**Single Qubit Gates – Detailed Explanation**

**1. What is a Qubit?**  
A qubit (quantum bit) is the smallest unit of quantum information.

* In classical computing, a bit can only be 0 or 1.
* In quantum computing, a qubit can be in a superposition — meaning it can be partly 0 and partly 1 at the same time.

We usually write the two basic quantum states as:  
|0⟩ and |1⟩

* |0⟩ means the qubit is in the zero state.
* |1⟩ means the qubit is in the one state.

Any qubit state can be written as:  
|ψ⟩ = α|0⟩ + β|1⟩  
where:

* α and β are complex numbers (probability amplitudes).
* |α|² + |β|² = 1 (probabilities must add up to 1).

**2. What is a Single Qubit Gate?**  
A single qubit gate is an operation that changes the state of one qubit.

Mathematically:

* Qubit states are represented as 2×1 column vectors:  
  |0⟩ = [1, 0]ᵀ  
  |1⟩ = [0, 1]ᵀ
* Single qubit gates are represented as 2×2 unitary matrices (U†U = I, so probabilities are preserved).
* Applying a gate means matrix multiplication:  
  |ψ'⟩ = U |ψ⟩

**3. Examples of Single Qubit Gates**  
**(a) Pauli-X Gate (Quantum NOT gate)**  
Matrix:  
[ 0 1 ]  
[ 1 0 ]  
Effect: Flips |0⟩ ↔ |1⟩.  
X|0⟩ = |1⟩, X|1⟩ = |0⟩

**(b) Pauli-Y Gate**  
Matrix:  
[ 0 -i ]  
[ i 0 ]  
Effect: Similar to X but introduces a phase factor i.

**(c) Pauli-Z Gate (Phase Flip)**  
Matrix:  
[ 1 0 ]  
[ 0 -1 ]  
Effect: Leaves |0⟩ unchanged but flips the sign of |1⟩.

**(d) Hadamard Gate (H)**  
Matrix:  
(1/√2) [ 1 1 ]  
[ 1 -1 ]  
Effect: Creates superposition:  
H|0⟩ = (|0⟩ + |1⟩)/√2  
H|1⟩ = (|0⟩ - |1⟩)/√2

**(e) Phase Gate (S)**  
Matrix:  
[ 1 0 ]  
[ 0 i ]  
Effect: Adds a phase i to |1⟩.

**4. Why Single Qubit Gates Matter**

* They are the basic building blocks of quantum algorithms.
* When combined with multi-qubit gates (like CNOT), they can perform any quantum computation.
* They control rotation and phase in quantum states.

Multiple Qubit Gates – Detailed Explanation

1. What are Multiple Qubit Gates?  
   Multiple qubit gates are quantum operations that act on two or more qubits at the same time.

* While single qubit gates change the state of only one qubit, multiple qubit gates can change the state of multiple qubits in a way that can create **entanglement**.
* Entanglement is a quantum property where the states of qubits become linked, meaning the state of one qubit depends on the state of another, even if they are far apart.

1. Representation of Multi-Qubit States

* For two qubits, the possible basis states are:  
  |00⟩, |01⟩, |10⟩, |11⟩
* Each basis state is represented as a 4×1 column vector:  
  |00⟩ = [1, 0, 0, 0]ᵀ  
  |01⟩ = [0, 1, 0, 0]ᵀ  
  |10⟩ = [0, 0, 1, 0]ᵀ  
  |11⟩ = [0, 0, 0, 1]ᵀ
* Multi-qubit gates are represented as unitary matrices of size 2ⁿ × 2ⁿ, where n is the number of qubits.

1. Examples of Common Multiple Qubit Gates

(a) Controlled-NOT Gate (CNOT)  
Matrix:  
[ 1 0 0 0 ]  
[ 0 1 0 0 ]  
[ 0 0 0 1 ]  
[ 0 0 1 0 ]  
Effect:

* The first qubit is the control qubit, the second is the target qubit.
* If the control qubit is |1⟩, the target qubit is flipped.
* If the control qubit is |0⟩, nothing changes.

(b) Controlled-Z Gate (CZ)  
Matrix:  
[ 1 0 0 0 ]  
[ 0 1 0 0 ]  
[ 0 0 1 0 ]  
[ 0 0 0 -1 ]  
Effect:

* If both qubits are |1⟩, the amplitude gets multiplied by -1 (phase flip).

(c) SWAP Gate  
Matrix:  
[ 1 0 0 0 ]  
[ 0 0 1 0 ]  
[ 0 1 0 0 ]  
[ 0 0 0 1 ]  
Effect:

* Swaps the states of two qubits.
* |01⟩ becomes |10⟩ and vice versa.

(d) Toffoli Gate (CCNOT)  
Matrix: 8×8 matrix (not written fully here due to size).  
Effect:

* Has two control qubits and one target qubit.
* If both controls are |1⟩, the target qubit is flipped.
* This gate is important for quantum error correction and reversible classical logic.

1. Why Multiple Qubit Gates Matter

* They are essential for creating entanglement, which gives quantum computers their advantage.
* They allow conditional operations (one qubit’s state controls what happens to another).
* Together with single qubit gates, they can perform any quantum computation.

Quantum Circuits – Detailed Explanation

1. Introduction  
   A quantum circuit is a sequence of quantum operations (gates and measurements) applied to one or more qubits to perform a computation.

* Similar to classical logic circuits, quantum circuits use gates to process information.
* The key difference is that quantum gates manipulate qubits, which can be in superposition and entangled states.
* A quantum circuit diagram visually shows the sequence of operations on qubits over time.

1. Components of a Quantum Circuit

(a) Qubits

* The horizontal lines in a quantum circuit represent qubits.
* Qubits start in an initial state, usually |0⟩, and evolve as gates are applied.

(b) Quantum Gates

* Operations that change the state of qubits.
* Single qubit gates: act on one qubit (e.g., X, Y, Z, H, S, T).
* Multiple qubit gates: act on two or more qubits (e.g., CNOT, CZ, SWAP).

(c) Measurement

* At the end (or sometimes during the circuit), qubits are measured.
* Measurement collapses the quantum state into a classical outcome (0 or 1).

1. Reading a Quantum Circuit Diagram

* Time flows from left to right.
* Each horizontal wire is a qubit.
* Boxes or symbols along a wire represent gates applied to that qubit.
* Vertical lines connecting wires represent multi-qubit gates.

Example:  
• A Hadamard gate (H) followed by a CNOT gate can create an entangled state.

1. Example – Creating a Bell State  
   Step 1: Start with two qubits in |00⟩.  
   Step 2: Apply a Hadamard gate to the first qubit to create superposition: (|0⟩ + |1⟩)/√2.  
   Step 3: Apply a CNOT gate with the first qubit as control and second as target.  
   Result: The two qubits are now in the Bell state: (|00⟩ + |11⟩)/√2.

Circuit diagram (text representation):  
q0 — H —■—  
|  
q1 ————X—

1. Importance of Quantum Circuits

* They are the blueprint for running algorithms on quantum computers.
* They define the order and type of operations to achieve the desired quantum computation.
* Quantum circuits can be simulated on classical computers for testing before execution on real hardware.

**Bell States**

Bell states are **special entangled two-qubit states** that cannot be separated into individual qubit states. They are the simplest and most famous examples of *quantum entanglement*.

**Why they matter**

* They form a *maximally entangled basis* for two qubits.
* Any two-qubit state can be expressed as a combination of Bell states.
* They’re crucial for quantum communication protocols like **quantum teleportation** and **superdense coding**.

**The Four Bell States**

These are usually written as:

1. **Φ⁺** = (|00⟩ + |11⟩) / √2
2. **Φ⁻** = (|00⟩ − |11⟩) / √2
3. **Ψ⁺** = (|01⟩ + |10⟩) / √2
4. **Ψ⁻** = (|01⟩ − |10⟩) / √2

Here:

* **|00⟩** means *both qubits are 0*
* **|11⟩** means *both qubits are 1*
* **|01⟩** means *first qubit is 0, second is 1*, etc.
* The 1/√2 keeps the probabilities normalized.

**Key property:**  
If you measure one qubit of a Bell state, the other qubit’s state is instantly determined, no matter how far apart they are.

**Creating a Bell State**

Let’s make **Φ⁺** step-by-step:

1. **Start:** |00⟩ (two qubits both in 0 state)
2. **Hadamard on qubit 1:** Turns first qubit into superposition: (|0⟩ + |1⟩) / √2 ⊗ |0⟩
3. **CNOT:** Control = qubit 1, Target = qubit 2
   * If control is |0⟩, target stays the same.
   * If control is |1⟩, target flips.  
     → Result: (|00⟩ + |11⟩) / √2

**Quantum Teleportation**

Quantum teleportation is a **method to send the *state* of a qubit from one place to another** without physically sending the qubit itself — and without violating relativity.

**Main idea**

1. **Pre-share entanglement:**  
   Alice and Bob share a pair of qubits in a Bell state (Φ⁺, for example).
2. **Alice’s side:**  
   She has another qubit with an unknown state **|ψ⟩ = α|0⟩ + β|1⟩** that she wants to send to Bob.
3. **Bell measurement:**  
   Alice performs a special measurement (Bell basis measurement) on her unknown qubit and her half of the Bell pair.
4. **Send classical bits:**  
   Alice sends Bob **two classical bits** of information about her measurement result.
5. **Bob’s correction:**  
   Depending on the bits, Bob applies one of four unitary operations (I, X, Z, or XZ) to his qubit.  
   After correction, Bob’s qubit is exactly in the original state |ψ⟩.

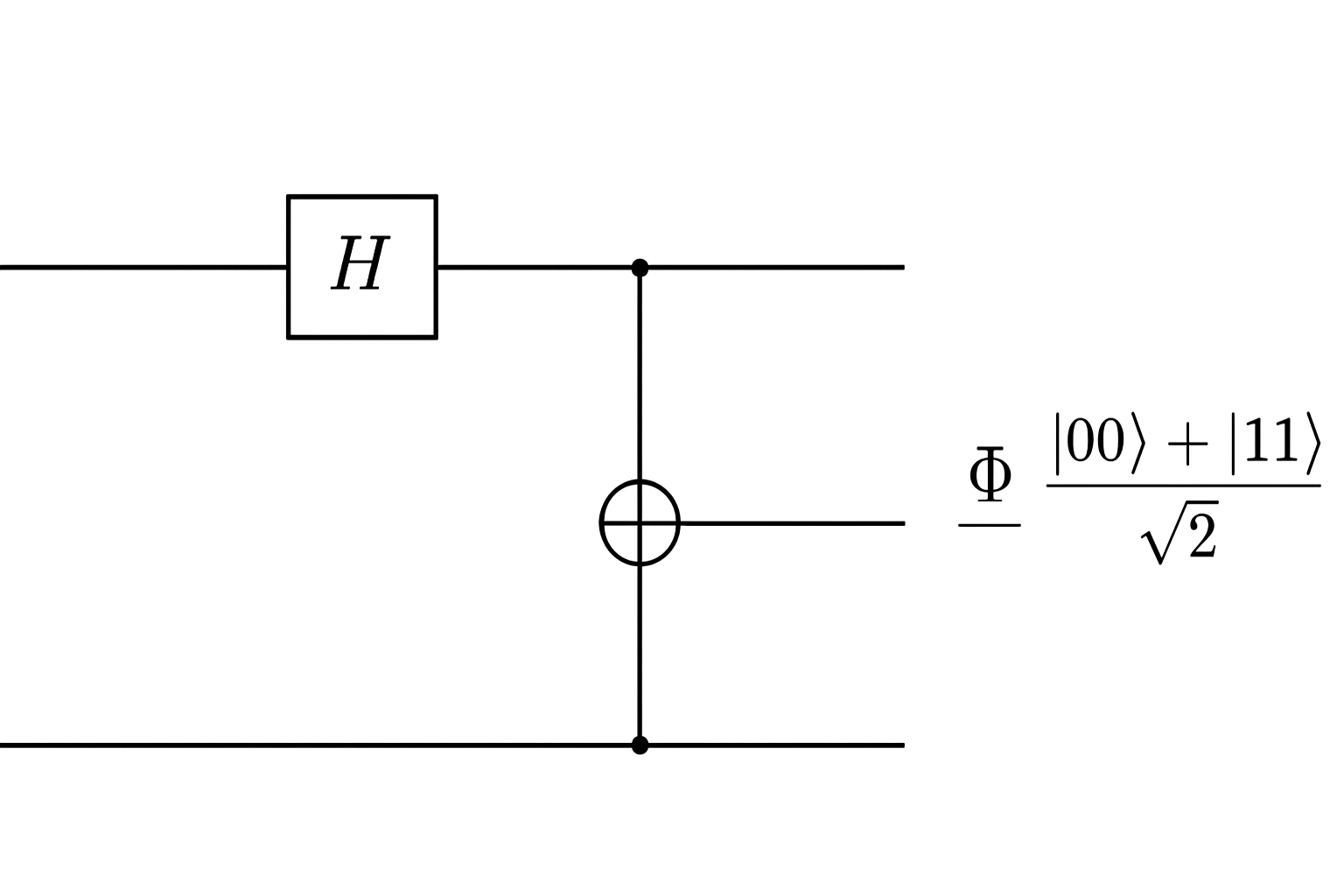
**Step-by-Step Teleportation Process**

**Initial setup:**

* Alice has qubits: Q1 = |ψ⟩, Q2 = her half of Bell pair.
* Bob has qubit: Q3 = his half of Bell pair.

**Procedure:**

1. **Alice entangles her two qubits** (Q1 and Q2) using CNOT + Hadamard.
2. **She measures Q1 and Q2** → gets two classical bits (00, 01, 10, or 11).
3. **She sends those bits** to Bob over a normal classical channel.
4. **Bob applies correction** based on the bits:
   * 00 → I (do nothing)
   * 01 → X (flip)
   * 10 → Z (phase flip)
   * 11 → XZ (flip + phase flip)
5. **Now Bob’s qubit (Q3)** is in the exact state |ψ⟩.



**Hilbert Space in Quantum Computing – Easy Explanation**

In **quantum computing**, a **Hilbert space** is the mathematical space where all possible quantum states of a system are represented.  
It is like the "universe" of all possible configurations your qubits can have.

**Key Points (Quantum Computing View):**

* **State Representation**
  + Each qubit is represented by a **vector** in a Hilbert space.
  + For example, a single qubit lives in a **2-dimensional Hilbert space**, spanned by **|0⟩** and **|1⟩**.
* **Multiple Qubits**
  + For *n* qubits, the Hilbert space has **2ⁿ dimensions**.
  + Example: 2 qubits → 4D Hilbert space (basis: |00⟩, |01⟩, |10⟩, |11⟩).
* **Inner Product**
  + The inner product in Hilbert space lets us find **probabilities** of measurement outcomes.
* **Completeness**
  + Any valid quantum state must exist inside this space, including **superposition** and **entangled states**.
* **Quantum Gates**
  + Operations on qubits (quantum gates) are **mathematical transformations** that move the state vector around in the Hilbert space without leaving it.

**Example:**

* **1 qubit** → Hilbert space: 2D  
  |ψ⟩ = α|0⟩ + β|1⟩  
  where α and β are complex numbers with |α|² + |β|² = 1.
* **2 qubits** → Hilbert space: 4D  
  |ψ⟩ = α|00⟩ + β|01⟩ + γ|10⟩ + δ|11⟩

**Products and Tensor Products – Easy Explanation (Quantum Computing)**

**1. Product States (Simple Product)**

* In quantum computing, if you have **two qubits** and they are **not entangled**, their combined state can be written as the **product** of their individual states.
* **Example:**
  + Qubit A: |ψ⟩ = α|0⟩ + β|1⟩
  + Qubit B: |φ⟩ = γ|0⟩ + δ|1⟩
  + Combined state: |ψ⟩ × |φ⟩ = (α|0⟩ + β|1⟩)(γ|0⟩ + δ|1⟩)

**2. Tensor Product**

* The **tensor product** (denoted ⊗) is the mathematical way to combine quantum states into a larger Hilbert space.
* It creates a new vector that contains **all possible combinations** of the basis states.

**Example:**  
1 qubit: |0⟩ = 1,01, 01,0  
1 qubit: |1⟩ = 0,10, 10,1

Tensor product:  
|0⟩ ⊗ |1⟩ = 1,01, 01,0 ⊗ 0,10, 10,1 = 0,1,0,00, 1, 0, 00,1,0,0  
This corresponds to the **two-qubit basis state** |01⟩.

**3. Why Tensor Product is Important in Quantum Computing**

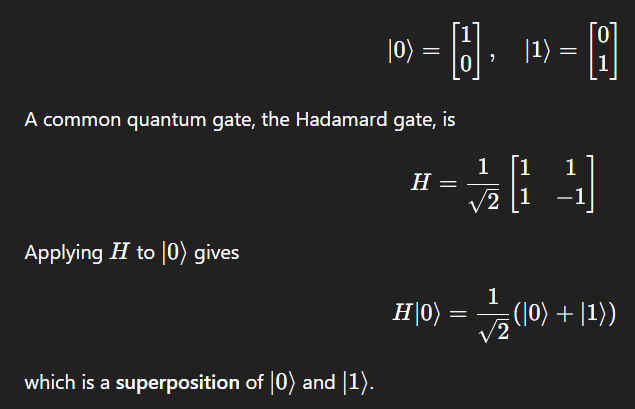
* It is how we mathematically **combine qubits** to form multi-qubit systems.
* If we have **n qubits**, their total Hilbert space is the **tensor product** of the Hilbert spaces of each qubit.
* This is why **n qubits** have **2ⁿ basis states**.

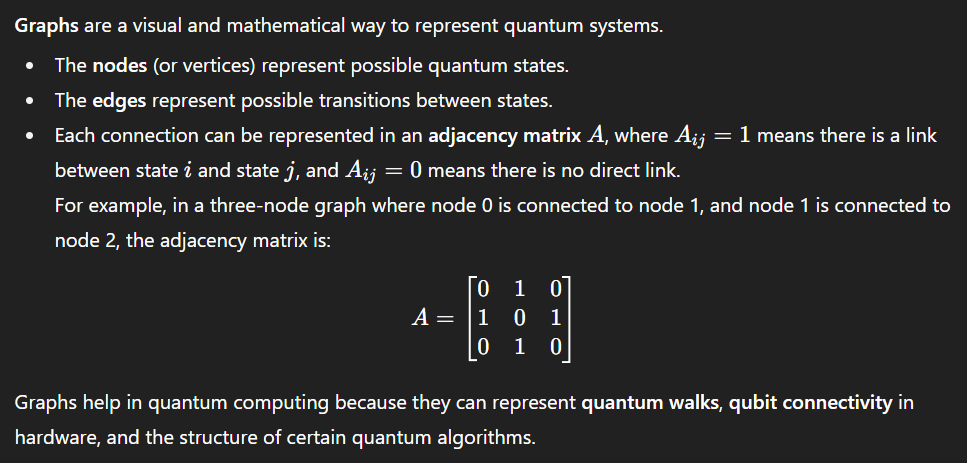
**Example for 2 Qubits:**  
Qubit 1: |ψ⟩ = α|0⟩ + β|1⟩  
Qubit 2: |φ⟩ = γ|0⟩ + δ|1⟩

Tensor product:  
|ψ⟩ ⊗ |φ⟩ = αγ|00⟩ + αδ|01⟩ + βγ|10⟩ + βδ|11⟩

**Matrices, Graphs, and Sums Over Paths (Quantum Computing)**

In quantum computing, **matrices** play an important role because they describe how quantum states evolve over time. A quantum state is written as a **vector**, and each quantum gate or operation is represented as a **unitary matrix**. Multiplying the matrix by the state vector changes the state according to the rules of quantum mechanics. For example, the computational basis states are:



****

**Sums over paths** is a concept that comes from quantum mechanics and is also used in quantum computing. It says that there can be many possible **paths** for a quantum system to go from one state to another.

* Each path has an **amplitude** (a complex number) which represents both the probability size and the phase of that path.
* The total amplitude from an initial state to a final state is found by **adding** the amplitudes of all possible paths.
* If two paths have amplitudes that point in the same direction, they **add up** (constructive interference).
* If they point in opposite directions, they can **cancel out** (destructive interference).

