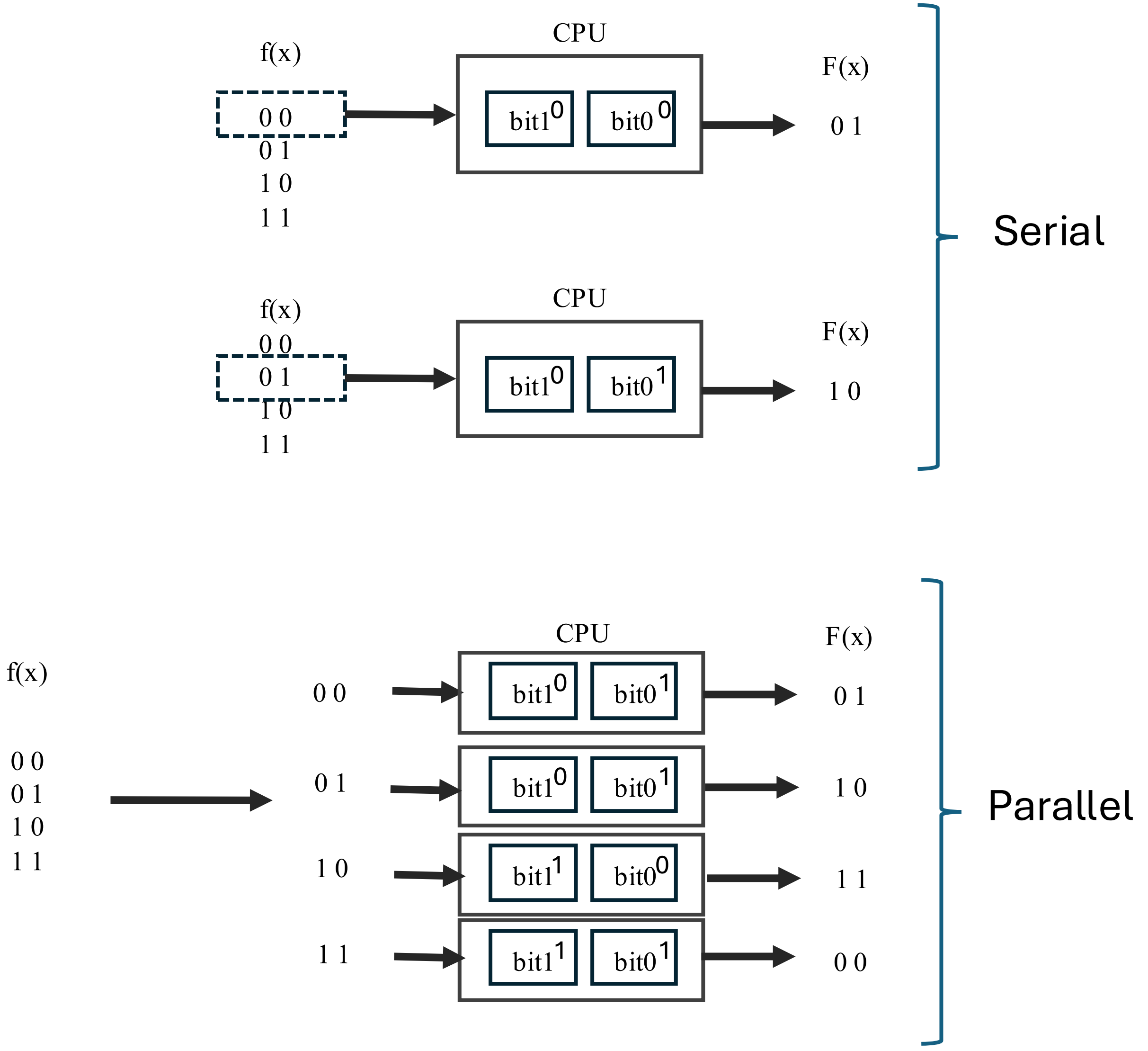
# Understanding Quantum Computing: Qubits, Gates, and Circuits

Quantum Computing
Parallelizing Data Processing Algorithm

Bhattaraprot Bhabhatsatam, Ph.D.

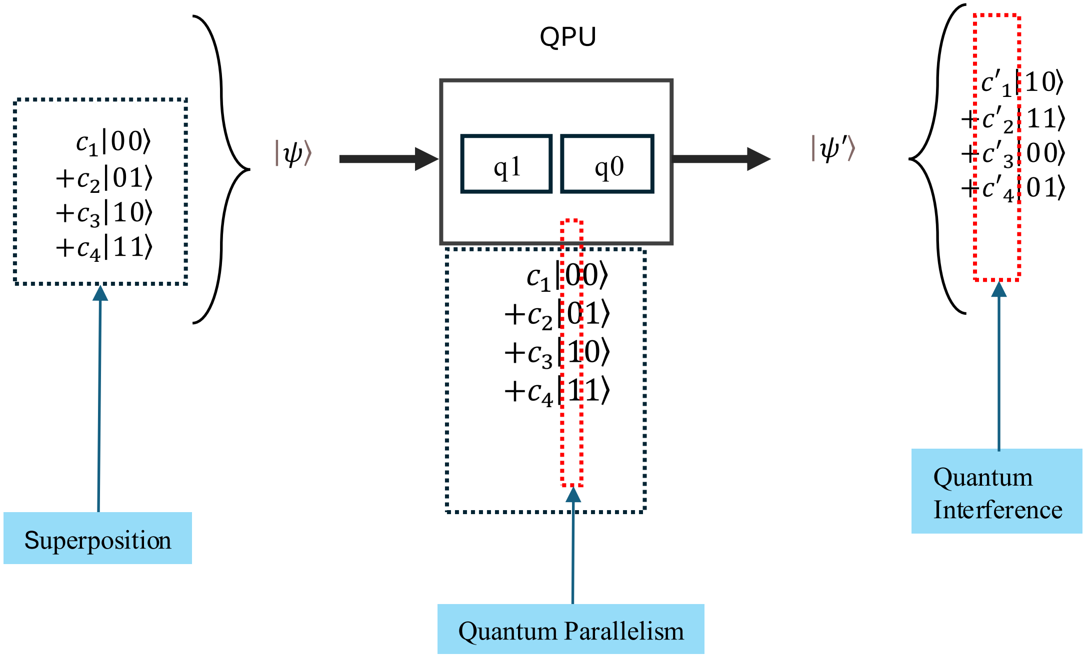
# Classical Computers — A Binary Input/Output Model

- **Input to CPU**
  - Data in **binary** form (bits)
  - Processed by **logical operations** (AND, OR, NOT, etc.) on registers
- **Output**
  - Also, in **binary** (e.g., 0 or 1, or sets of bits)
- **Key Point**
  - All processing is effectively in a binary, **deterministic** framework
  - Even parallel or multi-core approaches remain bound by binary logic



# Quantum Computing — Superposition & Wavefunction

- **Example**: 2 Qubits in Superposition
- **Actions Affect the Whole Wavefunction**
  - Applying a gate to one qubit transforms the **entire** wavefunction
  - This is fundamentally different from classical bitwise operations
- **Measurement**
  - **Single Shot**: Each measurement collapses the superposition to one outcome
  - **Probability Distribution**: Repeating the measurement many times (multiple "shots") builds a statistical distribution reflecting



# Key Quantum Concepts: Wavefunction & Phenomena

- **Quantum Wavefunction**
  - Represents **information** in quantum systems
  - Complex amplitudes (phase + magnitude)
  - Foundation for computing with qubits
- **Quantum Phenomena**
  - **Superposition**: Qubits can exist in multiple basis states simultaneously
  - **Entanglement**: Strong correlation between qubits that has no classical analogue
  - **Interference**: Amplitudes can reinforce or cancel out, affecting measurement outcomes
- **Quantum Parallelism**
  - Arises from these phenomena
  - Allows certain computations to evaluate many possibilities "at once"

# Designing for Quantum Advantage

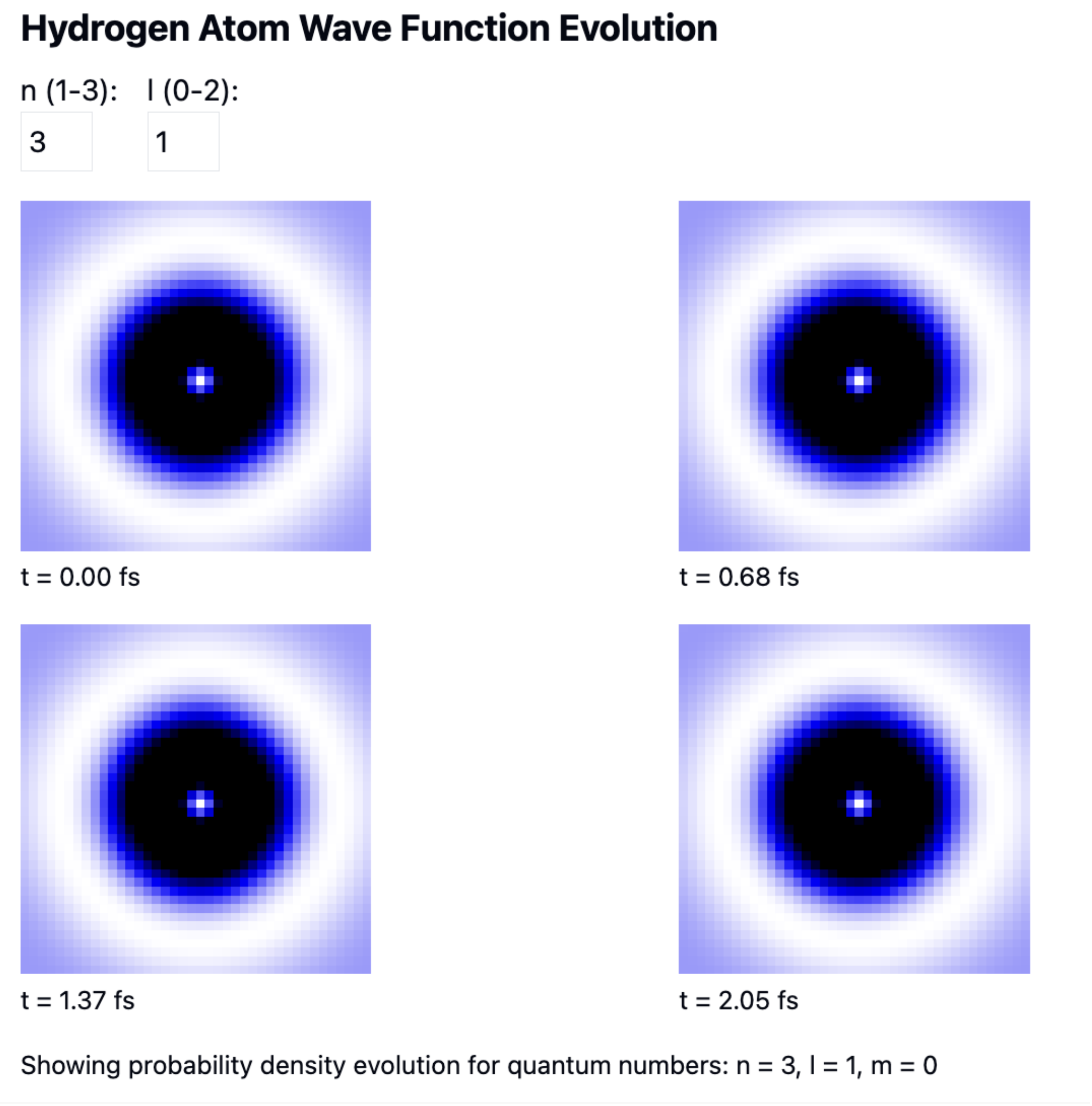
- **Key Takeaway**
  - The goal of quantum computing design is to **define and encode information** into a **quantum wavefunction**, then **exploit** superposition, entanglement, and interference to achieve **quantum parallelism** and potential speedups.
- **Focus**
  - *Harness* quantum phenomena as much as possible
  - *Architect* circuits or annealing processes to maximize quantum advantage

# Quantum Wave Equation & Quantum Wavefunction

- **What is the Quantum Wave Equation?**
  - **Schrödinger's Equation**: Governs how the quantum wavefunction $|\psi\rangle$ evolves over time.
  - Replaced classical deterministic trajectories with **probabilistic** descriptions.
- **Significance**
  - *Foundation of Quantum Mechanics*: By solving Schrödinger's equation, physicists uncovered quantum phenomena (superposition, entanglement, interference).
  - *Describes Dynamics*: Provides the rule for how $\psi(x,t)$changes under various potentials or interactions.
- **Key Insight**
  - **Wavefunction**: Encodes **all possible outcomes** a system can exhibit.
  - **Schrödinger's Equation**: Dictates **when and how** those possibilities evolve or interact.
  - **Quantum Computing**: Utilizes the wavefunction as a **mathematical tool** to describe qubit states and engineer quantum operations.
- **Reference**

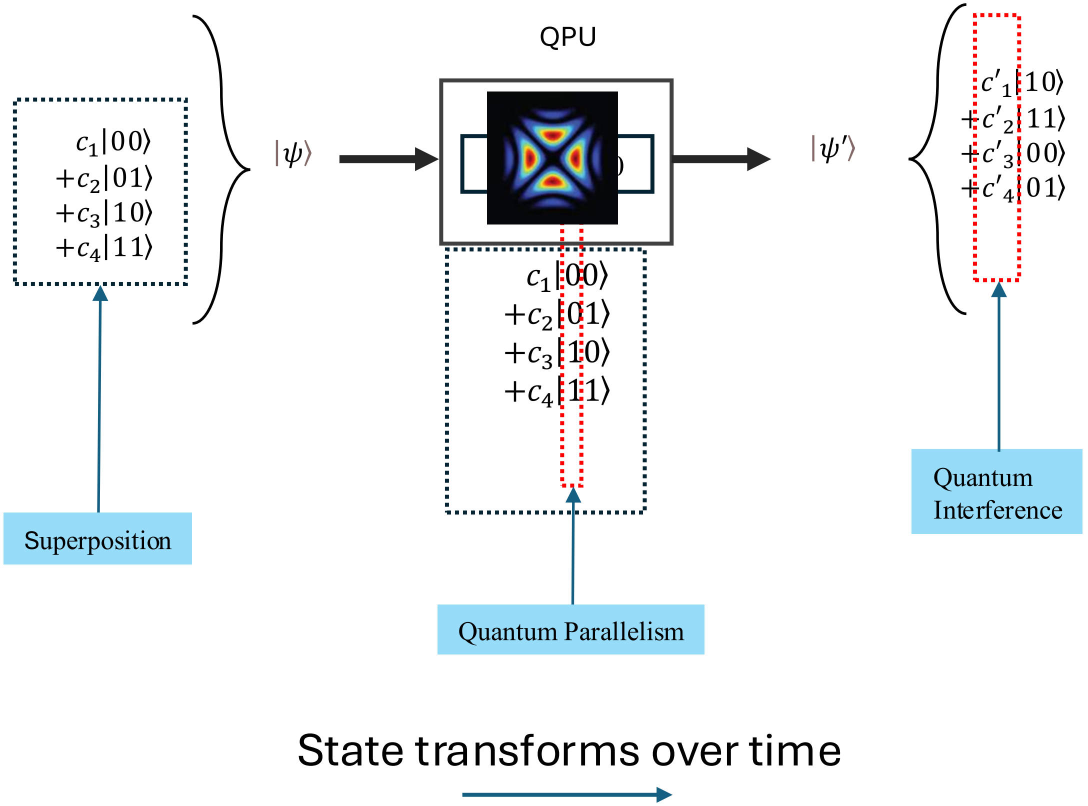
  - E. Schrödinger (1926), *Quantisierung als Eigenwertproblem*, *Annalen der Physik*.
  - R. P. Feynman, *The Feynman Lectures on Physics, Vol. III*.

# Visualizing Time Evolution of the Quantum Wavefunction



# Encoding Data into the Wave function

- **Wave function as Data**
  - We **encode** classical information into a **quantum wavefunction** $|\psi\rangle$
  - For two qubits, we might have
    - Sss
- Processing via Quantum Circuit
  - A **quantum circuit** transforms $|\psi\rangle$ over time
  - Different circuit types or models ⇒ different quantum computing paradigms
- **Mathematical Tools**
  - **Linear algebra** (vectors, matrices)
  - **Complex numbers** for phases and amplitudes
  - Bra–ket notation

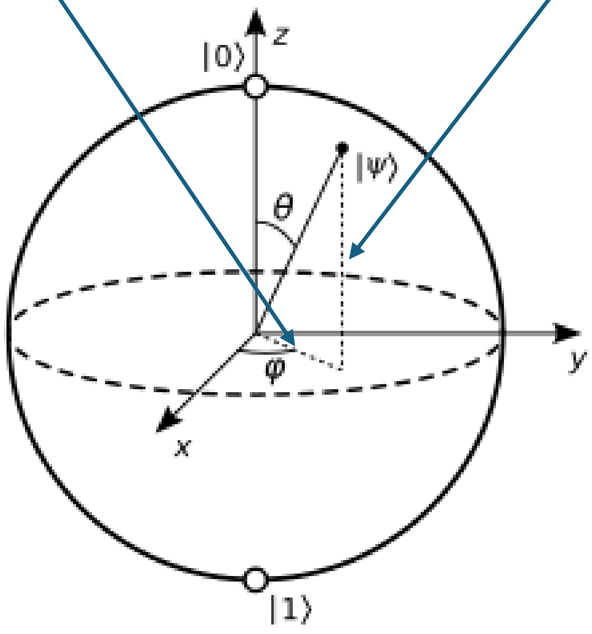


# Qubits, Bloch Sphere & Bra–Ket Notation

- **From Wavefunction to Qubit**
  - **Abstracting** quantum state into a **qubit**
- **Bra–Ket Notation**
  - **Dirac Notation:** $|\psi\rangle$ (ket) and $\langle\psi|$ (bra)
  - Represents **vectors** and **dual vectors** in Hilbert space
  - **Superposition :** $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ , with $|\alpha|^2 + |\beta|^2 = 1$
  - $|0\rangle$ and $|1\rangle$ form the standard basis
- **Equivalent Mathematical Forms**
  - **Vector Form:** $|\psi\rangle = \begin{pmatrix}\alpha \\ \beta\end{pmatrix}$
  - **Bra–Ket Form:** $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$
  - **State Vector :** $|\psi\rangle = (a + bi)|0\rangle + (c + di)|1\rangle$
  - **Bloch Sphere Representation :** $|\psi\rangle = \cos(\frac{\theta}{2})|0\rangle + e^{-i\varphi}\sin(\frac{\theta}{2})|1\rangle$

$\beta = \sin(\frac{\theta}{2})$

$\alpha = \cos(\frac{\theta}{2})$




$\theta$ = Polar angle

$\varphi$ = Azimuthal angle

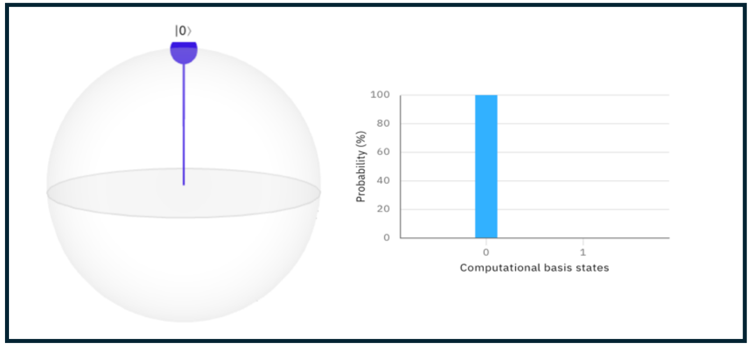
# Bloch Sphere: 1 qubit

$$|\psi\rangle = \cos(\tfrac{\theta}{2})|0\rangle + e^{-i\varphi}\sin(\tfrac{\theta}{2})|1\rangle$$

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$$

cos(0/2) = 1

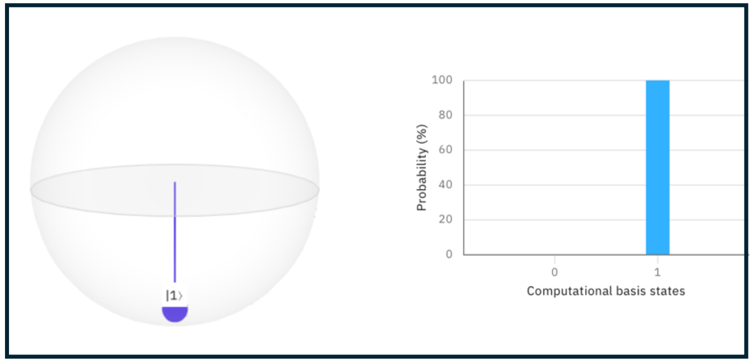
sin(0/2) = 0



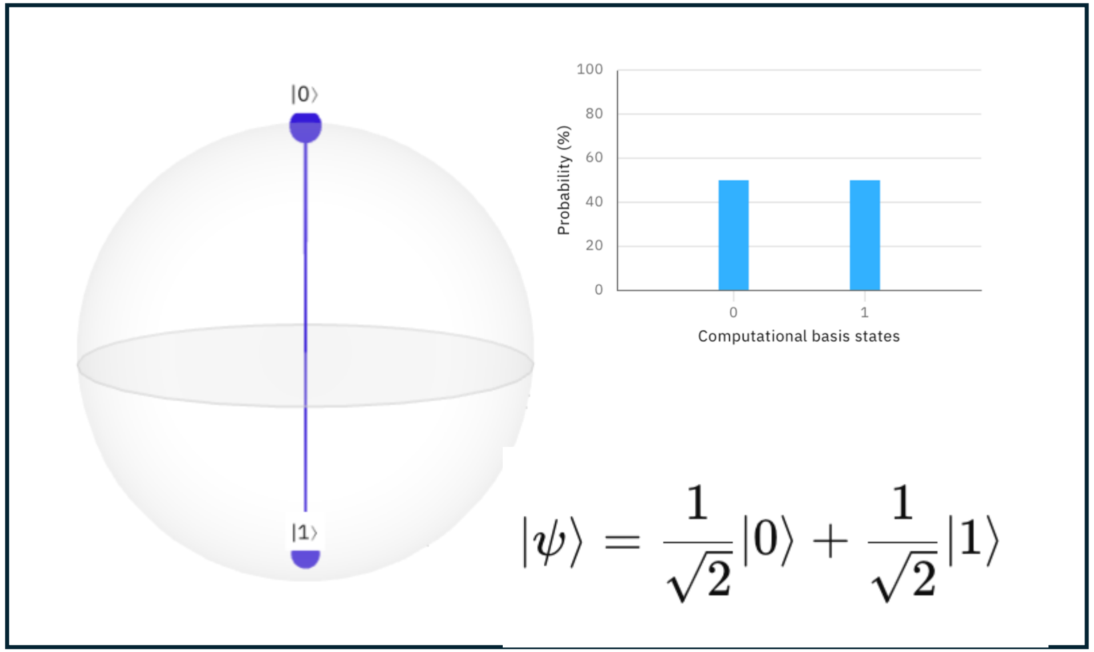

$$|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$$

cos(180°/2) = 0

sin(180°/2) = 1




Single Qubit, Basis State




Single Qubit, Superposition State

$\alpha = \frac{1}{\sqrt{2}}$, so $|\alpha|^2 = 0.5.$

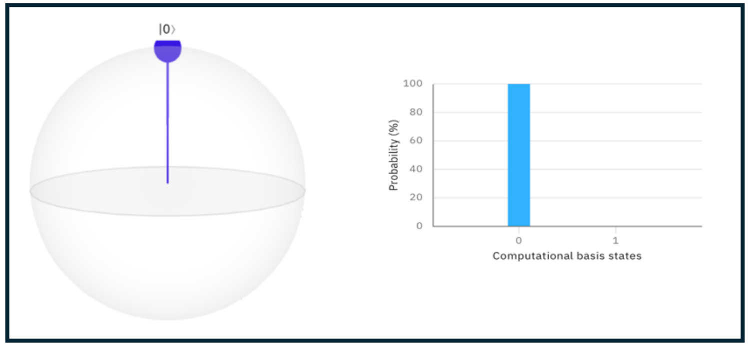
$\beta = \frac{1}{\sqrt{2}}$, so $|\beta|^2 = 0.5.$

# Bloch Sphere: single qubit

Basis State

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$$

cos(0/2) = 1
sin(0/2) = 0

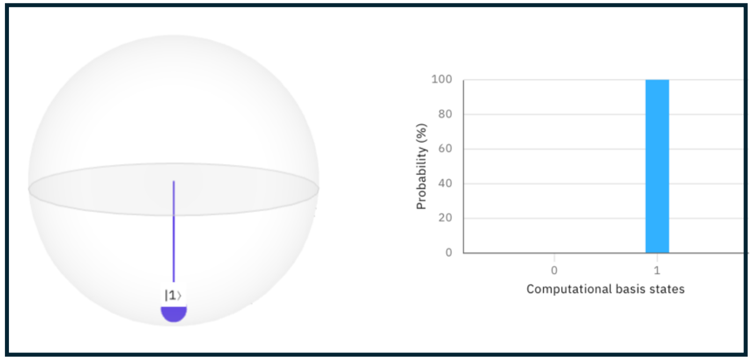



As column vector:
[1 + 0i]
[0 + 0i]

In basis notation:
1|0⟩ + 0|1⟩

$$|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$$

cos(180°/2) = 0
sin(180°/2) = 1




As column vector:
[0 + 0i]
[1 + 0i]

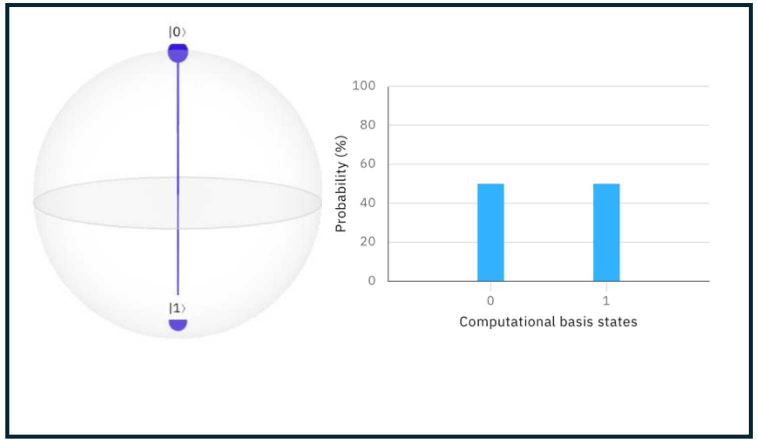
In basis notation:
0|0⟩ + 1|1⟩

Superposition State

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

$\alpha = \frac{1}{\sqrt{2}}$, so $|\alpha|^2 = 0.5.$

$\beta = \frac{1}{\sqrt{2}}$, so $|\beta|^2 = 0.5.$




As column vector:
[1/√2 + 0i]
[1/√2 + 0i]

In basis notation:
(1/√2)|0⟩ + (1/√2)|1⟩

# 2 Qubits

## General Two-Qubit State

$|\psi\rangle = [a + bi] = (a + bi)|00\rangle + (c + di)|01\rangle + (e + fi)|10\rangle + (g + hi)|11\rangle$

Where:

- $a^2 + b^2 + c^2 + d^2 + e^2 + f^2 + g^2 + h^2 = 1$ (normalization)
- Each component represents amplitude for respective basis state:
  - First: $(a + bi)$ for $|00\rangle$
  - Second: $(c + di)$ for $|01\rangle$
  - Third: $(e + fi)$ for $|10\rangle$
  - Fourth: $(g + hi)$ for $|11\rangle$

### $|00\rangle$ State

Vector form: Bra-ket form:

[1 + 0i] $|00\rangle = 1|00\rangle + 0|01\rangle + 0|10\rangle + 0|11\rangle$
[0 + 0i]
[0 + 0i]
[0 + 0i]

### $|01\rangle$ State

Vector form: Bra-ket form:

[0 + 0i] $|01\rangle = 0|00\rangle + 1|01\rangle + 0|10\rangle + 0|11\rangle$
[1 + 0i]
[0 + 0i]
[0 + 0i]

### $|10\rangle$ State

Vector form: Bra-ket form:

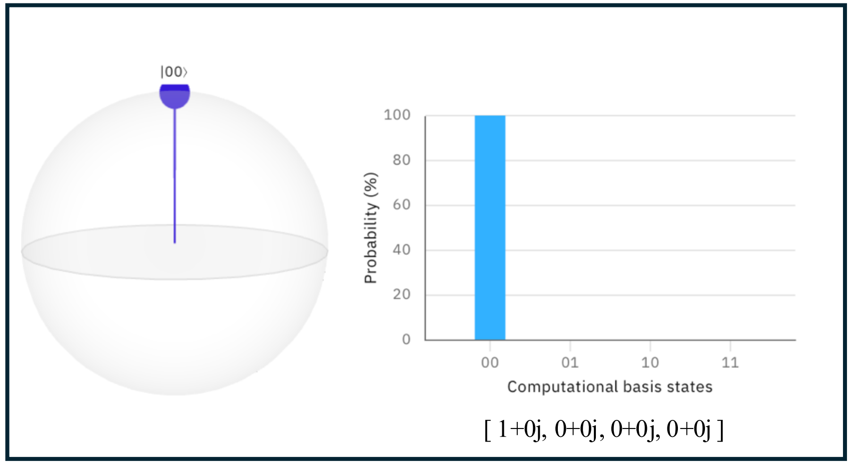
[0 + 0i] $|10\rangle = 0|00\rangle + 0|01\rangle + 1|10\rangle + 0|11\rangle$
[0 + 0i]
[1 + 0i]
[0 + 0i]

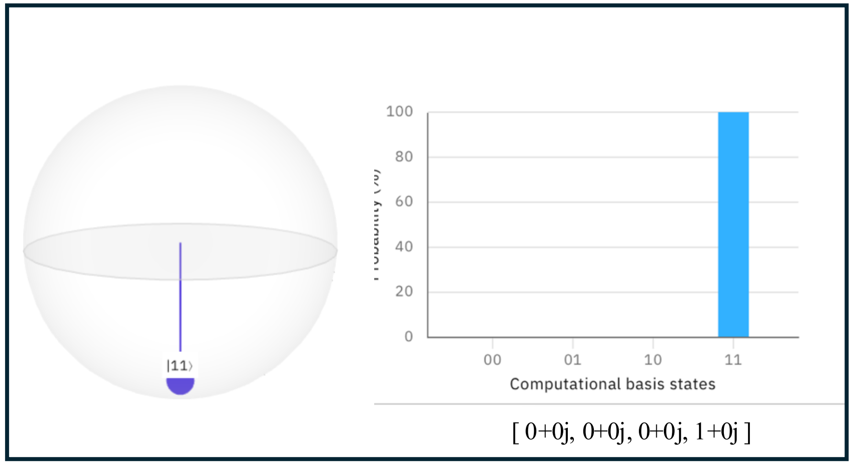
### $|11\rangle$ State

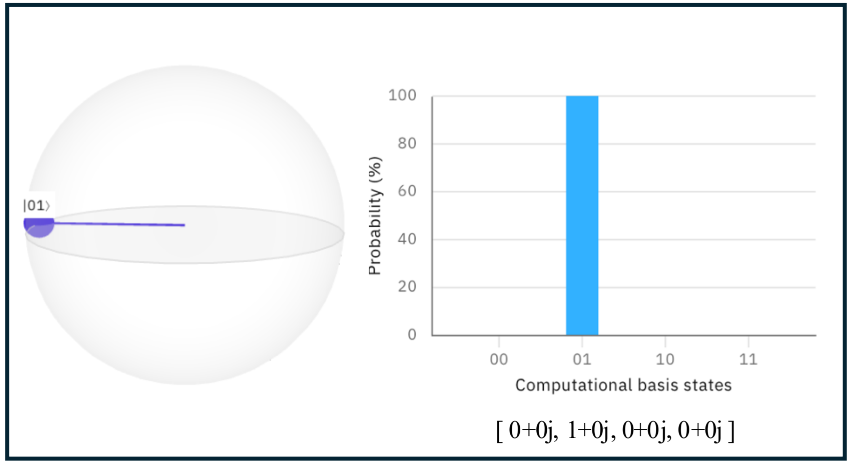
Vector form: Bra-ket form:

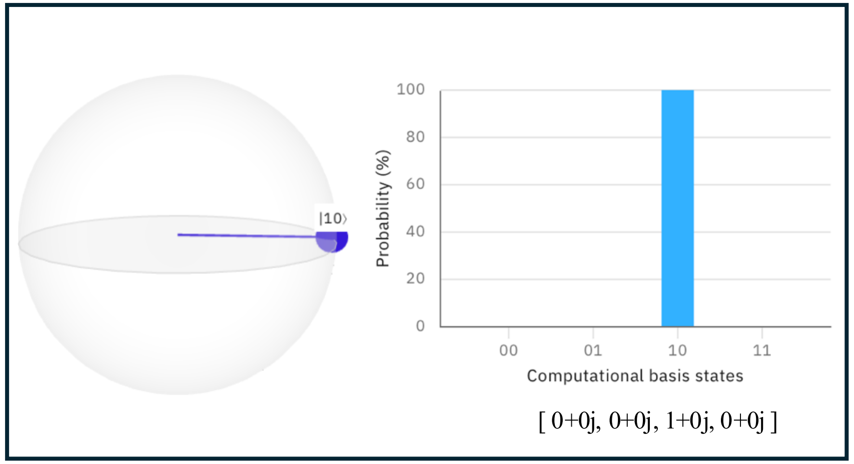
[0 + 0i] $|11\rangle = 0|00\rangle + 0|01\rangle + 0|10\rangle + 1|11\rangle$
[0 + 0i]
[0 + 0i]
[1 + 0i]

# 2 Qubits, Basis State



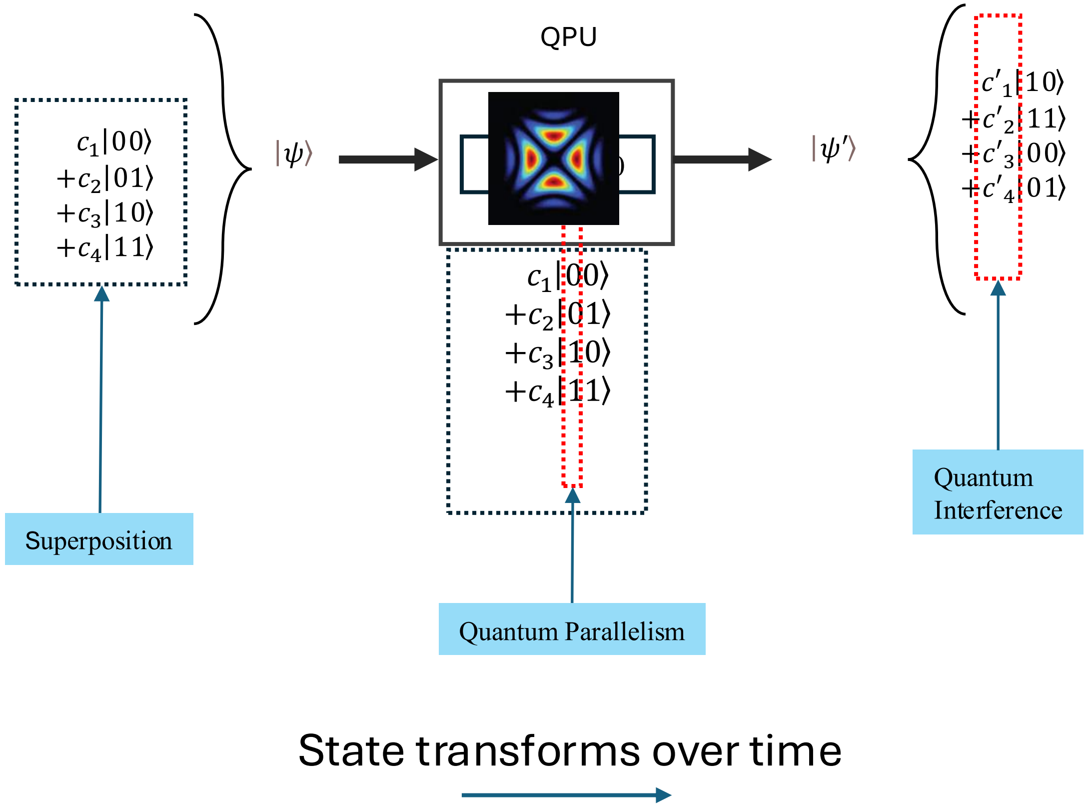

2 Qubits, Basis State

# Lab 2-1 Introduction — Exploring the IBM Quantum Composer

- **Getting Started**
  - **Create an Account**: Visit https://quantum.ibm.com and sign up for a free account.
- **Access the Quantum Composer**:
  - Log in and open the **Quantum Composer** tool.
- **Key Features to Explore**
  - **Circuit Composer:** Drag and drop quantum gates to build quantum circuits.
  - **State Visualization**: View the qubit's state on the **Bloch sphere** after each gate operation.
- **Objectives for This Lab**
  - **Read and Interpret Output States**: Observe how quantum states (state vectors) change after applying gates.
  - **Explore the Bloch Sphere**:

# Computing Units of a Quantum Computer

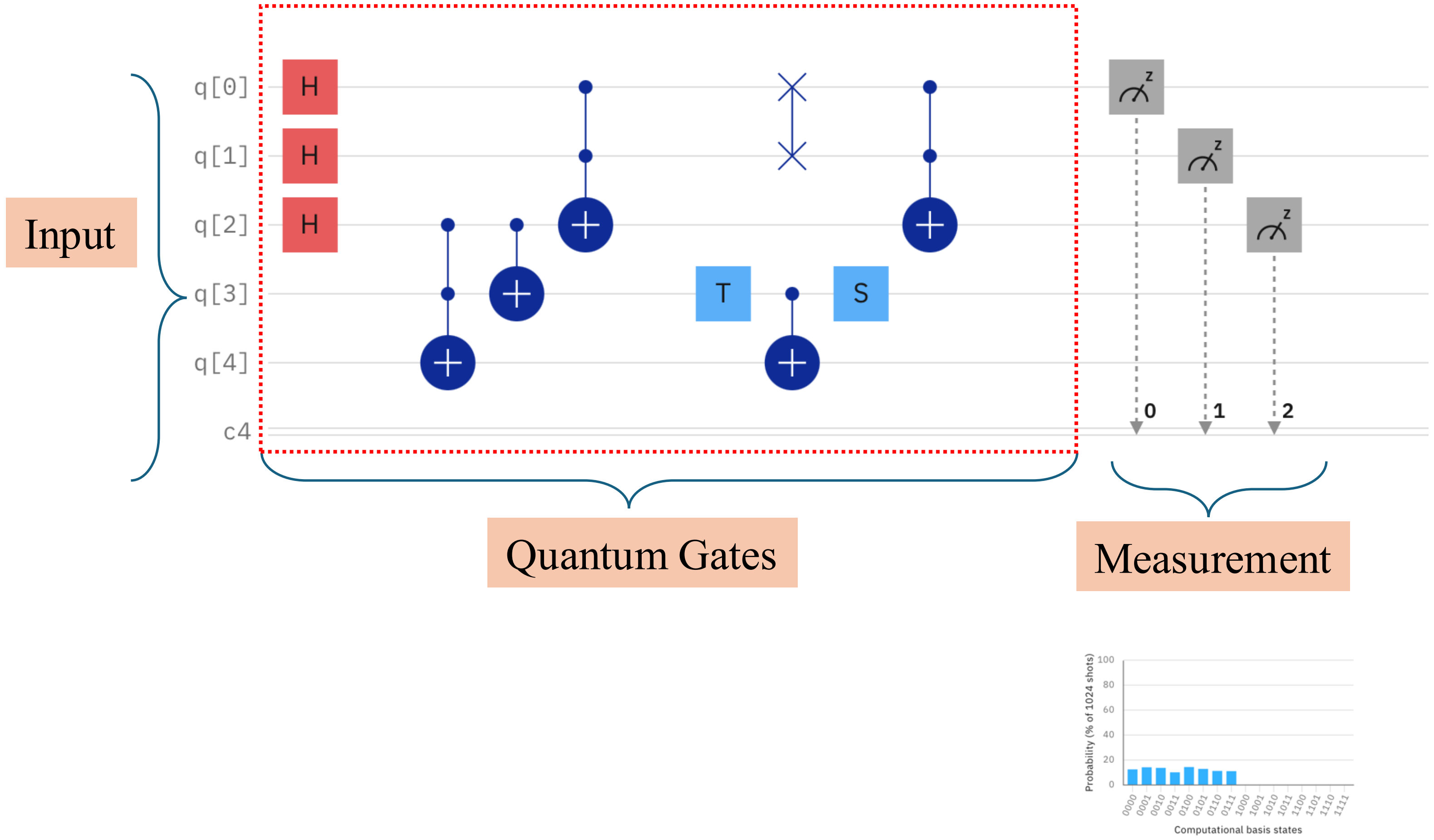
- **Quantum Computing Units**
  - A quantum computer's computing unit defines how it manipulates the quantum wavefunction to process algorithms.
  - These units determine the type of quantum computer, shaping both the hardware and the way we implement quantum algorithms.
- **Types of Quantum Computing Units**
  - **Gate-Based Quantum Computers**: Use quantum gates and circuits (e.g., IBM, Google, Rigetti).
  - **Quantum Annealers:** Solve optimization problems by finding energy minima (e.g., D-Wave).
  - **Measurement-Based Quantum Computers:** Use entangled states and measurements to compute.
  - **Topological Quantum Computers:** Encode information into topological states for fault tolerance (e.g., Microsoft's approach).
- **The type of computing unit dictates:**
  - How we design algorithms (gate-based, annealing, etc.)
  - Hardware capabilities and limitations
  - Efficiency for specific tasks



# Gate-Based Quantum Computers

- **What is a Gate-Based Quantum Computer?**
  - Processes quantum information using **quantum gates**.
  - Quantum gates operate on qubits to manipulate their states (superposition, entanglement, interference).
  - Like classical circuits but uses **unitary transformations** to modify the quantum wavefunction.
- **Mathematical Model**
  - Quantum gates are represented as **unitary matrices**
    - $\mathbb{U}(\boldsymbol{a\ unitary\ matrix}) \Rightarrow \mathbb{U}^{\dagger}\,\mathbb{U} = \boldsymbol{I}$
  - A quantum state evolves as: $|\psi'\rangle = \mathbb{U}|\psi\rangle$

# Gates-Based Quantum Computer



# The X Gate

- Mathematical Representation of the X Gate $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
- The X gate is the **quantum equivalent of a classical NOT gate**, flipping the qubit state $|0\rangle \to |1\rangle$ $|1\rangle \to |0\rangle$
- Applying X Gate to $|0\rangle$

Start with the qubit state:

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

Apply the X gate:

$$|\psi'\rangle = X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Result:

$$|\psi'\rangle = |1\rangle$$

**TABLE 2-2** Example of Quantum Gates

| Gate | Function | Matrix |
|---|---|---|
| I | No rotation | $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ |
| X | Rotate by π radians about the x-axis | $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ |
| Y | Rotate by π radians about the y-axis | $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ |
| Z | Rotate by π radians about the z-axis | $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ |
| S | Rotate by π/2 radians about z-axis | $S = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix}$ |
| H | Rotate by π radians about a diagonal in x-z plane | $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ |

**Quantum algorithms** are designed by arranging gates in specific sequences to manipulate the wavefunction and produce desired measurement outcomes.

# Grover's Circuit — Designing Quantum Algorithms

- **Grover's Circuit Example**
  - Demonstrates a **quantum algorithm** designed to solve a search problem.
- **Structure:**
  - **Input**: Encodes the initial state of the qubits.
  - **Gates**: A series of operations applied step by step to transform the wavefunction.
  - **Output**: Measurement collapses the state to reveal the result.

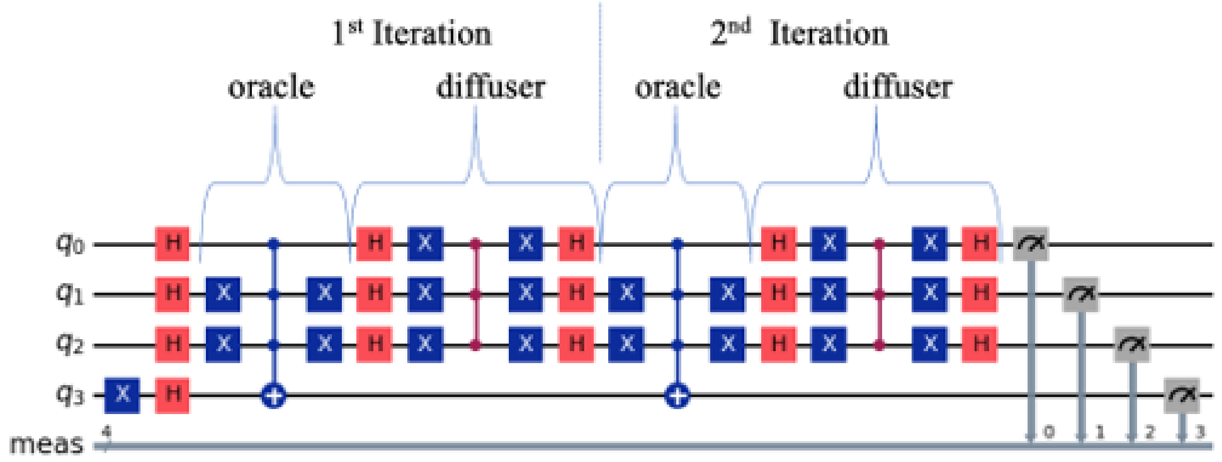



FIGURE 2-15 Quantum Circuit for Grover's Algorithm Implementation¶

# Lab 2-2 — Exploring Quantum Gates and Their Outputs

- **Lab Overview**
  - Explore how individual quantum gates manipulate qubit states.
  - Use the **IBM Quantum Composer** to visualize the effect of gates on qubits, both mathematically (state vectors) and geometrically (Bloch sphere).
- **What You'll Do**
  - Start with a single qubit in the $|0\rangle$ state
- Apply various quantum gates (e.g., X, H, Z, S, etc.) and observe:
  - **State Vector Updates**: How the qubit's mathematical representation changes.
  - **Bloch Sphere Visualization**: How the qubit's position on the Bloch sphere evolves.
  - Experiment with combinations of gates to understand their cumulative effects.
- **Goals of the Lab**
  - Develop intuition about how quantum gates manipulate qubit states.
  - Understand the relationship between gates, state vectors, and the Bloch sphere.
  - Build familiarity with the IBM Quantum Composer interface.

# Recommended Reading

- **Topics to Review**
  - **Linear Algebra**
    - Basics of vectors and matrices
    - Operations: addition, multiplication, dot product
    - Special matrices: unitary, Hermitian, and eigenvalues
  - **Quantum Phenomena**
    - Superposition
    - Entanglement
    - Quantum interference
- **Suggested Resources**
  - **Linear Algebra**
    - *"Linear Algebra and Its Applications"* by Gilbert Strang
    - Khan Academy: Linear Algebra Tutorials (online free resource)
    - 3Blue1Brown's YouTube series: *The Essence of Linear Algebra*
  - **Quantum Mechanics**
    - *"Quantum Mechanics: The Theoretical Minimum"* by Leonard Susskind
    - MIT OpenCourseWare: *Introduction to Quantum Mechanics* (online free resource)
    - IBM Quantum Documentation (link)

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

$$\alpha = ae^{i\phi_a}, \tag{2}$$

$$\beta = be^{i\phi_b}, \tag{3}$$

$$|\psi\rangle = a|0\rangle + be^{i\phi}|1\rangle, \tag{4}$$

$$\rightarrow \quad |\psi\rangle = \cos(\tfrac{\theta}{2})|0\rangle + e^{-i\phi}\sin(\tfrac{\theta}{2})|1\rangle$$

The normalization constraint of quantum states, $|\alpha|^2 + |\beta|^2 = 1$, allows you to parametrize $a$ and $b$ as

$$a = \cos(\theta/2), \tag{5}$$

$$b = \sin(\theta/2), \tag{6}$$

where $\theta$ goes from 0 to $\pi$.