MODULE 3:
QUANTUM GATES AND CIRCUITS

CONTENTS

QUANTUM GATES

Quantum gates are the fundamental operations in quantum computing, manipulating qubits through unitary transformations. Unlike classical gates, they are reversible and leverage superposition and phase to enable complex computations. Quantum gates are mainly classified into two types.

- Single - Qubit Gates
- Multi - Qubit Gates

Single-Qubit Gates:

In quantum computing, single qubit gates are operations that act on a single qubit, changing its state by applying a unitary transformation (a reversible matrix with complex entries).

- A qubit is represented as: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad \text{where } |\alpha|^2 + |\beta|^2 = 1$
- Single qubit gates are represented by 2×2 unitary matrices.

QUANTUM GATES

Pauli Gates:

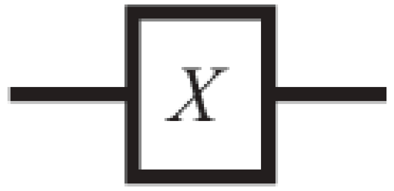
Named after physicist Wolfgang Pauli, the Pauli gates are the X gate, Y gate, and Z gate. Pauli gates perform a rotation of 180 degrees around the X, Y, and Z axes of the Bloch sphere, respectively.

Pauli-X Gate (NOT Gate):

This gate is analogous to the NOT gate in classical computing. It flips the state of the qubit from |0⟩ to |1⟩ or from |1⟩ to |0⟩.

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

Pauli-X (NOT)



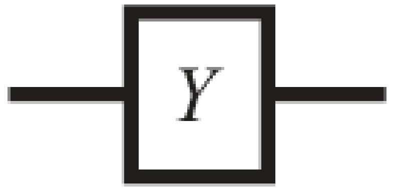
Ex: $X|0\rangle = |1\rangle, X|1\rangle = |0\rangle$

Pauli-Y Gate (NOT Gate):

This gate is equivalent to applying both X and Z gates and a global phase. It performs a bit and phase flip.

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

Pauli-Y



Ex: $Y|0\rangle = i|1\rangle, Y|1\rangle = -i|0\rangle$

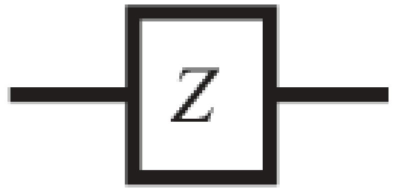
QUANTUM GATES

Pauli-Z Gate:

This gate flips the phase of the |1⟩ state, leaving the |0⟩ state unchanged.

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

Pauli-Z (phase flip)



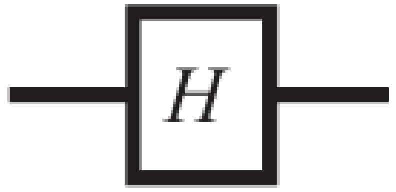
Ex: $Z|0\rangle = |0\rangle, Z|1\rangle = -|1\rangle$

Hadamard gate:

This gate creates a superposition state by transforming the |0⟩ state into an equal superposition of the |0⟩ and |1⟩ states.

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

Hadamard



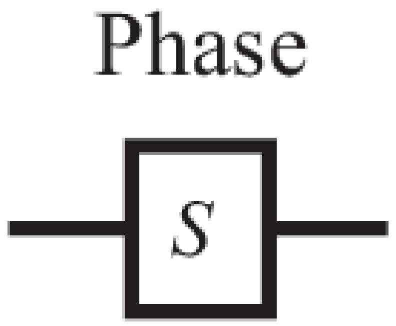
Ex: $H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$

QUANTUM GATES

Phase Gate(S Gate):

This gate is a 90-degree phase shift gate that introduces a phase shift of π/2 radians to the |1⟩ state.

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

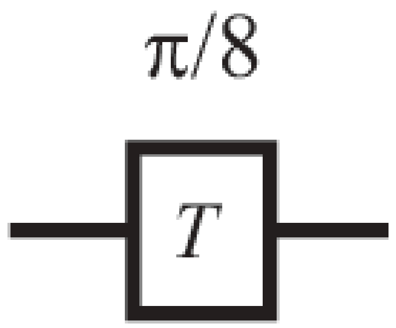


Ex: $S|1\rangle = i|1\rangle$

T Gate(π/8 Gate):

This gate is a 45-degree phase shift gate that introduces a phase shift of π/4 radians to the |1⟩ state.

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

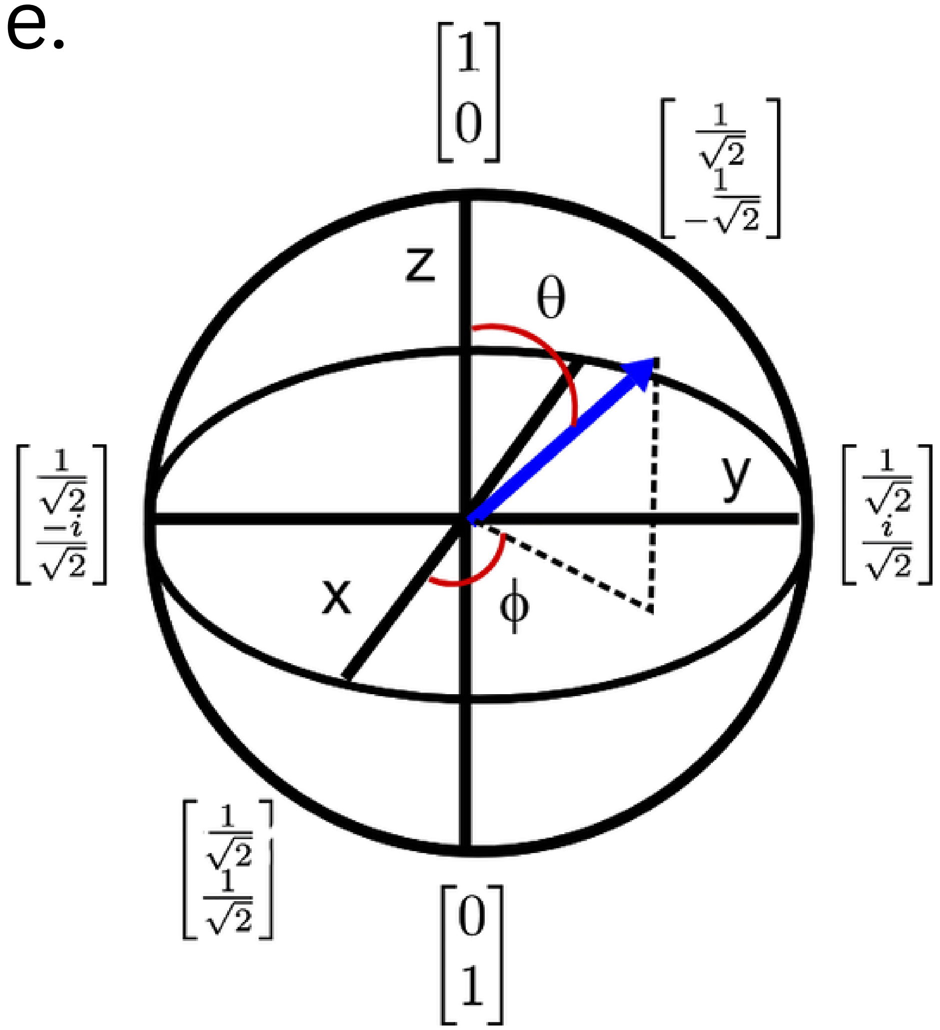


Ex: $T|1\rangle = e^{i\pi/4}|1\rangle$

QUANTUM GATES

Bloch sphere:

The Bloch sphere is a 3D geometric representation of a single qubit's quantum state, where the state is shown as a point on the surface of a unit sphere. The position is defined by angles θ (theta) and ϕ (phi). Quantum gates are unitary operations that transform the qubit's state by rotating or reflecting it on the Bloch sphere, thereby changing its superposition and phase.



X gate: 180° around X-axis → flips |0⟩ to |1⟩.
Y gate: 180° around Y-axis → flips state with phase change.
Z gate: 180° around Z-axis → flips the phase of |1⟩.
Hadamard (H): Rotates |0⟩ to the equator → creates equal superposition.
S gate: 90° around Z-axis → adds phase iii to |1⟩.
T gate: 45° around Z-axis → adds phase eiπ/4e^{i\pi/4}eiπ/4 to |1⟩.

QUANTUM GATES

Multi-Qubit Gates:

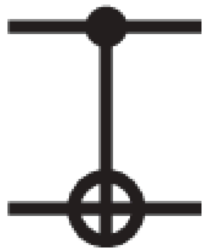
Multi-qubit gates are a type of quantum gate that act on two or more qubits simultaneously. These gates enable the creation and manipulation of entangled states, which are essential for performing complex quantum computations.

CNOT (Controlled-NOT) gate:

A two-qubit gate that performs a NOT operation on the target qubit if - and only if - the control qubit is in the state $|1\rangle$. The matrix in the basis $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$

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

CNOT



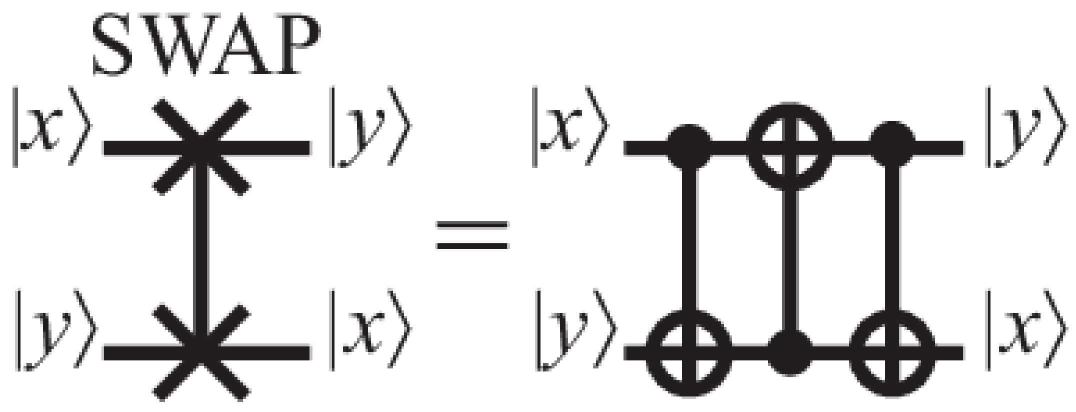
Ex: **CNOT**($|10\rangle$) = $|11\rangle$
(Control = 1 → flip target)

QUANTUM GATES

SWAP gate:

The SWAP gate is a two-qubit quantum gate that exchanges the states of two qubits. If the input qubits are in states |a⟩ and |b⟩, the SWAP gate outputs them as |b⟩ and |a⟩.

$$\mathrm{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Ex: SWAP(|01⟩) = |10⟩

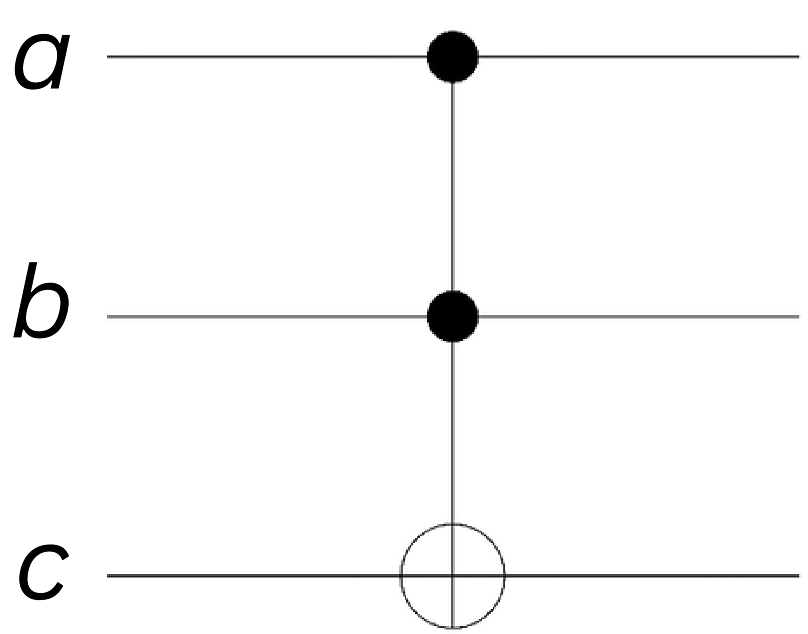
Toffoli gate:

A three-qubit gate that performs a NOT operation on the target qubit only if both of the control qubits are in state |1⟩. It is also known as the CCNOT gate (Controlled-Controlled-NOT) and is a key building block for many quantum algorithms.

QUANTUM GATES

It is an 8×8 matrix in basis from $|000\rangle$ to $|111\rangle$.

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



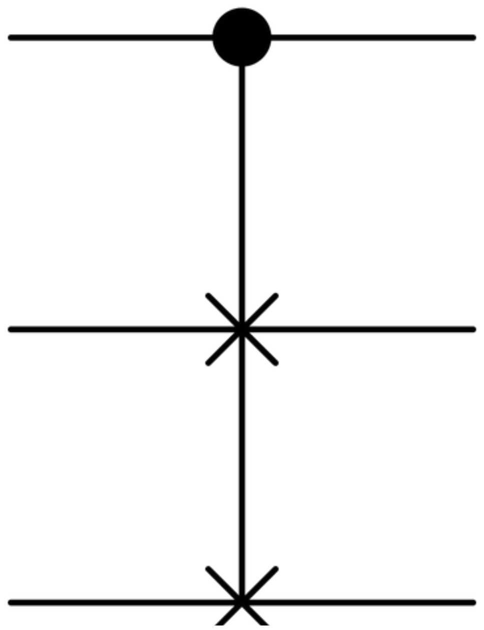
***Ex:* Toffoli**($|110\rangle$) = $|111\rangle$
(Control 1 = 1, Control 2 = 1 → flip target)

Fredkin gate:

A three-qubit gate that swaps the states of two target qubits only if the control qubit is in the state $1\rangle$. It is also known as the CSWAP gate (Controlled-SWAP).
it is also an 8×8 matrix (basis from $|000\rangle$ to $|111\rangle$).

QUANTUM GATES

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



Ex: Fredkin($|101\rangle$) = $|110\rangle$

Uses & Importance of Quantum Gates:

- Single-qubit gates create superposition and control the phase of individual qubits.
- They allow precise manipulation of qubit states on the Bloch sphere.
- Multi-qubit gates enable entanglement between qubits, essential for quantum communication.
- Gates like CNOT, Toffoli, and Fredkin perform conditional operations based on qubit states.
- Single and multi-qubit gates together form universal gate sets for any quantum computation.
- They are used in key quantum algorithms like Shor’s and Grover’s.
- They play a vital role in quantum error correction and maintaining qubit reliability.

BUILDING QUANTUM CIRCUITS

Introduction to Quantum circuits

What is a Quantum Circuit?

A quantum circuit is a graphical way to represent a sequence of quantum gates and measurements used to perform a quantum computation. In a quantum circuit, single-qubit gates are typically represented as boxes with a text label indicating the type of gate.

A quantum circuit is a model for quantum computation, consisting of:

Qubits: The quantum equivalent of classical bits, initialized in a specific state (e.g., $|0\rangle$ or a superposition).

Quantum Gates: Operations that manipulate the state of qubits (e.g., single-qubit gates like X, H, Z or multi-qubit gates like CNOT).

Measurements: Operations that extract classical information from qubits, collapsing their quantum state.

BUILDING QUANTUM CIRCUITS

Key Components of Quantum Circuits

1. **Initialization:**

   Qubits are typically initialized in the $|0\rangle$ state.
   Superposition states can be created using gates like the Hadamard (H) gate to prepare qubits for computation.

2. **Quantum Gates:**

   Single-Qubit Gates:

   - **X Gate:** Flips a qubit's state ($|0\rangle \leftrightarrow |1\rangle$).
   - **H Gate:** Creates superposition, mapping $|0\rangle$ to $(|0\rangle + |1\rangle)/\sqrt{2}$ and $|1\rangle$ to $(|0\rangle - |1\rangle)/\sqrt{2}$.
   - **Z Gate:** Applies a phase flip, leaving $|0\rangle$ unchanged and mapping $|1\rangle$ to $-|1\rangle$.
   - **Y Gate:** Combines X and Z effects, applying a phase and state flip.
   - **S Gate:** Applies a $\pi/2$ phase shift to $|1\rangle$.
   - **T Gate:** Applies a $\pi/4$ phase shift to $|1\rangle$.

BUILDING QUANTUM CIRCUITS

Multi-Qubit Gates:

- **CNOT Gate:** Flips the target qubit if the control qubit is |1⟩.
- **SWAP Gate:** Exchanges the states of two qubits.
- **Toffoli Gate:** A controlled-controlled-NOT gate with two control qubits.
- **Fredkin Gate:** A controlled-SWAP gate, swapping two target qubits if the control is |1⟩.

Gates are applied sequentially to transform the quantum state.

3. Measurements:

- Measurement collapses a qubit's state to either |0⟩ or |1⟩, with probabilities determined by the quantum state.
- In a circuit, measurements are typically performed at the end to extract classical results.
- Represented in diagrams as a meter symbol connected to a classical bit.

BUILDING QUANTUM CIRCUITS

Steps to Build a Quantum Circuit

1. **Define the Problem:**
    a. Identify the quantum algorithm or computation (e.g., Deutsch-Jozsa, Grover's search, quantum teleportation).
    b. Determine the number of qubits and classical bits required.
2. **Initialize Qubits:**
    a. Start with qubits in $|0\rangle$ or prepare specific initial states using gates like H for superposition.
3. **Apply Quantum Gates:**
    a. Choose a sequence of gates to implement the desired algorithm.
    b. Use single-qubit gates for local transformations and multi-qubit gates for entanglement or controlled operations.

Example: To create an entangled Bell state, apply an H gate to one qubit followed by a CNOT gate with the second qubit as the targe

BUILDING QUANTUM CIRCUITS

3. Apply Quantum Gates:

- Choose a sequence of gates to implement the desired algorithm.
- Use single-qubit gates for local transformations and multi-qubit gates for entanglement or controlled operations.
- Example: To create an entangled Bell state, apply an H gate to one qubit followed by a CNOT gate with the second qubit as the target.

4. Incorporate Measurements:

- Add measurement operations to extract the final result.
- Ensure measurements align with the algorithm's output requirements (e.g., measuring all qubits or specific ones).
- Run on a quantum computer if available, accounting for hardware constraints like connectivity or gate fidelity.

BUILDING QUANTUM CIRCUITS

5. Optimize the Circuit:

Minimize the number of gates to reduce errors (quantum computers are noisy).

Combine or cancel redundant gates (e.g., two consecutive X gates cancel out).

Use gate identities to simplify circuits (e.g., HZH = X).

6. Simulate or Execute:

Test the circuit using a quantum simulator (e.g., Qiskit, Cirq) to verify correctness.

Tools for Building Quantum Circuits

- **Qiskit (IBM):** Python-based framework for designing, simulating, and running quantum circuits.
- **Cirq (Google):** Open-source library for quantum circuit design and simulation.
- **Q# (Microsoft):** Programming language for quantum algorithms.
- **Quirk:** Browser-based quantum circuit simulator for quick prototyping.

THANK YOU