Week-2 meet

Soumik & Divyansh

IIT BOMBAY, June 5, 2025

Outline

Quantum Circuits – Let's Build Quantum Machines!

What's This session About?

This is where we level up. We're building the quantum equivalent of classical circuits — with qubits, not bits. Think logic gates, but with superposition and entanglement.

Quantum Circuits – Let's Build Quantum Machines!

What's This session About?

This is where we level up. We're building the quantum equivalent of classical circuits — with qubits, not bits. Think logic gates, but with superposition and entanglement.

- **Quantum Gates** – The basic ops like Hadamard, Pauli, and CNOT.
- **Controlled Operations** – Gates that act depending on other qubits.
- **Measurement** – Observing qubits without breaking your algorithm (too much).
- **Universality** – Building ANY quantum operation from a small set of gates.
- **Simulation & Complexity** – Why quantum circuits aren't just cool — they're powerful.

Outline

Quantum Algorithms: Overview

What's a Quantum Algorithm?

- Starts with $|0\rangle^{\otimes n}$
- Applies a sequence of unitary gates: $|\psi_{\text{final}}\rangle = U_d \cdots U_2 U_1 |0\rangle^{\otimes n}$
- Ends with a measurement in the computational basis

Why Do We Care?

- Some quantum algorithms outperform the best classical ones.
- Only two types known to offer super-polynomial speedups:
  1. Quantum Fourier Transform (e.g., Shor's algorithm)
  2. Quantum Search (e.g., Grover's algorithm)

Two Main Quantum Algorithm Families

Quantum Fourier Transform (QFT)

- Used in factoring, period-finding
- Core to Shor's algorithm
- Time complexity: $O(n^2)$

QFT-Based $\downarrow$ Shor

Grover's Algorithm

- Search in unstructured databases
- Finds marked item in $O(\sqrt{N})$
- Quadratic speedup

Grover-Based $\downarrow$ Unstructured Search

Outline

Single Qubit Operations – Let's Go Quantum Solo

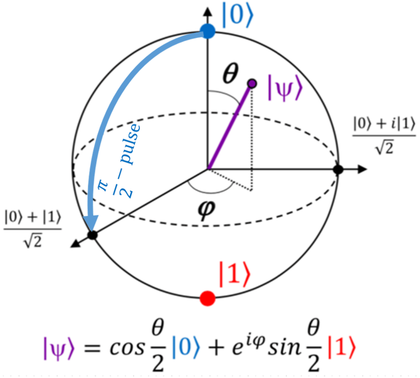
The Basics

- A qubit state is: $|\psi\rangle = a|0\rangle + b|1\rangle$ with $|a|^2 + |b|^2 = 1$
- Single-qubit operations are 2×2 unitary matrices
- They rotate the state on the Bloch sphere

Single Qubit Operations – Let's Go Quantum Solo

The Basics

- A qubit state is: $|\psi\rangle = a|0\rangle + b|1\rangle$ with $|a|^2 + |b|^2 = 1$
- Single-qubit operations are 2×2 unitary matrices
- They rotate the state on the Bloch sphere



Pauli Gates – Quantum Logic Basics

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

- X: bit-flip (quantum NOT)
- Z: phase-flip
- Y: combo of both
- All three generate 180° rotations around Bloch sphere axes

H, S, and T – Special Single-Qubit Gates

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}, \quad S = \begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix}, \quad T = \begin{bmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{bmatrix}$$

- H: puts $|0\rangle$ into superposition
- S: 90° phase shift, T: 45° phase shift ($\pi/8$ gate)
- Useful relations: $S = T^2$, $H = \frac{X+Z}{\sqrt{2}}$

Bloch Sphere + Rotations

- Any qubit state: $|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\phi}\sin(\theta/2)|1\rangle$
- Corresponds to point on unit sphere (Bloch vector):

$$(\cos\phi\sin\theta,\ \sin\phi\sin\theta,\ \cos\theta)$$

- Rotation gates:

$$R_x(\theta) = e^{-i\theta X/2}, \quad R_y(\theta) = e^{-i\theta Y/2}, \quad R_z(\theta) = e^{-i\theta Z/2}$$

Rotation Gates – Matrix Style

$$R_x(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

Rotation Gates – Matrix Style

$$R_x(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

Bonus Fact: $T = R_z(\pi/4)$ (up to a global phase)

Every Gate is Just Fancy Rotation

Euler Decomposition

Any 1-qubit unitary U can be written as:

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta)$$

Every Gate is Just Fancy Rotation

Euler Decomposition

Any 1-qubit unitary U can be written as:

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta)$$

Corollary: There exist A, B, C such that:

$$U = e^{i\alpha} AXBXC, \quad ABC = I$$

This is big when you want to build gates using just rotations + Pauli-X!

Exercises

1. Show that $HXH = Z$, $HZH = X$, $HYH = -Y$
2. Prove: $\exp(iAx) = \cos(x)I + i\sin(x)A$ when $A^2 = I$
   Use this to verify the rotation matrices.
3. Express the Hadamard gate in the form:

$$U = e^{i\alpha}AXBXC$$

Outline

Controlled Operations – If A, Then B

What Are They?

- Controlled operations are the quantum version of conditional logic.
- Most famous: Controlled-NOT (CNOT) or CX gate.
- Acts on two qubits:

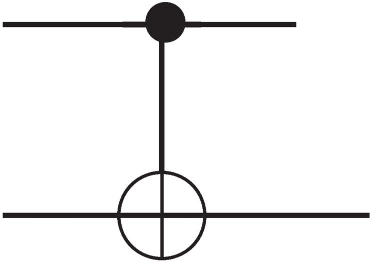
$$|c\rangle|t\rangle \mapsto |c\rangle|t \oplus c\rangle$$

Controlled Operations – If A, Then B

What Are They?

- Controlled operations are the quantum version of conditional logic.
- Most famous: Controlled-NOT (CNOT) or CX gate.
- Acts on two qubits:

$$|c\rangle|t\rangle \mapsto |c\rangle|t \oplus c\rangle$$



CNOT Gate – Matrix View

Matrix in Computational Basis

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

- Target is flipped only if control is $|1\rangle$

CNOT Gate – Matrix View

Matrix in Computational Basis

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

- Target is flipped only if control is $|1\rangle$

Fun Fact: CNOT is its own inverse: $\text{CNOT}^2 = I$

Controlled-U Gate – Generalizing CNOT

$$|c\rangle|\psi\rangle \mapsto |c\rangle U^c|\psi\rangle$$

- If $c = 0$: nothing happens.
- If $c = 1$: apply U to target.

Controlled-U Gate – Generalizing CNOT

$$|c\rangle|\psi\rangle \mapsto |c\rangle U^c|\psi\rangle$$

- If $c = 0$: nothing happens.
- If $c = 1$: apply U to target.

Matrix:

$$\mathrm{CU} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{bmatrix}$$

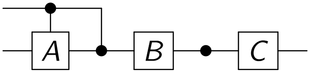
Controlled-U Using 1-Qubit Gates

Important

Any U can be written as:

$$U = e^{i\alpha} AXBXC \quad \text{with } ABC = I$$

So controlled-U can be built like:



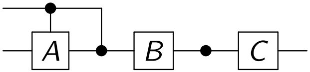
Controlled-U Using 1-Qubit Gates

Important

Any U can be written as:

$$U = e^{i\alpha} AXBXC \quad \text{with } ABC = I$$

So controlled-U can be built like:



This decomposition works because $XBX = B^{-1}$ (when $X^2 = I$)

Toffoli and $C^n(U)$– Many Controls, One Target

- Toffoli $= C^2(X) =$ controlled-controlled-NOT
- General $C^n(U)$: apply U if all n control qubits $= 1$
- Formally:

$$C^n(U)|x_1 \cdots x_n\rangle|\psi\rangle = |x_1 \cdots x_n\rangle U^{x_1 x_2 \cdots x_n}|\psi\rangle$$

Toffoli and $C^n(U)$– Many Controls, One Target

- Toffoli $= C^2(X) =$ controlled-controlled-NOT
- General $C^n(U)$: apply U if all n control qubits $= 1$
- Formally:
$$C^n(U)|x_1 \cdots x_n\rangle|\psi\rangle = |x_1 \cdots x_n\rangle U^{x_1 x_2 \cdots x_n}|\psi\rangle$$

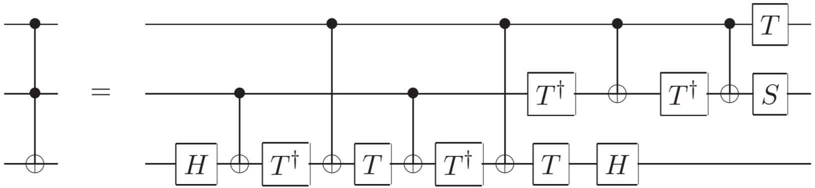
Matrix size: $2^{n+1} \times 2^{n+1}$
Used for: Universal logic, error correction, Grover's diffusion operator

Toffoli with Basic Gates

- You can build Toffoli using only:
  - Hadamard H
  - Phase S, T
  - CNOT

Toffoli with Basic Gates

- You can build Toffoli using only:
  - Hadamard H
  - Phase S, T
  - CNOT



Try It Yourself!

1. Find the 4×4 unitary matrix for a CNOT gate.
2. Use Hadamard + Controlled-Z to construct a CNOT gate.
3. Construct a Controlled-$R_x(\theta)$ gate using CNOT + 1-qubit gates.

Try It Yourself!

1. Find the 4×4 unitary matrix for a CNOT gate.
2. Use Hadamard + Controlled-Z to construct a CNOT gate.
3. Construct a Controlled-$R_x(\theta)$ gate using CNOT + 1-qubit gates.

Bonus: Why does CNOT $= HZH$ (up to basis change)?

Outline

Measurement – Where Qubits Spill Their Secrets

- Measurement converts a quantum state into a classical outcome.
- Usually in the **computational basis**: $|0\rangle, |1\rangle$
- Denoted by:

$$\text{Measure } |\psi\rangle = \sum_i a_i |i\rangle \quad \rightarrow \quad i \text{ with probability } |a_i|^2$$

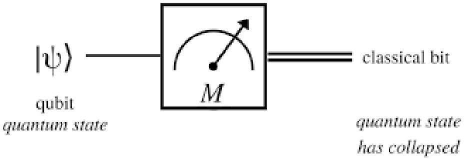
- After measurement: system collapses to $|i\rangle$

Measurement – Where Qubits Spill Their Secrets

- Measurement converts a quantum state into a classical outcome.
- Usually in the **computational basis**: $|0\rangle, |1\rangle$
- Denoted by:

$$\text{Measure } |\psi\rangle = \sum_i a_i |i\rangle \quad \rightarrow \quad i \text{ with probability } |a_i|^2$$

- After measurement: system collapses to $|i\rangle$



(Symbol: meter or box with arrow to classical bit)

Mathematical Formalism of Measurement

Measurement Operators

$$\{M_m\} \quad \text{such that} \quad \sum_m M_m^\dagger M_m = I$$

Mathematical Formalism of Measurement

Measurement Operators

$$\{M_m\} \quad \text{such that} \quad \sum_m M_m^\dagger M_m = I$$

If the system is in state $|\psi\rangle$, outcome m occurs with probability:

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle$$

Post-measurement state becomes:

$$\frac{M_m|\psi\rangle}{\sqrt{p(m)}}$$

Projective Measurements – The Common Case

- For computational basis:

$$M_0 = |0\rangle\langle 0|, \quad M_1 = |1\rangle\langle 1|$$

- Then for state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$:

$$p(0) = |\alpha|^2, \quad p(1) = |\beta|^2$$

- After measurement, state is either $|0\rangle$ or $|1\rangle$

When to Measure?

- At the end: to extract the answer.
- During: to control future operations based on measured values.
- Can be used for:
  - Probabilistic post-selection
  - Adaptive measurements (e.g., quantum error correction)
  - Teleportation and intermediate collapses

When to Measure?

- At the end: to extract the answer.
- During: to control future operations based on measured values.
- Can be used for:
  - Probabilistic post-selection
  - Adaptive measurements (e.g., quantum error correction)
  - Teleportation and intermediate collapses

Quantum Circuit Model allows:

- Unitary gates
- Measurement gates (classical output)
- Classical control of future gates

Why Not Just Use Unitaries?

- Measurement is **irreversible** and probabilistic.
- Unitaires are **reversible** and deterministic.
- You need measurement to:
  - Collapse entanglement
  - Output classical info
  - Apply classical feedback

Why Not Just Use Unitaries?

- Measurement is **irreversible** and probabilistic.
- Unitaires are **reversible** and deterministic.
- You need measurement to:
  - Collapse entanglement
  - Output classical info
  - Apply classical feedback

Takeaway: You can simulate measurement with unitaries *plus ancilla* + *tracing*, but it's messy.

Exercises – Test Your Quantum Reflexes

1. Suppose you measure qubit 1 of the state:

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

   What's the post-measurement state if the result is 0? What if it's 1?

2. Show that measurement in the computational basis is equivalent to applying:

$$\sum_i |i\rangle\langle i|$$

   as the projectors.

3. Construct a measurement operator that distinguishes between $|+\rangle$ and $|-\rangle$.

Outline

- Motivation
- Overview
- Single Qubit Operations
- Controlled Operations
- Measurements
- Universality
- Algorithm

Universal Quantum Gates – What's the Big Deal?

Definition

A set of quantum gates is **universal** if:

Any unitary U can be approximated to arbitrary accuracy $\varepsilon > 0$

using only gates from that set.

Universal Quantum Gates – What's the Big Deal?

Definition

A set of quantum gates is **universal** if:

Any unitary U can be approximated to arbitrary accuracy $\varepsilon > 0$

using only gates from that set.

- Classical analogy: NAND is universal for Boolean logic.
- Quantum version? We want a minimal set that does *everything*.
- Example universal set:
$$\{H, S, T, \mathrm{CNOT}\}$$

Step 1: Two-Level Unitary Matrices

- Any $d \times d$ unitary matrix U can be decomposed into simpler unitaries acting on just 2 dimensions at a time.
- These are called **two-level unitaries**.

Step 1: Two-Level Unitary Matrices

- Any $d \times d$ unitary matrix U can be decomposed into simpler unitaries acting on just 2 dimensions at a time.
- These are called **two-level unitaries**.

Example: 3×3 Case

$$U = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & j \end{bmatrix} \Rightarrow U = U_1^\dagger U_2^\dagger U_3^\dagger$$

where each U_i is a two-level unitary.

Key Idea: Reduce one row/column at a time like Gaussian elimination!

Step 2: 1-Qubit + CNOTs = Enough

- Any two-level unitary can be implemented using:

$$\text{1-qubit gates} \quad + \quad \text{CNOTs}$$

- This means: $\{\text{CNOT}, \text{all 1-qubit unitaries}\}$ is universal.

Step 2: 1-Qubit + CNOTs = Enough

- Any two-level unitary can be implemented using:

$$\text{1-qubit gates} \quad + \quad \text{CNOTs}$$

- This means: $\{\text{CNOT}, \text{all 1-qubit unitaries}\}$ is universal.

So now the challenge becomes:

How do we approximate ANY 1-qubit unitary using a small set?

Step 3: Discrete Universal Gate Set

- We want to go from: All 1-qubit unitaries $\rightarrow$ small fixed gate set
- Turns out, the following is enough:

$$\{H, S = R_z(\pi/2), T = R_z(\pi/4), \text{CNOT}\}$$

- With these, you can approximate any $U \in U(2^n)$ to arbitrary accuracy

Step 3: Discrete Universal Gate Set

- We want to go from: All 1-qubit unitaries $\rightarrow$ small fixed gate set
- Turns out, the following is enough:

$$\{H, S = R_z(\pi/2), T = R_z(\pi/4), \text{CNOT}\}$$

- With these, you can approximate any $U \in U(2^n)$ to arbitrary accuracy

Why Is This Wild?

Even though this set is finite, it's dense in unitary space. You just need long enough circuits!

The Solovay–Kitaev Theorem – Efficient Approximations

Theorem

Any 1-qubit unitary U can be approximated to accuracy ε using:

$$O(\log^c(1/\varepsilon)) \text{ gates from a finite universal set}$$

where $c \approx 2$

The Solovay–Kitaev Theorem – Efficient Approximations

Theorem

Any 1-qubit unitary U can be approximated to accuracy ε using:

$$O(\log^c(1/\varepsilon)) \text{ gates from a finite universal set}$$

where $c \approx 2$

- Before SK: naïve methods took $\sim 1/\varepsilon$ gates.
- Now: polylog overhead – a quantum compiler's dream.
- Practical implication: your gate set doesn't need to be big, just smart.

Exercise Time – Build Your Brain!

1. Build a circuit to add two 2-bit numbers mod 4:

$$|x, y\rangle \mapsto |x, x + y \mod 4\rangle$$

2. Prove: $\cos(\theta) = \frac{3}{5} \Rightarrow \theta$ is irrational multiple of 2π
3. Show that any circuit using only H, S, T gates results in a matrix with entries of form $2^{-k/2} \cdot \mathbb{Z}[i]$

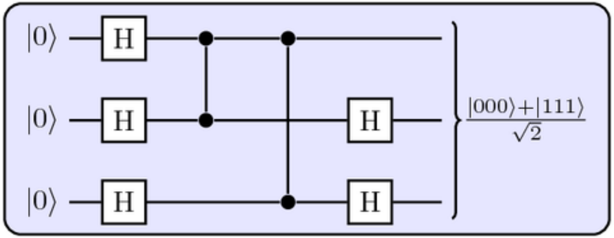
Outline

Representing Quantum Circuits

- Quantum algorithms = sequences of unitary gates + measurements
- Circuits use:
  - Horizontal wires = qubits
  - Boxes = gates
  - Time flows left to right

Representing Quantum Circuits

- Quantum algorithms = sequences of unitary gates + measurements
- Circuits use:
  - Horizontal wires = qubits
  - Boxes = gates
  - Time flows left to right



(Illustrate: 3-qubit input to Hadamard chain)

Cost of a Quantum Circuit

Key metrics:

- **Gate count:** Total number of gates (like runtime)
- **Circuit depth:** Max number of sequential steps
- **Width:** Number of qubits used

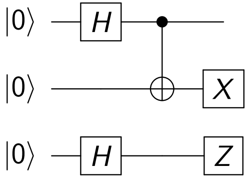
Cost of a Quantum Circuit

Key metrics:

- **Gate count:** Total number of gates (like runtime)
- **Circuit depth:** Max number of sequential steps
- **Width:** Number of qubits used

Example

A depth-3 circuit acting on 3 qubits:



Standard Gate Symbols

gate-symbols.png

Standard Gate Symbols

gate-symbols.png

Depth vs. Gate Count – Can We Parallelize?

- Quantum gates on different qubits can run in parallel.
- Two gates on the same qubit must be serialized.
- **Goal:** Minimize depth without increasing gate count too much.

Depth vs. Gate Count – Can We Parallelize?

- Quantum gates on different qubits can run in parallel.
- Two gates on the same qubit must be serialized.
- **Goal:** Minimize depth without increasing gate count too much.

Example: Parallel Hadamards

$$|0\rangle \;-\!\boxed{H}\!- \quad \Rightarrow \quad \text{Depth } = 1$$

$$|0\rangle \;-\!\boxed{H}\!-$$

$$|0\rangle \;-\!\boxed{H}\!-$$

Variants of the Circuit Model

- **Classical control:** Measure $\rightarrow$ control next gate (e.g., teleportation)
- **Mid-circuit measurement:** Useful in error correction, resets
- **Ancilla qubits:** Temporary workspace qubits (can be reset)
- **Measurement-based models:** Gates replaced by adaptive measurements (e.g., MBQC)

Variants of the Circuit Model

- **Classical control:** Measure $\rightarrow$ control next gate (e.g., teleportation)
- **Mid-circuit measurement:** Useful in error correction, resets
- **Ancilla qubits:** Temporary workspace qubits (can be reset)
- **Measurement-based models:** Gates replaced by adaptive measurements (e.g., MBQC)

Important: These variants are computationally equivalent!

Comparing Classical and Quantum Circuits

Classical:

- Bits (0 or 1)
- Deterministic gates (AND, OR, NOT)
- Irreversible

Quantum:

- Qubits
- Unitary gates (reversible)
- Superposition + entanglement

Try These!

1. Design a 3-qubit quantum circuit that applies Hadamard to all and flips the last qubit if the first two are $|1\rangle$.
2. Convert the following circuit into gate matrix form:

$$H \otimes I \cdot \text{CNOT}_{12} \cdot I \otimes X$$

3. What is the minimum depth of a circuit that applies a Hadamard on each of 5 qubits followed by a single Toffoli?

Circuit Equivalences

- Two circuits are equivalent if they produce the same final output state for any input.
- Why care?
  - Optimization
  - Implementation ease
  - Understanding structure

Circuit Equivalences

- Two circuits are equivalent if they produce the same final output state for any input.
- Why care?
  - Optimization
  - Implementation ease
  - Understanding structure

Example:

$$HXH = Z, \quad HZH = X, \quad HYH = -Y$$

Key Equivalences – Single Qubit Remix

Useful conjugation identities:

$$HXH = Z$$
$$HZH = X$$
$$HYH = -Y$$

Key Equivalences – Single Qubit Remix

Useful conjugation identities:

$$HXH = Z$$
$$HZH = X$$
$$HYH = -Y$$

These follow from:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{and } X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Transformations via Basis Change

Phase gate: $S = R_z(\pi/2), \ T = R_z(\pi/4)$

Transformations via Basis Change

Phase gate: $S = R_z(\pi/2)$, $T = R_z(\pi/4)$
Conjugate with Hadamard:

$$HR_z(\theta)H = R_x(\theta)$$

Example:

$$HTH = R_x(\pi/4)$$

This lets you rotate around X using Z-axis rotation + basis swap.

Rotations on the Bloch Sphere

- Composing two rotations = one new rotation
- Angle and axis found by:

$$c_{12} = c_1 c_2 - s_1 s_2 \hat{n}_1 \cdot \hat{n}_2$$

$$s_{12} \hat{n}_{12} = s_1 c_2 \hat{n}_1 + c_1 s_2 \hat{n}_2 - s_1 s_2 \hat{n}_2 \times \hat{n}_1$$

- $c_i = \cos(\theta_i/2), s_i = \sin(\theta_i/2)$

Rotations on the Bloch Sphere

- Composing two rotations = one new rotation
- Angle and axis found by:

$$c_{12} = c_1 c_2 - s_1 s_2 \hat{n}_1 \cdot \hat{n}_2$$

$$s_{12} \hat{n}_{12} = s_1 c_2 \hat{n}_1 + c_1 s_2 \hat{n}_2 - s_1 s_2 \hat{n}_2 \times \hat{n}_1$$

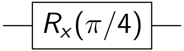
- $c_i = \cos(\theta_i/2), s_i = \sin(\theta_i/2)$

This is crucial when understanding approximate gate decompositions!

Circuit Identities in Practice

Original:

Equivalent:



Circuit Identities in Practice

Original:

$$-\boxed{H}-\boxed{T}-\boxed{H}-$$

Equivalent:

$$-\boxed{R_x(\pi/4)}-$$

Why this matters: Helps with gate scheduling and simplification.

Exercise Time – Flex These Equivalences

1. Prove: $HXH = Z$ and $HTH = R_x(\pi/4)$
2. Simplify this circuit:
$$H \to T \to H \to T^\dagger \to H$$
3. Using Bloch sphere rotation formulas, compute the net rotation of:
$$R_x(\pi/4)R_y(\pi/2)$$