Quantum Computing Basics

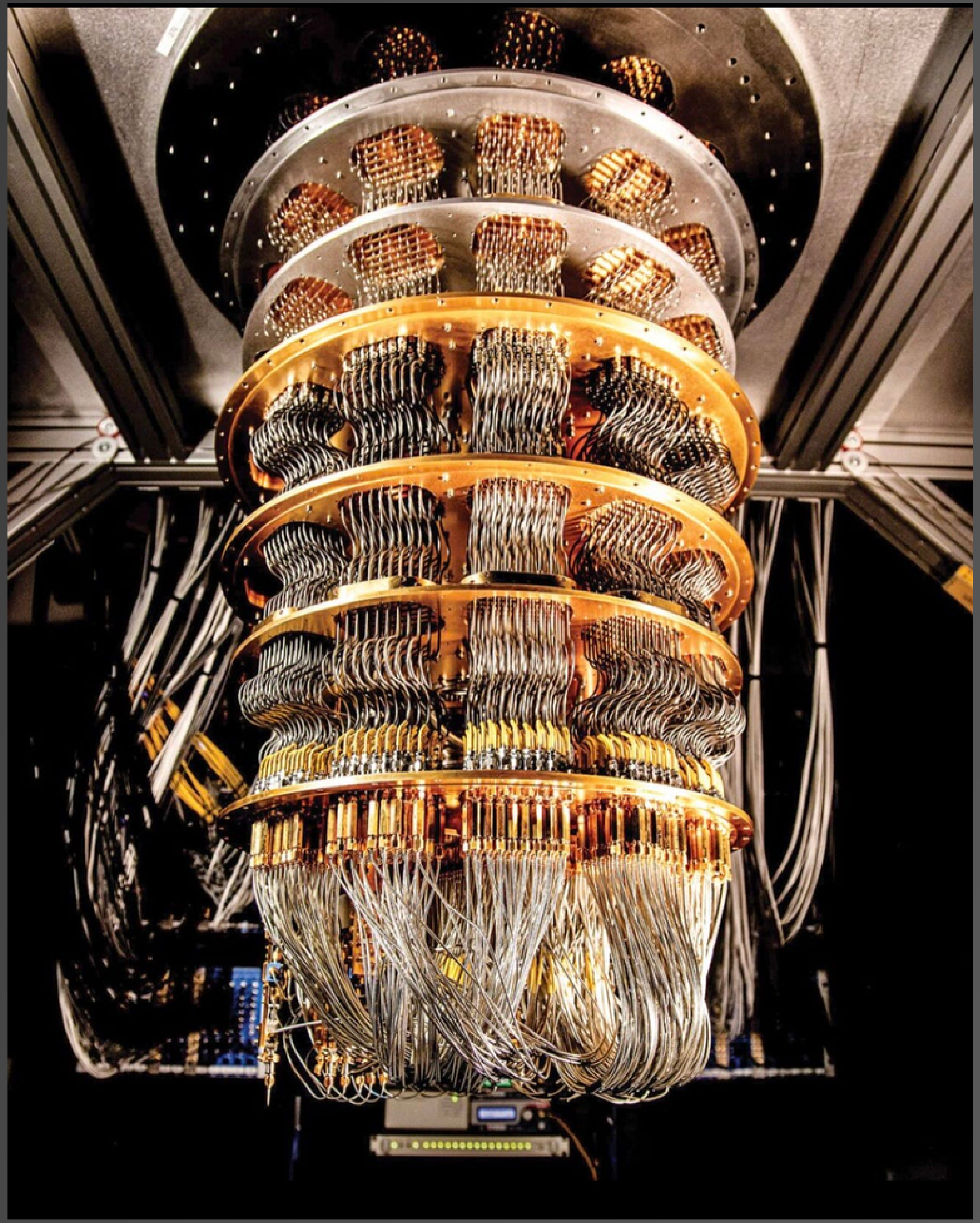
Presented by: Inderpreet Singh

Quantum Computing

- Quantum computing is a new paradigm of computing that leverages the principles of **quantum mechanics (entanglement and superposition)** to process information
- Quantum computing uses quantum bits, or **qubits**, to encode information as **0s, 1s, or both 0 and 1 simultaneously.**
- Has many applications in cryptography/cybersecurity, optimization problems, drug discovery etc.

Quantum Computers

- Quantum computers work on quantum mechanics principle of **entanglement and superposition**.
- Quantum computers need extremely low temperatures **(-273.15°C)**
- Quantum computers are susceptible to **noise** so lots of quantum error correction is required
- The current state of quantum computing is referred to as **NISQ ("Noisy Intermediate-Scale Quantum”) era**
- The highest qubit computer we have today is around **1000+ qubits**
- **Decoherence** is a major problem for quantum computers

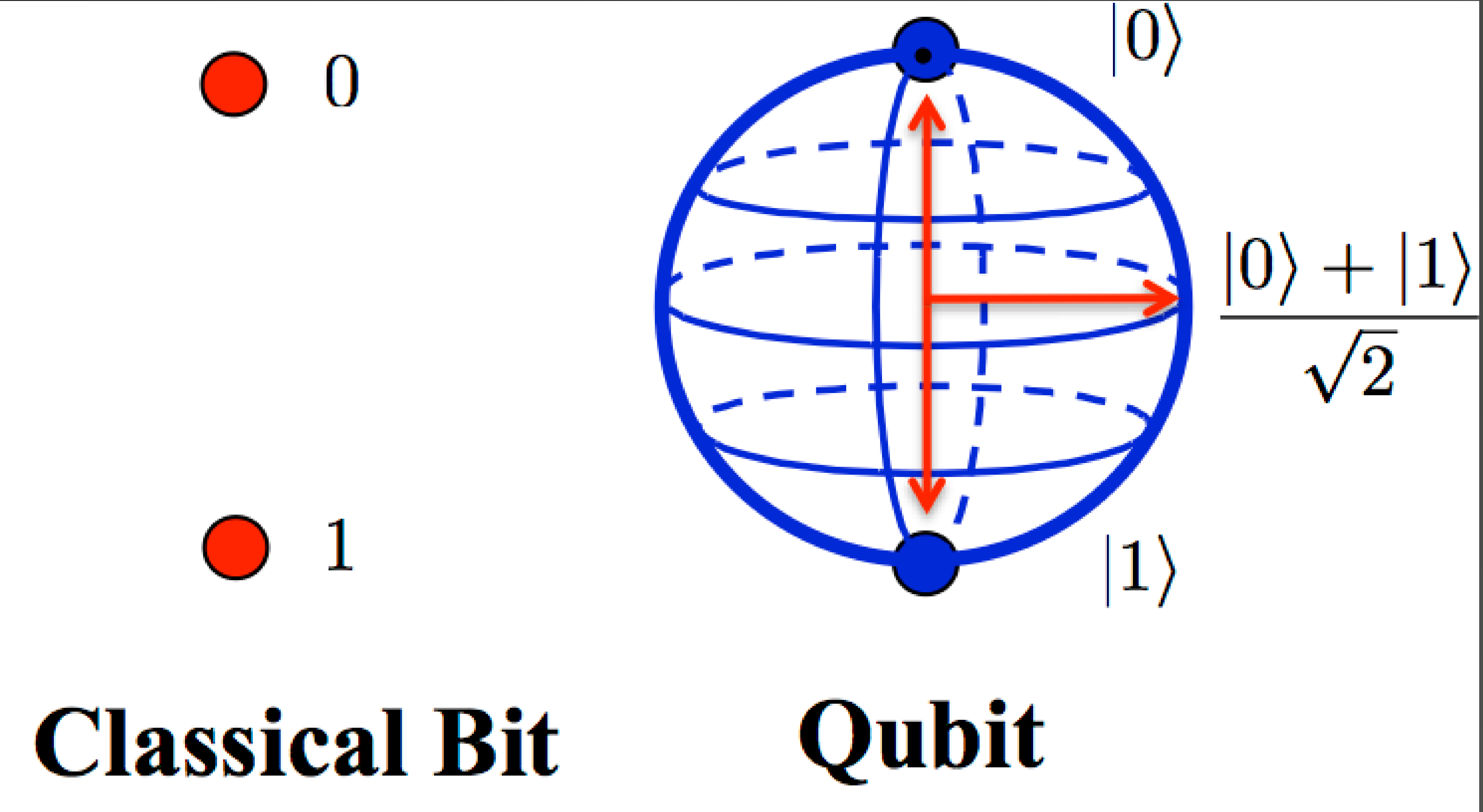


Qubits

In Quantum Computing, the information is stored in form of **qubits**

Qubits

In Quantum Computing, the information is stored in form of **qubits**



Representing Qubits

A qubit state in Dirac notation, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, can be represented in matrix form as a column vector:

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

The basis states $|0\rangle$ and $|1\rangle$ themselves can be represented in matrix form as:

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

Matrix Multiplication

- The process of multiplying two matrices by taking the dot product of rows of the first matrix with the columns of the second matrix.
- If we have two matrices A and B, their product is denoted as $C=AB$.
- For the multiplication to be valid, the number of columns in the first matrix must equal the number of rows in the second matrix. If A is an $m \times n$ matrix and B is an $n \times p$ matrix, their product C will be an $m \times p$ matrix.

Matrix Multiplication

- The process of multiplying two matrices by taking the dot product of rows of the first matrix with the columns of the second matrix.
- If we have two matrices A and B, their product is denoted as C=AB.
- For the multiplication to be valid, the number of columns in the first matrix must equal the number of rows in the second matrix. If A is an $m \times n$ matrix and B is an $n \times p$ matrix, their product C will be an $m \times p$ matrix.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax + by \\ cx + dy \end{pmatrix}$$

Quantum Gates

- Just as logic gates are the **fundamental building blocks** of classical circuits, quantum gates serve as the essential components of quantum circuits.

- A quantum gate receives one or more qubits, performs a **specific quantum operation**, and outputs the qubits in a new state.

- Key Types of Quantum Gates
  - Hadamard Gate (H): Creates superposition, turning a definite state into a combination of possibilities.
  - Pauli Gates (X, Y, Z): Flip qubits in various ways, similar to classical NOT gates but with quantum twists.
  - Controlled NOT Gate (CNOT): Entangles two qubits, flipping the second (target) qubit if the first (control) qubit is in state 1.

Quantum Gates in Matrix Form

- Quantum gates, which are operations applied to qubits, are represented by matrices.
- When a quantum gate is applied to a qubit, the operation is represented by the matrix multiplication of the gate matrix with the qubit state vector.

Quantum Gates in Matrix Form

- Quantum gates, which are operations applied to qubits, are represented by matrices.
- When a quantum gate is applied to a qubit, the operation is represented by the matrix multiplication of the gate matrix with the qubit state vector.

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

Quantum Gates in Matrix Form

- **Pauli-X Gate** (Quantum NOT Gate):

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

- **Pauli-Y Gate:**

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

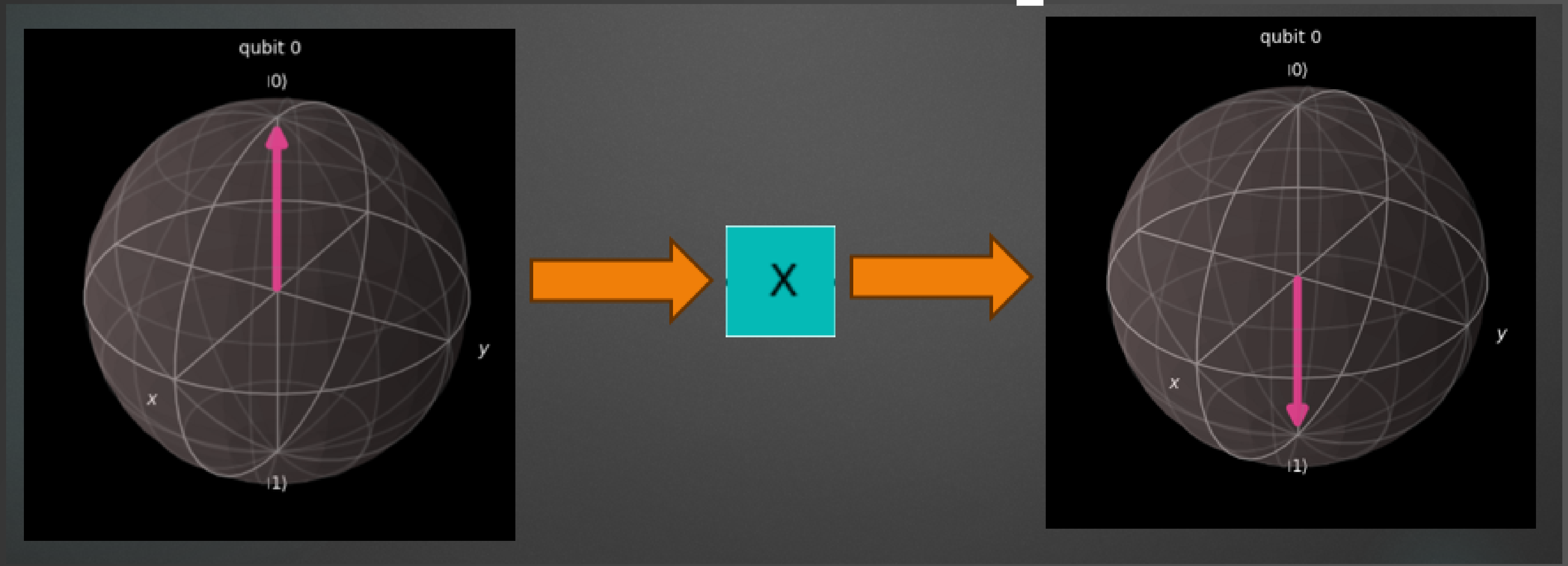
- **Pauli-Z Gate:**

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

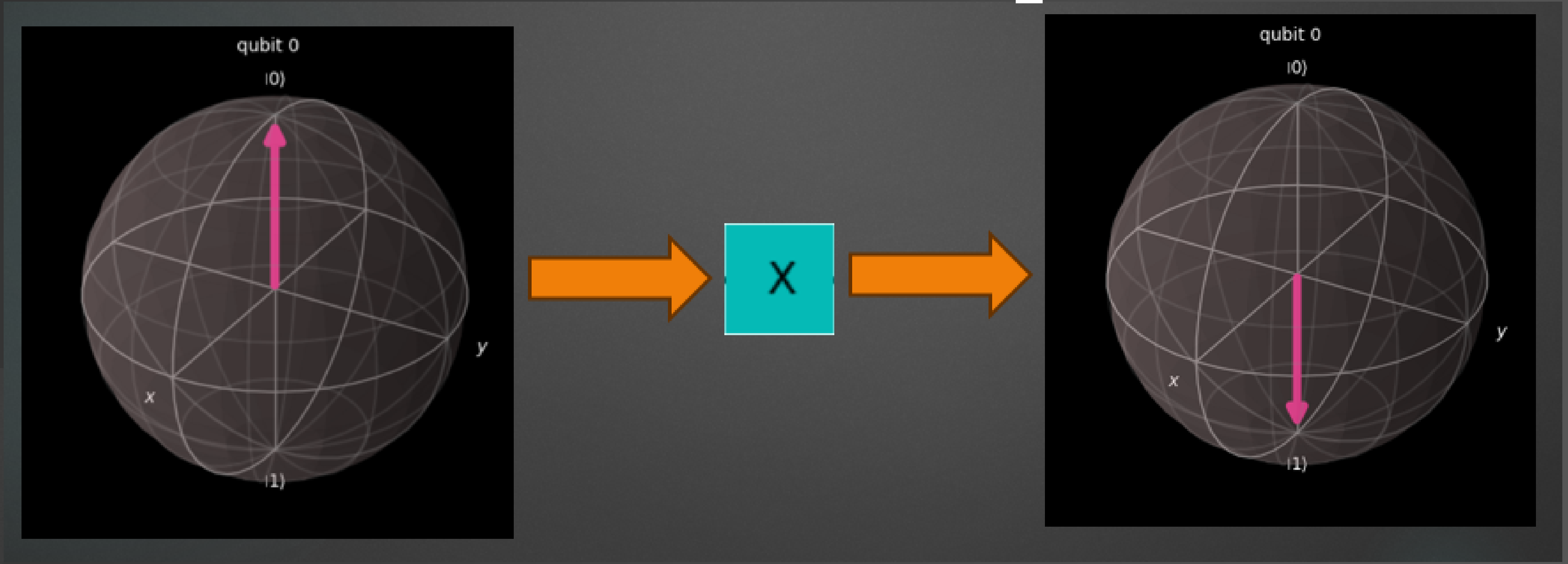
- **Hadamard Gate:**

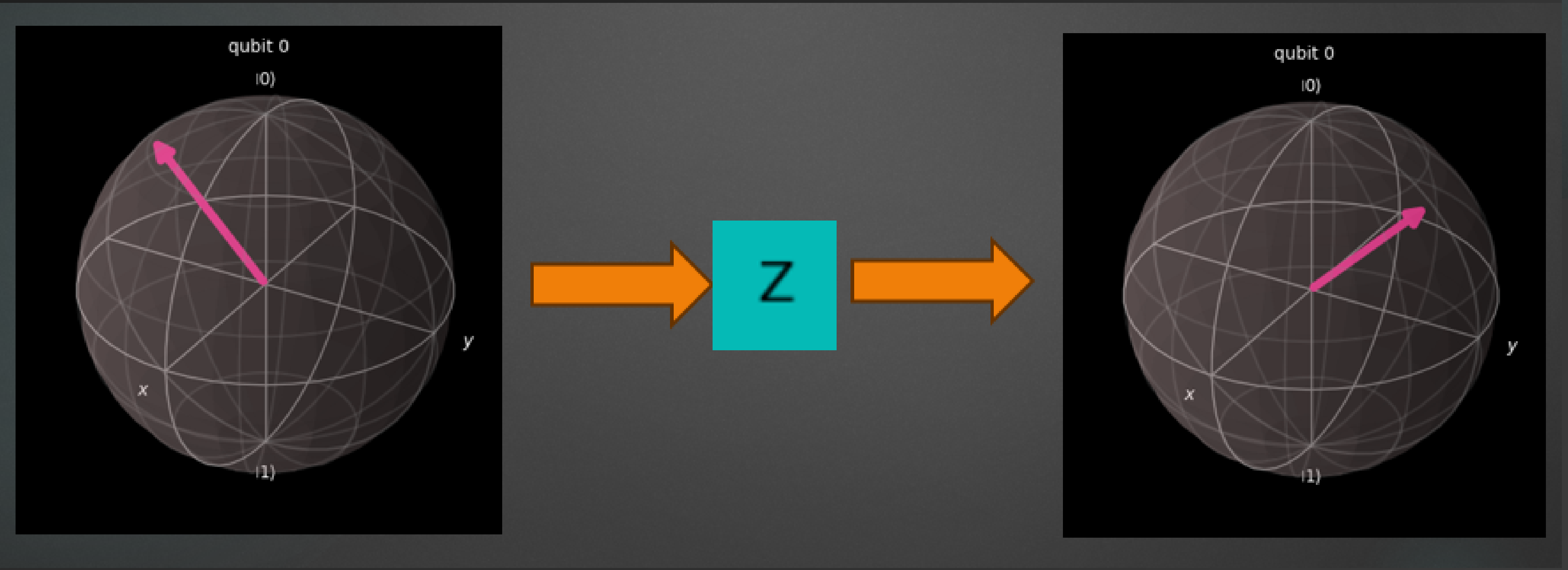
$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Quantum Gates in Bloch Sphere

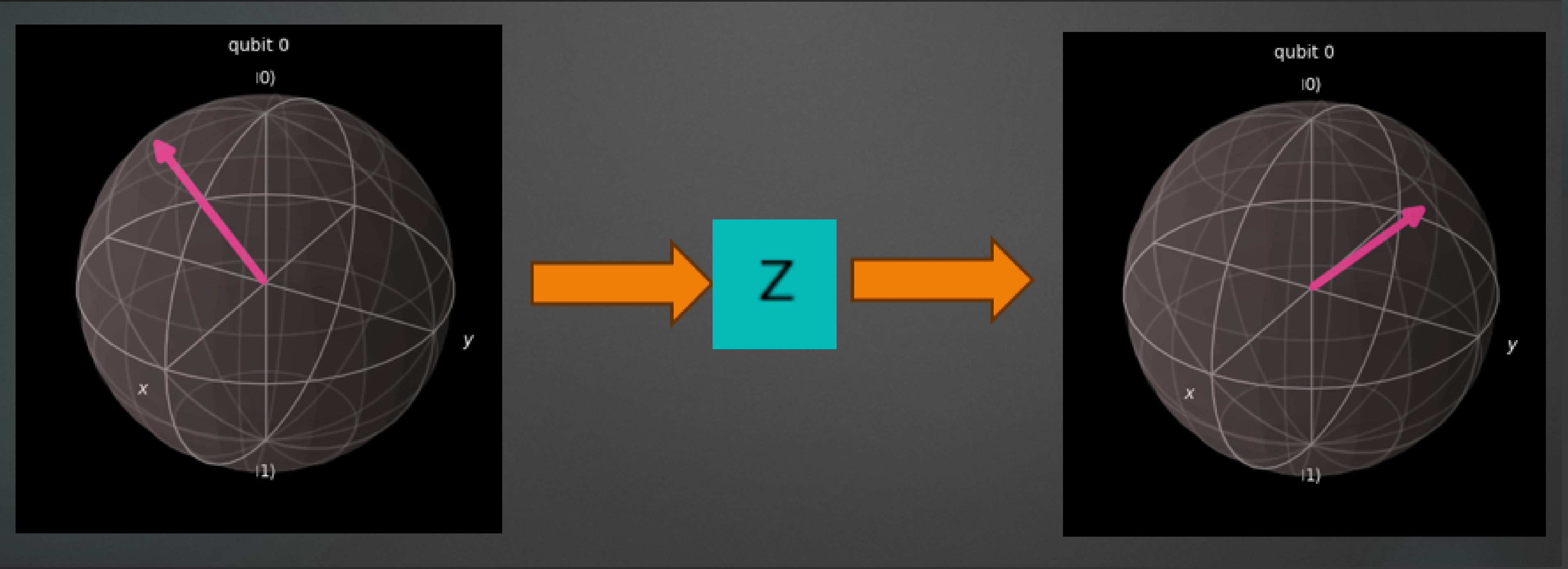


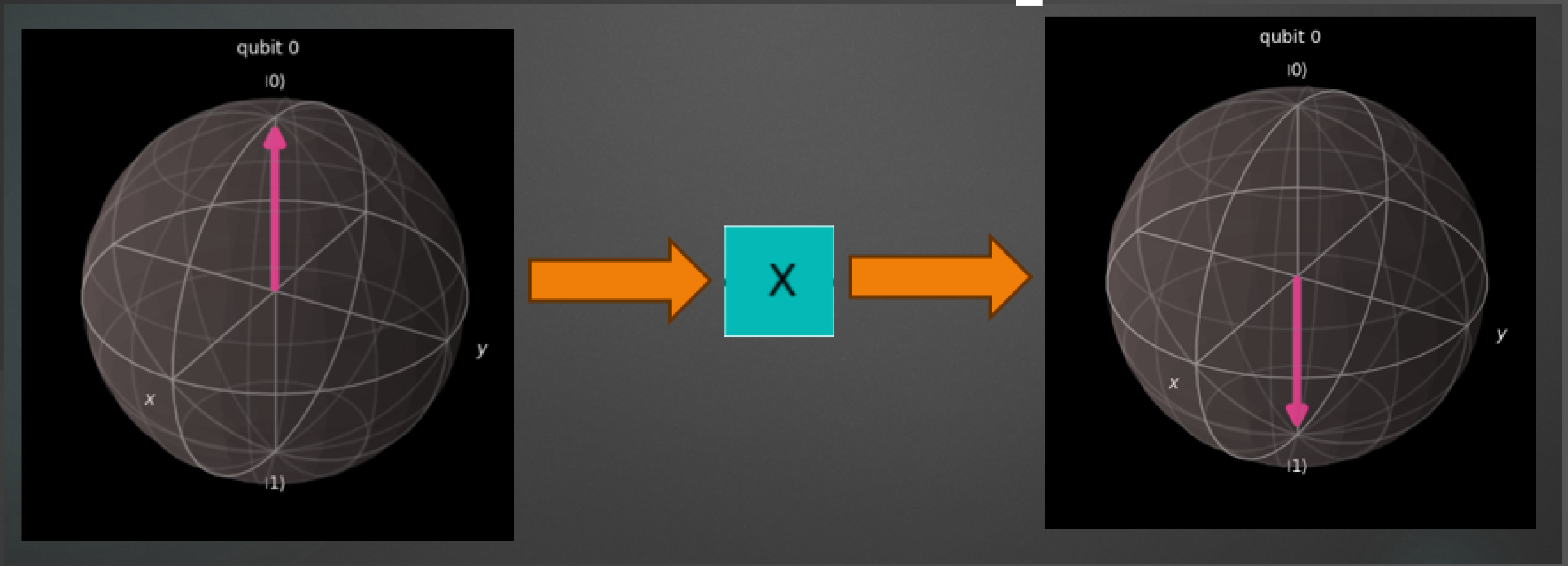
Quantum Gates in Bloch Sphere

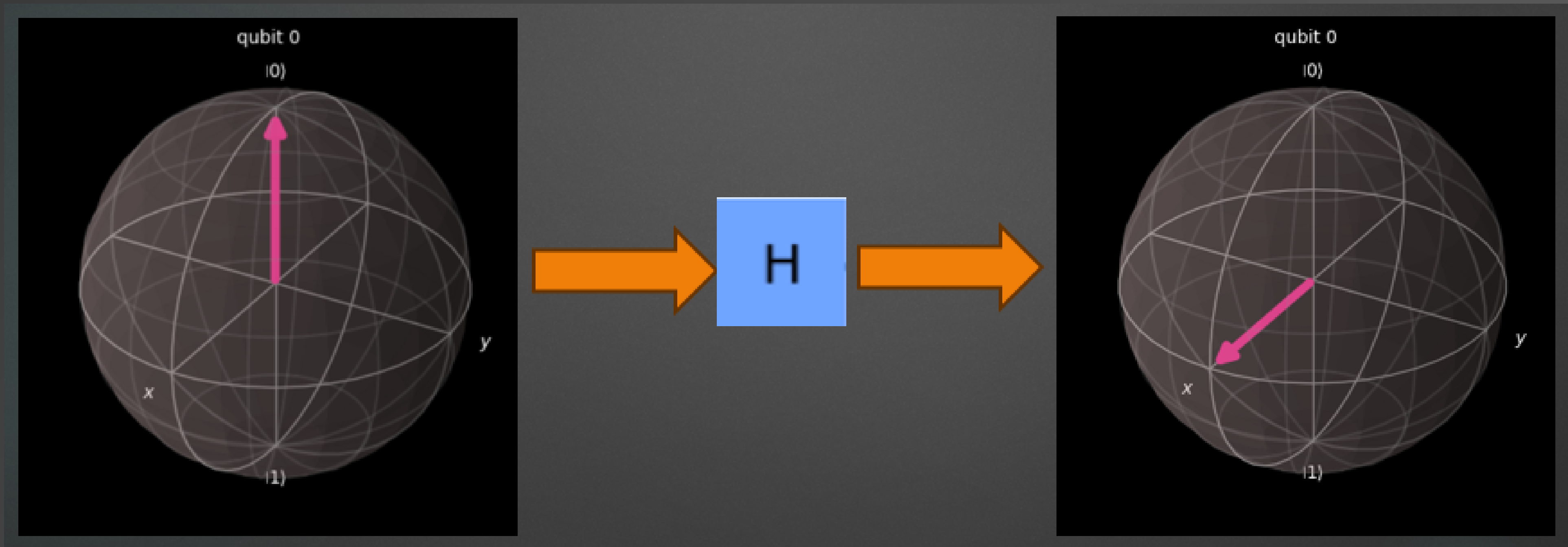




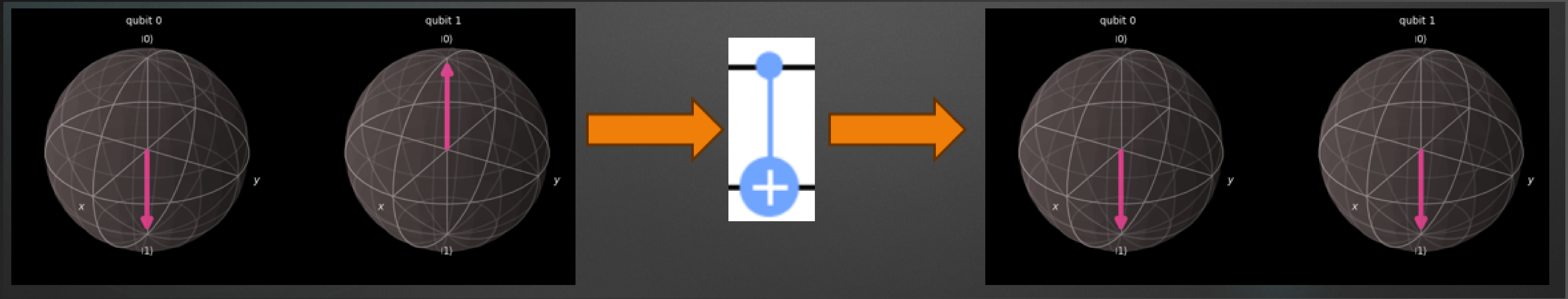
Quantum Gates in Bloch Sphere







Quantum Gates in Bloch Sphere

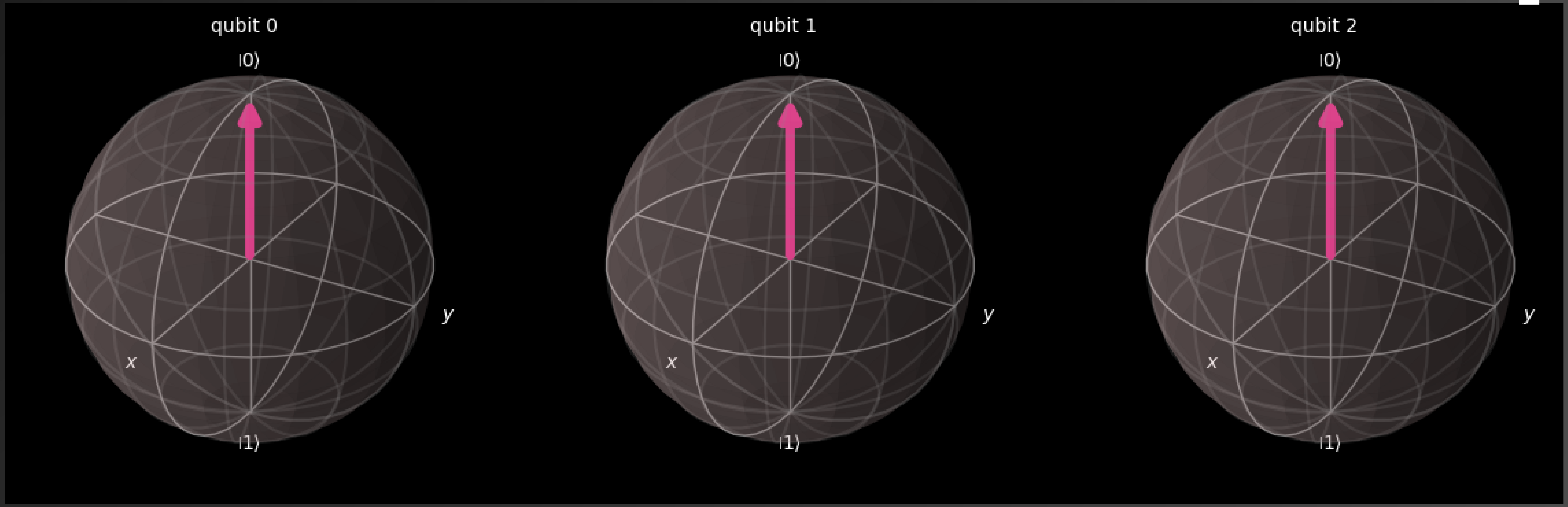


The CNOT gate is a mulit-qubit gate that consists of two qubits. The first qubit is known as the control qubit and the second is known as the target qubit. If the control qubit is |1〉 then it will flip the targets qubit state from |0〉 to |1〉 or vice versa.

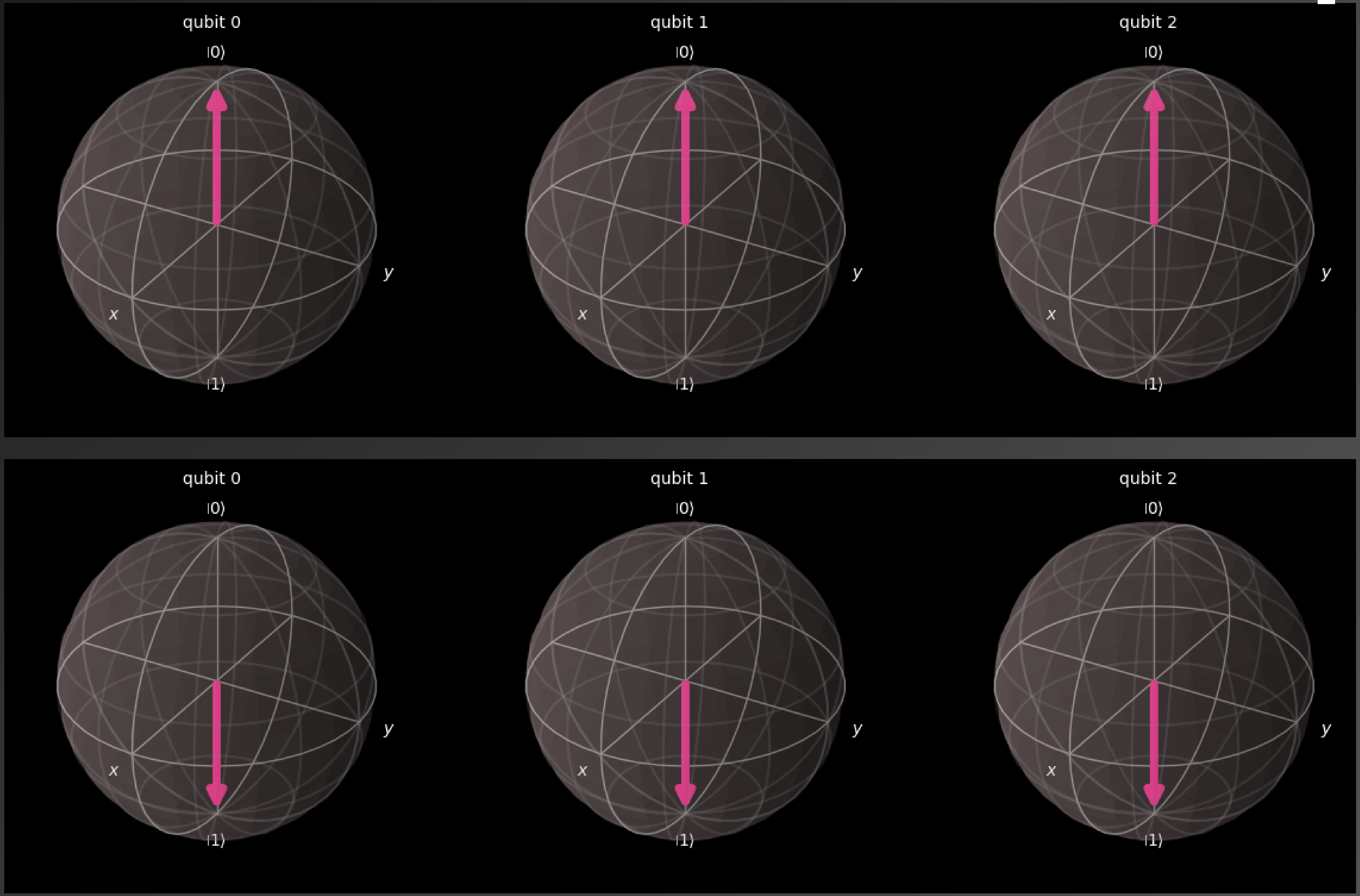
Quantum Gates in Bloch Sphere

The **Toffoli gate**, also known as the Controlled-Controlled-NOT gate or **CCNOT**, is a universal quantum gate with three qubits - two control qubits and one target qubit. The Toffoli gate flips the state of the target qubit if, and only if, both control qubits are in the state $|1\rangle|1\rangle$

Quantum Gates in Bloch Sphere

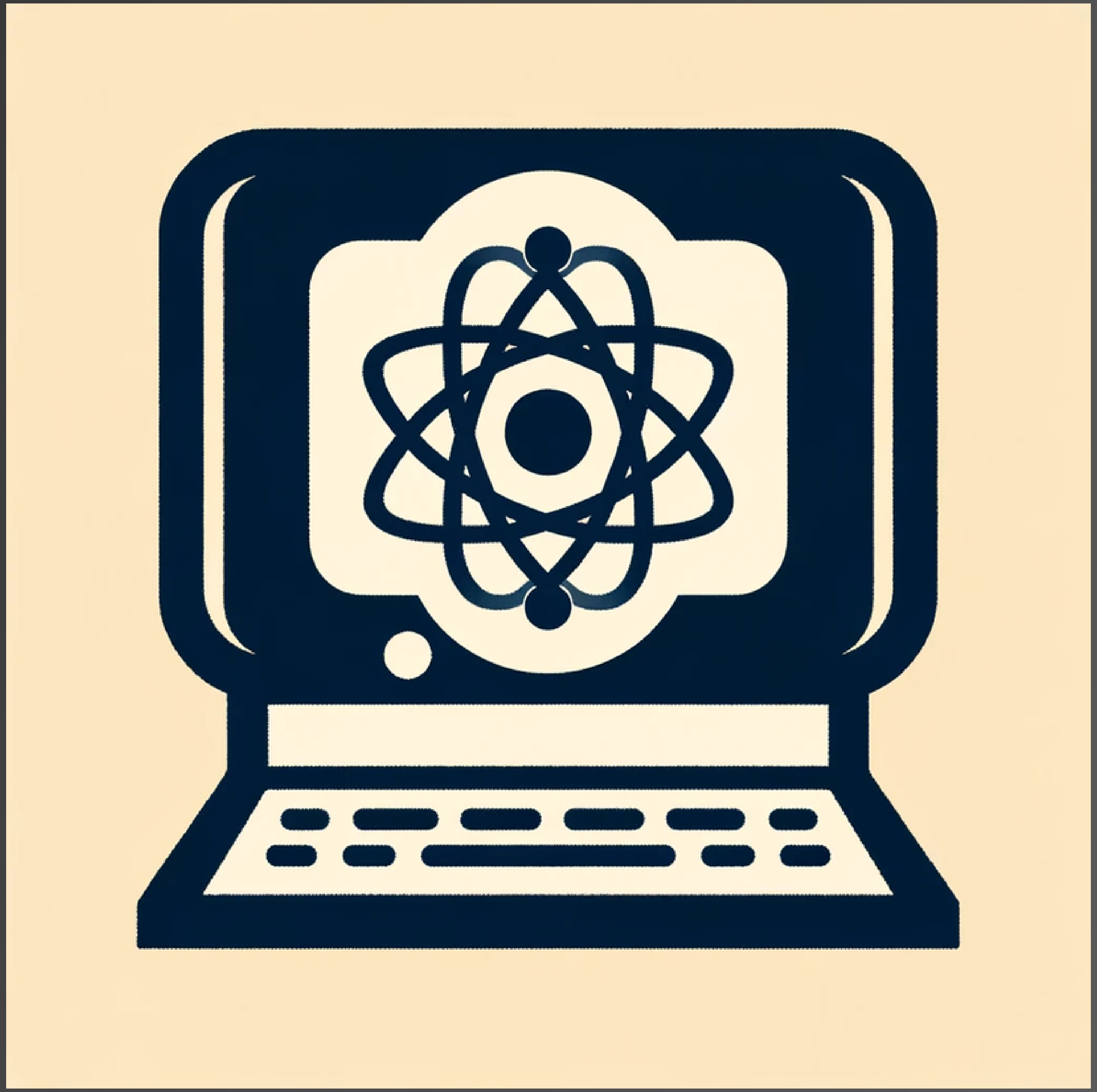


Quantum Gates in Bloch Sphere



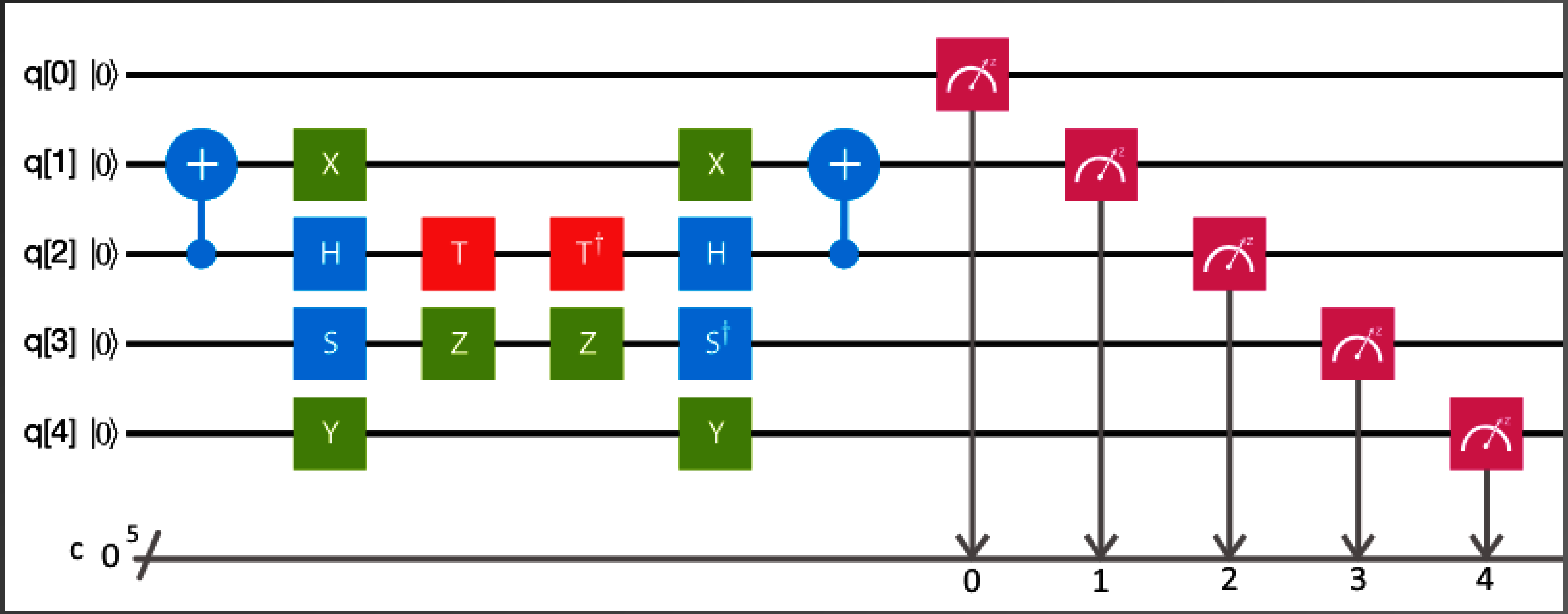
Quantum Simulators

- Quantum computers are **limited and expensive**.
- You need **time and money** to run your program on an actual quantum computer
- Quantum simulators are tools that replicate the **behaviors of quantum systems** within the constraints of classical computers.
- They allow researchers to **test and iterate** on complex quantum algorithms without the need for a full-scale quantum computer.
- Example Qiskit Aer



Quantum Circuits

- Quantum circuits are **series of quantum gates linked together**, where the output of one gate becomes the input for the next.
- Gates are arranged in a sequence or in parallel configurations, depending on the computation. The **sequence and type of gates determine the circuit's function**.



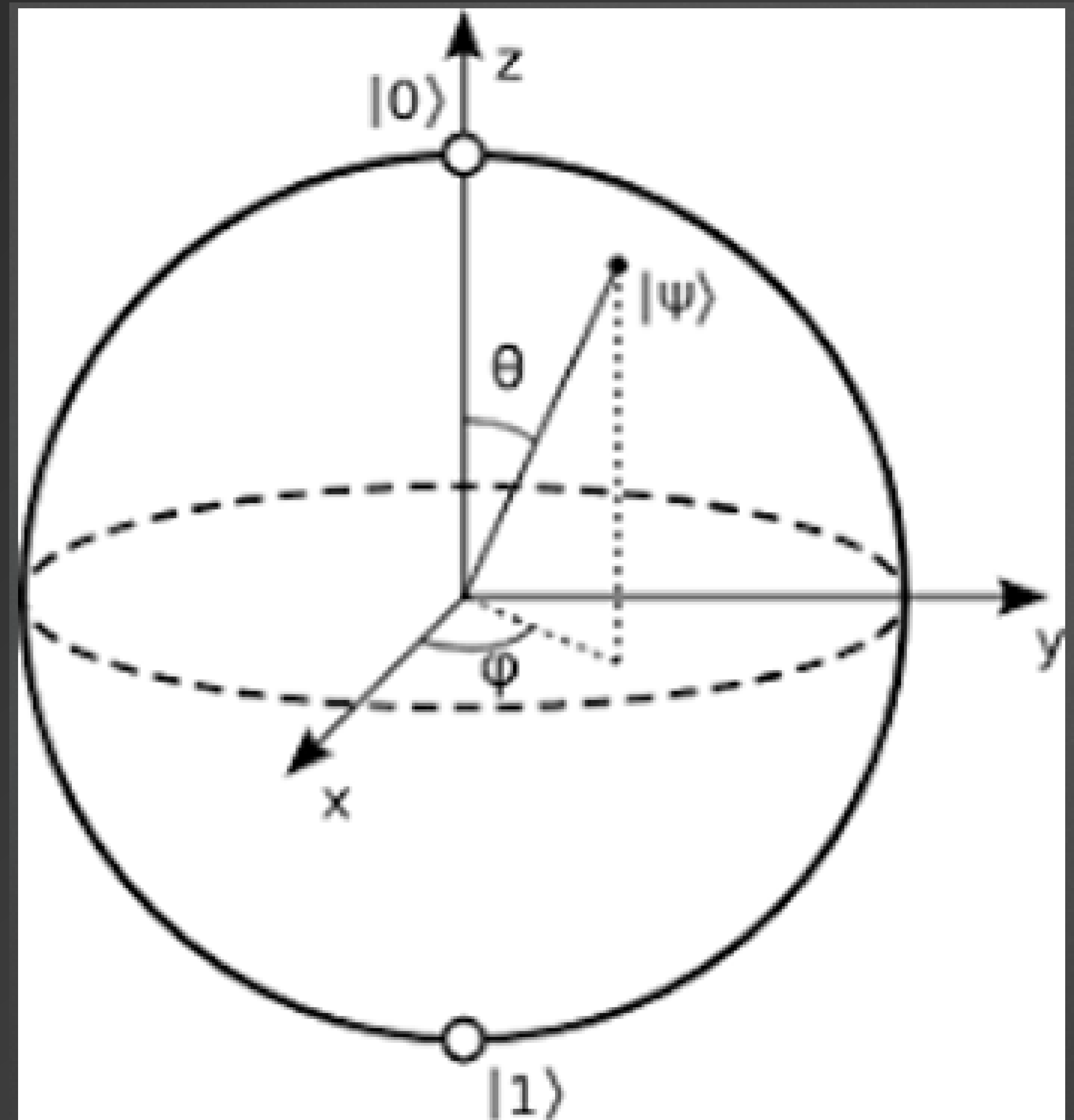
Qiskit: The Quantum Toolkit

- Qiskit is an open-source quantum computing software development framework that allows users to work with quantum computers at a higher level.
- It provides tools for creating, simulating, and running quantum circuits on real quantum hardware or simulators.

Visualization

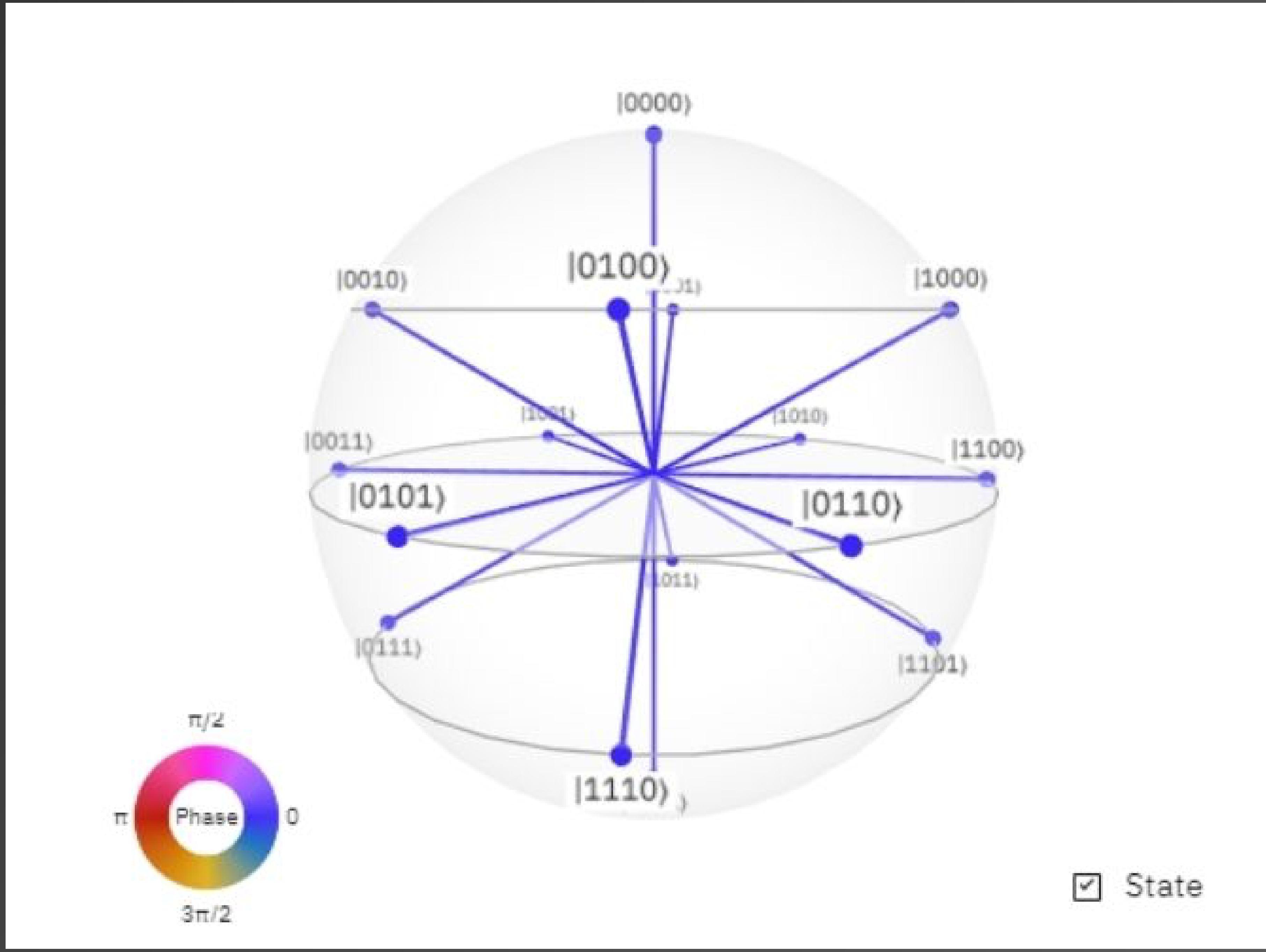
Bloch Sphere

- The Bloch Sphere is a three-dimensional representation used to visualize **the state of a single qubit.**
- It depicts every possible state of a qubit as points **on the surface of sphere.**



Visualization
Q Sphere

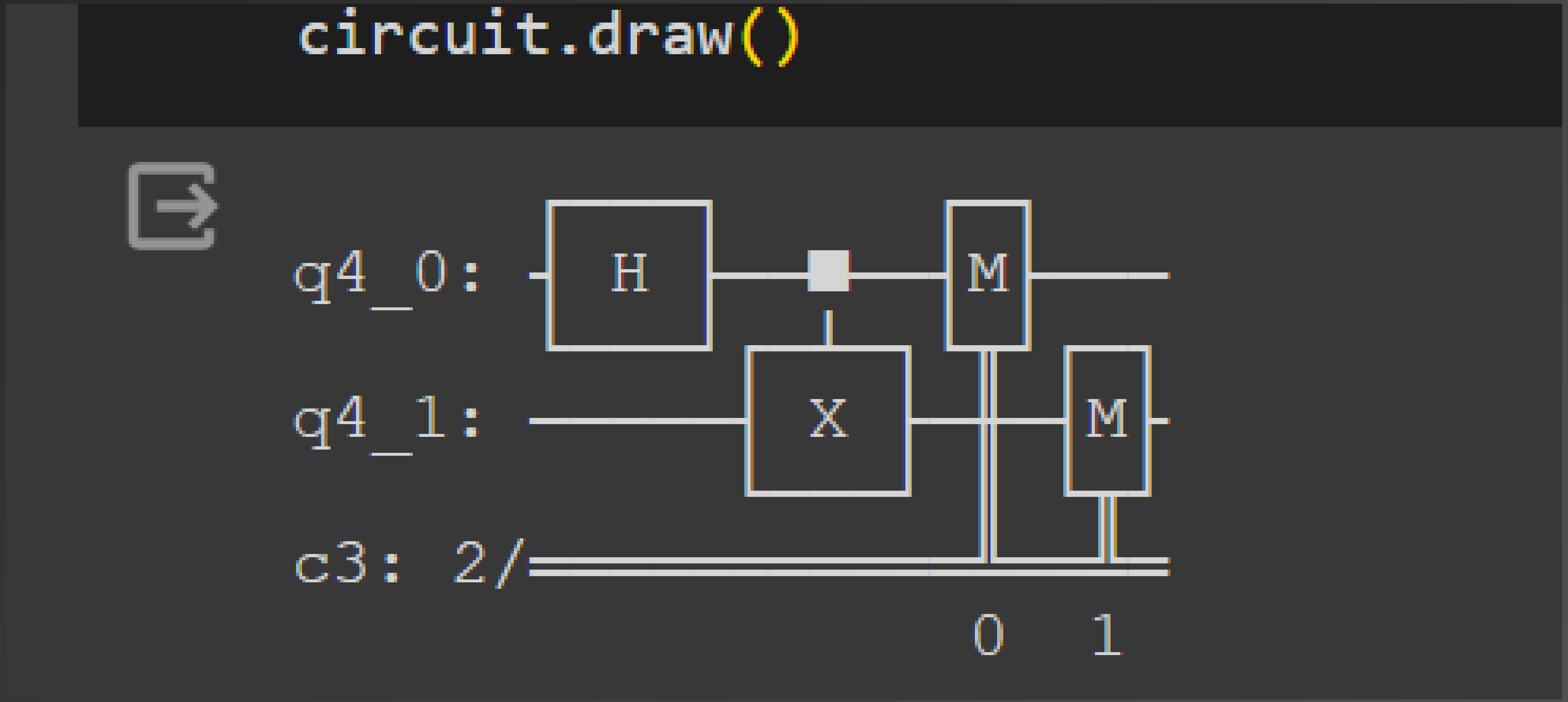
- The Q Sphere extends the concept of the Bloch Sphere to visualize **states of multiple qubits.**
- It represents complex multi-qubit states and their relationships in a visually intuitive manner.
- Q-sphere is about visualizing state up to few qubits, and the Bloch sphere is about visualizing a single qubit.



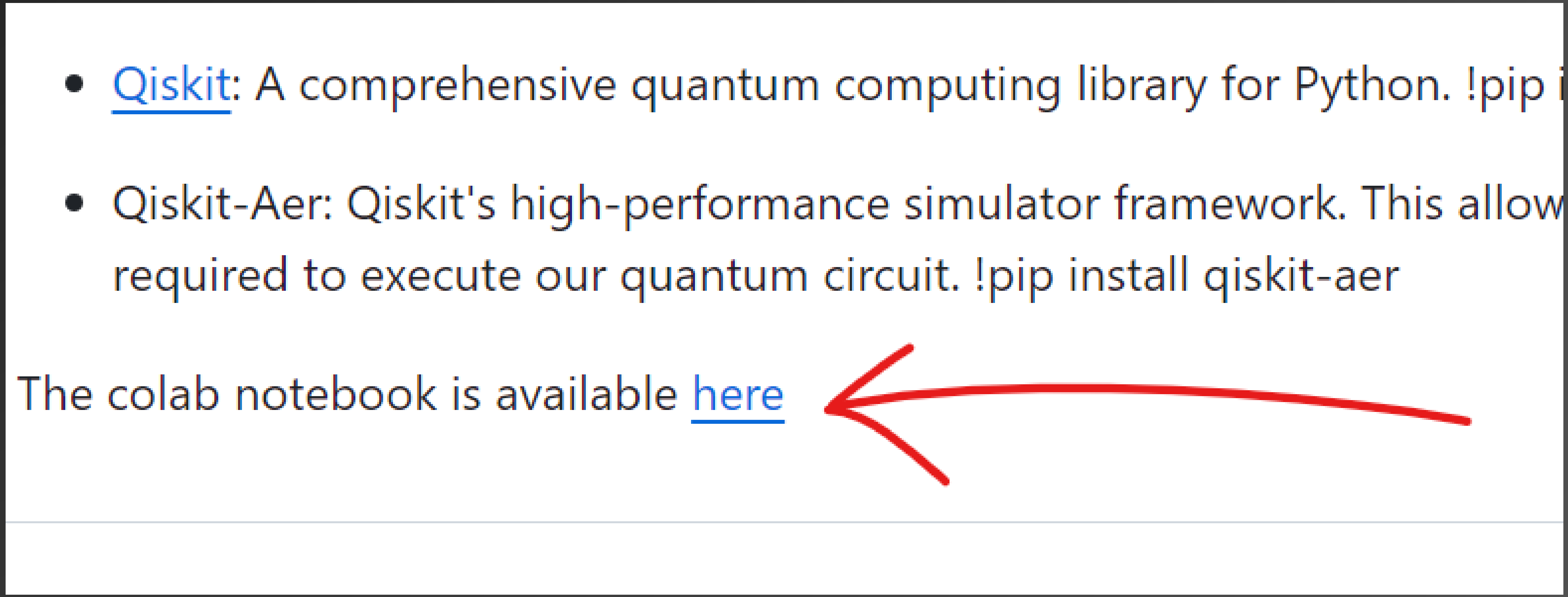
Visualization

Quantum Circuit

- These diagrams are schematic representations showing the **sequence of quantum gates** (the operations) applied to qubits (the lines).



- Go to **https://github.com/inderpreet1390/qc**
- Navigate to **Qiskit-Update** folder
- Click on this link at the bottom



- Qiskit: A comprehensive quantum computing library for Python. !pip i
- Qiskit-Aer: Qiskit's high-performance simulator framework. This allow required to execute our quantum circuit. !pip install qiskit-aer

The colab notebook is available here

IBM Quantum

https://quantum.ibm.com/

IBM Quantum is a platform for learning and prototyping quantum circuits in Qiskit and implementing them on simulators or even real quantum computers