's state** in a way that aligns with the no-cloning and no-deleting theorems; in other words, information is neither duplicated nor destroyed, only transformed.

Measurement: Irreversible Collapse of State

Measurement is a fundamentally different operation from logic gates because it is **irreversible**. When a qubit is measured, **information about its state is extracted**, and as a consequence, the **qubit's quantum state collapses** to one of the basis states $|0\rangle$ or $|1\rangle$. This collapse causes the **loss of superposition** and, if entanglement is involved, the **loss of entanglement** as well (explained later).

Important points:

- Measurement provides **probabilistic outcomes based on the amplitudes** of the quantum state.
- After measurement, the qubit is **no longer in a superposition**; it is deterministically in $|0\rangle$ or $|1\rangle$.

- This operation is **non-unitary** and hence **non-reversible**.

Initialization: Preparing Known States

The **Initialization** of qubits is the process of **setting them into a known, well-defined state**, most commonly $|0\rangle$ or $|1\rangle$. Initialization is essential because quantum algorithms require precisely defined input states.

Key facts:

- Initialization can *often* be **implemented via measurement**: by measuring a qubit and then applying a gate if necessary, we can prepare it in the desired state.
- As with measurement, initialization **collapses the quantum state**, making it **irreversible**.
- While measurement discards the quantum coherence, **initialization ensures a clean starting point** for computations.

Comparison between Initialization and Measurement

- **Purpose**
  - **Measurement**: To extract information about the qubit's state.
  - **Initialization**: To prepare the qubit in a known state ($|0\rangle$ or $|1\rangle$).
- **Reversibility**
  - **Measurement**: Irreversible, collapses superposition.
  - **Initialization**: Irreversible, collapses superposition to set a state.
- **Effect on Qubit**
  - **Measurement**: Collapses to $|0\rangle$ or $|1\rangle$ randomly (probabilistic).
  - **Initialization**: Collapses to a specific state, often $|0\rangle$ (deterministic or controlled).
- **Information Gained**
  - **Measurement**: Yes, outcome of measurement is known.
  - **Initialization**: No, goal is not to gain information, just set state.
- **How Implemented**
  - **Measurement**: Via a measurement operator.
  - **Initialization**: Often via measurement + gate correction.

- **Post-operation Use**
  - **Measurement**: Qubit may be discarded or reused depending on outcome.
  - **Initialization**: Qubit is now ready for computation.

In other words:

- **Measurement** tells we **what state the qubit is** in.
- **Initialization sets the qubit** to a desired state so we can **start a computation**.

2.2 Quantum Logic Gates Overview

In quantum computing, quantum logic gates are the primary tools used to **manipulate qubits**. These gates perform **unitary transformations** on the state of one or more qubits, meaning they **preserve the norm** of the quantum state and are **reversible**. The design of quantum algorithms and the execution of quantum circuits rely entirely on the sequential application of these gates.

Types of Quantum Gates

Quantum Gates can be classified into:

- **Single-Qubit Gates**: These gates operate on **individual qubits**, modifying their state in isolation.
- **Multiple-Qubit Gates**: These gates act on **two or more qubits simultaneously**, allowing for entanglement and complex correlations between qubits.

In this section we will focus on single qubit gates.

Mathematical Framework: Qubit as Vectors, Gates as Matrices

A key distinction of quantum logic compared to classical logic is that **qubits are vectors**, while **quantum gates are matrices**.

- A **qubit's state**, as we discussed on page 12, is a **vector in a 2-dimensional complex vector space**, often written as:

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle \qquad \text{with } |a|^2 + |b|^2 = 1$$

- A **Quantum Gate** that acts on a single qubit is a 2×2 **complex matrix**, specifically a **unitary matrix**, denoted by U.

  The **action of a gate** U on a qubit $|\psi\rangle$ is simply **matrix-vector multiplication**:

$$|\psi'\rangle = U\,|\psi\rangle \tag{2}$$

  This transforms the qubit into a **new quantum state**, while ensuring that **probabilities remain normalized**.

Unitarity: The Fundamental Constraint

A quantum gate **must be unitary**, which means:

$$U^\dagger U = UU^\dagger = I \tag{3}$$

Where:

- $U^\dagger$ is the **Hermitian transpose** (conjugate transpose) of U.
- I is the **identity matrix**.

Why is it important? Because **unitarity ensures reversibility**: every gate has an inverse, and information is never lost, only transformed. This is in contrast to measurement, which is non-unitary and irreversible.

Why must gates be unitarity?

1. **No-Cloning Theorem**: It is impossible to create an identical copy of an unknown qubit. Unitary transformations respect this by not duplicating information.
2. **No-Deleting Theorem**: We cannot delete quantum information arbitrarily. Since unitarity gates are invertible, they do not erase information.
3. **Preservation of Probability**: The normalization condition $|a|^2 + |b|^2 = 1$ must hold after any operation, and only unitarity matrices preserve this norm.

Physical Intuition

In classical circuits, logic gates like AND, OR, NOT process bits deterministically. In quantum circuits:

- Gates **rotate** the quantum state on the **bloch sphere**.
- These rotations correspond to **unitary transformations**.
- The **physical implementation** of quantum gates may vary across hardware (e.g., superconducting qubits, trapped ions), but **mathematically**, the **same unitary matrix describes the gate**.

Quantum gates do not correspond to physical gates in the classical sense. They are **abstract operations** realized by **controlled interactions** in quantum systems.

2.3 Main Single-Qubit Gates

2.3.1 Identity Gate (I)

The **Identity Gate**, denoted I, is the most elementary gate in quantum computing. Its action on a qubit is trivial, yet it **serves important roles in quantum circuits**, particularly when managing multi-qubit systems or timing synchronization.

Matrix Representation

The Identity gate is represented by the 2×2 **identity matrix**:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{4}$$

Applying the Identity gate to any qubit **leaves the state unchanged**. Formally, for a qubit in state $|\psi\rangle$:

$$I\,|\psi\rangle = |\psi\rangle$$

Physical Interpretation

Although it may appear useless at first glance, the Identity gate has **conceptual and practical utility**:

- ✔ It acts as a **placeholder** or "***do nothing***" **operation** when required by **circuit timing**.
- ✔ In **multi-qubit circuits**, it allows one **qubit to remain unchanged** while other qubits are operated on.
- ✔ In **matrix composition** of gates, it serves to **align dimensions** when performing **tensor products** (e.g., $I \otimes H$ applies Hadamard to the second qubit only).

Bloch Sphere Perspective

In the Bloch sphere (remember? page 19), the **identity gate does not rotate**. The **qubit's position** on the sphere remains **fixed**, reinforcing the idea that no transformation has taken place.

2.3.2 Pauli-X (NOT) Gate

The **Pauli-X Gate**, often called the **quantum NOT gate**, is one of the *fundamental* single-qubit quantum gates. It is denoted X and **corresponds to the σ_X Pauli matrix** in quantum mechanics. This gate **flips the state** of a qubit, **similar to how a classical NOT gate inverts a bit**.

Remark: Pauli Matrix

In quantum mechanics, the **Pauli matrices** are **a set of three** 2×2 **Hermitian and unitary matrices** that represent **spin operators** for spin-$\frac{1}{2}$ particles and are widely used in quantum computing to describe single-qubit gates. They are:

$$\begin{aligned} \sigma_X &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ \sigma_Y &= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \\ \sigma_Z &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \end{aligned}$$

Key properties:

- Hermitian: $\sigma_i^\dagger = \sigma_i$ (observable-related)
- Unitary: $\sigma_i^\dagger \sigma_i = I$ (reversible transformations)
- Involution: $\sigma_i^2 = I$
- Non-commuting: $[\sigma_i, \sigma_j] = 2i\varepsilon_{ijk}\sigma_k$ (with ε the Levi-Civita symbol)

Matrix Representation

$$X = \sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{5}$$

This **matrix swaps** the $|0\rangle$ and $|1\rangle$ **basis states**:

$$X|0\rangle = |1\rangle \qquad \text{and} \qquad X|1\rangle = |0\rangle \tag{6}$$

Action on a General Qubit

Let's consider a qubit in the general state:

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

Applying the X gate:

$$X|\psi\rangle = aX|0\rangle + bX|1\rangle = a|1\rangle + b|0\rangle = b|0\rangle + a|1\rangle$$

As a result, the amplitudes a and b are **swapped**.

Geometric Interpretation: Rotation on the Bloch Sphere

The Pauli-X gate corresponds to a **rotation around the x-axis by π radians** ($180°$) on the Bloch sphere.

To understand this, recall the **general qubit state in spherical coordinates**:

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

After a π rotation around the x-axis:

- $\theta \to \pi - \theta$
- $\phi \to -\phi$

This **rotation mirrors the qubit across the x-axis**, flipping its position between $|0\rangle$ and $|1\rangle$, and adjusting the phase accordingly.

Key Properties of X Gate

1. **Involution**. Applying X twice returns the original state:

   $$XX = X^2 = I$$

   This **reflects reversibility** and **matches the intuition of two** $180°$ **rotations** around x-axis equaling no rotation.

2. **Entanglement Tool**. The X gate is crucial in **entangling qubits** when combined with CNOT in multi-qubit circuits.

3. **Measurement Preparation**. Used to **flip the measurement outcome**, e.g., preparing $|1\rangle$ from $|0\rangle$.

2.3.3 Pauli-Z (Phase Flip) Gate

The **Pauli-Z Gate**, also called the **Phase Flip Gate**, is one of the core **Pauli operators** used in quantum computing. It acts **only on the phase** of the qubit, **leaving the $|0\rangle$ amplitude unchanged** but **flipping the sign of the $|1\rangle$ amplitude**.

Matrix Representation

$$Z = \sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{7}$$

Action on a General Qubit

For a general state:

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle$$

Applying Z:

$$Z\,|\psi\rangle = aZ\,|0\rangle + bZ\,|1\rangle = a\,|0\rangle - b\,|1\rangle$$

As a result, Z **flips the sign of the $|1\rangle$ amplitude**, this is why it's called a **phase flip**. Finally, it's trivial to show that:

$$Z\,|0\rangle = |0\rangle \quad \text{(unchanged)}, \qquad Z\,|1\rangle = -\,|1\rangle \quad \text{(sign flip)}$$

Geometric Interpretation: Rotation around the z-axis

On the Bloch sphere, the Pauli-Z gate performs a π **radians** $(180°)$ **rotation around the z-axis**. In spherical coordinates:

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

After Z rotation:

- The angle $\phi \to \phi + \pi$
- This phase shift **changes the sign of the complex amplitude for $|1\rangle$**

Since **measurement probabilities depend on $|a|^2$ and $|b|^2$**, which remain unchanged, the Z gate **does not affect measurement outcomes** in the standard basis. It only affects **interference and future gate interactions**.

Interpretation and Use Cases

- **Phase Shift**: Z is a special case of a phase shift gate, shifting the phase by π.
- **No effect on $|0\rangle$**: Useful for controlled operations, where only $|1\rangle$ paths acquire a phase.

- **Circuit Simplification**: Often used in **error correction** and **phase-sensitive algorithms**.

Key Properties

1. **Involution**: $Z^2 = I$
2. **Diagonal Matrix**: Easy to compute and simulate.
3. **Commutes with Z and Phase gates**, but **anticommutes with X and Y**.

2.3.4 Pauli-Y Gate

The **Pauli-Y Gate**, denoted Y, is one of the three fundamental **Pauli matrices** in quantum mechanics. It is less intuitive than Pauli-X and Z, but equally important, especially for its role in complex rotations and quantum interference.

Matrix Representation

$$Y = \sigma_Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

Like the other Pauli gates, Y is **Hermitian** and **unitary**, which guarantees that it is **observable** (measurable) and **reversible**.

Given a qubit in state:

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle$$

Apply the Y gate:

$$Y\,|\psi\rangle = aY\,|0\rangle + bY\,|1\rangle = a\,(-i\,|1\rangle) + b\,(i\,|0\rangle) = ib\,|0\rangle - ia\,|1\rangle$$

So the **amplitudes are swapped and multiplied by** $\pm i$. This shows Pauli-Y **introduces a complex phase shift** alongside the swap.

Geometric Interpretation: Rotation on the Bloch Sphere

The Pauli-Y gate corresponds to a **rotation around the y-axis by π radians** ($180°$). The effect on Bloch sphere coordinates:

- $\theta \to \pi - \theta$
- $\psi \to \pi - \psi$

This rotates the qubit across the y-axis, **altering both amplitudes and their phases**.

Pauli-Y as a Composite Gate

Interestingly, $Y = iXZ$. Let's verify this:

$$XZ = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Now, multiply by i:

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

This identity confirms that **Pauli-Y can be viewed as a combination of Pauli-X and Pauli-Z** (page 28) **with a global phase factor** i, which does not affect measurement outcomes.

Key Properties of Y Gate

1. Involution: $Y^2 = I$
2. Anticommutation: $XY = -YX$
3. Phase Sensitive: introduces imaginary coefficients, unlike X and Z.

2.3.5 Phase Gate (S)

The **Phase Gate (S)** is a single-qubit gate that **adds a phase shift of $\frac{\pi}{2}$ (90°) to the amplitude of the $|1\rangle$ state**. It **leaves $|0\rangle$ unchanged**, like the Pauli-Z gate, but instead of a sign flip (π phase), it **introduces a complex phase factor** i.

Matrix Representation

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \tag{8}$$

Action on a General Qubit

For a state:

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle$$

Apply S:

$$S\,|\psi\rangle = a\,|0\rangle + ib\,|1\rangle$$

Only the $|1\rangle$ **amplitude gains a phase of** i, affecting interference but **not measurement probabilities** in the computational basis.

$$S\,|0\rangle = |0\rangle \quad \text{(unchanged)}, \qquad S\,|1\rangle = i\,|1\rangle \quad \left(\text{phase } + \frac{\pi}{2}\right)$$

Geometric Interpretation: Rotation around z-axis $\left(\frac{\pi}{2}\right)$

On the Bloch sphere, S performs a rotation of $\frac{\pi}{2}$ radians around the z-axis. This rotation **shifts the phase angle** $\phi \to \phi + \frac{\pi}{2}$, altering **how the qubit interferes** in later operations, especially when **Hadamard gates** are involved.

Relationship to Pauli-Z

Two applications of S equal Z:

$$S \cdot S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & i^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = Z$$

Thus, $S^2 = Z$, this is a **useful identity** in circuit simplifications.

Applications and Significance

- **Superposition Control**: S **adjusts relative phases**, crucial in algorithms like Quantum Fourier Transform and Grover's Search.
- **Building Block**: S helps construct **higher-order phase gates**, such as the T gate, which introduces a $\frac{\pi}{4}$ phase.
- **Error Correction**: Used in **stabilizer codes** for **fault-tolerant quantum computing**.

Key Properties

1. **Unitary**: $S^\dagger S = I$
2. **Not Hermitian**: Unlike Pauli gates, $S \neq S^\dagger$
3. **Diagonal Matrix**: Efficient in simulation.

2.3.6 Hadamard Gate (H)

The **Hadamard Gate (H)** **transforms a qubit from a definite state** ($|0\rangle$ or $|1\rangle$) **into a superposition of both**. It is the **gateway to quantum parallelism**, allowing quantum computers to **process multiple possibilities simultaneously**.

Matrix Representation

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{9}$$

This matrix is **unitary** and **Hermitian**: $H^\dagger = H$ and $H^2 = I$.

Action on Basis States

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

The $|+\rangle$ and $|-\rangle$ **states** are known as the **Hadamard basis**, or **superposition states**, with **equal probability amplitudes** for $|0\rangle$ and $|1\rangle$.

Bloch Sphere Interpretation

On the Bloch sphere, the **Hadamard gate performs two sequential rotations**:

1. Rotation around the y-axis by $\frac{\pi}{2}$ radians ($90°$).
2. Followed by a rotation around the x-axis by π radians ($180°$).

This brings $|0\rangle$ to the $+x$-axis ($|+\rangle$) and $|1\rangle$ to the $-x$-axis ($|-\rangle$). As a result, **qubit moves from pole to equator**, entering **maximum superposition**.

Superposition and Measurement

After applying H to $|0\rangle$, the resulting qubit is:

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

Upon measurement:

- **Probability of 0**:
$$\left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}$$
- **Probability of 1**:
$$\frac{1}{2}$$

This **equal probability** reflects **maximum uncertainty**, a hallmark of **quantum superposition**.

Key Properties of H Gate

1. **Involution**: $H^2 = I \rightarrow$ applying H **twice restores the original state**.
2. **Self-adjoint**: $H = H^\dagger \rightarrow$ H is its own **inverse and adjoint**.
3. **Basis Change**: H converts between the **computational basis** $\{|0\rangle, |1\rangle\}$ and the **Hadamard basis** $\{|+\rangle, |-\rangle\}$.

Applications

- **Quantum Parallelism**: Prepares qubits for **simultaneous computation paths**.
- **Quantum Algorithms**: Found in Grover's Search, Quantum Fourier Transform, Shor's Algorithm.
- **Entanglement Creation**: Combined with CNOT, H creates Bell states.
- **Interference**: Enables **constructive and destructive interference** in algorithms.

2.4 Properties

Single-qubit gates exhibit a rich set of mathematical and geometric properties, all of which stem from their nature as unitary transformations on a two-level quantum system. Understanding these properties is key to **predicting gate behavior**, **designing quantum circuits**, and **analyzing quantum algorithms**.

1. All Quantum Gates Are Equivalent to Rotations

A profound insight in quantum computing is that **every single-qubit unitary gate corresponds to a rotation of the qubit's state vector on the Bloch sphere**.

- **Bloch Sphere** is a geometric representation of a qubit's state, where any point on the sphere represents a valid pure state.
- A **unitary operation** on qubit results in **rotating the vector** on this sphere **without changing its length** (preserving probability).

Some examples of rotations:

- `X Gate` → Rotation π around x-axis (section 2.3.2, page 26)
- `Z Gate` → Rotation π around z-axis (section 2.3.3, page 28)
- `Y Gate` → Rotation π around y-axis (section 2.3.4, page 30)
- `S Gate` → Rotation $\frac{\pi}{2}$ around z-axis (section 2.3.5, page 32)
- `H Gate` → Rotation $\frac{\pi}{2}$ around y, then π around x (section 2.3.6, page 34)

Why is this important? Quantum computation is fundamentally about **state rotations**, understanding gates as rotations helps visualize **interference**, **phase shifts**, and **entanglement creation**.

2. Reversibility: Applying a Gate Twice

Many single-qubit gates, especially the Pauli gates, exhibit the property of **involution**: applying the same gate twice **returns the qubit to its original state**. Mathematically it can be expressed as:

$$X^2 = Y^2 = Z^2 = I$$

This is due to the fact that **two** 180° **rotations around the same axis bring the vector back** to its original position on the Bloch sphere.

Proof that X Gate has involution.

$$X^2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$$

These identities confirm that **Pauli gates are their own inverses**, and **all unitary gates are reversible**. QED

3. Global Phases Do Not Matter

Quantum gates may **introduce a global phase** (a constant complex factor $e^{i\phi}$) to a state. Physically, **global phases are unobservable**, meaning they do **not affect measurement probabilities**. For example:

- States $|\psi\rangle$ and $e^{i\phi}|\psi\rangle$ are physically indistinguishable.
- The Y gate $= iXZ$, where i is a global phase that can be ignored in practice.

4. Composite Gates and Commutativity

- Gates can be **combined** by **matrix multiplication** (note: non commutative in general).
- **Order matters**: $AB \neq BA$ for most gates.
- Exception: **Gate Z and Gate S** are **commuted** because they are both **diagonal phase gates**.

Property	Pauli Gates (X, Y, Z)	Phase Gate (S)	Hadamard (H)
Rotation Axis	x, y, z (π radians)	z $\left(\frac{\pi}{2}\text{radians}\right)$	y $\left(\frac{\pi}{2}\right)$ + x (π)
Involution $G^2 = I$	✔	✘ ($\mathbf{S^2}$ = Z)	✔
Self-adjoint $G = G^\dagger$	✔	✘	✔
Unitary	✔	✔	✔
Phase Introduction	Z, S	i for $\|1\rangle$	✔ (interference)
Creates Superposition?	✘ (X, Y, Z alone)	✘	✔

Table 1: Summary of Properties.

2.5 When Does a Gate Create Superposition?

In this section, our goal is to determine **what structural properties a quantum gate must have to create superposition** when applied to a **basis state** ($|0\rangle$ or $|1\rangle$).

Let's consider a generic 2×2 gate A, given by:

$$A = \begin{pmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{pmatrix}$$

This matrix has a zero in the lower-left corner, suggesting no mixing from $|0\rangle$ to $|1\rangle$ via the first column. In other words:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow A|0\rangle = \begin{pmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} a_{11} \\ 0 \end{pmatrix}$$

That zero blocks (in the lower-left corner) blocks any $|1\rangle$ component from arising when we apply A to $|0\rangle$. So no superposition is created from $|0\rangle$, it stays "pure". Note that to create a superposition, the output of $A|0\rangle$ must contain both $|0\rangle$ and $|1\rangle$ components, so both entries are non-zero.

Step 1: Apply A to $|0\rangle$

$$A|0\rangle = \begin{pmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} a_{11} \\ 0 \end{pmatrix}$$

The output is purely $|0\rangle$, **no superposition**.

Step 2: Apply A to $|1\rangle$

$$A|1\rangle = \begin{pmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} a_{12} \\ a_{22} \end{pmatrix}$$

We have both $|0\rangle$ and $|1\rangle$ components, **potential superposition**. But now the key insight: **superposition requires that both amplitudes (coefficients) are non-zero**. This only happens if $a_{12} \neq 0$ and $a_{22} \neq 0$.

Proof that Gate A cannot create a superposition

Proof that Gate A cannot create a superposition. Quantum gates operate according to the principles of quantum mechanics, unitarity and reversibility. Therefore, Gate A must satisfy the following unitarity condition:

$$A^\dagger A = I$$

Let's simplify the discussion by assuming A is real, so $A^\dagger = A^T$. We compute $A^T A$:

$$A^T = \begin{pmatrix} a_{11} & 0 \\ a_{12} & a_{22} \end{pmatrix} \qquad A = \begin{pmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{pmatrix}$$

$$A^T A = \begin{pmatrix} a_{11}^2 & a_{11}a_{12} \\ a_{11}a_{12} & a_{12}^2 + a_{22}^2 \end{pmatrix}$$

Set equal to identity:

$$A^T A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

From this, we get:

1. $a_{11}^2 = 1 \Rightarrow a_{11} = \pm 1$
2. $a_{11} a_{12} = 0 \Rightarrow a_{12} = 0$
3. $a_{12}^2 + a_{22}^2 = 1 \Rightarrow a_{22} = \pm 1$

The superposition is eliminated by $a_{12} = 0$, because both columns must have non-zero elements. QED

In conclusion, **to create superposition, a gate must have non-zero elements in both columns**. In fact, gates like Hadamard H, with all non-zero elements, can create superposition. However, diagonal gates like Z or S cannot produce superposition.

2.6 Single-Qubit Quantum Circuits

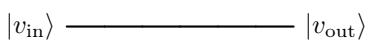
A **Quantum Circuit** is a **mathematical model** for quantum computation, **composed of sequential operations applied to qubits**. These operations include initializations, unitary gates, and measurements. In this section, we focus on circuits involving a single qubit, which form the foundation for understanding multi-qubit systems.

Circuit Elements

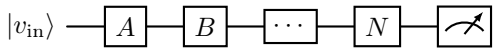
1. **Initialization**
   - Qubit is prepared in a **known state**, typically $|0\rangle$ or $|1\rangle$.
   - Essential **starting point** for any computation.
2. **Gates**
   - Single-qubit **unitary operations** (e.g., X, Z, H, S).
   - Transform the qubit's state **reversibly**.
3. **Measurement**
   - **Irreversible process**.
   - Collapses the qubit to $|0\rangle$ or $|1\rangle$ with probabilities dictated by the quantum state's amplitudes.

Quantum Circuit Diagram: Visual Language

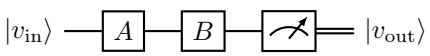
- **Horizontal lines** represent the **timeline of a single qubit**.



- **Boxes** on the line represent **gates** (letters) or **measurements** (tachometer icon).



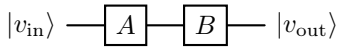
- **Double lines** (if present) represent **classical information** (e.g., measurement result).



- Left to right: The **circuit evolves over time from left (input) to right (output)**.

Gate Sequence: Serial Gates

Consider two gates A and B applied in sequence:



1. First apply A, then B.
2. Mathematically, this is expressed as:

$$v_{\text{out}} = BAv_{textin}$$

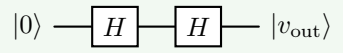
3. The **combined effect** of the two gates is equivalent to a single gate C = BA:

$$v_{\text{out}} = Cv_{\text{in}} = BAv_{\text{in}}$$

Matrix multiplication is from right to left, which may seem counterintuitive. The **last gate applied** (B) is **on the left** of the product **BA**.

Example 1: Hadamard followed by Hadamard

Let's compute H followed by H to $|0\rangle$:



1. Apply H:

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

2. Applying H again, we **restore the original state due to the involution properties** ($H^2 = I$):

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

2.7 Outer Product of Kets

In quantum mechanics, **Bra-Ket notation** (introduced by Dirac) is a concise and powerful tool for representing quantum states and operations. It uses **kets** $|\psi\rangle$ for **vectors** (states), and **bras** $\langle\psi|$ for their **Hermitian adjoints** (dual vectors).

Given two kets:

$$|a\rangle = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \qquad |b\rangle = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

The **Outer Product** $|a\rangle\langle b|$ is a **matrix**, formed by:

$$|a\rangle\langle b| = |a\rangle \otimes \langle b| = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \begin{pmatrix} \overline{b_1} & \overline{b_2} \end{pmatrix} = \begin{pmatrix} a_1\overline{b_1} & a_1\overline{b_2} \\ a_2\overline{b_1} & a_2\overline{b_2} \end{pmatrix} \tag{10}$$

Where the symbol $\otimes$ is the **tensor product operator**. This operation **maps a vector to a matrix**.

The **Inner Product** $\langle b|a\rangle$ is a scalar:

$$\langle b|a\rangle = \overline{b_1}a_1 + \overline{b_2}a_2 \tag{11}$$

So:

- **Inner product** $\rightarrow$ **projection**, result is number.
- **Outer product** $\rightarrow$ **matrix**, result is operator.

Associativity between outer and inner products

$$|a\rangle \cdot \langle b|c\rangle = \langle b|c\rangle \cdot |a\rangle \tag{12}$$

This is a fundamental identity in quantum mechanics, showing how outer products interact with inner products.

Proof the Associativity. Let's define:

$$|a\rangle = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \qquad |b\rangle = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \qquad |c\rangle = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$$

1. Compute inner product $\langle b|c\rangle$:

   $$\langle b|c\rangle = \overline{b_1}c_1 + \overline{b_2}c_2 = \alpha$$

   Where α is a scalar value.

2. Multiply scalar α with $\langle a|$:

   $$\alpha\langle a| = \alpha \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} \alpha \cdot a_1 \\ \alpha \cdot a_2 \end{pmatrix} = \begin{pmatrix} (\overline{b_1}c_1 + \overline{b_2}c_2) \cdot a_1 \\ (\overline{b_1}c_1 + \overline{b_2}c_2) \cdot a_2 \end{pmatrix} = (\overline{b_1}c_1 + \overline{b_2}c_2) \cdot \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

3. Compute outer product $|a\rangle\langle b|$ (matrix):

   $$|a\rangle\langle b| = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \begin{pmatrix} \overline{b_1} & \overline{b_2} \end{pmatrix} = \begin{pmatrix} a_1\overline{b_1} & a_1\overline{b_2} \\ a_2\overline{b_1} & a_2\overline{b_2} \end{pmatrix}$$

4. Multiply $|a\rangle\langle b|$ by $|c\rangle$:

$$(|a\rangle\langle b|)|c\rangle = \begin{pmatrix} a_1\overline{b_1} & a_1\overline{b_2} \\ a_2\overline{b_1} & a_2\overline{b_2} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} a_1\left(\overline{b_1}c_1 + \overline{b_2}c_2\right) \\ a_2\left(\overline{b_1}c_1 + \overline{b_2}c_2\right) \end{pmatrix}$$

Factor out the scalar $\left(\overline{b_1}c_1 + \overline{b_2}c_2\right)$:

$$\left(\overline{b_1}c_1 + \overline{b_2}c_2\right) \cdot \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \langle b|c\rangle \cdot |a\rangle$$

This proves that **outer product applied to a ket is equivalent to inner product $\langle b|c\rangle$ scaled by $|a\rangle$.** QED

2.8 Measurement

In quantum computing, **measurement** is a special type of operation that differs fundamentally from unitary gates. While gates perform reversible transformations, **measurement is irreversible**. It allows us to **gain information about the state of a qubit**, but at the price of **collapsing the quantum state**.

Measurement Operator: Matrix Form

A **measurement operator** is a **non-unitary**, **non-invertible** matrix. For a measurement **along a specific direction**, represented by a ket $|k\rangle$, the associated **projector** is:

$$M_k = |k\rangle\langle k| \tag{13}$$

This is called a **Projection Operator**, it projects any vector **onto the direction of** $|k\rangle$.

Effect of Measurement

Given a qubit $|\psi\rangle$, applying M_k yields the **(unnormalized) projected vector**:

$$|\psi_k\rangle = M_k\,|\psi\rangle = |k\rangle\,\langle k|\psi\rangle = \langle k|\psi\rangle\,|k\rangle$$

To obtain the **new normalized state**, we **divide by the square root of the probability**:

$$|\psi_k^{\text{norm}}\rangle = \frac{M_k \cdot |\psi\rangle}{\sqrt{\langle\psi\,|M_k|\,\psi\rangle}} = \frac{\langle k|\psi\rangle \cdot |k\rangle}{\sqrt{|\langle k|\psi\rangle|^2}} = |k\rangle$$

So after measurement, the qubit **collapses** to $|k\rangle$, the **eigenstate corresponding to the measurement outcome**.

Remark: $\langle\psi\,|M_k|\,\psi\rangle$

This is called a quadratic form in linear algebra, and in quantum mechanics it represents the **probability of the measurement outcome** k when **measuring the qubit** $|\psi\rangle$ with **measurement operator** M_k. Suppose a qubit ψ:

$$|\psi\rangle = \begin{pmatrix} a \\ b \end{pmatrix} \qquad \text{with } |a|^2 + |b|^2 = 1$$

If we apply the measurement operator M_0:

$$M_0\,|\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ 0 \end{pmatrix}$$

And we finally calculate the bra $\langle\psi|$:

$$\langle\psi|\begin{pmatrix} a \\ 0 \end{pmatrix} = \begin{pmatrix} \overline{a} & \overline{b} \end{pmatrix}\begin{pmatrix} a \\ 0 \end{pmatrix} = \overline{a}\cdot a = |a|^2$$

In other words, this is the probability of measuring 0: p_0.

Probability of Outcome

The **probability of measuring the qubit in the state** $|k\rangle$ is:

$$p_k = \langle\psi| M_k |\psi\rangle = |\langle k|\psi\rangle|^2 = \left|\overline{k_1} \cdot \psi_1 + \overline{k_2} \cdot \psi_2\right|^2 \tag{14}$$

This is the **Born rule**, a fundamental postulate of quantum mechanics. In quantum computing we have:

$$\begin{aligned} p_0 &= \langle\psi| M_0 |\psi\rangle = |\langle 0|\psi\rangle|^2 = |1 \cdot \psi_1 + \cancel{0 \cdot \psi_2}| \\ p_1 &= \langle\psi| M_1 |\psi\rangle = |\langle 1|\psi\rangle|^2 = |\cancel{0 \cdot \psi_1} + 1 \cdot \psi_2| \end{aligned}$$

Standard Basis Measurement

In quantum computing, we usually measure in the **computational basis** $\{|0\rangle, |1\rangle\}$. The **projectors** are:

$$\begin{aligned} M_0 &= |0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \\ M_1 &= |1\rangle\langle 1| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \end{aligned} \tag{15}$$

For a **general qubit** $|\psi\rangle = a|0\rangle + b|1\rangle$, the **probabilities** are:

$$p_0 = |a|^2 \qquad p_1 = |b|^2 \tag{16}$$

After measurement, the qubit **collapses to** $|0\rangle$ **with probability** $|a|^2$, or to $|1\rangle$ **with probability** $|b|^2$.

Key Properties of Measurement Operator

- **Idempotent**: once applied, **applying it again does nothing**:

  $$M_k^2 = M_k$$

  A simple proof:

  $$M_k^2 = |k\rangle\,|k\rangle\langle k|\,\langle k| = |k\rangle\,\langle k| = M_k$$

  This reflects that **after projection, the qubit is already in the state** $|k\rangle$.

- **Non-unitary** and **Non-reversible**: measurement **destroys superposition** and **cannot be undone**.

3 Multiple Qubit Gates

3.1 Multiple Qubit States

In quantum computing, the description of systems involving multiple qubits relies on a mathematical operation known as the **tensor product**. This operation enables us to **formally construct the joint state of two or more qubits starting from their individual states**.

★ The Tensor Product

Suppose we have two qubits, each in its own superposition:

$$|v_A\rangle = a_0 |0\rangle + a_1 |1\rangle \qquad \text{and} \qquad |v_B\rangle = b_0 |0\rangle + b_1 |1\rangle$$

Our **goal** is to **describe the joint state of the system**, i.e., the probability amplitudes for:

- Both qubits being in state $|0\rangle$
- Qubit A in $|0\rangle$ and qubit B in $|1\rangle$
- Qubit A in $|1\rangle$ and qubit B in $|0\rangle$
- Both in $|1\rangle$

To construct the combined state mathematically, we use the **Tensor Product** $\otimes$. Give the vectors of v_A and v_B:

$$|v_A\rangle = a_0 \begin{bmatrix}1\\0\end{bmatrix} + a_1 \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}a_0\\a_1\end{bmatrix} \qquad |v_B\rangle = b_0 \begin{bmatrix}1\\0\end{bmatrix} + b_1 \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}b_0\\b_1\end{bmatrix}$$

The tensor product $|v_A\rangle \otimes |v_B\rangle$ gives:

$$\begin{aligned}
|v_A\rangle \otimes |v_B\rangle &= (a_0 |0\rangle + a_1 |1\rangle) \otimes (b_0 |0\rangle + b_1 |1\rangle) \\
&= \begin{bmatrix}a_0\\a_1\end{bmatrix} \otimes \begin{bmatrix}b_0\\b_1\end{bmatrix} \\
&= a_0b_0 |00\rangle + a_0b_1 |01\rangle + a_1b_0 |10\rangle + a_1b_1 |11\rangle \\
&= a_0b_0 \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} + a_0b_1 \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} + \\
&\quad a_1b_0 \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} + a_1b_1 \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} \\
&= a_0b_0 \begin{bmatrix}1\\0\\0\\0\end{bmatrix} + a_0b_1 \begin{bmatrix}0\\1\\0\\0\end{bmatrix} + a_1b_0 \begin{bmatrix}0\\0\\1\\0\end{bmatrix} + a_1b_1 \begin{bmatrix}0\\0\\0\\1\end{bmatrix} \\
&= \begin{bmatrix}a_0b_0\\a_0b_1\\a_1b_0\\a_1b_1\end{bmatrix}
\end{aligned} \tag{17}$$

This four-dimensional vector fully describes the joint system of two qubits in the computational basis. The tensor product **preserves the linearity** of the system and ensures that the **resulting state is still a valid quantum state**.

About the notation, the ket $|00\rangle$ can be explicitly rewritten as the tensor product $|0\rangle \otimes |0\rangle$. These two notations are the same.

% Measurement Probabilities

Each coefficient in the final four-dimensional vector corresponds to an **amplitude for a specific measurement outcome**. For instance:

- The probability of observing $|00\rangle$ is $|a_0 b_0|^2$
- The probability of observing $|01\rangle$ is $|a_0 b_1|^2$
- The probability of observing $|10\rangle$ is $|a_1 b_0|^2$
- The probability of observing $|11\rangle$ is $|a_1 b_1|^2$

This reflects the Born rule: the probability of an outcome is the square modulus of the corresponding amplitude (page 45).

Normalization of Two-Qubit Quantum State

The normalization of a quantum state means that the **total probability of all possible outcomes must sum to 1**. The reason we need normalization is due to the **Born rule**: the probability of measuring a quantum state $|\psi\rangle$ and finding it in basis state $|i\rangle$ is $|c_i|^2$. Since total probability must equal 1, we must have:

$$\sum_i |c_i|^2 = 1$$

Normalization of two-qubit quantum state. Given two normalized single-qubit states:

$$|v_A\rangle = a_0 |0\rangle + a_1 |1\rangle \qquad |v_B\rangle = b_0 |0\rangle + b_1 |1\rangle$$

We know that (because they are normalized, and the total probability of all possible outcomes must sum to 1):

$$|a_0|^2 + |a_1|^2 = 1 \qquad |b_0|^2 + |b_1|^2 = 1$$

The combined state, via the tensor product, is:

$$|v_{AB}\rangle = a_0 b_0 |00\rangle + a_0 b_1 |01\rangle + a_1 b_0 |10\rangle + a_1 b_1 |11\rangle$$

The question is: **does this state remain normalized?** We need to verify that the sum of the squared moduli (i.e., squared magnitudes) of the coefficients equals 1:

$$|a_0 b_0|^2 + |a_0 b_1|^2 + |a_1 b_0|^2 + |a_1 b_1|^2 = 1$$

We use the property of modulus:

$$|a_i b_j|^2 = |a_i|^2 \cdot |b_j|^2$$

Hence:

- $|a_0 b_0|^2 = |a_0|^2 |b_0|^2$
- $|a_0 b_1|^2 = |a_0|^2 |b_1|^2$
- $|a_1 b_0|^2 = |a_1|^2 |b_0|^2$
- $|a_1 b_1|^2 = |a_1|^2 |b_1|^2$

We regroup the terms:

$$|a_0|^2 \left(|b_0|^2 + |b_1|^2\right) + |a_1|^2 \left(|b_0|^2 + |b_1|^2\right)$$

Factor out:

$$\left(|a_0|^2 + |a_1|^2\right) \cdot \left(|b_0|^2 + |b_1|^2\right)$$

Since both v_A and v_B are normalized:

$$|a_0|^2 + |a_1|^2 = 1 \qquad |b_0|^2 + |b_1|^2 = 1$$

So: $1 \cdot 1 = 1$. QED

The combined two-qubit state $v_{AB} = v_A \otimes v_B$ is indeed **normalized**, provided the original qubits are normalized. This confirms that the **tensor product of two normalized states is itself a normalized state**, a crucial result that ensures consistency.

3.2 Introduction to Multiple Qubit Gates

In quantum computing, the transition from single-qubit to multi-qubit systems represents a fundamental leap in computational expressiveness. While single-qubit gates manipulate isolated quantum bits, the true power of quantum computation emerges only when we consider operations on multiple qubits. This is because **multi-qubit gates allow for the creation and manipulation of quantum entanglement**, a uniquely quantum phenomenon with no classical analog, and a critical resource for quantum advantage.

The behavior of multi-qubit systems is governed by the **tensor product** structure of quantum mechanics. That is, the state space of a system composed of multiple qubits is described by the tensor product of the state spaces of the individual qubits. For example, a system of two qubits is represented by a vector in a four-dimensional Hilbert space $\mathbb{C}^2 \otimes \mathbb{C}^2 = \mathbb{C}^4$, and similarly, an n-qubit system is described by a vector in $\mathbb{C}^{2^n}$. This **exponential growth in state space dimensionality is one of the key reasons why quantum computers can**, in principle, **outperform classical ones for certain tasks**.

Multi-qubit gates are therefore defined as **unitary transformations acting on these higher-dimensional vector spaces**. A simple extension of single-qubit operations to the multi-qubit context is given by applying gates in a factorized way (e.g., applying a Hadamard gate on the first qubit and an identity on the second corresponds to $H \otimes I$). However, more powerful and interesting operations arise when gates do not decompose as tensor products of single-qubit gates. These non-separable gates, such as the Controlled-NOT (CNOT) gate, are capable of generating entangled states, and as such, are essential for universal quantum computation.

Thus, the **study of multi-qubit gates** is not merely a generalization of single-qubit logic; it is the **gateway to fundamentally quantum phenomena**. Understanding these gates, their representations, and their implications is central to designing quantum algorithms and circuits that go beyond classical capabilities.

3.3 Tensor Product of Quantum Gates

In a quantum system composed of multiple qubits, it is common to **apply gates independently and simultaneously to different qubits**. This operations is known as parallel execution, and it is described formally using the **Tensor Product of Quantum Gates**.

Motivation

Suppose we have two qubits in individual states:

$$|v_A\rangle = a_0 |0\rangle + a_1 |1\rangle \qquad |v_B\rangle = b_0 |0\rangle + b_1 |1\rangle$$

The **joint state** of the system is represented using the tensor product (page 46):

$$|v_{AB}\rangle = |v_A\rangle \otimes |v_B\rangle$$

Now assume we wish to apply:

- A gate A to the first qubit $|v_A\rangle$
- A gate B to the second qubit $|v_B\rangle$

How do we describe this *combined operation*?

Formal Definition

The combined transformation on the system is represented by the **tensor product of the operators**:

$$(A \otimes B)(v_A \otimes v_B)$$

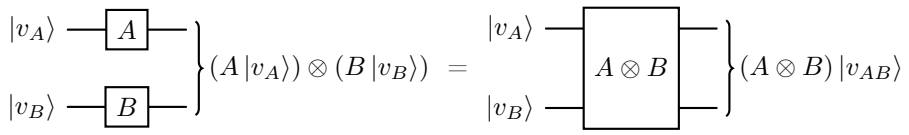
This is **not the same as applying** A and B **separately** and then multiplying the results. It's a structured, mathematically-defined operation where:

- A acts only on the first qubit
- B acts only on the second qubit

The overall effect on the joint state v_{AB} is captured by applying the composite operator $A \otimes B$.

Circuit Interpretation

In a quantum circuit diagram, each horizontal line represent a qubit, and gates are applied along these lines. The parallel application of gates is captured mathematically by $A \otimes B$, and visually by **applying gates side-by-side in the circuit**.



Applying a Gate to One Qubit

Let's say we have two qubits:

$$|v_{AB}\rangle = v_A \otimes v_B$$

We want to apply:

- No operation to qubit A (i.e., we want it to stay as it is)
- Gate B to qubit B

But since we're working in the **joint system**, we need to describe the total gate as acting on the full state vector in $\mathbb{C}^4$. We cannot apply gate B directly on the 2nd qubit in isolation.

To formalize "*do nothing*" to qubit A, we use the **Identity Gate**:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{18}$$

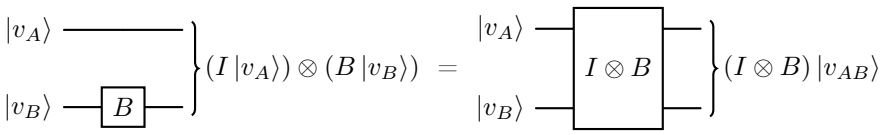
Then, the correct total operation on the two-qubit state is:

$$(I \otimes B)\,(v_A \otimes v_B)$$

This notation ensures that:

- I acts on qubit A and **leaves it unchanged**
- B is applied to qubit B
- The full operator still acts on the **tensor product state space** $\mathbb{C}^4$

In the circuit diagram, this is shown as:



Even if we don't draw the identity gate, it is **implicitly applied** to all untouched qubits, because we are still working in the joint space.

Hadamard Transform on Multiple Qubits

In Section 2.3.6 page 34, we saw how to apply the Hadamard operation to a single qubit system. Now that we are on multiple qubit systems, we want to understand how to use the Hadamard operation.

The **Hadamard Gate (H)** creates superposition states. When applied to each qubit in a system, it **transforms a basis state into a uniform superposition of all possible states**. To apply the Hadamard gate to **two qubits in parallel**, we compute the **tensor product of two Hadamard matrices**:

$$H^{\otimes 2} = H \otimes H$$

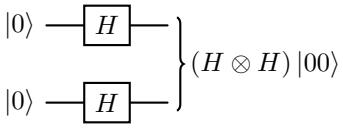
This results in a 4×4 matrix:

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

For example, if we want to apply the Hadamard gate to the state $|00\rangle$, mathematically we have:

$$H^{\otimes 2} |00\rangle = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

This is a **uniform superposition over all possible two-qubit basis states**. From a circuit point of view:



The **Hadamard operation can be generalized** to n qubits:

$$H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n - 1} |i\rangle \tag{19}$$

This creates a uniform quantum state across all 2^n possibilities. Something **impossible in classical computing** without explicitly listing all combinations.

3.4 Controlled NOT (CNOT) Gate

The **Controlled-NOT (CNOT)** gate is one of the most fundamental and widely used **two-qubit gates** in quantum computing. It serves as a canonical example of a **quantum gate that cannot be decomposed into single-qubit operations**, and it plays a central role in the generation of entanglement, error correction, and universal gate constructions.

How does it work?

The CNOT gate operates on two qubits:

- The **control qubit** $|c\rangle$
- The **target qubit** $|t\rangle$

Its defining action is **conditional flipping of the target qubit**. If the control qubit is in the state:

- $|1\rangle$, the CNOT gate **applies the Pauli-X (NOT) gate** (page 26) to the target (so the target qubit flips and 0 becomes 1 and 1 becomes 0).
- $|0\rangle$, the **target is left unchanged**.

This behavior can be expressed algebraically as:

$$\text{CNOT}\,|c\rangle\,|t\rangle = |c\rangle\,|t \oplus c\rangle \tag{20}$$

Where:

- Qubit Input: two-qubit basis state $|c\rangle|t\rangle$ ($|c\rangle\,|t\rangle = |c\,t\rangle$).
- CNOT operator: if $c = 0$, leave t unchanged, otherwise, flip t.
- XOR $\oplus$ symbol: denotes addition modulo 2.

$$t \oplus c = \begin{cases} 0 & \text{if } t = c \\ 1 & \text{if } t \neq c \end{cases}$$

It flips t if and only if $c = 1$. The **Addition modulo 2**, often denoted as $\oplus$, is a simple binary operation where the result is the remainder after dividing the sum of two bits by 2. It's also known as the **exclusive OR (XOR)** operation.

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

So, $t \oplus c$ means: "flip the target qubit t if and only if the control qubit c is 1".

- The control qubit $|c\rangle$ stays the same.
- The target qubit $|t\rangle$ becomes $|t \oplus c\rangle$ (flipped if $c = 1$).

Matrix Representation

The matrix representation of the CNOT gate is:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{21}$$

It operates on **two-qubit states** ordered as:

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$$

This matrix is clearly **unitary**, preserving the norm of quantum states, and it **interchanges the basis** vectors $|10\rangle$ and $|11\rangle$ while **leaving** $|00\rangle$ and $|01\rangle$ **unchanged**, consistent with its definition.

For example, let a general two-qubit quantum state written as a linear combination of basis states:

$$|\psi\rangle = c_0 |00\rangle + c_1 |01\rangle + c_2 |10\rangle + c_3 |11\rangle$$

Where:

$$c_0, c_1, c_2, c_3 \in \mathbb{C} \quad \text{and} \quad \sum |c_i|^2 = 1$$

Now apply the CNOT gate:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \\ c_3 \\ c_2 \end{bmatrix}$$

The amplitudes of the $|10\rangle$ and $|11\rangle$ components have been swapped. This corresponds to flipping the target qubit only when the control is $|1\rangle$:

$$|\psi\rangle = c_0 |00\rangle + c_1 |01\rangle + c_2 |11\rangle + c_3 |10\rangle$$

Who is the Control Qubit?

We have discussed that the CNOT gate operates on two qubits: control $|c\rangle$ and target $|t\rangle$. In the operations, *how can we understand which is the control qubit or the target qubit?* By **convention**, the **control qubit is the most significant qubit**, i.e., the **leftmost one** in the tensor product $|c\,t\rangle$. So:

- $|00\rangle$: control $= 0$, target $= 0$
- $|01\rangle$: control $= 0$, target $= 1$
- $|10\rangle$: control $= 1$, target $= 0$
- $|11\rangle$: control $= 1$, target $= 1$

Thus, in a state like:

$$|\psi\rangle = c_0 |00\rangle + c_1 |01\rangle + c_2 |10\rangle + c_3 |11\rangle$$

The **first qubit is the control**, and the **second is the target**.

Generate Entanglement

An important aspect of the CNOT gate is its **ability to generate entanglement**. For instance, if the control qubit is prepared in a superposition state using a Hadamard gate H, such that the input to the CNOT is:

$$(H \otimes I) |00\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |10\rangle)$$

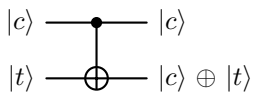
Then applying the **CNOT produces the entangled Bell state**:

$$\text{CNOT} \left(\frac{1}{\sqrt{2}} (|00\rangle + |10\rangle) \right) = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$$

This outcome exemplifies how **CNOT combined with single-qubit gates can prepare maximally entangled states**, a prerequisite for many quantum protocols including teleportation and superdense coding.

Visual representation of a CNOT

Finally, in quantum circuit diagrams, the CNOT is represented with a solid dot on the control qubit and a $\oplus$ symbol on the target qubit, connected by a vertical line. This visual notation helps emphasize the logical dependence between the two qubits.



3.5 Generic Controlled Gate

After introducing the CNOT gate as controlled-X operation, we generalize the idea to any unitary gate. In quantum computing, we often want to **apply a certain gate U to a target qubit v_B**, but **only if the control qubit v_A is in state $|1\rangle$**. This leads us to the concept of a generic controlled gate, usually denoted CU or C_U.

The **Generic Controlled Gate** has the matrix form:

$$CU = C_U = \begin{bmatrix} I & 0 \\ 0 & U \end{bmatrix} \tag{22}$$

This matrix acts on a two-qubit system, where:

- I is the 2×2 **identity** matrix
- U is the *any* **single-qubit unitary gate** (Pauli-X, Z, Y, ...)
- The **upper block** (top-left 2×2 block) acts when the **control qubit** is $|0\rangle$
- The **lower block** (bottom-right 2×2 block) applies U when the **control qubit** is $|1\rangle$

This form is **not separable**, so the operator C_U cannot be written as a tensor product of two individual single-qubit gates: $CU \neq A \otimes B$. The reason is simple:

- A **tensor product** like $I \otimes U$ **always** applies U to the second qubit **regardless** of the state of the first qubit.
- But C_U applies U to the second qubit **only if the first qubit is** $|1\rangle$.

So C_U is a **conditional operation** that links the behavior of one qubit to the value of the other.

How it works

Suppose we have two qubits:

$$v_A = a_0 |0\rangle + a_1 |1\rangle \qquad v_B = b_0 |0\rangle + b_1 |1\rangle$$

Their joint state is:

$$v_{AB} = v_A \otimes v_B = \begin{bmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{bmatrix}$$

Then the result of applying the generic controlled gate is:

$$C_U (v_A \otimes v_B) = \begin{bmatrix} I & 0 \\ 0 & U \end{bmatrix} \begin{bmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{bmatrix}$$

This matrix selectively applies U to the last two components (i.e., when $v_A = 1$), leaving the first two unchanged.

1. **Control is $|0\rangle$**. If $v_A = |0\rangle$, then:

   - $a_0 = 1$
   - $a_1 = 0$

   Consequently:

$$v_{AB} = \begin{bmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{bmatrix} = \begin{bmatrix} 1 \cdot b_0 \\ 1 \cdot b_1 \\ 0 \cdot b_0 \\ 0 \cdot b_1 \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ 0 \\ 0 \end{bmatrix}$$

   Using the generic controlled gate, there is **no change**:

$$C_U\left(v_{AB}\right) = \begin{bmatrix} I \cdot \begin{matrix} b_0 \\ b_1 \end{matrix} \\ U \cdot \begin{matrix} 0 \\ 0 \end{matrix} \end{bmatrix}$$

   Because the upper block is multiplied by the identity matrix. So the generic controlled gate **does nothing** when the control qubit is in state $|0\rangle$.

2. **Control is $|1\rangle$**. If $v_A = |1\rangle$, then:

   - $a_0 = 0$
   - $a_1 = 1$

   Now:

$$v_{AB} = \begin{bmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{bmatrix} = \begin{bmatrix} 0 \cdot b_0 \\ 0 \cdot b_1 \\ 1 \cdot b_0 \\ 1 \cdot b_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b_0 \\ b_1 \end{bmatrix}$$

   Using the generic controlled gate:

$$C_U\left(v_{AB}\right) = \begin{bmatrix} I \cdot \begin{matrix} 0 \\ 0 \end{matrix} \\ U \cdot \begin{matrix} b_0 \\ b_1 \end{matrix} \end{bmatrix}$$

   So the gate U is **only applied to the target qubit when the control qubit is** $|1\rangle$.

3.6 SWAP Gate

The **SWAP gate** is a two-qubit gate that **exchanges the states of the two qubits**. That is, if qubit A is in state v_A and qubit B in state v_B, then after applying SWAP, their roles are reversed.

Matrix Representation

The SWAP gate is defined by the following 4×4 unitary matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{23}$$

This matrix operates on the basis states in the usual order:

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$$

And swaps the coefficients of $|01\rangle$ and $|10\rangle$, leaving $|00\rangle$ and $|11\rangle$ unchanged.

How it works

Suppose we have two qubits in superposition:

$$v_A = a_0 |0\rangle + a_1 |1\rangle \qquad v_B = b_0 |0\rangle + b_1 |1\rangle$$

Their joint state is:

$$v_{AB} = v_A \otimes v_B = a_0 b_0 |00\rangle + a_0 b_1 |01\rangle + a_1 b_0 |10\rangle + a_1 b_1 |11\rangle$$

After applying the SWAP gate:

$$\text{SWAP}(v_{AB}) = a_0 b_0 |00\rangle + a_1 b_0 |01\rangle + a_0 b_1 |10\rangle + a_1 b_1 |11\rangle$$

But this result can be obtained also doing $v_B \otimes v_A$:

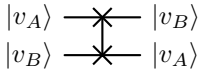
$$v_{BA} = b_0 a_0 |00\rangle + b_0 a_1 |01\rangle + b_1 a_0 |10\rangle + b_1 a_1 |11\rangle$$

Note that the coefficients a_i and b_j are scalar: $a_i \cdot b_j = b_j \cdot a_i$. So the SWAP gate effectively implements:

$$v_A \otimes v_B \xrightarrow{\text{SWAP gate}} v_B \otimes v_A \tag{24}$$

Circuit Representation

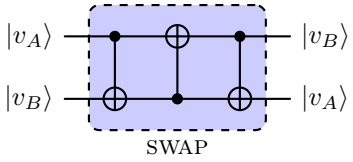
In circuit diagrams, the SWAP gate is often draw using **crossed lines with $\times$ symbols**:



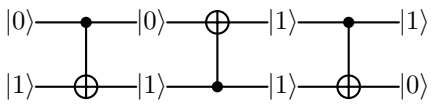
★ SWAP as Three CNOTs

An important implementation detail is that **SWAP can be decomposed into three CNOT gates** (page 53). This is crucial **because SWAP is not a native gate on most quantum hardware**, but CNOT often is.

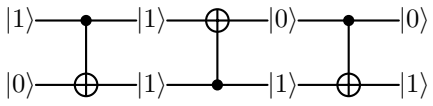
The decomposition is represented graphically as:



1. First CNOT:
   - Control qubit: v_A
   - Target qubit: v_B
2. Second CNOT:
   - Control qubit: v_B
   - Target qubit: v_A
3. Third CNOT:
   - Control qubit: v_A
   - Target qubit: v_B



And:



3.7 Toffoli Gate (CCNOT)

The **Toffoli Gate**, or **CCNOT (Controlled-Controlled-NOT)**, is a **three-qubit gate** where **two qubits act as controls** and **one as the target**. It generalizes the behavior of the CNOT gate by requiring **both control qubits** to be in the state $|1\rangle$ before the NOT operation is applied to the target.

Matrix Representation

The matrix form of the Toffoli gate is an 8×8 identity matrix with only two elements swapped:

$$\text{CCNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

This matrix swpas the $|110\rangle \leftrightarrow |111\rangle$ entries. In other words, all basis states remain unchanged except:

- $|110\rangle$ becomes $|111\rangle$
- $|111\rangle$ becomes $|110\rangle$

Gate Behavior

The CCNOT gate can be described as follows:

- **Inputs**: v_0, v_1 as *control qubits*, and v_2 as *target qubit*.
- **Output**:

$$v_2 \to v_2 \oplus (v_0 \wedge v_1)$$

This means:

- If v_0 logically combined with v_1 returns true, then flip v_2. In other words, flip if and only if v_0 and v_1 are 1.
- Otherwise, leave v_2 unchanged.

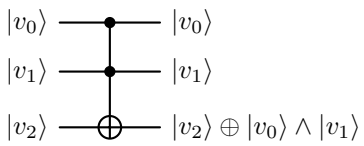
This behavior is **nonlinear** in the classic sense, and it's what makes Toffoli especially powerful: it can **simulate universal classical logic** within a quantum system. So any logic circuit that we can build using AND, OR and NOT gates, can also be implemented using only Toffoli gates.

Circuit Notation

In quantum circuit diagrams, the CCNOT gate is drawn with:

- Two control dots on the top two lines (for v_0, v_1)
- A $\oplus$ symbol on the third line (for the target qubit v_2)
- Vertical lines connecting the three

This shows the conditional behavior clearly: the **NOT is triggered only when both controls are active**.



3.8 Foundations of Universal Quantum Circuits

To implement arbitrary quantum computations, we must answer two fundamental questions:

1. **Which gates are sufficient** to build any quantum circuit?
2. **How do we mathematically model** a circuit built from those gates?

This section addresses both: it first introduces the concept of **universal gate sets**, and then explains how to formally represent quantum circuits as **matrices** acting on the full system state.

Universal Quantum Gate Sets

Quantum computation occurs within the framework of unitary transformations on quantum states. However, since the space of all unitary operations is infinite (continuous), we cannot construct every possible unitary matrix exactly with a finite number of gates. Instead, we aim for **approximate universality**: the ability to approximate any unitary operation to arbitrary precision using a small set of elementary gates.

A **Universal Quantum Gate Set** is a finite collection of quantum gates from which **any unitary operation** on any number of qubits can be approximated arbitrarily well.

Example 1

One of the most commonly used universal sets includes:

$$\{H, T, \text{CNOT}\}$$

- H: Hadamard gate, creates superposition
- T: $\frac{\pi}{8}$ phase gate, a non-Clifford gate necessary for universality
- CNOT: a two-qubit entangling gate

With these gates alone, it is possible to approximate any quantum circuit. This is backed by the Solovay-Kitaev, which ensures efficient approximation of unitaries using such a discrete set.

The set of Clifford gates alone is **not universal**. To achieve universality, we must include **at least one non-Clifford gate**, like T gate. Without it, the resulting circuits can be efficiently simulated classically, and therefore cannot exhibit true quantum advantage.

Modelling Quantum Circuits as Matrices

Quantum circuits are not only visual tools; they correspond to precise mathematical transformations. Every complete circuit defines a **unitary matrix** acting on the system's joint state vector. Understanding how to construct this matrix from a sequence of gates is crucial for analysis and simulation.

When building the matrix of a circuit composed of multiple gates and qubits, we follow three formal rules:

1. **Tensor product across qubits**. Gates applied simultaneously on different qubits are composed using the **tensor product**. For example:

$$A \text{ on qubit } 1, \quad B \text{ on qubit } 2 \Rightarrow A \otimes B$$

2. **Matrix product along time (sequential composition)**. Gates applied **in sequence** (over time) are composed via **matrix multiplication**, applied **from right to left**. This reflects the order of function composition.

3. **Identity padding for untouched qubits**. If a gate acts only on some of the qubits, we insert the **identity matrix** I for those that are not affected, to preserve the tensor product structure.

3.9 Entanglement

In a multi-qubit quantum system, **Entanglement** refers to the **phenomenon in which the quantum state of each qubit cannot be described independently of the state of the orders**. Entangled states are **non-separable**, meaning they cannot be expressed as the tensor product of individual qubit states.

- **The state of a n-qubit system is described by 2^n complex amplitudes.**

  Why? Because a general quantum state is a **superposition of all basis states**. For n qubits, there are 2^n possible basis states (e.g., $|000\rangle$, $|001\rangle$, ..., $|111\rangle$). So a general state is:

  $$|\psi\rangle = c_0 |0\ldots0\rangle + c_1 |0\ldots1\rangle + \cdots + c_{2^n-1} |1\ldots1\rangle$$

  Where:

  - $c_0, c_1, \ldots, c_{2^n-1} \in \mathbb{C}$.
  - So we need 2^n complex numbers $= 2 \cdot 2^n = 2^{n+1}$ real numbers.

- **Accounting for normalization and the irrelevance of global phase, it takes $2^{n+1} - 2$ real numbers to fully specify the state.**

  If we want to **completely describe the quantum state of an n-qubit system**, so that we can simulate it, reconstruct it, or predict its evolution, **we need to know $2^{n+1} - 2$ real numbers**.

  In other words, if someone gives us:

  - A list of $2^{n+1} - 2$ real values
  - And tells us "these specify the quantum state"

  Then:

  1. We can **reconstruct the full state vector**
  2. We know everything we need to calculate outcomes of **any measurable**
  3. We can simulate unitary evolution, or compute entanglement, etc.

Example 2: $n = 1$ qubit

The state is:

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

That's 4 real numbers (2 complex), but minus:

- 1 for normalization: $|\alpha|^2 + |\beta|^2 = 1$
- 1 for global phase: $e^{i\theta}$ is physically irrelevant

So, to fully describe a single qubit, we need:

$$2^{1+1} - 2 = 2 \text{ real numbers}$$

And this make sense, because we can describe any pure qubit with two real angles θ, ϕ on the block sphere (page 19).

Why 2^{n+1}?

- 2^n complex amplitudes for an n-qubit system (because there are 2^n basis states).
- Each complex number is made up of 2 real numbers: the real part and the imaginary part. So we have to add one to n.

Why do we subtract 2?

- **Normalization**. **Quantum states must always be normalized**:

$$\sum_{i=0}^{2^n-1} |c_i|^2 = 1$$

  This is **one constraint** on the set of amplitudes. Since it's a real-valuated equation, it removes 1 real degree of freedom from the total.
- **Global Phase**. In quantum mechanics, multiplying a quantum state by a global phase $e^{i\theta}$ doesn't change anything physically measurable:

$$|\psi\rangle \quad \text{and} \quad e^{i\theta} |\psi\rangle$$

  Are **physically indistinguishable**, they lead to the **same measurement outcomes**. So we treat two such states as equivalent. That means we're overcounting one extra complex degree of freedom, which corresponds to the global phase. This removes another real parameter (the angle θ).

- **In contrast, n independent (separable, then not entangled) qubits would only require $2n$ real parameters.**

  What about independent qubits? If the qubits are not entangled, each qubit can be written separately as:

$$|\psi_i\rangle = \alpha_i |0\rangle + \beta_i |1\rangle$$

  Each qubit needs:

  - $\alpha_i, \beta_i \in \mathbb{C}$
  - But again, we subtract 2 real numbers (normalization and global phase) for each qubit.

  So one single qubit state needs **2 real parameters**.

- **Since $2^n \gg 2n$, most quantum states are entangled.**

  What does this mean? If a state can be written as:

  $$v_1 \otimes v_2 \otimes \cdots \otimes v_n$$

  Then it only uses $2n$ real numbers. But a general n-qubit state needs $2^{n+1} - 2 \gg 2n$ real numbers. So: **almost all quantum states cannot be written as tensor products of single-qubit states**. These are the entangled states.

  How is this possible? The **full space** of quantum states is **exponentially larger** $(2^{n+1} - 2)$ than the subspace of separable states $(2n)$. We can think of the space of all quantum states as a huge-dimensional sphere of radius 1 (because of normalization).

  – The set of separable states is a tiny curved surface embedded inside that space.
  – It's so tiny that if we pick a random point on the full sphere (a random quantum state), we will almost surely land outside the separable surface.

  The probability of randomly choosing a separable state from the set of all quantum states is essentially **zero**.

  Note that this is not a mathematical trick! This is **experimentally observable**. If we:

  – Generate random quantum states (e.g. using random circuits)
  – Or simulate the uniform distribution over the Hilbert space

  We'll find that:

  – With high probability, the state cannot be written as a product of individual qubit states.
  – In fact, the expected entanglement entropy of a randomly chosen pure state is very close to maximal.

Example 3: Bell State

One of the canonical examples of an entangled state is the **Bell state**:

$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}} \left(|00\rangle + |11\rangle\right) \tag{25}$$

Suppose we try to write this as product of two qubits:

- Let $v_A = a_0 |0\rangle + a_1 |1\rangle$
- Let $v_A = b_0 |0\rangle + b_1 |1\rangle$

Their tensor product becomes:

$$v_A \otimes v_B = a_0 b_0 |00\rangle + a_0 b_1 |01\rangle + a_1 b_0 |10\rangle + a_1 b_1 |11\rangle$$

To match the Bell state, we would require:

$$a_0 b_0 = \frac{1}{\sqrt{2}}, \quad a_1 b_1 = \frac{1}{\sqrt{2}}, \quad a_0 b_1 = 0, \quad a_1 b_0 = 0$$

But if $a_0 b_1 = 0$, then at least one of a_0 or b_1 must be zero. Similarly, if $a_1 b_0 = 0$, one of a_1 or b_0 must be zero. This leads to a contradiction, because we also need $a_0 b_0 \neq 0$ and $a_1 b_1 \neq 0$. Therefore, no such a_0, a_1, b_0, b_1 exist.
Conclusion: the Bell state cannot be written as a tensor product of two individual qubit states, it is entangled.

The Nature of Entanglement

Entanglement is a **structural property** of the quantum state, **not dependent on the basis**. This **contrasts with superposition**, which is basis-dependent.

- Whether a state is entangled depends on the decomposition of the system.
- A state can be entangled with respect to one partitioning of subsystems, and separable under another.
- In most contexts, when we say **a state is entangled**, we refer to its **entanglement with respect to the individual qubits** (standard tensor product structure).

So we can change the basis of the individual qubits (rotate them, use Hadamard basis), but we will **not** be able to write Bell state state as $|\psi_A\rangle \otimes |\psi_B\rangle$. No matter what basis we use, we will **not** be able to "break" the entanglement. Entanglement is about whether the state **can be written as a product of parts**, this is a property of the **whole structure**, not of how we write it.

Entangling Gates

A gate is said to be **entangling** if it is **capable of producing entangled states from separable inputs**. Not all two-qubit gates are entangling.

- A **generic 2-qubit gate** is a 4×4 unitary matrix, with 16 complex entries (minus constraint from unitarity).
- Two single-qubit gates acting independently on two qubits use only $2 \times 4 = 8$ complex parameters.

Thus, **most two-qubit gates are not tensor products** of single-qubit gates. If a gate cannot be written as $U_1 \otimes U_2$, it is **non-separable** and **can generate entanglement**.

Example 4: CNOT as an Entangling Gate

The CNOT gate can transform separable states into entangled states, for example:

$$\text{CNOT}\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |0\rangle\right) = \frac{|00\rangle + |11\rangle}{\sqrt{2}} = \left|\Phi^+\right\rangle \quad \text{(Bell state)}$$

This demonstrates the key role of CNOT in quantum-algorithms, it **creates entanglement**, which is a resource for quantum advantage.

What are Bell States?

Bell States are **four special two-qubit states**:

$$\begin{aligned} |\Phi^+\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) & |\Phi^-\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\Psi^+\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) & |\Psi^-\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \end{aligned} \tag{26}$$

That are:

- **Maximally entangled**: measuring one qubit fully determines the other.
- **Orthonormal**: they form a basis of the two-qubit Hilbert space.
- **Used in fundamental protocols** like teleportation, superdense coding, and entanglement-based quantum cryptography.

Bell states represent **all possible maximally entangles superpositions of two-qubit basis states**.

Why are they important?

Bell states are not just theoretical constructs, they are:

- **Fundamental to quantum computing**: often used as building blocks for more complex states.
- **Hard to simulate classically**: their entanglement implies that classical representations cannot compress them efficiently.
- **Maximally entangled**: each individual qubit, when observed in isolation, appears **completely random**, even though the total state is pure and deterministic.

When a Bell state is measured on one qubit, the outcome of the second qubit is **fully correlated**, but until that measurement happens, each qubit looks like it is in a **completely mixed state** (i.e. maximum uncertainty). This **contrast with unentangled (separable)** states, where each qubit can be described independently and has less uncertainty when observed locally.

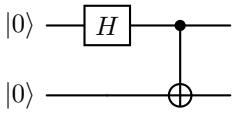
How to create Bell states

Bell states can be created using a **simple 2-gate quantum circuit**:

1. Apply a **Hadamard gate to the first qubit**:

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

2. Apply a **CNOT gate**, with:
   - Control: qubit 1
   - Target: qubit 2



The final output is:

$$\text{CNOT}\,(H \otimes I)\,(|00\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \left|\Phi^+\right\rangle$$

So this circuit creates the Φ^+ Bell state starting from $|00\rangle$. Mathematical:

1. Compute $H \otimes I$, i.e., apply Hadamard to the first qubit:

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

2. Then apply the CNOT operator to that result:

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

3. Multiply:

$$\text{CNOT} \cdot (H \otimes I)\,|00\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Example 5: Bell States in the Hadamard Basis

In this exercise, we express the Bell state $|\Phi^+\rangle$ in the Hadamard basis instead of the computational basis.

Bell state $|\Phi^+\rangle$ in the Hadamard basis. Given:

$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

We aim to re-express this state using the Hadamard basis $\{|+\rangle, |-\rangle\}$, where:

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

1. **Invert the definitions**. To convert from the computational basis to the Hadamrd basis, we invert the above equations to express $|0\rangle$ and $|1\rangle$ in terms of $|+\rangle$ and $|-\rangle$:
   - $|0\rangle = \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)$
   - $|1\rangle = \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle)$
2. **Substitute into $|00\rangle$ and $|11\rangle$**. We compute the tensor products using the expression above.
   - Compute $|00\rangle$:

$$\begin{aligned} |00\rangle &= |0\rangle \otimes |0\rangle \\ &= \left(\frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)\right) \otimes \left(\frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)\right) \\ &= \frac{1}{2}(|++\rangle + |+-\rangle + |-+\rangle + |--\rangle) \end{aligned}$$

   - Compute $|11\rangle$:

$$\begin{aligned} |11\rangle &= |1\rangle \otimes |1\rangle \\ &= \left(\frac{1}{\sqrt{2}}(|+\rangle - |-\rangle)\right) \otimes \left(\frac{1}{\sqrt{2}}(|+\rangle - |-\rangle)\right) \\ &= \frac{1}{2}(|++\rangle - |+-\rangle - |-+\rangle + |--\rangle) \end{aligned}$$

3. **Add the two to get $|\Phi^+\rangle$**. Now add the two results:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Substituting:

$$\begin{aligned}
|\Phi^+\rangle &= \frac{1}{\sqrt{2}}\left[\frac{1}{2}\left(|{+}{+}\rangle + |{+}{-}\rangle + |{-}{+}\rangle + |{-}{-}\rangle\right)\right. \\
&\qquad \left. + \frac{1}{2}\left(|{+}{+}\rangle - |{+}{-}\rangle - |{-}{+}\rangle + |{-}{-}\rangle\right)\right] \\
&= \frac{1}{\sqrt{2}} \cdot \frac{1}{2}\left(2\,|{+}{+}\rangle + 0\,|{+}{-}\rangle + 0\,|{-}{+}\rangle + 2\,|{-}{-}\rangle\right) \\
&= \frac{1}{\sqrt{2}} \cdot \frac{1}{2}\left(2\,|{+}{+}\rangle + 2\,|{-}{-}\rangle\right) \\
&= \frac{1}{\sqrt{2}}\left(|{+}{+}\rangle + |{-}{-}\rangle\right)
\end{aligned}$$

Thus, in the Hadamard basis, we have:

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

Which is a Bell-like state expressed in the Hadamard basis instead of the computational basis. QED

3.10 Measurement in Multi-Qubit Systems

We begin with a general two-qubit state:

$$|v_C\rangle = |v_A v_B\rangle = c_0 |00\rangle + c_1 |01\rangle + c_2 |10\rangle + c_3 |11\rangle$$

This is an entangled or general superposed state over the two-qubit basis. But the question is: ***What happens to the state of the system if we measure only qubit A?***

Decomposition Strategy

The idea is to **separate the global state** into parts conditioned on the state of qubit A:

$$|v\rangle = c_{01} |0\rangle \otimes \frac{c_0 |0\rangle + c_1 |1\rangle}{c_{01}} + c_{23} |1\rangle \otimes \frac{c_2 |0\rangle + c_3 |1\rangle}{c_{23}}$$

Where:

- $c_{01} = \sqrt{|c_0|^2 + |c_1|^2}$
- $c_{23} = \sqrt{|c_2|^2 + |c_3|^2}$

This expression means: the total state is a **superposition of two conditional branches**:

- One where qubit A is $|0\rangle$, and qubit B is in a normalized superposition of $|0\rangle$, $|1\rangle$.
- One where qubit A is $|1\rangle$, and qubit B is in another normalized state.

What happens after a Measurement?

1. Case 1. **Qubit A is measured as $|0\rangle$**:
   - The outcome collapses the state: $c_2 = 0$, $c_3 = 0$
   - The system is no longer a superposition, it's now in:

     $$|v_B\rangle = \frac{c_0 |0\rangle + c_1 |1\rangle}{c_{01}}$$

     Which is a valid normalized single-qubit state (i.e., $|v_B\rangle\langle v_B| = 1$)
2. Case 2. **Qubit A is measured as $|1\rangle$**:
   - Similarly, $c_0 = 0$, $c_1 = 0$
   - New state of B becomes:

     $$|v_B\rangle = \frac{c_2 |0\rangle + c_3 |1\rangle}{c_{23}}$$

Measuring **one qubit** in an entangled system **instantaneously collapses the entire state**, even though **only one qubit** is directly observed.

This feels strange because **measurement is local**, yet the **effect is nonlocal**, it determines the state of the other qubit, even if it's far away! This is a **manifestation of quantum entanglement** and is deeply **linked to** the phenomenon of **quantum nonlocality**.

Example 6: Bell state

Let's apply this idea to the Bell state:

$$|v\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

This can be rewritten as:

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

So we're in the entangled state $|\Phi^+\rangle$.

Now, we measure qubit A. If we get **outcome**:

- $|0\rangle$, the global state collapses to $|00\rangle \Rightarrow$ qubit B is now in state $|0\rangle$.
- $|1\rangle$, the state collapses to $|11\rangle \Rightarrow$ qubit B is in state $|1\rangle$.

References

[1] Cremonesi Paolo. Quantum computing. Slides from the HPC-E master's degree course on Politecnico di Milano, 2024.

Index