सत्यमेव जयते

Department of Science & Technology
(Government of Gujarat)

IEEE, KSV
STUDENT BRANCH

IEEE Signal Processing Society
Kadi Sarva Vishwavidyalaya
Student Branch Chapter

GUJCOST sponsored and IEEE-SPS supported

OFFLINE SHORT TERM TRAINING PROGRAM ON

For Registration:
https://forms.gle/QYHLH4nhKDeC672G6

Quantum Computing:

Principles, Algorithms & Applications

29-12-2025 To 02-01-2026

This presentation is available at http://www.github.com/hbpatel1976/QuantumComputing

Dr. Hiren B. Patel
Principal, Vidush Somany Institute of Technology and Research
Kadi Sarva Vishwavidyalaya, Sarva Vidyalaya Kelavani Mandal
URL: www.hbpatel.in

Acknowledgement

- **Udemy**'s "*Quantum Computing and Introduction to Quantum Machine Learning*" October 2024.
- **NPTEL**-AICTE FDP on "*Introduction to Quantum Computing: Quantum Algorithms and Qiskit*" during July-August 2024.
- AICTE's **ATAL FDP** on "*Quantum Machine Learning*" at CHRIST UNIVERSITY Bangalore in Nov-2024.
- **YouTube**: "*Quantum Computing Course – Math and Theory for Beginners*" at freeCodeCamp.org (https://youtu.be/tsbCSkvHhMo)

Presentation Outline

- Introduction
- Essential Mathematics behind Quantum Computing
  - Complex Numbers, Linear Algebra, Matrices
- Quantum Bits (Qubits)
  - Single Qubit, Multiple Qubits, Operations on Qubits
- Quantum Phenomena
  - Entanglement, Phase kickback
- Quantum Algorithms

Introduction

- Applications of Quantum Computing
  - Artificial Intelligence, Machine Learning, Natural Language Processing (NLP)
  - Cybersecurity / Cryptography / Blockchain
  - Drug Discovery, Molecular Modelling and Chemical Research
  - Fundamental Science and Mathematics
  - Optimization (Logistic, Finance, Manufacturing, Supply Chain, Portfolio, Search, Flight path, Fuel Consumption, Radar signal processing, satellite positioning)
  - Modelling (Material, Finance, Climate, Weather)

Introduction

- Why Quantum Computing?
- Problem Hardness / Complexity [Time/Space]
  - N
  - N log (N)
  - N^2
  - N^5
  - 2^N
  - 7^N

Introduction: Classical Vs Quantum Computing

Classical Computing	Quantum Computing
Binary Digits (Bits)	Quantum Bits (Qubits)
States: 0, 1	$\|\psi> \ = \alpha\|0>+ \beta\|1>$, States: |0>, |1>, α, β: Complex Numbers, +: Superposition
State: Deterministic	State: Probabilistic Probability of |0>: $\|\alpha\|^2$, Probability of |0>: $\|\beta\|^2$
Logical Gates: NOT, AND, OR, XOR	Gates: Pauli X Gate (Matrix/Vector representation)
Algebra: Boolean	Algebra: Linear
Operation: Irreversible	Operation: Reversible
State Space Size: Linear (N)	State Space Size: Exponential (2^N)

Numbers: Real, Imaginary, Complex

Real Numbers

$X^2 = 4$

$\sqrt{X^2} = \sqrt{4}$

$X = \pm 2$

Imaginary Numbers

$X^2 = -4$

$\sqrt{X^2} = \sqrt{-4}$

$X = \pm\sqrt{-4}$

$X = \pm\sqrt{4}\sqrt{-1}$

$X = \pm 2\sqrt{-1}$

Let $i = \sqrt{-1}$

$X = \pm 2i$

Numbers: Real, Imaginary, Complex

Complex Numbers = Real Numbers + Imaginary Numbers

$a + i\,b$ where $a, b \in R$

E.g.

3 + 4 I

-5 –2 i

$\sqrt{3} + i\sqrt{5}$

Numbers: Real, Imaginary, Complex

Operations on Complex Numbers

Addition:

$(3+2i)+(2-4i)$

$=5-2i$

Multiplication:

$(3+2i)+(2-4i)$

$=5+4i-12i-12i^2$

$=5-8i-12(\sqrt{-1})^2$

$=5-8i+12$

$=17-8i$

Numbers: Real, Imaginary, Complex

Complex Conjugate: Negate the imaginary part

$(a+ib)$

$(a+ib)^*$

$=(a-ib)$

Example:

$(-2+3i)$

$(-2+3i)^*$

$=(-2-3i)$

Numbers: Real, Imaginary, Complex

Complex Conjugate: Multiplication of any complex number with its complex conjugate will always be a real number

Example:

Number: $(-2+3i)$

$(-2+3i) \text{ x } (-2+3i)^*$

$= (-2+3i) \text{ x } (-2-3i)$

$= 4 -6i + 6i - 9i^2$

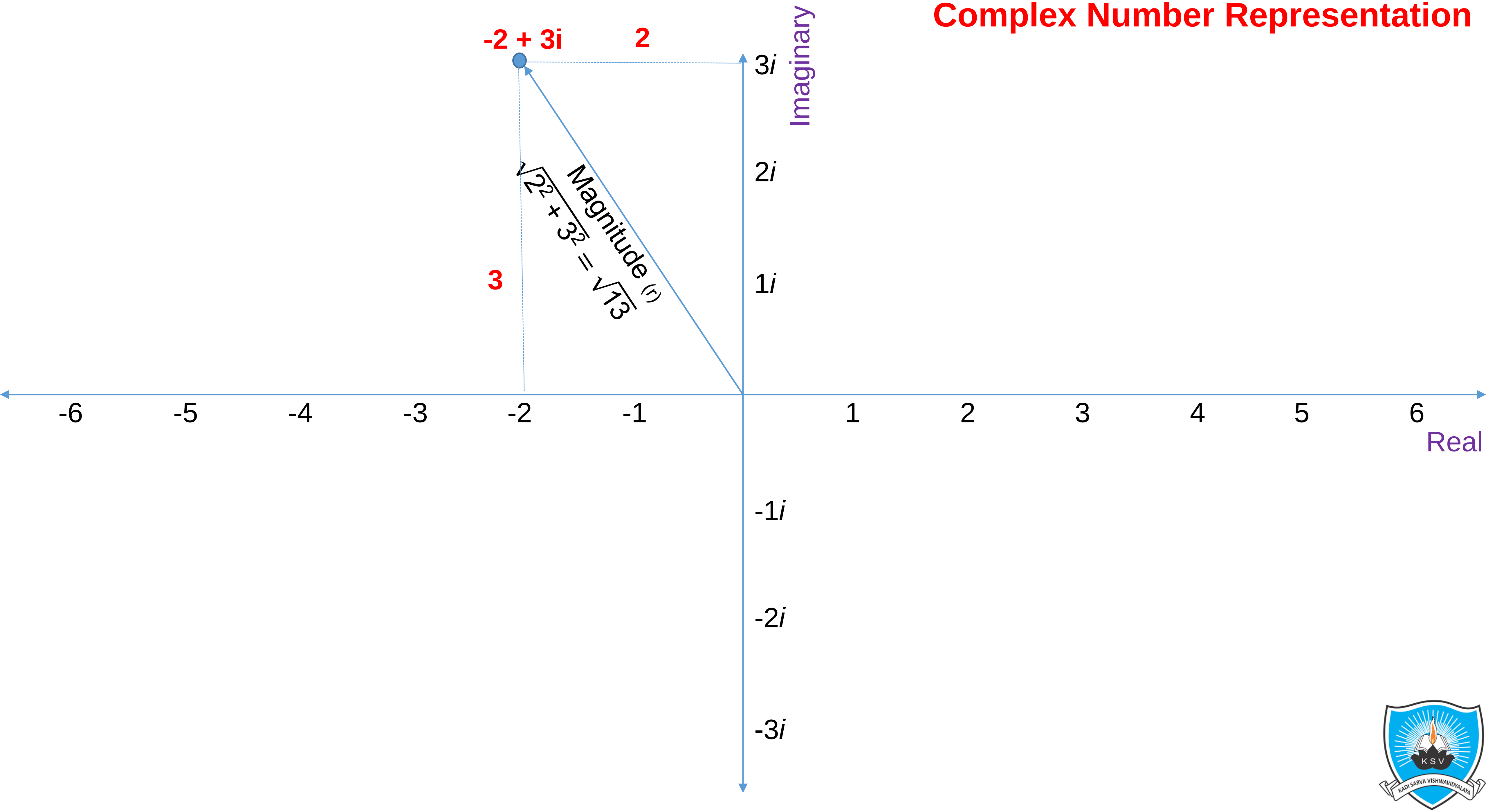
$= 4 – 9i^2$

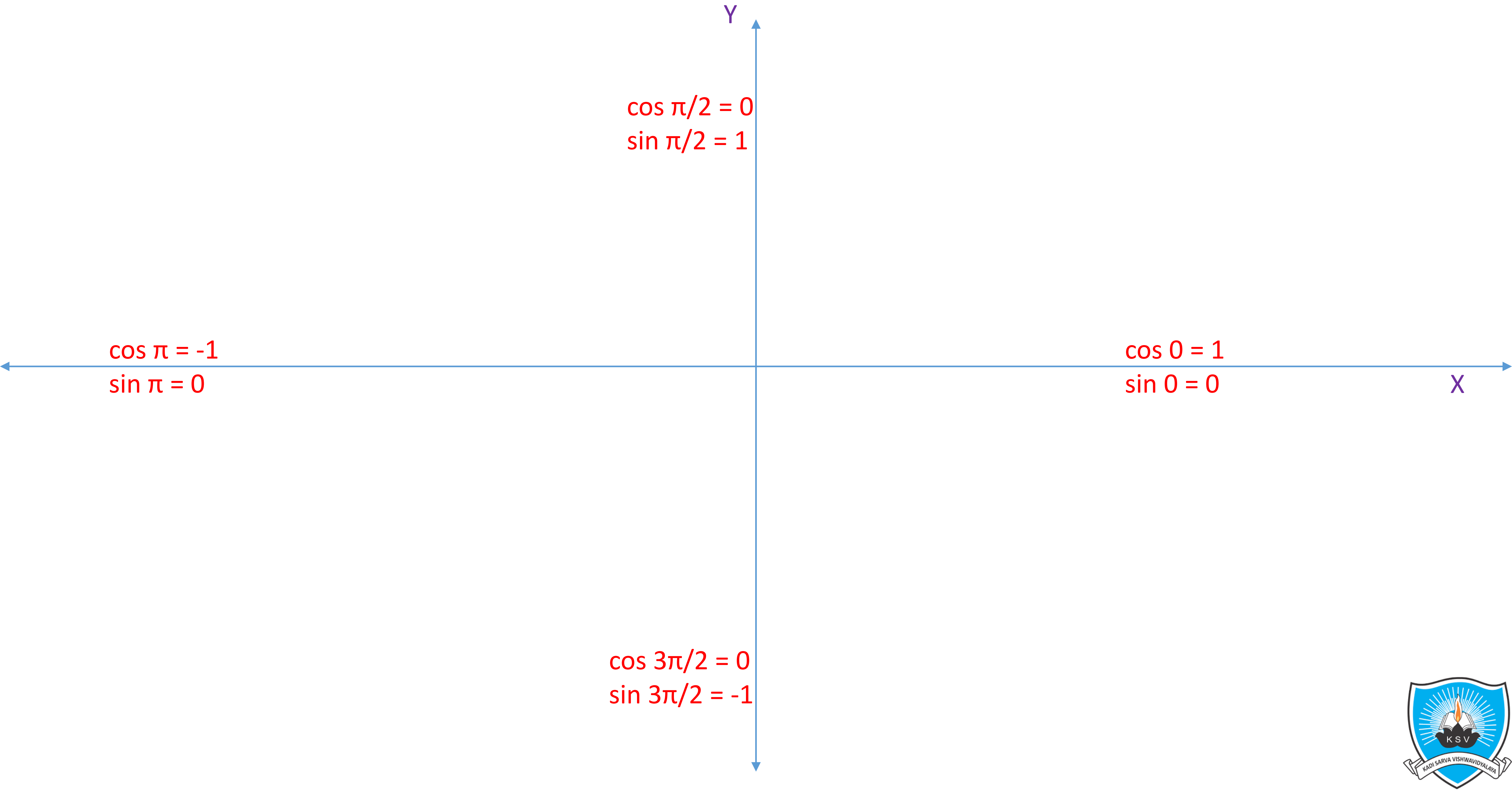
$= 4 + 9$

$= 13$

Complex Numbers on a Number Plane

Complex Number Representation





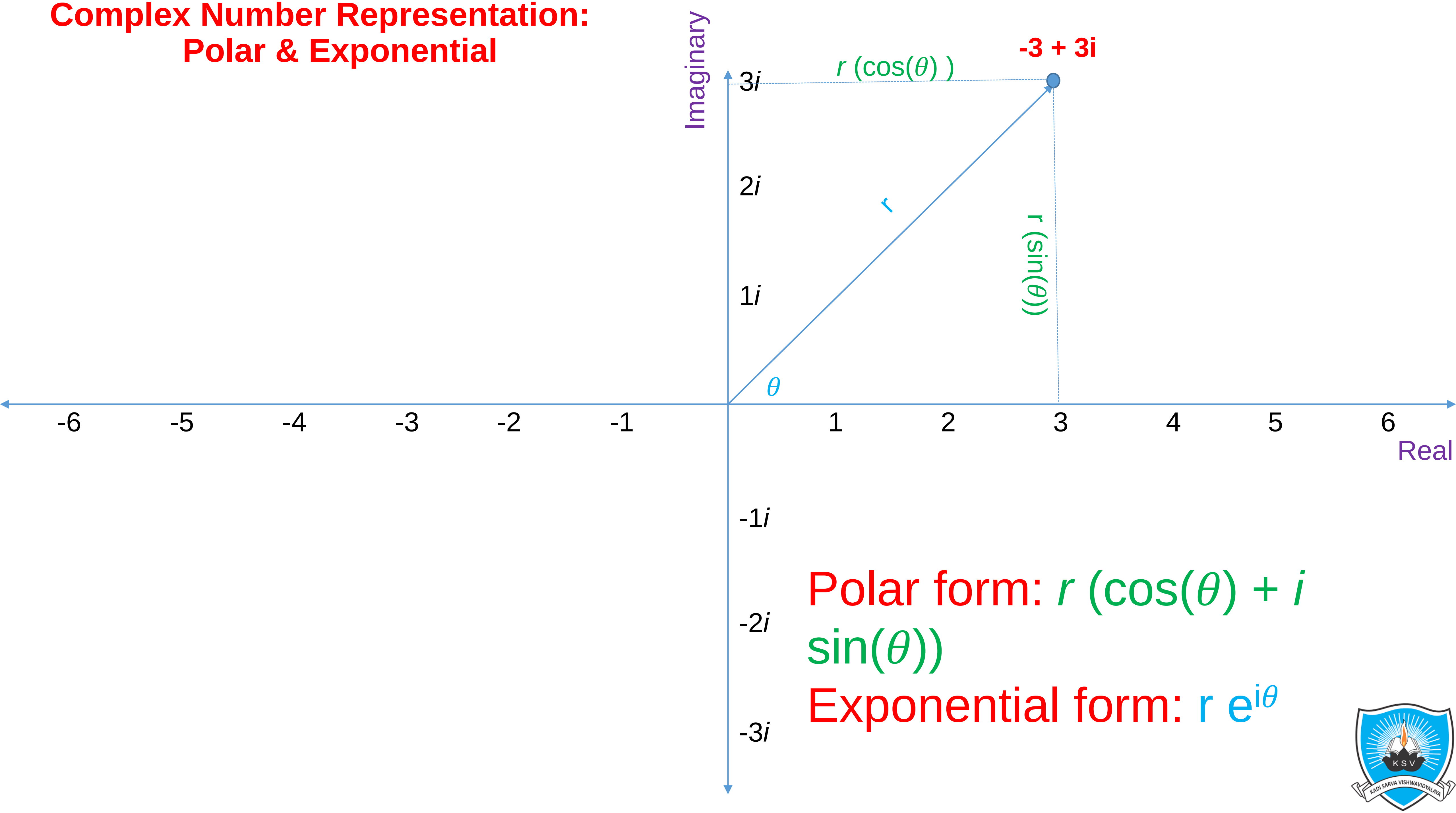
$\cos \pi/2 = 0$
$\sin \pi/2 = 1$

$\cos \pi = -1$
$\sin \pi = 0$

$\cos 0 = 1$
$\sin 0 = 0$

$\cos 3\pi/2 = 0$
$\sin 3\pi/2 = -1$

Complex Number Representation: Polar & Exponential



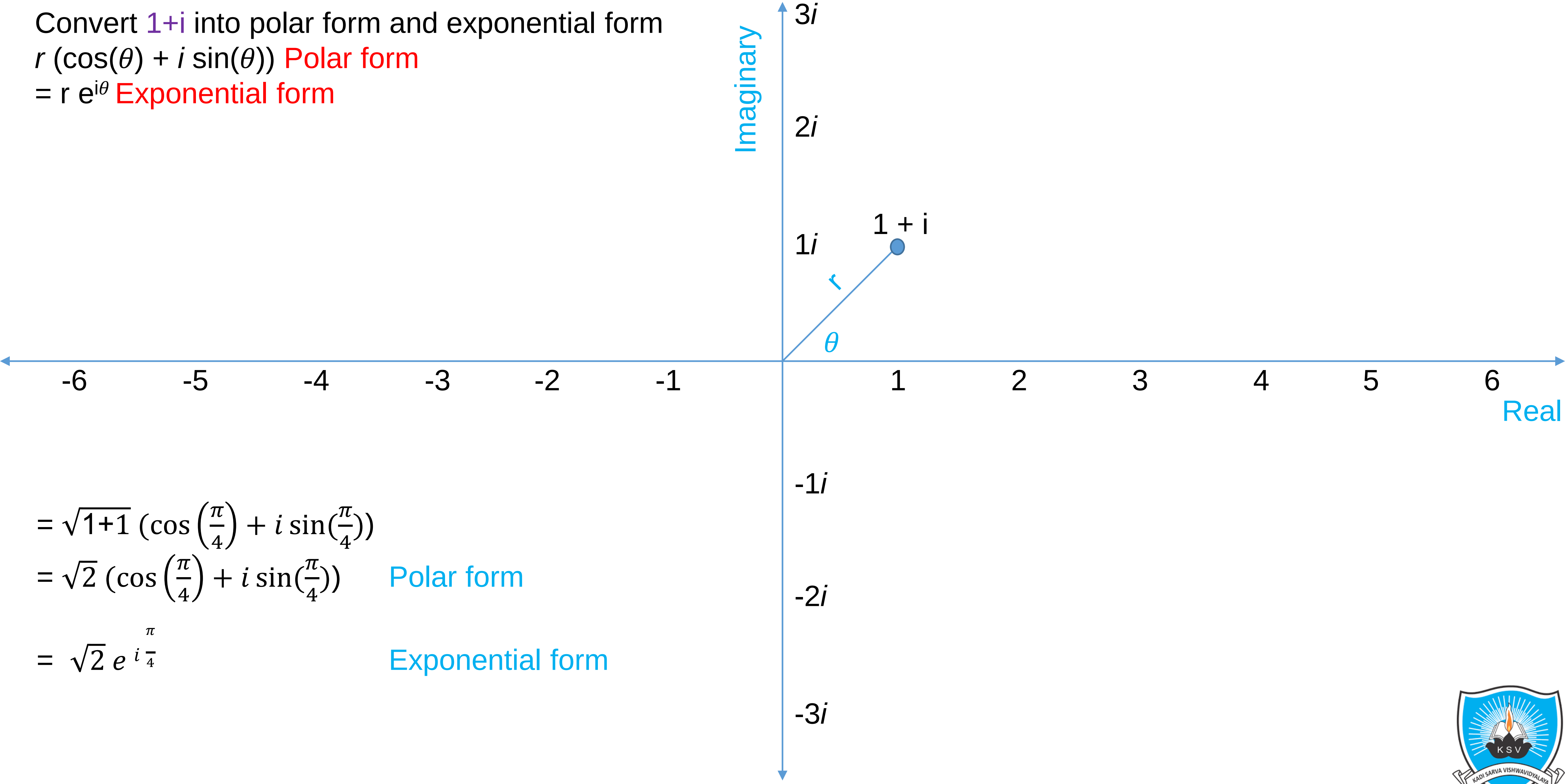
Polar form: $r\ (\cos(\theta) + i\ \sin(\theta))$

Exponential form: $r\ e^{i\theta}$

Convert 1+i into polar form and exponential form

$r\,(\cos(\theta) + i\,\sin(\theta))$ Polar form

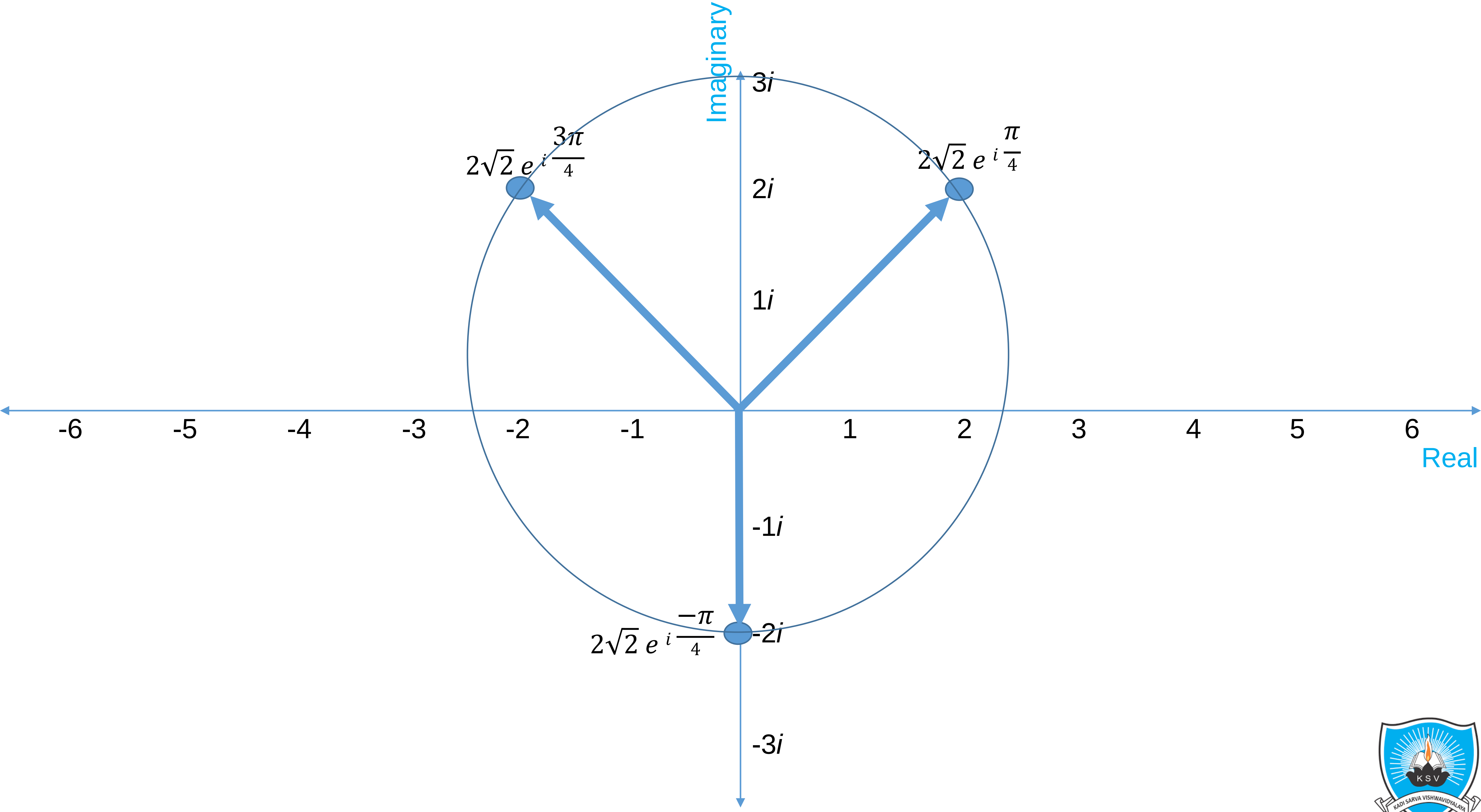
$= r\,e^{i\theta}$ Exponential form



$= \sqrt{1+1}\,(\cos\left(\frac{\pi}{4}\right) + i\,\sin(\frac{\pi}{4}))$

$= \sqrt{2}\,(\cos\left(\frac{\pi}{4}\right) + i\,\sin(\frac{\pi}{4}))$ Polar form

$= \sqrt{2}\,e^{i\frac{\pi}{4}}$ Exponential form

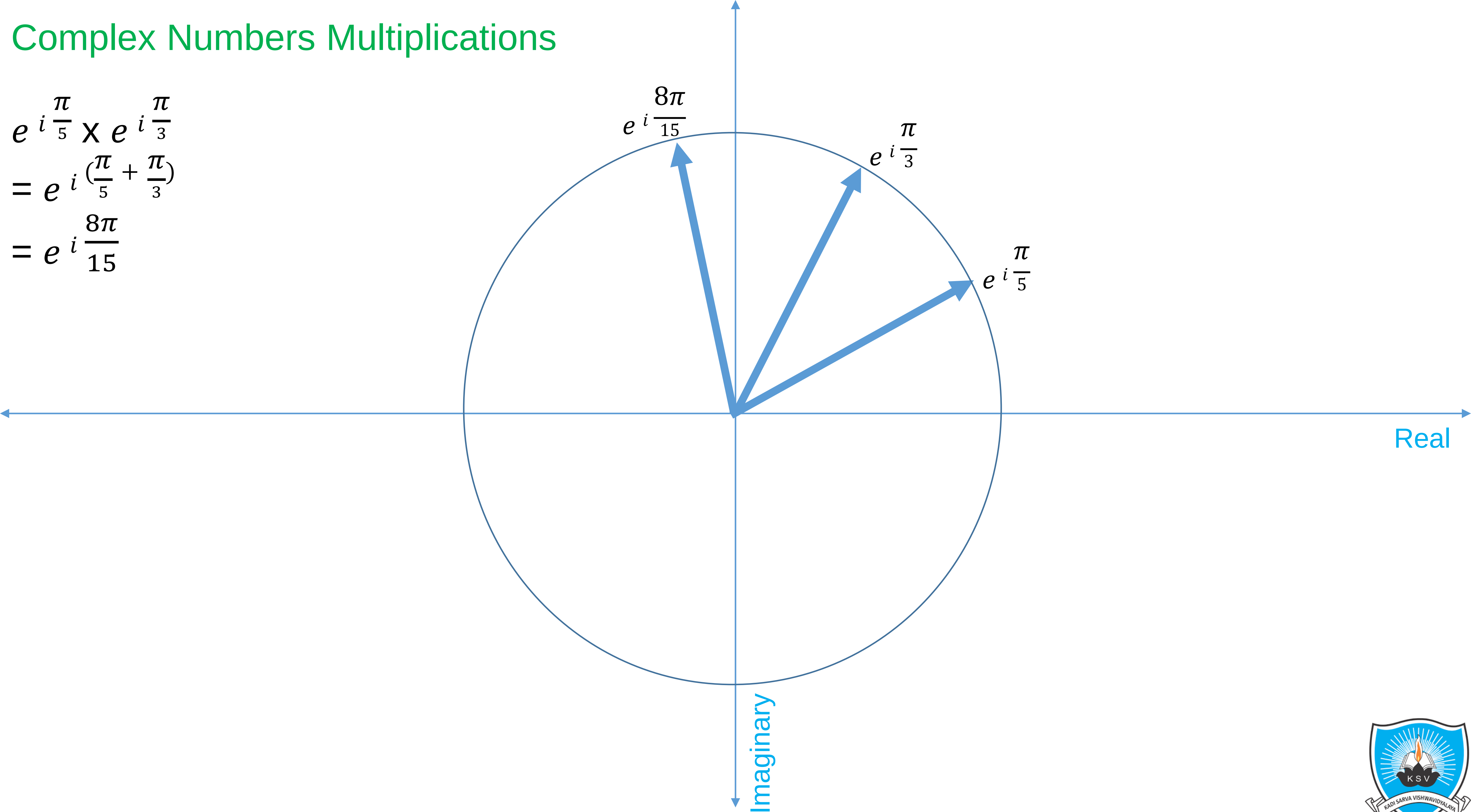

Imaginary
3i
2i
1i
-1i
-2i
-3i
-6
-5
-4
-3
-2
-1
1
2
3
4
5
6
Real
$2\sqrt{2}\ e^{\,i\frac{3\pi}{4}}$
$2\sqrt{2}\ e^{\,i\frac{\pi}{4}}$
$2\sqrt{2}\ e^{\,i\frac{-\pi}{4}}$

Complex Numbers Multiplications

$$e^{i\frac{\pi}{5}} \text{ x } e^{i\frac{\pi}{3}}$$

$$= e^{i\left(\frac{\pi}{5}+\frac{\pi}{3}\right)}$$

$$= e^{i\frac{8\pi}{15}}$$

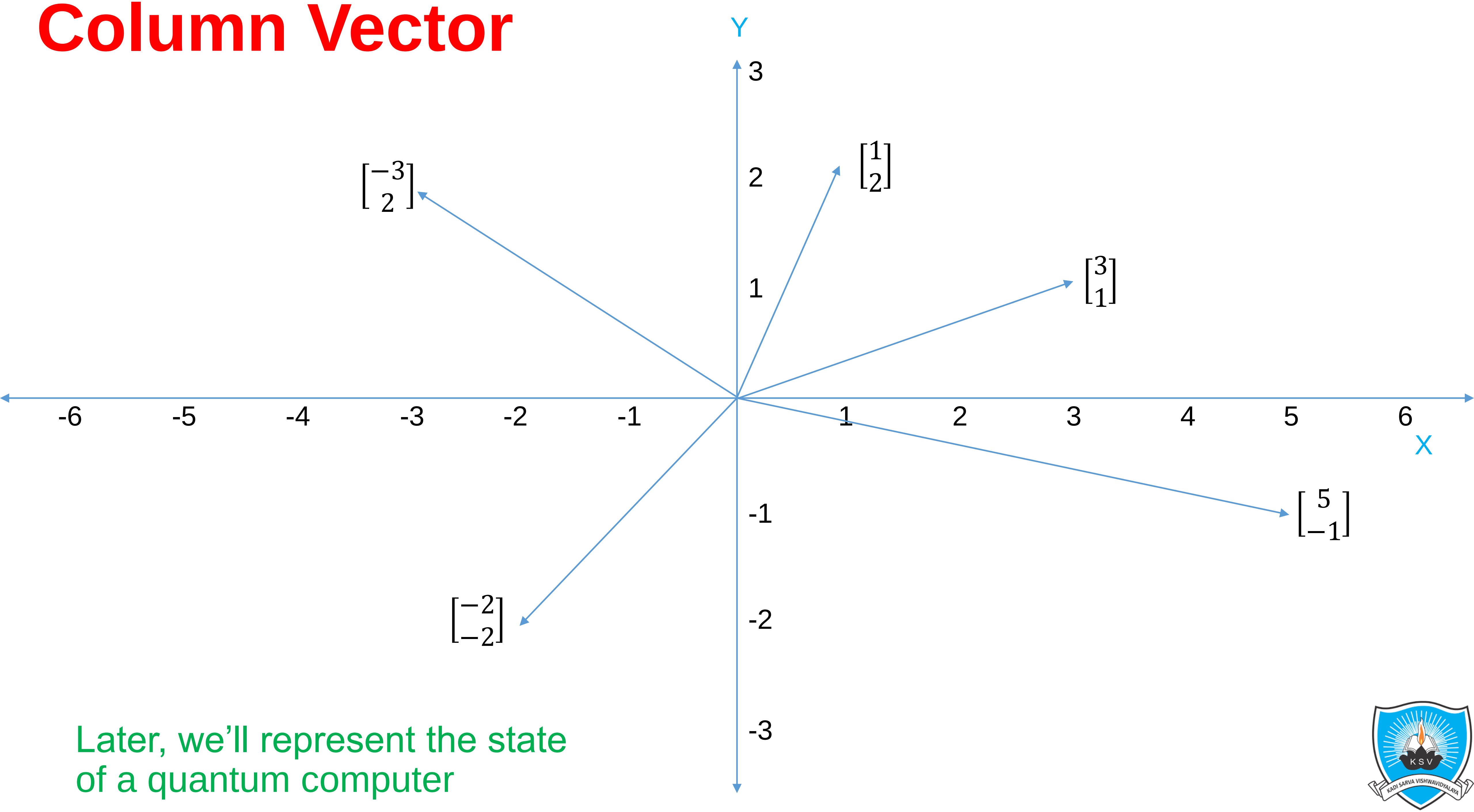


Matrices

Operation: Addition, Multiplication, Scaler

Single column matrix: Column Vector

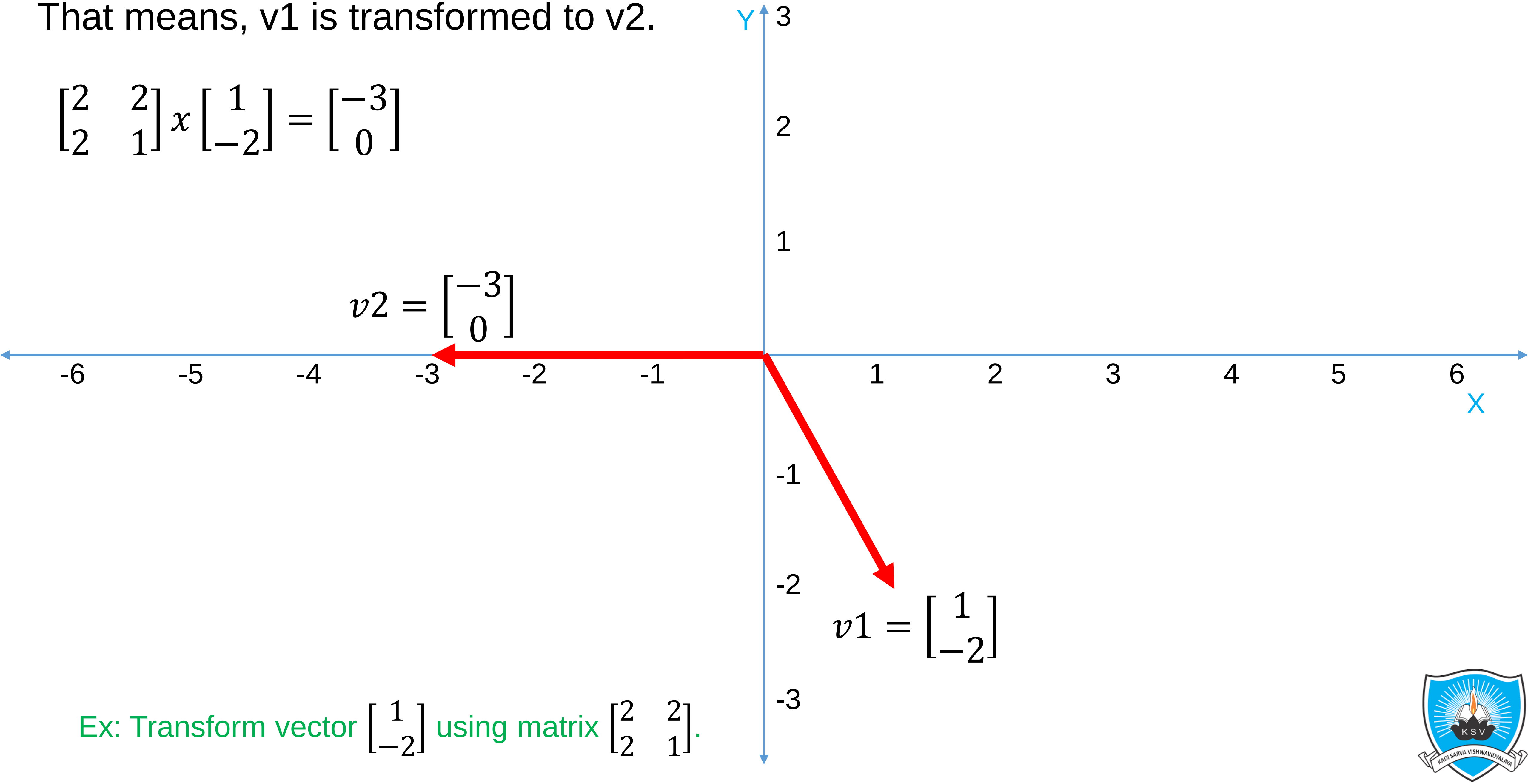
Column Vector


Y
X
3
2
1
-1
-2
-3
-6
-5
-4
-3
-2
-1
1
2
3
4
5
6
$\begin{bmatrix}-3\\2\end{bmatrix}$
$\begin{bmatrix}1\\2\end{bmatrix}$
$\begin{bmatrix}3\\1\end{bmatrix}$
$\begin{bmatrix}5\\-1\end{bmatrix}$
$\begin{bmatrix}-2\\-2\end{bmatrix}$

Later, we’ll represent the state of a quantum computer

When we multiple a column vector (E.g. v1) by matrix, we get another vector (E.g. v2). That means, v1 is transformed to v2.

$$\begin{bmatrix} 2 & 2 \\ 2 & 1 \end{bmatrix} x \begin{bmatrix} 1 \\ -2 \end{bmatrix} = \begin{bmatrix} -3 \\ 0 \end{bmatrix}$$


Y
X
3
2
1
-1
-2
-3
-6
-5
-4
-3
-2
-1
1
2
3
4
5
6
$v2 = \begin{bmatrix} -3 \\ 0 \end{bmatrix}$
$v1 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$

Ex: Transform vector $\begin{bmatrix} 1 \\ -2 \end{bmatrix}$ using matrix $\begin{bmatrix} 2 & 2 \\ 2 & 1 \end{bmatrix}$.

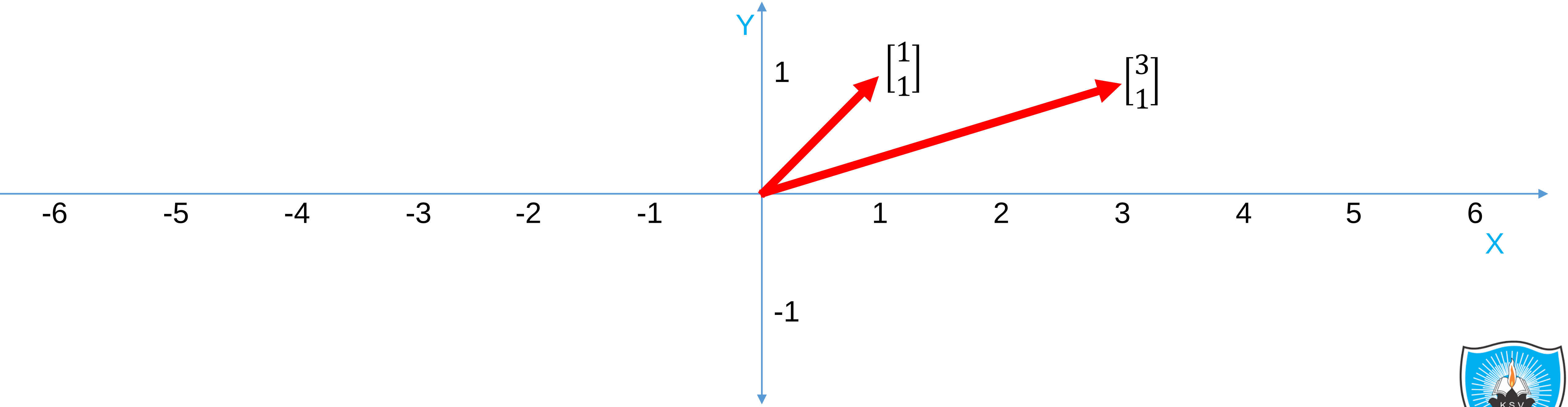
Identity Matrix

$I = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$ $I = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$ $I\ x\ A = A$ $\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\mathrm{x}\\ y\end{bmatrix} = \begin{bmatrix}x\\ y\end{bmatrix}$

Matrix Inverse

$A\ x\ A^{-1} = I$

$A = \begin{bmatrix}1 & 2\\ 2 & -1\end{bmatrix}$ $\begin{bmatrix}1\\ 1\end{bmatrix}$ $A\ x\ I = \begin{bmatrix}3\\ 1\end{bmatrix}$ $A^{-1} = \begin{bmatrix}\frac{1}{5} & \frac{2}{5}\\ \frac{2}{5} & -\frac{1}{5}\end{bmatrix}$ $\begin{bmatrix}\frac{1}{5} & \frac{2}{5}\\ \frac{2}{5} & -\frac{1}{5}\end{bmatrix}$ x $\begin{bmatrix}3\\ 1\end{bmatrix}$ = $\begin{bmatrix}1\\ 1\end{bmatrix}$



Complex Conjugate: Negate the imaginary part

$$A = \begin{bmatrix} 2+3\mathrm{i} & 0 \\ 5 & 3-i \end{bmatrix} \qquad A^{*} = \begin{bmatrix} 2-3\mathrm{i} & 0 \\ 5 & 3+i \end{bmatrix}$$

Matrix Transpose: $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}^{T} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$

$[A^{*}]^{T} = [A^{T}]^{*} = A^{\dagger}$ A dagger [Example in next slide(s)]

Unitary Matrix $U\,U^{\dagger} = I$ i.e. $U^{\dagger}$is inverse of U [Example in next slide(s)]

Hermitian Matrix $H = H^{\dagger}$ [Example in next slide(s)]

original matrix
$$\begin{bmatrix} 2+3i & i & 6-4i \\ 7 & 2-3i & -i \end{bmatrix}$$

complex conjugate
$$\begin{bmatrix} 2-3i & -i & 6+4i \\ 7 & 2+3i & i \end{bmatrix}$$

conjugate transpose
$$\begin{bmatrix} 2-3i & 7 \\ -i & 2+3i \\ 6+4i & i \end{bmatrix}$$

$$\begin{tikzcd}
M \quad \begin{bmatrix} 2+3i & i & 6-4i \\ 7 & 2-3i & -i \end{bmatrix} \arrow[d, "\text{take the conjugate transpose}"] \\
M^H \text{ or } M^{\dagger} \quad \begin{bmatrix} 2-3i & 7 \\ -i & 2+3i \\ 6+4i & i \end{bmatrix}
\end{tikzcd}$$

$$M^H = M$$

Hermitian Matrices

$$M = \begin{bmatrix} 2 & 1 - i \\ 1 + i & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 1 + i \\ 1 - i & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 1 - i \\ 1 + i & 3 \end{bmatrix}$$

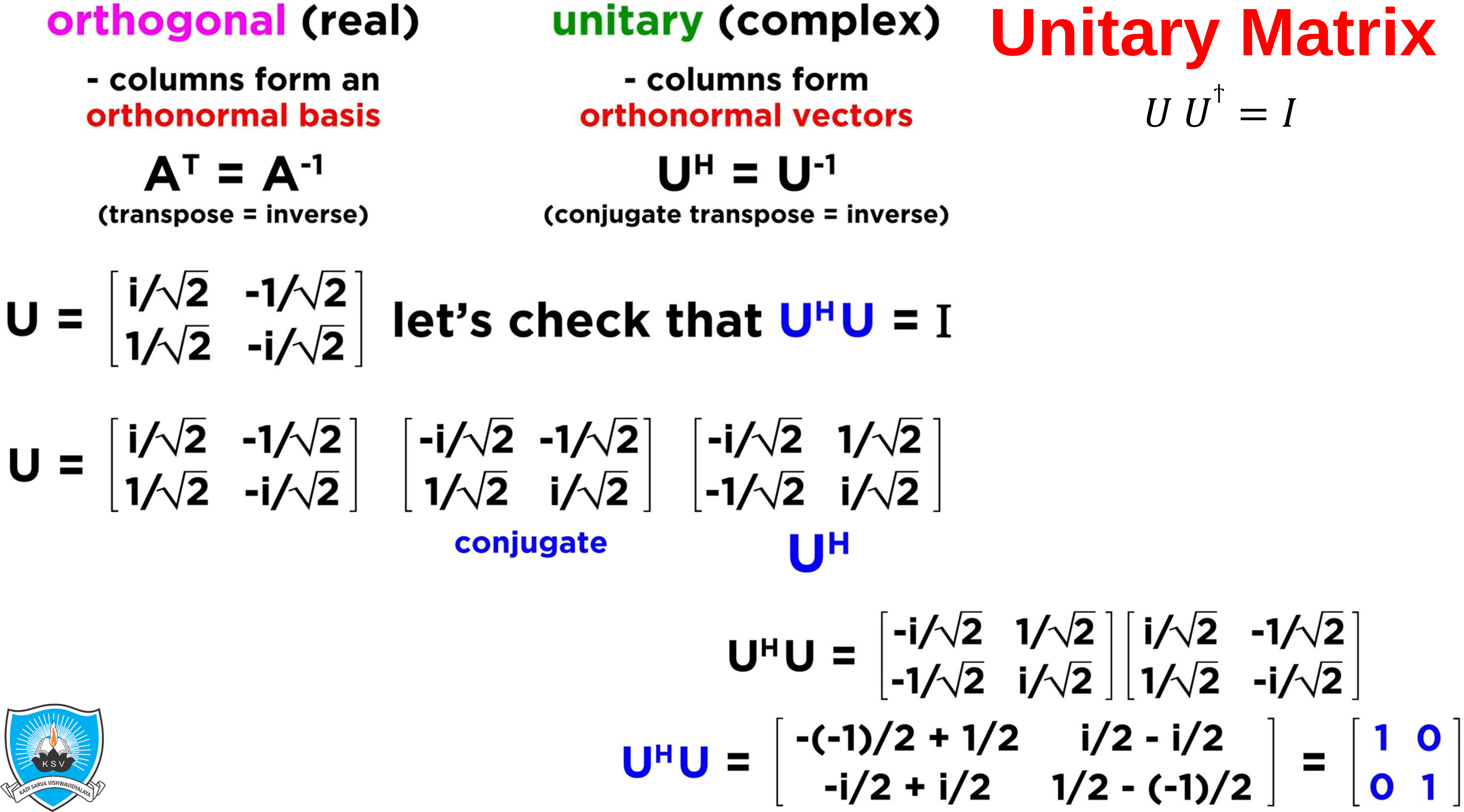
complex conjugate

conjugate transpose

(same as M)

Unitary Matrix

$$U\,U^{\dagger} = I$$



orthogonal (real)

- columns form an **orthonormal basis**

$$A^T = A^{-1}$$

(transpose = inverse)

unitary (complex)

- columns form **orthonormal vectors**

$$U^H = U^{-1}$$

(conjugate transpose = inverse)

$$U = \begin{bmatrix} i/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & -i/\sqrt{2} \end{bmatrix}$$ let's check that $U^H U = I$

$$U = \begin{bmatrix} i/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & -i/\sqrt{2} \end{bmatrix} \quad \underbrace{\begin{bmatrix} -i/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & i/\sqrt{2} \end{bmatrix}}_{\text{conjugate}} \quad \underbrace{\begin{bmatrix} -i/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & i/\sqrt{2} \end{bmatrix}}_{U^H}$$

$$U^H U = \begin{bmatrix} -i/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & i/\sqrt{2} \end{bmatrix} \begin{bmatrix} i/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & -i/\sqrt{2} \end{bmatrix}$$

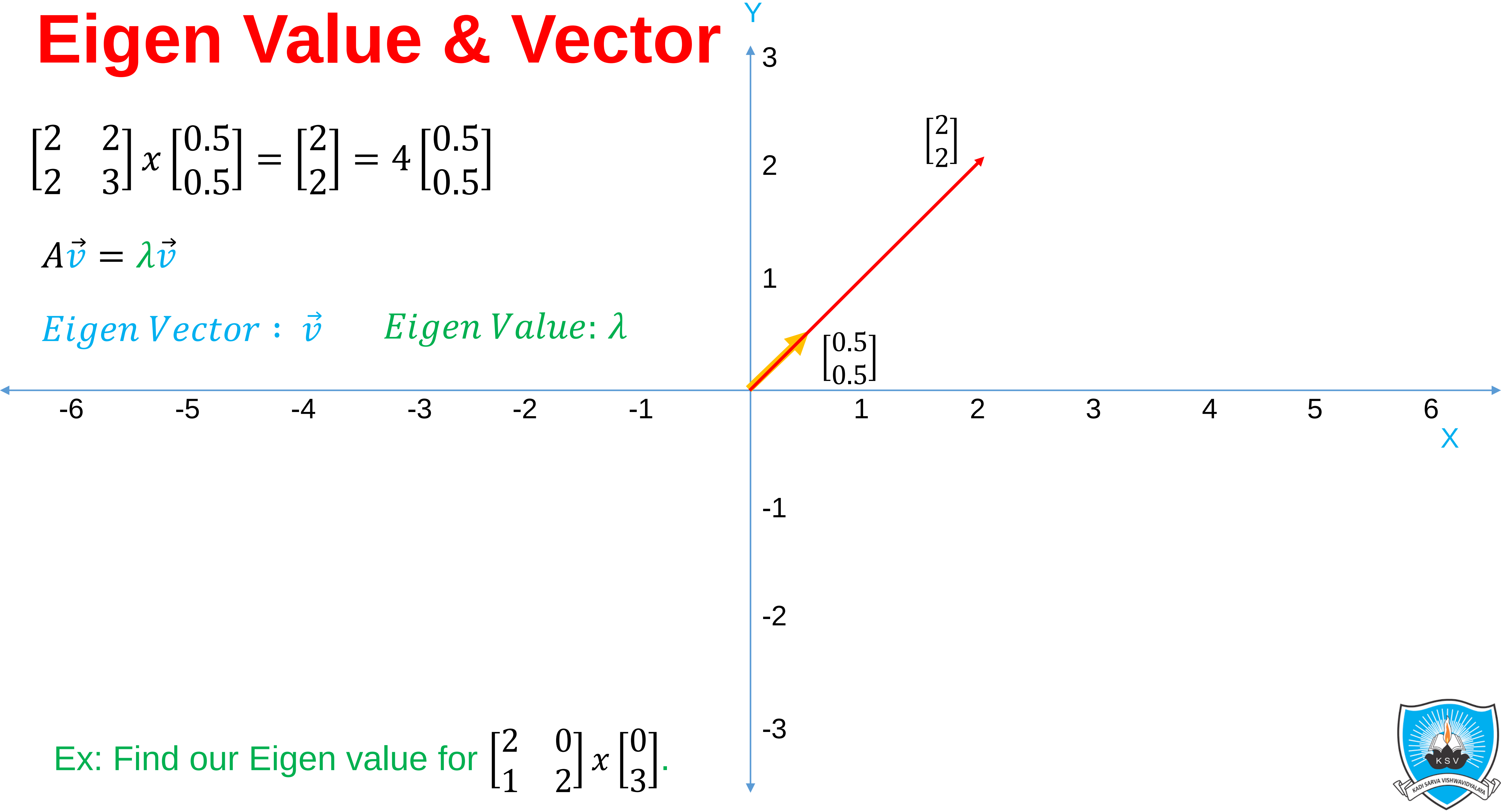
$$U^H U = \begin{bmatrix} -(-1)/2 + 1/2 & i/2 - i/2 \\ -i/2 + i/2 & 1/2 - (-1)/2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Eigen Value & Vector

$$\begin{bmatrix} 2 & 2 \\ 2 & 3 \end{bmatrix} x \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} = 4 \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

$$A\vec{v} = \lambda\vec{v}$$

$Eigen\,Vector : \ \vec{v}$ $\quad$ $Eigen\,Value: \lambda$



Y
3
2
1
-1
-2
-3
-6
-5
-4
-3
-2
-1
1
2
3
4
5
6
X
$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$
$\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$

Ex: Find our Eigen value for $\begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix} x \begin{bmatrix} 0 \\ 3 \end{bmatrix}$.

Qubit (Quantum Bit): Dirac (Bra-Ket) Notation

ket notation: $|0>\ |1>$

$|0> \ = \begin{pmatrix}1\\0\end{pmatrix}, |1> \ = \begin{pmatrix}0\\1\end{pmatrix}$ [Vector representation]

$|\psi> \ = \alpha|0>+\ \beta|1>$

$\begin{pmatrix}\alpha\\\beta\end{pmatrix}$ Element in a complex Hilbert (vector) space.

A Hilbert space is the mathematical framework used to describe the state space of quantum systems. Loosely speaking, it is a complete vector space equipped with an inner product, which allows you to measure angles and lengths of vectors. In quantum computing, these vectors represent quantum states.

bra notation: $< 0|< 1|$

bra (row vector) is a conjugate transpose of a *ket* (column vector).

Qubit (Quantum Bit)

Classical Computer: Bit: Can be either 0 or 1.
Quantum Computer: Qubit: Can be both 0 or 1, at the same time.

Qubit: Any quantum particle that has two states. For example, photon of the light can be polarized horizontally or vertically.

$$|0> \ = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1> \ = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Superposition: Quantum particle is in two states simultaneously.

$$|\psi> \ = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

α: How much the qubit is in $|0>$ state.
β: How much the qubit is in $|1>$ state.

$$|0> \ = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1> \ = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$\alpha = 1$ $(100\%$ the qubit is in $|0>$ state)
0% the qubit is in $|1>$ state

$\beta = 1$ $(100\%$ the qubit is in $|1>$ state)
0% the qubit is in $|0>$ state

Qubit Measurement

When the Qubit is in superposition, it can have both values, 0 and 1. However, when we measure it, it collapses to one of these two states, and gives result of either 0 or 1.

Then, you may ask, what is the role of α and β, then?

α and β are the probability of measuring 0 or 1, respectively.

Probability of measuring $|\psi>$ as 0 is: $|\alpha|^2$
Probability of measuring $|\psi>$ as 1 is: $|\beta|^2$

$$|\psi> = \begin{pmatrix}\alpha \\ \beta\end{pmatrix} = \begin{pmatrix}\frac{\sqrt{3}}{2} \\ \frac{1}{2}\end{pmatrix}$$

Probability of measuring $|\psi>$ as 0 is= $|\alpha|^2$ = ¾ = 75%
Probability of measuring $|\psi>$ as 1 is= $|\beta|^2$ = ¼ = 25%

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

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

$$|\alpha|^2 + |\beta|^2 = 1$$

$\alpha = 1$ (100% the qubit is in $|0>$ state)
0% the qubit is in $|1>$ state

$\beta = 1$ (100% the qubit is in $|1>$ state)
0% the qubit is in $|0>$ state

Dirac Notation

Ex: Convert this into diract notation. $|\psi> \ = \begin{pmatrix} \frac{1}{2} \\ \frac{2\sqrt{3}}{4} \end{pmatrix}$

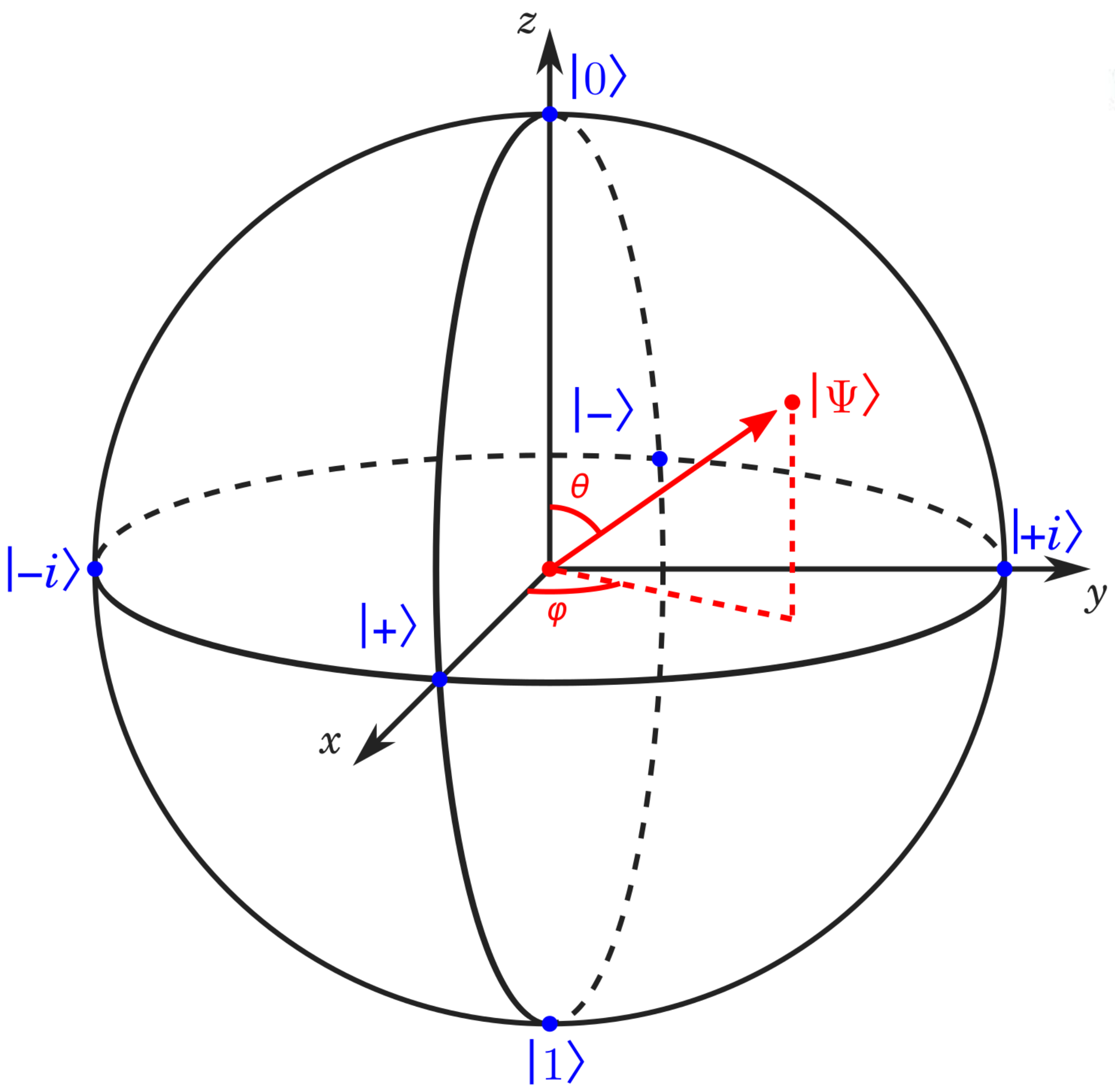
$$|\psi> \ = \binom{\alpha}{\beta}$$

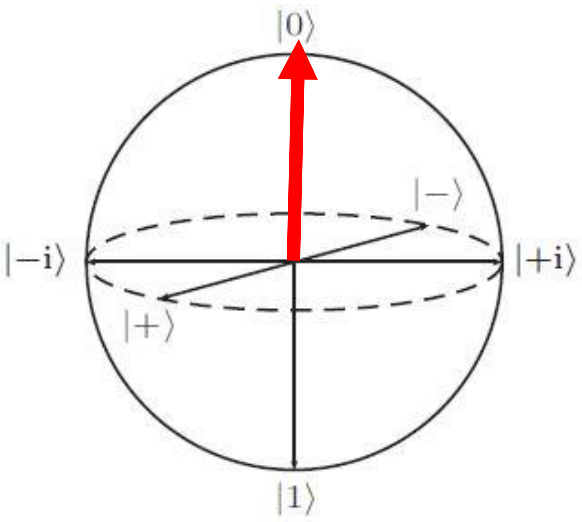
$$|\psi> \ = \binom{\alpha}{\beta} = \binom{\alpha}{0} + \binom{0}{\beta}$$

$$|\psi> \ = \alpha \binom{1}{0} + \beta \binom{0}{1}$$

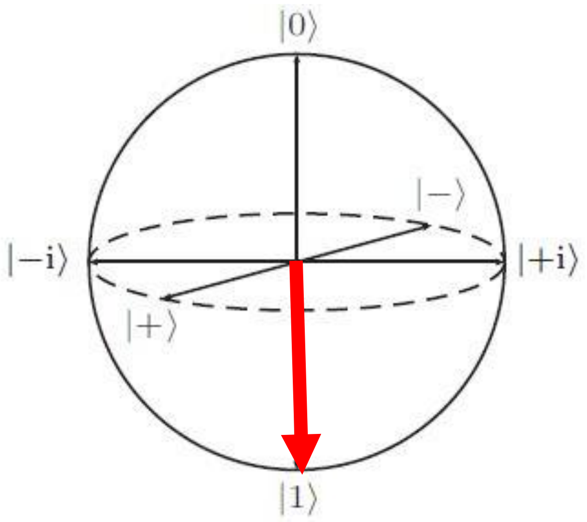
$$|\psi> \ = \underbrace{\alpha|0>+ \ \beta|1>}_{\text{(Dirac Notation)}}$$

Bloch Sphere

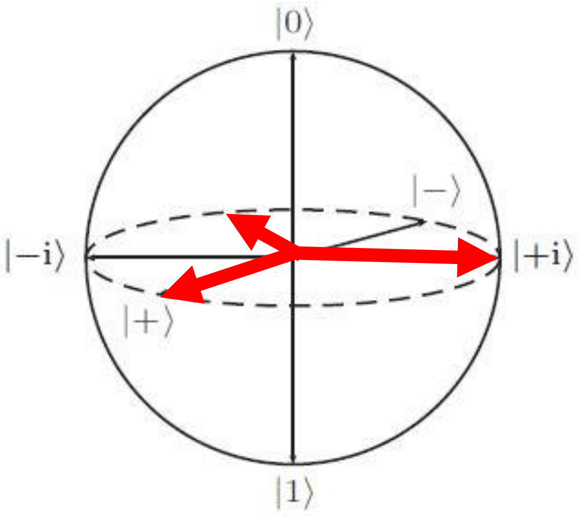




|0>

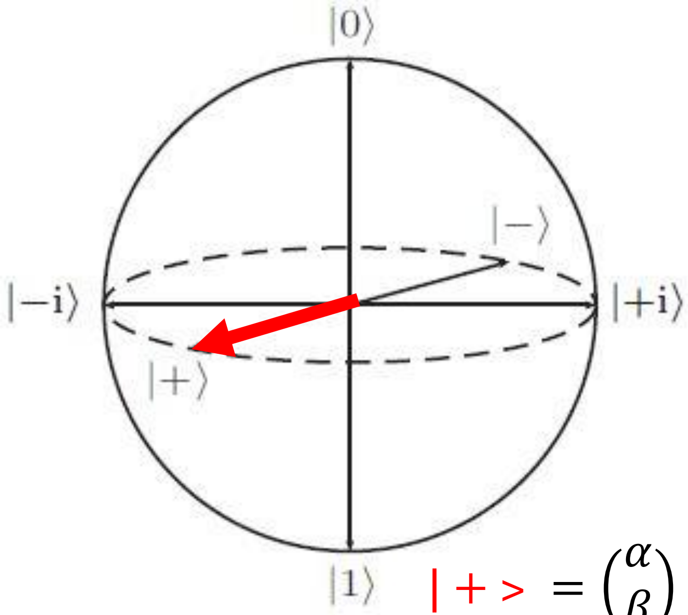


|1>

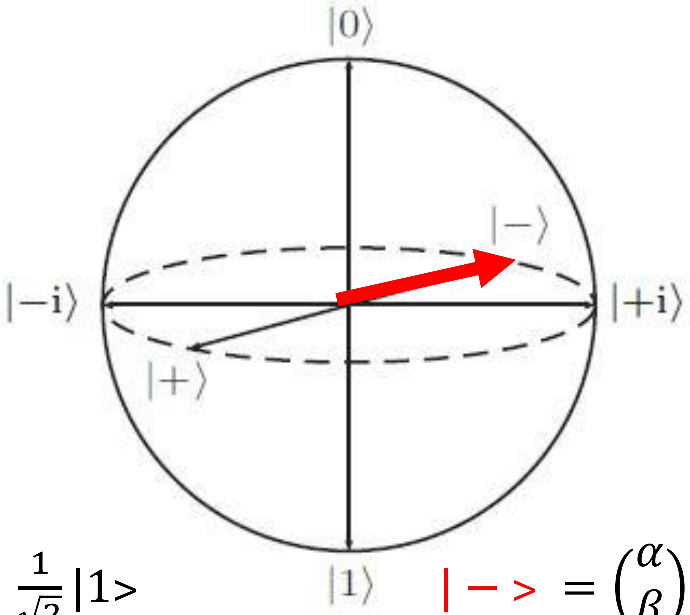


$$|\psi> \; = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

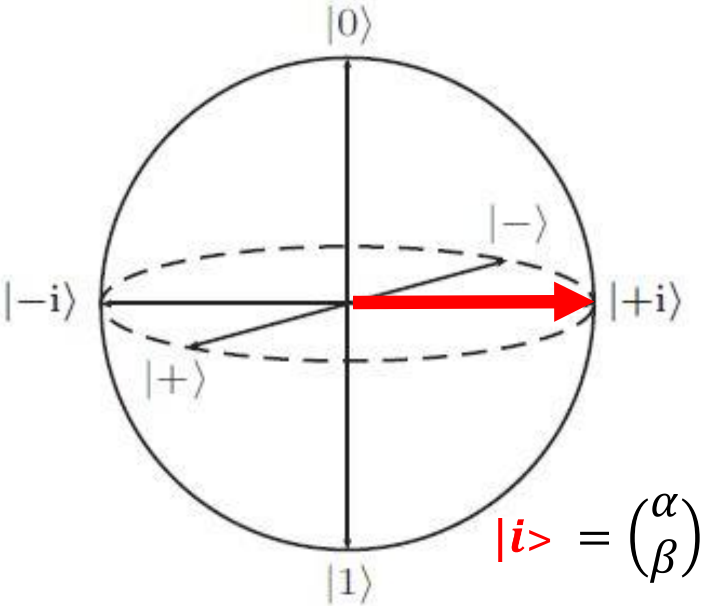
Bloch Sphere



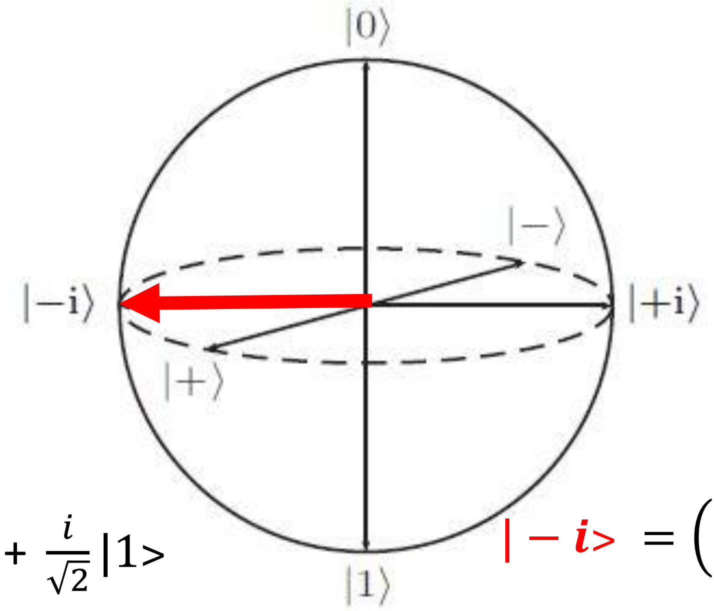
$$|+> \;= \binom{\alpha}{\beta} = \begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{bmatrix} = \frac{1}{\sqrt{2}}|0> + \frac{1}{\sqrt{2}}|1>$$



$$|-> \;= \binom{\alpha}{\beta} = \begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{bmatrix} = \frac{1}{\sqrt{2}}|0> - \frac{1}{\sqrt{2}}|1>$$



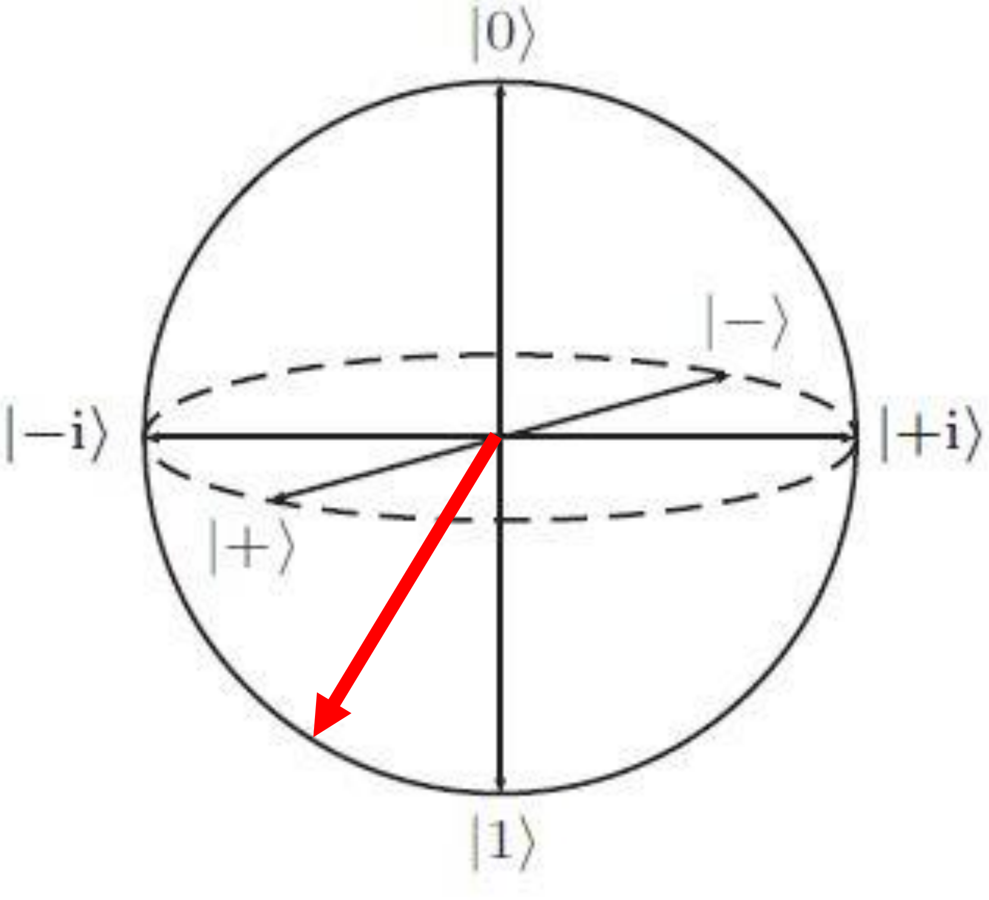
$$|i> \;= \binom{\alpha}{\beta} = \begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{bmatrix} = \frac{1}{\sqrt{2}}|0> + \frac{i}{\sqrt{2}}|1>$$



$$|-i> \;= \binom{\alpha}{\beta} = \begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{bmatrix} = \frac{1}{\sqrt{2}}|0> - \frac{i}{\sqrt{2}}|1>$$

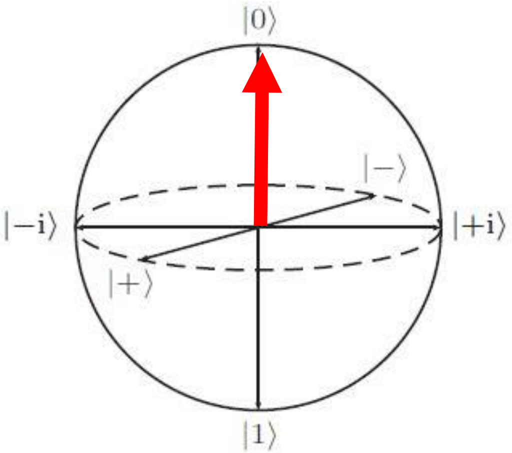
Bloch Sphere

Ex: Locate the state $|\psi> \ = \begin{pmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{pmatrix}$ on the Bloch sphere

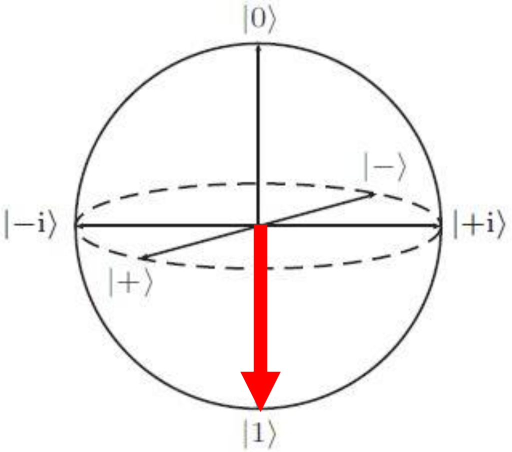


1/4 (25%) Chance of being measured as 0
3/4 (75%) Chance of being measured as 1

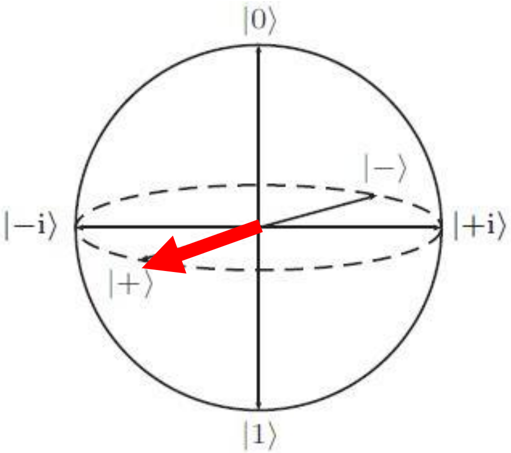
Gates



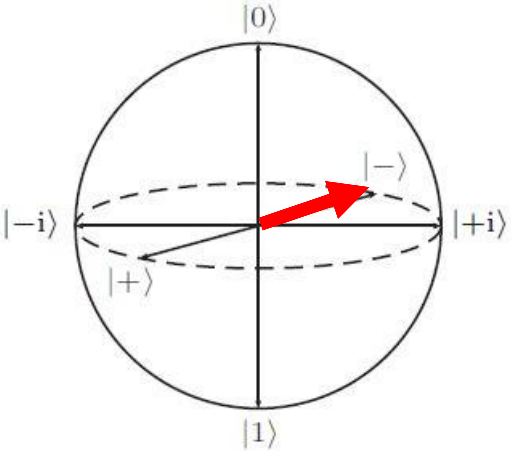
X Gate



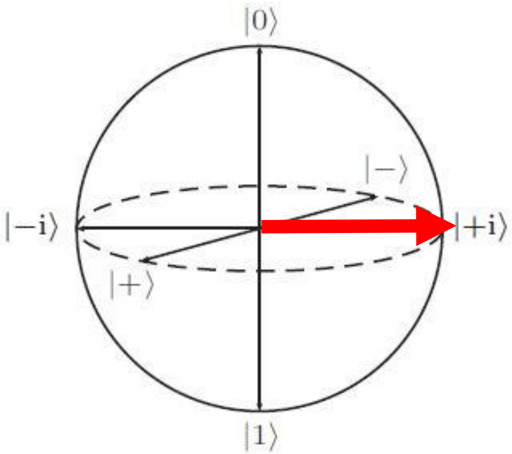
X |0> = |1>



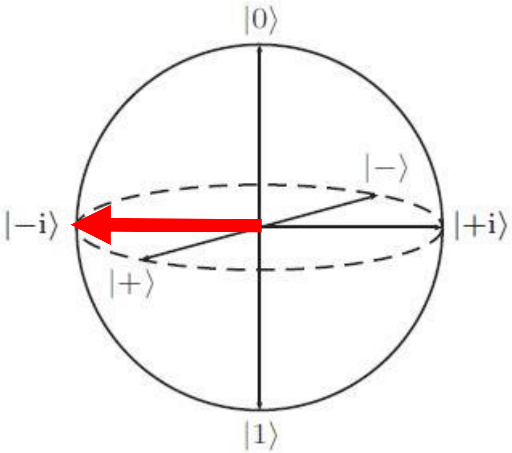
X Gate



X |+> = |->



X Gate



X |i> = |-i>

X gate flips the qubit 180 (π) radians around the X axis on Bloch Sphere

Y/Z gate flips the qubit 180 (π) radians around the Y/Z axis (respectively) on Bloch Sphere

Gates: X, Y, Z

Y gate flips the qubit 180 (π) radians around the Y axis on Bloch Sphere

Z gate flips the qubit 180 (π) radians around the Z axis on Bloch Sphere

Applying the same gate two times would result into the original state.

This means, X, Y and Z gates are **reversible** or inverse of its own.

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

Gates: X, Y, Z

$$X\ |\psi> = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$$

Ex:*Prove that applying X gate to* $|0>$ *results into* $|1>$

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

Ex:*Prove that applying X gate to* $|1>$ *results into* $|0>$

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

Arbitrary Gate

$$Let\ U = \begin{bmatrix} \mathrm{a} & \mathrm{b} \\ \mathrm{c} & \mathrm{d} \end{bmatrix}\ be\ an\ arbitary\ gate$$

$$U|0> = \begin{pmatrix} a \\ c \end{pmatrix} \qquad U|0> = a|0> + b|1>$$

$$U|1> = \begin{pmatrix} b \\ d \end{pmatrix} \qquad U|1> = b|0> + d|1>$$

$$Let\ |\psi> = \alpha|0> +\beta|1> \qquad U|\psi> = U(\alpha|0> +\beta|1>)$$

$$U|\psi> = \ \alpha U|0> +\beta U|1>$$

Example:

$$Let\ us\ apply\ Y\ gate\ \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix}\ to\ |\psi> = \frac{\sqrt{3}}{2}|0> + \frac{1}{2}|1>$$

$$\mathrm{Y}|\psi> = \mathrm{Y}(\frac{\sqrt{3}}{2}|0> + \frac{1}{2}|1>)$$

$$\mathrm{Y}|\psi> = \frac{\sqrt{3}}{2}\mathrm{Y}|0> + \frac{1}{2}\mathrm{Y}|1>$$

$$\mathrm{Y}|\psi> = \frac{\sqrt{3}}{2}\begin{pmatrix} 0 \\ i \end{pmatrix} + \frac{1}{2}\begin{pmatrix} -i \\ 0 \end{pmatrix}$$

$$\mathrm{Y}|\psi> = \frac{\sqrt{3}}{2}i\begin{pmatrix} 0 \\ 1 \end{pmatrix} - \frac{1}{2}\mathrm{i}\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\mathrm{Y}|\psi> = \frac{\sqrt{3}}{2}i\begin{pmatrix} 0 \\ 1 \end{pmatrix} - \frac{1}{2}\mathrm{i}\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

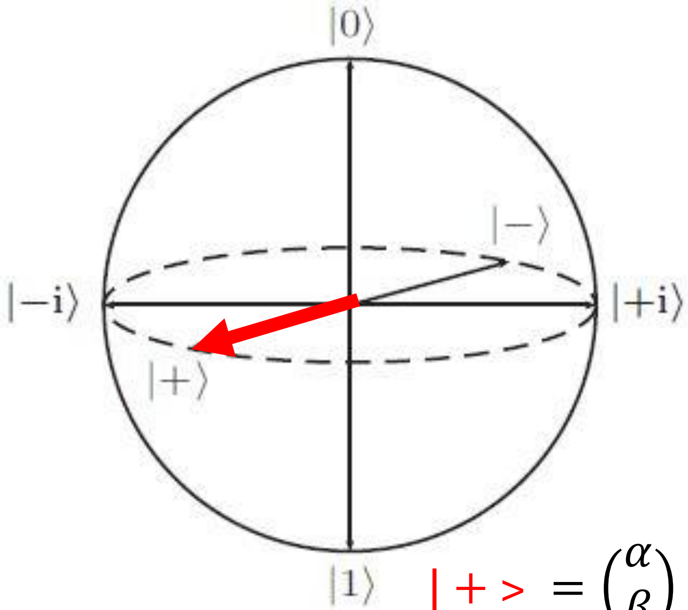
$$\mathrm{Y}|\psi> = \frac{\sqrt{3}}{2}i|1> - \frac{1}{2}\mathrm{i}|0>$$

$$Ex\text{: } Let\ us\ apply\ Z\ gate\ \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\ to\ |\psi> = \alpha|0> + \beta|1>$$

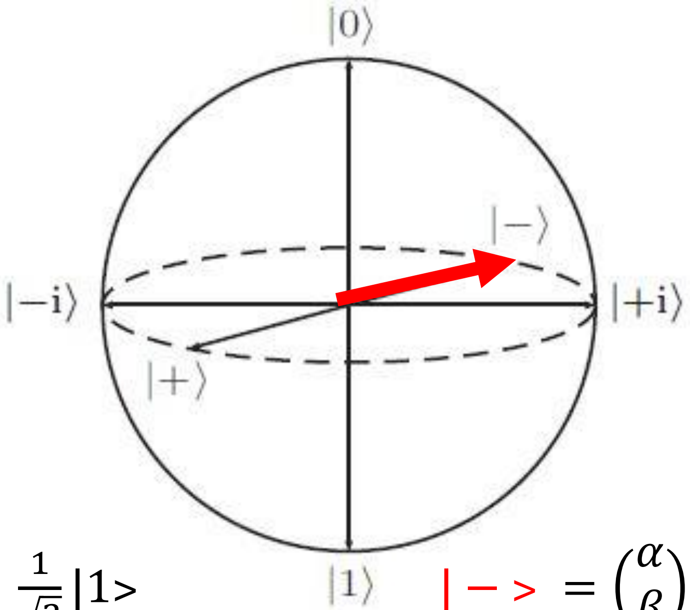
$$\mathrm{ANS}\ = \alpha|0> - \beta|1>$$

Hadamard (H) Gate

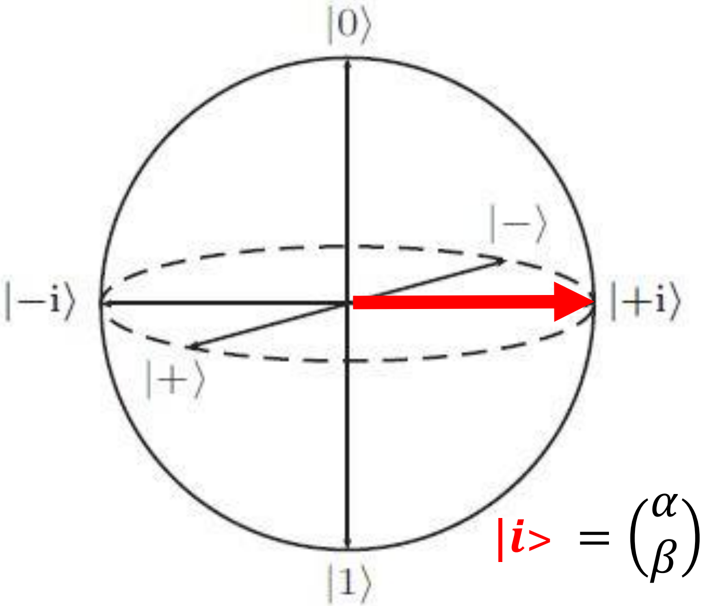
Bloch Sphere



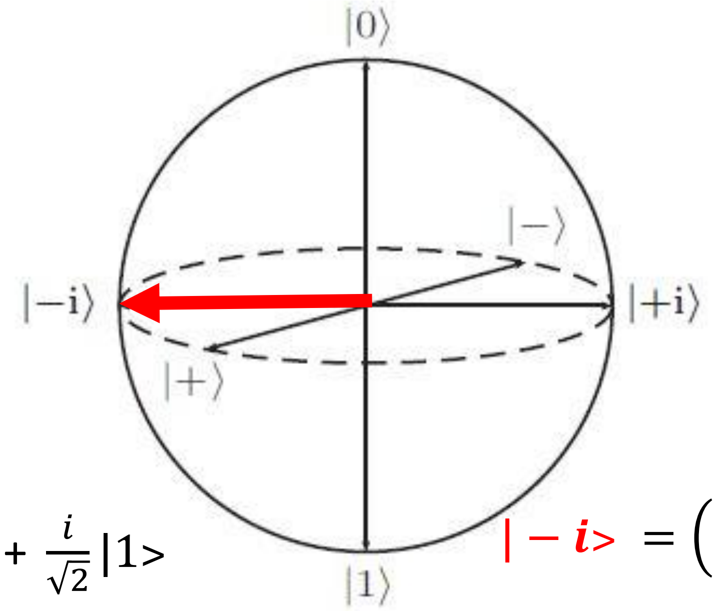
$$|+> = \binom{\alpha}{\beta} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}|0> + \frac{1}{\sqrt{2}}|1>$$



$$|-> = \binom{\alpha}{\beta} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}|0> - \frac{1}{\sqrt{2}}|1>$$



$$|i> = \binom{\alpha}{\beta} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}|0> + \frac{i}{\sqrt{2}}|1>$$



$$|-i> = \binom{\alpha}{\beta} = \begin{bmatrix} 1 \\ \frac{\sqrt{2}}{\frac{1}{\sqrt{2}}} \end{bmatrix} = \frac{1}{\sqrt{2}}|0> - \frac{i}{\sqrt{2}}|1>$$

Hadamard (H) Gate:

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

$$|+> \; = \binom{\alpha}{\beta} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}|0> + \frac{1}{\sqrt{2}}|1>$$

$$|i> \; = \binom{\alpha}{\beta} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}|0> + \frac{i}{\sqrt{2}}|1>$$

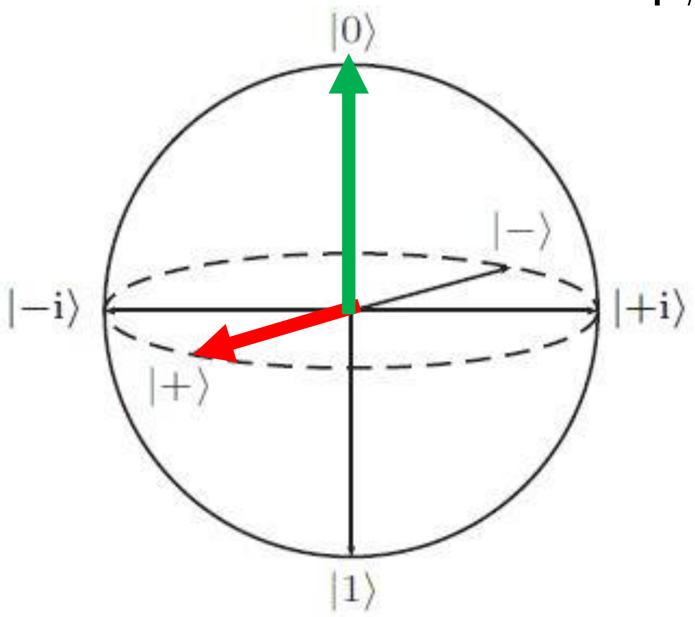
$$|-> \; = \binom{\alpha}{\beta} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}|0> - \frac{1}{\sqrt{2}}|1>$$

$$|-i> \; = \binom{\alpha}{\beta} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}|0> - \frac{i}{\sqrt{2}}|1>$$

$H|0> \rightarrow |+>$ $\qquad$ $H|+> \rightarrow |0>$

Hadamard is inverse of its own (Reversible)

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}|0> + \frac{1}{\sqrt{2}}|1> = |+>$$



$H|1> \rightarrow |->$ $\qquad$ $H|-> \rightarrow |1>$

S & T Gate

$$S = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{bmatrix} \qquad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}$$

Multiple Qubits

Tensor Product $\otimes$

Two Qubits in 0 states are represented using $|0> \otimes |0> \Rightarrow |00>$

Two Qubits in superposition

$$(\alpha|0> +\beta|1>) \otimes (\gamma|0> +\delta|1>) = \alpha\gamma|00> + \alpha\delta|01> + \beta\gamma|10> + \beta\delta|11>$$

$Probability\ of\ measuring\ |00>\ is\ |\alpha\gamma|^2$

Ex: Combine: $(\frac{1}{\sqrt{2}}|0> + \frac{1}{\sqrt{2}}|1>) \otimes (\frac{\sqrt{3}}{2}|0> + \frac{1}{2}|1>)$

Multiple Qubits

Ex: Combine: $(\frac{1}{\sqrt{2}}|0> + \frac{1}{\sqrt{2}}|1>) \otimes (\frac{\sqrt{3}}{2}|0> + \frac{1}{2}|1>)$

$$\frac{1}{\sqrt{2}}\frac{\sqrt{3}}{2}|00> + \frac{1}{\sqrt{2}}\frac{1}{2}|01> + \frac{1}{\sqrt{2}}\frac{\sqrt{3}}{2}|10> + \frac{1}{\sqrt{2}}\frac{1}{2}|11>$$

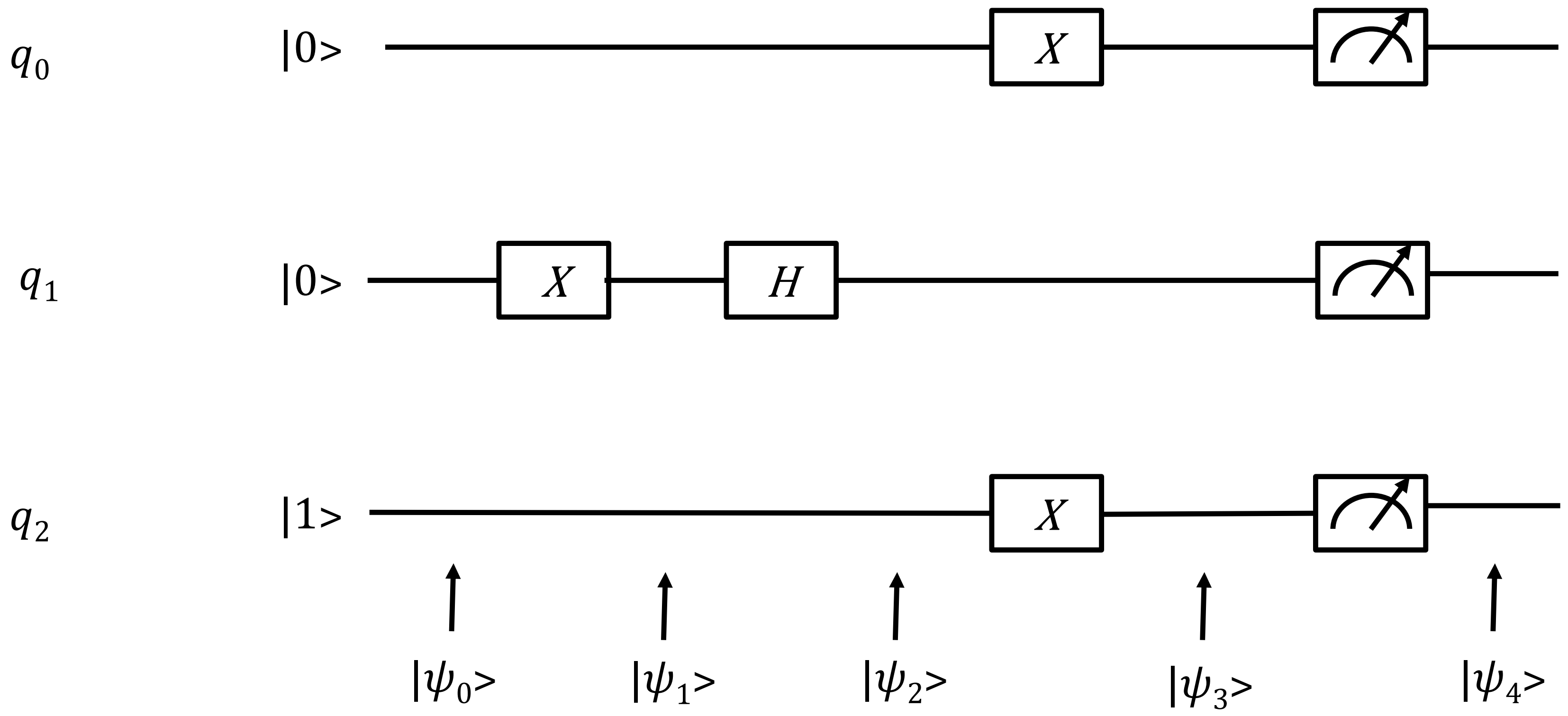
$$\frac{\sqrt{3}}{2\sqrt{2}}|00> + \frac{1}{2\sqrt{2}}|01> + \frac{\sqrt{3}}{2\sqrt{2}}|10> + \frac{1}{2\sqrt{2}}|11>$$

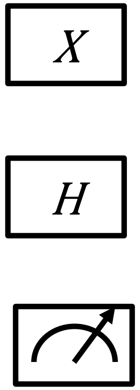
$Probability\ of\ measuring\ |00>\ is\ \left|\frac{\sqrt{3}}{2\sqrt{2}}\right|^2 = \frac{3}{8}$ = 37.5%

$Probability\ of\ measuring\ |01>\ is\ \left|\frac{1}{2\sqrt{2}}\right|^2 = \frac{1}{8}$ = 12.5%

$Probability\ of\ measuring\ |10>\ is\ \left|\frac{\sqrt{3}}{2\sqrt{2}}\right|^2 = \frac{3}{8}$ = 37.5%

$Probability\ of\ measuring\ |11>\ is\ \left|\frac{1}{2\sqrt{2}}\right|^2 = \frac{1}{8}$ = 12.5%





X Gate

Hadamard Gate

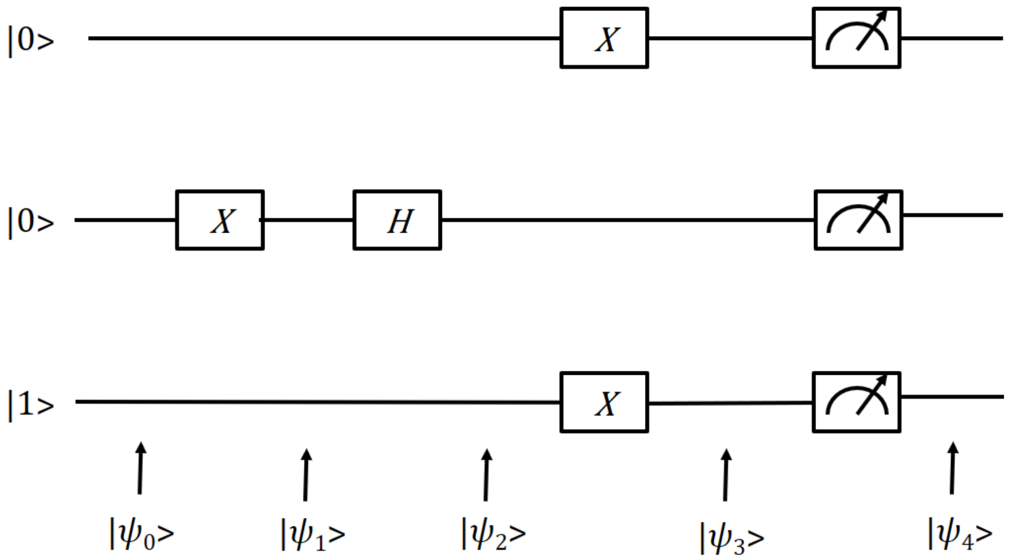
Measurement

Quantum Circuits

$$|\psi_0> = |001>$$

$$|\psi_1> = |011>$$

$$|\psi_2> = |0 - 1>$$



$$|\psi_2> = |0 \otimes (\frac{1}{\sqrt{2}}|0> - \frac{1}{\sqrt{2}}|1>) \otimes 1>$$

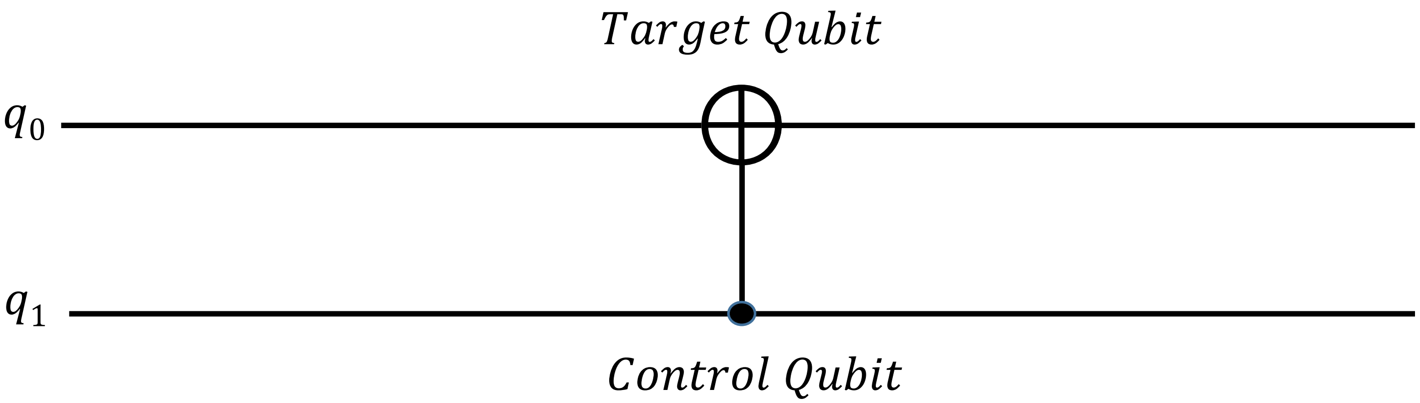
$$|\psi_2> = \frac{1}{\sqrt{2}}(|001> - \frac{1}{\sqrt{2}}|011>)$$

$$|\psi_3> = \frac{1}{\sqrt{2}}(|100> - \frac{1}{\sqrt{2}}|110>)$$

$$|\psi_4> =$$
$$|100> (50\% \; Probability)$$
$$|110> (50\% \; Probability)$$

Multi-Qubit Gates: CNOT, Toffoli, Controlled Gates

CNOT: Controlled NOT Gate



$If\ the\ Control\ Qubit\ is\ 1, the\ Target\ bit\ is\ flipped.$

CNOT: Controlled NOT Gate

$Ex: Apply\ CNOT\ to\ \left(\frac{\sqrt{3}}{2}\left|00>+\frac{1}{2}\right|01>+\frac{1}{\sqrt{2}}\left|10>+\frac{1}{4}\right|11>\right),$

$assuming\ first\ bit\ as\ a\ control\ and\ second\ bit\ as\ a\ target\ bit$

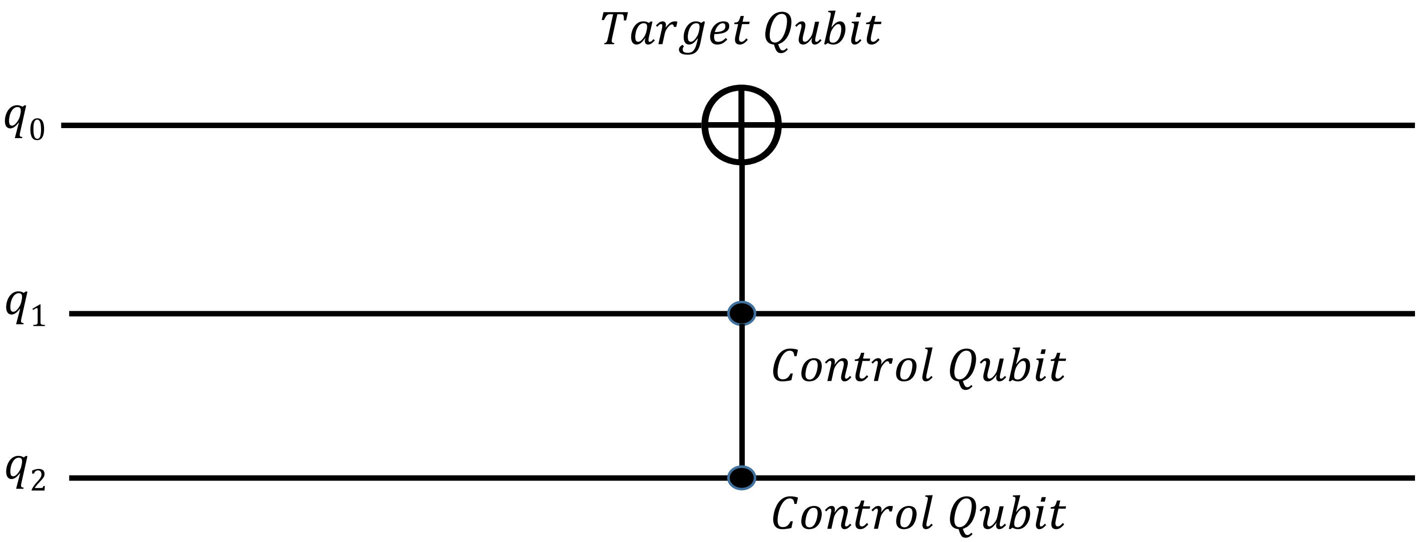
$$CNOT\left(\frac{\sqrt{3}}{2}\left|00>+\frac{1}{2}\right|01>+\frac{1}{\sqrt{2}}\left|10>+\frac{1}{4}\right|11>\right)$$

$$\left(\frac{\sqrt{3}}{2}CNOT\left|00>+\frac{1}{2}CNOT\right|01>+\frac{1}{\sqrt{2}}CNOT\left|10>+\frac{1}{4}CNOT\right|11>\right)$$

$$\left(\frac{\sqrt{3}}{2}\left|00>+\frac{1}{2}\right|01>+\frac{1}{\sqrt{2}}\left|11>+\frac{1}{4}CNOT\right|10>\right)$$

$Ex: Apply\ CNOT\ to\ \left(\frac{\sqrt{3}}{2}\left|001>+\frac{1}{2}\right|010>\right), assuming\ third\ bit\ as\ a\ control\ and\ second\ bit\ as\ a\ target\ bit$

Toffoli Gate: Controlled NOT Gate with 2 control qubits



$If\ all\ the\ Control\ Qubit\ are\ 1, the\ Target\ bit\ is\ flipped.$

Toffoli Gate: Controlled NOT Gate with 2 control qubits

$Ex: Apply\ Toffoli\ to \left(\frac{1}{\sqrt{2}}\left|0011> + \frac{1}{\sqrt{2}}\right| 0110>\right),$

$assuming\ second\ and\ third\ bits\ as\ control\ and\ fourth\ bit\ as\ a\ target\ bit$

$$Toffoli\left(\frac{1}{\sqrt{2}}\left|0011> + \frac{1}{\sqrt{2}}\right| 0110>\right)$$

$$\left(\frac{1}{\sqrt{2}}Toffoli\left|0011> + \frac{1}{\sqrt{2}}Toffoli\right| 0110>\right)$$

$$\left(\frac{1}{\sqrt{2}}\left|0011> + \frac{1}{\sqrt{2}}Toffoli\right| 0111>\right)$$

Measuring a qubit

Find out the probability of measuring second qubit as 1 in the state $(\frac{1}{2}|00> + \frac{1}{4}|01> + \frac{e^{i\frac{\pi}{2}}}{\sqrt{2}}|10> + \frac{\sqrt{3}}{4}|11>)$

$(\frac{1}{2}|00> + \frac{1}{4}|0\mathbf{1}> + \frac{e^{i\frac{\pi}{2}}}{\sqrt{2}}|10> + \frac{\sqrt{3}}{4}|1\mathbf{1}>)$ There are two super-positions where the second qubit is 1

Let us sum up their probability: $|\frac{1}{4}|^2 + |\frac{\sqrt{3}}{4}|^2$

$|\frac{1}{16}| + |\frac{3}{16}|$

$|\frac{4}{16}|$

$|\frac{1}{4}|$

25%

Measuring a qubit

How do we know what the state collapses to, once we measure a qubit?
Once we measure a qubit, it collapses from superposition and becomes a measurement.

Example: Let us assume that in the given state, we measure the first qubit to be 1.

$$(\frac{1}{2}|00> + \frac{1}{4}|01> + \frac{1}{\sqrt{2}}|10> + \frac{\sqrt{3}}{4}|11>)$$

As we know, the first qubit collapses to 1, and hence, we may ignore the state where the first qubit is zero.

$$\frac{1}{\sqrt{2}}|10> + \frac{\sqrt{3}}{4}|11>$$

Now to measure the total probability, let us add the remaining probability $|\frac{1}{2}| + |\frac{3}{16}|$

Since, the addition of all the probability should be 1, here it is 11/16, which is not 1. This is due to the fact that we ignored few states.

Here comes a normalization constant A, to bring the total probability to 1.

A $(\frac{1}{\sqrt{2}}|10> + \frac{\sqrt{3}}{4}|11>)$ $\quad (\frac{A}{\sqrt{2}})^2 + \left(\frac{A\sqrt{3}}{4}\right)^2 = 1$ $\quad A = 4/11$ $\quad \frac{4}{11}(\frac{1}{\sqrt{2}}|10> + \frac{\sqrt{3}}{4}|11>)$ $\quad \frac{4}{\sqrt{22}}|10> + \frac{\sqrt{3}}{\sqrt{11}}|11>$

Measuring a qubit

Example: Let us assume that in the given state, we measure the middle qubit to be 0.

$$|\psi_0> = (\frac{1}{2}|000> + \frac{1}{2}|001> + \frac{1}{2}|010> + \frac{1}{2}|101>)$$

$$|\psi_0> = (\frac{1}{2}|000> + \frac{1}{2}|001> + \frac{1}{2}|010> + \frac{1}{2}|101>)$$

$$|\psi_0> = A\ (\frac{1}{2}|000> + \frac{1}{2}|001> + \frac{1}{2}|101>)$$

$$|\frac{A}{2}|^2 + |\frac{A}{2}|^2 + |\frac{A}{2}|^2 = 1$$

$$\frac{3A^2}{4} = 1$$

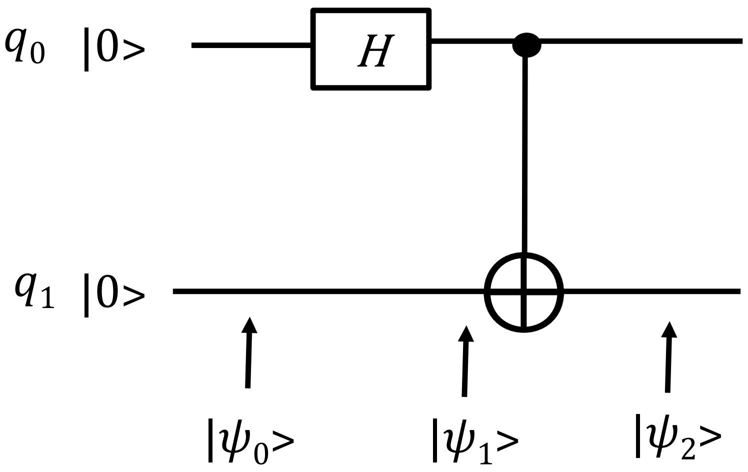
$$A = \frac{2}{\sqrt{3}}$$

$$|\psi_0> = \frac{2}{\sqrt{3}}\ (\frac{1}{2}|000> + \frac{1}{2}|001> + \frac{1}{2}|101>)$$

$$|\psi_0> = \frac{1}{\sqrt{3}}|000> + \frac{1}{\sqrt{3}}|001> + \frac{1}{\sqrt{3}}|101>)$$

Entanglement and Bell States

Entanglement



$$|\psi_0> = |00>$$

$$|\psi_1> = \mathrm{H}|0>|0>$$

$$|\psi_1> = | + 0>$$

$$|\psi_1>=(\frac{1}{\sqrt{2}}|0> + \frac{1}{\sqrt{2}}|1>) \otimes |0>$$

$$|\psi_1>=(\frac{1}{\sqrt{2}}|00> + \frac{1}{\sqrt{2}}|10>)$$

$$|\psi_2>=(\frac{1}{\sqrt{2}}CNOT|00> + \frac{1}{\sqrt{2}}CNOT|10>)$$

$$|\psi_2>=(\frac{1}{\sqrt{2}}|00> + \frac{1}{\sqrt{2}}|11>)$$

Conclusion: If we measure one of the qubit to 0, then other would collapse into 0.
And If we measure one of the qubit to 1, then other would collapse into 1.
Hence, without at looking both of the qubits, merely, by looking at one of the qubits, we can say the state of the other qubit.

Entangle States

A state is entangled if it can not be factored

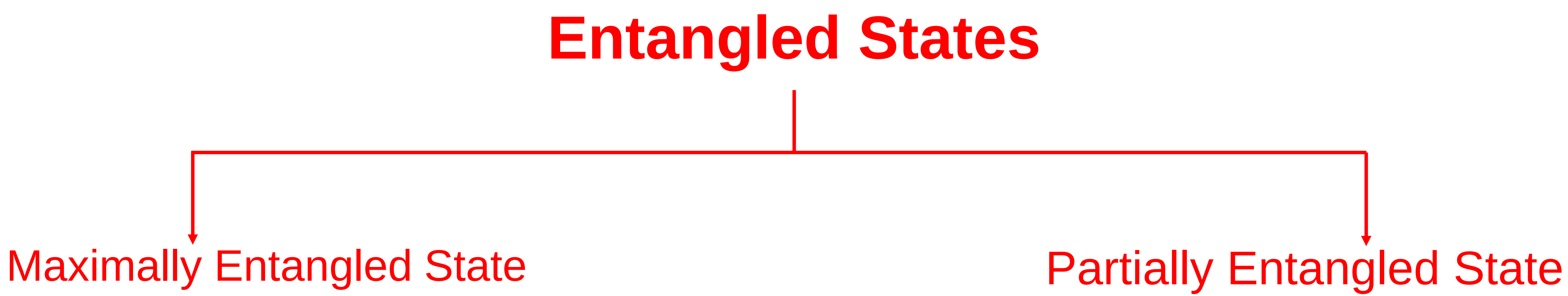
A state is entangled if it can not be factored into tensor products of individual qubits

For instance, the state $(\frac{\sqrt{3}}{2\sqrt{5}}|00> + \frac{1}{2\sqrt{5}}|01> + \frac{\sqrt{3}}{\sqrt{5}}|10> + \frac{1}{\sqrt{5}}|11>)$ in not in entangled state as, it can be factored into

$$(\frac{1}{\sqrt{5}}|0> + \frac{2}{\sqrt{5}}|1>) \otimes (\frac{\sqrt{3}}{2}|0> + \frac{1}{2}|1>)$$

But, the state $\frac{1}{\sqrt{2}}(|000> + |011>)$ is in entangled.

Entangled States



Maximally Entangled State

We can certainly determine state of one qubit from other

Example: $|\psi_1> = \frac{1}{\sqrt{2}}(|00> + |11>)$

$$\left.\begin{aligned}
|\phi^{+}> &= \frac{1}{\sqrt{2}}(|00> + |11>)\\
|\phi^{-}> &= \frac{1}{\sqrt{2}}(|00> - |11>)\\
|\Psi^{+}> &= \frac{1}{\sqrt{2}}(|01> + |10>)\\
|\Psi^{-}> &= \frac{1}{\sqrt{2}}(|01> - |10>)
\end{aligned}\right\} \text{Bell States}$$

Partially Entangled State

By measuring one of the qubits, the amplitude of other qubit is affected

Example: $(\frac{\sqrt{3}}{\sqrt{5}}|00> + \frac{1}{\sqrt{5}}|01> + \frac{1}{2\sqrt{5}}|10> + \frac{\sqrt{3}}{2\sqrt{5}}|11>)$

If we measure the first qubit as 0, the state collapses to:

$|0> \otimes \boxed{(\frac{\sqrt{3}}{2}|0> + \frac{1}{2}|1>)}$ $\longleftarrow$ Second qubit

If we measure the first qubit as 1, the state collapses to:

$|1> \otimes \boxed{(\frac{1}{2}|0> + \frac{\sqrt{3}}{2}|1>)}$ $\longleftarrow$ Second qubit

Super Dense Coding

Super Dense Coding

This quantum protocol allows to send two bits of classical information (00, 01, 10, 11) using 1 qubit.

Alice (sender) and Bob (receiver) maximally entangle two qubits.

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

Alice wants to send 00, she does nothing: Qubit $|\psi> = \frac{1}{\sqrt{2}}(|00> + |11>)$

Alice wants to send 01, she applies X gate to her qubit $|\psi> = \frac{1}{\sqrt{2}}(|10> + |01>)$

Alice wants to send 10, she applies Z gate to her qubit $|\psi> = \frac{1}{\sqrt{2}}(|00> - |11>)$

Alice wants to send 11, she applies both X & Z gate to her qubit $|\psi> = \frac{1}{\sqrt{2}}(|10> - |01>)$

Now, Bob has both the qubit in one of these four states,

$$|\psi> = \frac{1}{\sqrt{2}}(|00> + |11>)\ ,\ |\psi> = \frac{1}{\sqrt{2}}(|10> + |01>)\ ,\ |\psi> = \frac{1}{\sqrt{2}}(|00> - |11>)\ ,\ |\psi> = \frac{1}{\sqrt{2}}(|10> - |01>)$$

Super Dense Coding

Bob, now applies CNOT as first qubit as control and second qubit as target

Bob, then applies H gate on the left qubit.

$$|\psi> = \frac{1}{\sqrt{2}}(|00> + |11>) \xrightarrow{CNOT} |\psi> = \frac{1}{\sqrt{2}}(|00> + |10>) \xrightarrow{H} \mathrm{H}|+> |0> = |00 >$$

$$|\psi> = \frac{1}{\sqrt{2}}(|10> + |01>) \xrightarrow{CNOT} |\psi> = \frac{1}{\sqrt{2}}(|11> + |01>) \xrightarrow{H} \mathrm{H}|+> |1> = |01 >$$

$$|\psi> = \frac{1}{\sqrt{2}}(|00> - |11>) \xrightarrow{CNOT} |\psi> = \frac{1}{\sqrt{2}}(|00> - |10>) \xrightarrow{H} \mathrm{H}|-> |0> = |10 >$$

$$|\psi> = \frac{1}{\sqrt{2}}(|10> - |01>) \xrightarrow{CNOT} |\psi> = \frac{1}{\sqrt{2}}(|11> - |01>) \xrightarrow{H} \mathrm{H}|-> |1> = |11 >$$

Bob, now measures the qubits and he’ll have |00> or |01> or |10> or |11>

Reversible Function

Reversible Function

NOT	
Input=x	Output=f(x)
0	1
1	0

AND	
Input=x	Output=f(x)
00	0
01	0
10	0
11	1

OR	
Input=x	Output=f(x)
00	0
01	1
10	1
11	1

XOR	
Input=x	Output=f(x)
00	0
01	1
10	1
11	0

A function f is **reversible**, if given f(x), we can find x

Which of the above gate is reversible?

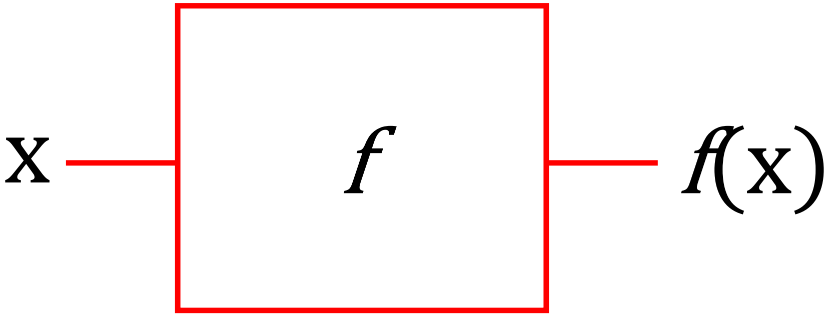
Check whether the following function is reversible?

Function F	
Input=x	Output=f(x)
00	01
01	00
10	11
11	10

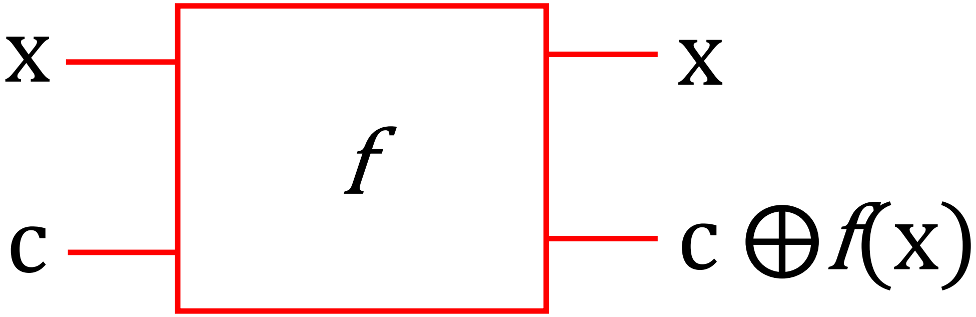
Reversible Function

A function f is **reversible**, if given f(x), we can find x

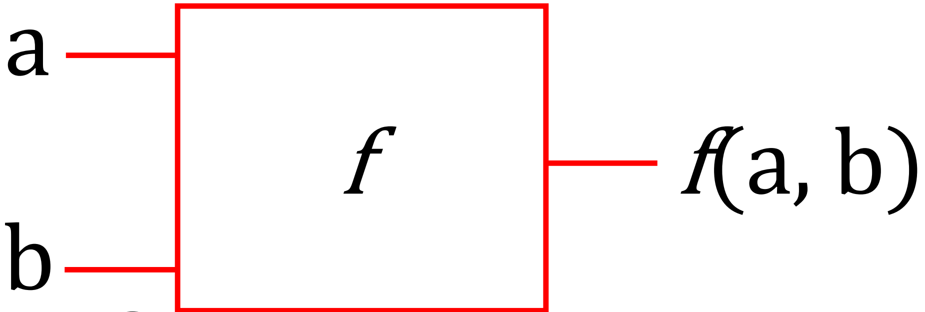
To make a function f is **reversible**, we need add extra information



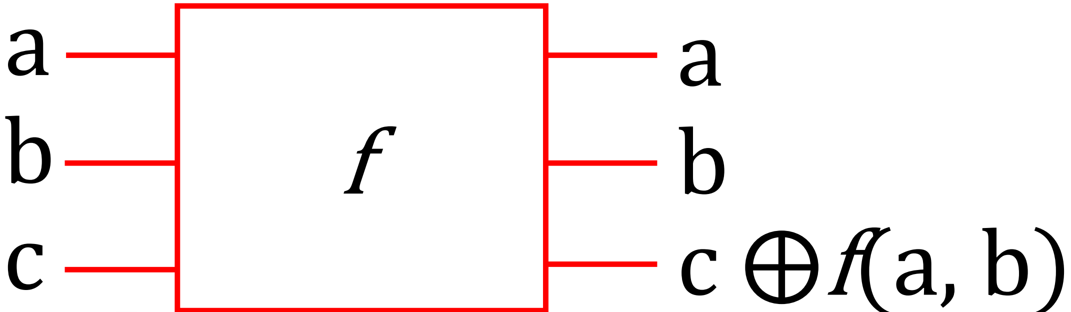
Standard Classical Operations



Reversible Classical Operations

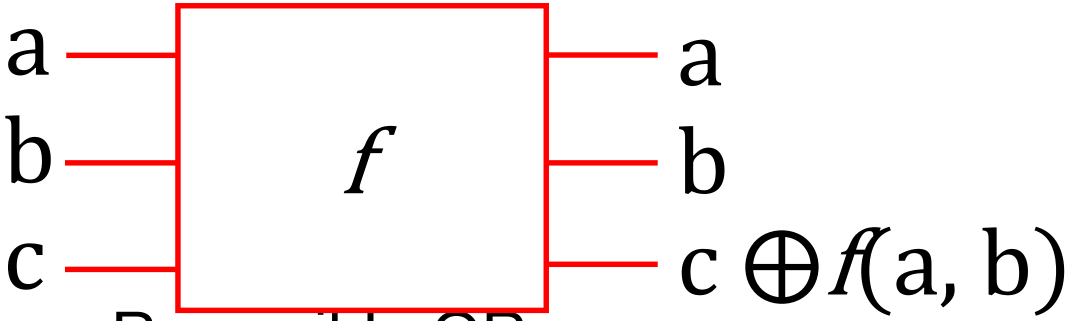


Standard OR Gate



Reversible OR Gate

Reversible Function



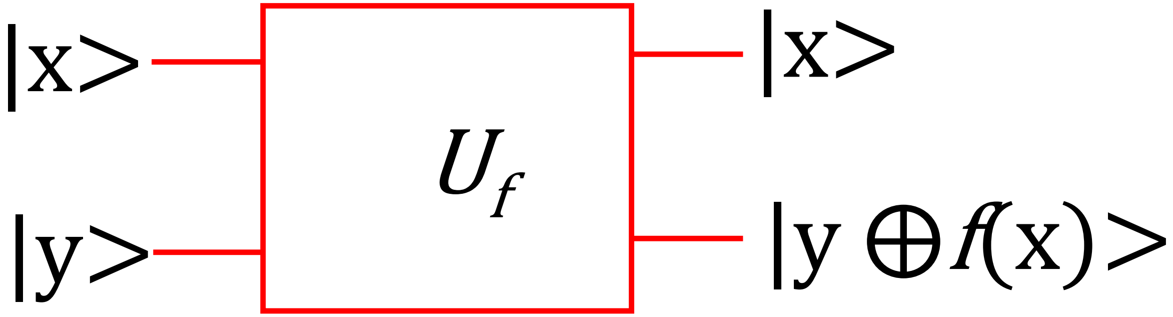
Reversible OR Gate

a	b	c	a	b	$c \oplus f(a, b)$
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	1	0

Why all these?

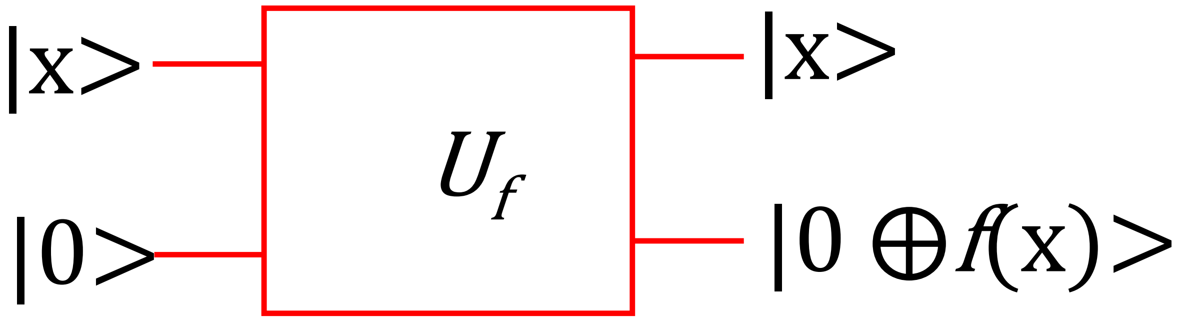
In Quantum Computing, all operations must be **reversible**.

Functions on Quantum Computers / Phase Oracle



Standard Quantum Function / Oracle

$$U_f\,|x>\,|y> = |x>\,|y \oplus f(x)>$$



$$U_f\,|x>\,|0> = |x>\,|0 \oplus f(x)>$$

$$U_f\,|x>\,|0> = |x>\,|f(x)>$$

$U_f\,|x>\,|->$

$U_f\,|x> \frac{1}{\sqrt{2}}(|0> - |1>)$

$U_f\,\frac{1}{\sqrt{2}}(|x>|0> - |x>|1>)$

$\frac{1}{\sqrt{2}}(U_f|x>|0> - U_f|x>|1>)$

$\frac{1}{\sqrt{2}}(|x>|f(x)> - |x>|\overline{f(x)}>)$

if f(x) = 0, $\frac{1}{\sqrt{2}}(|x>|0> - |x>|1>) \Rightarrow$ **$|x>|->$**

if f(x) = 1, $\frac{1}{\sqrt{2}}(|x>|1> - |x>|0>)$

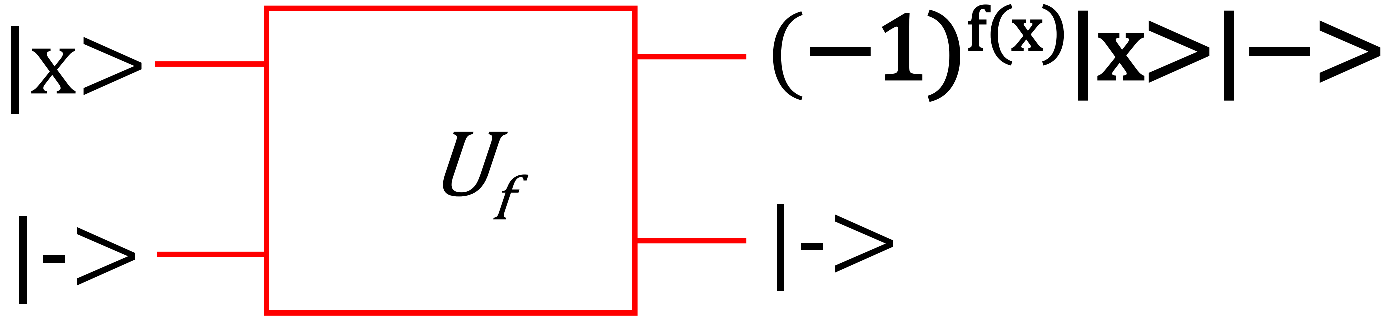
$\Rightarrow -\frac{1}{\sqrt{2}}(|x>|0> - |x>|1>)$

$\Rightarrow$ **$-|x>|->$**

Combining both, $U_f\,|x>\,|-> = (-1)^{f(x)}|x>|->$

Phase Oracle

$$U_f\,|x>\,|->\,=(-1)^{f(x)}|x>|->$$



Instead of the function output f(x) being XORed with the |-> register a phase of $(-1)^{f(x)}$ was applied to the input to the function (|x>)

When we create a function in this way, with the output in this minus state, we call it a *Phase Oracle*

Phase Oracle will be useful in Quantum Algorithms

No-Cloning Theorem

In standard computer, we can simply read a value and copy it to other location/register.

But, in quantum computer, if we have a state, $Let\ |\psi> = \alpha|0> +\beta|1>$

And if we don't know the value of α and β, then we can not copy the state of to other qubit.

Deutch's Algorithm

Deutch's Algorithm

What problem does the Deutch's Algorithm solve?

Let’s say, we have a function f that takes {0,1} as input and {0,1} as output.
f : {0, 1} → {0, 1}

Our task is to find out, whether the function f is balance or constant?

If a function is constant, it returns the same output, irrespective of the input: $f(0) = f(1)$

Constant 1	
x	f(x)
0	1
1	1

Constant 0	
x	f(x)
0	0
1	0

Balance function returns 0 for the half of the inputs and 1 for the other half of the inputs.

Balance (NOT)	
X	f(x)
0	1
1	0

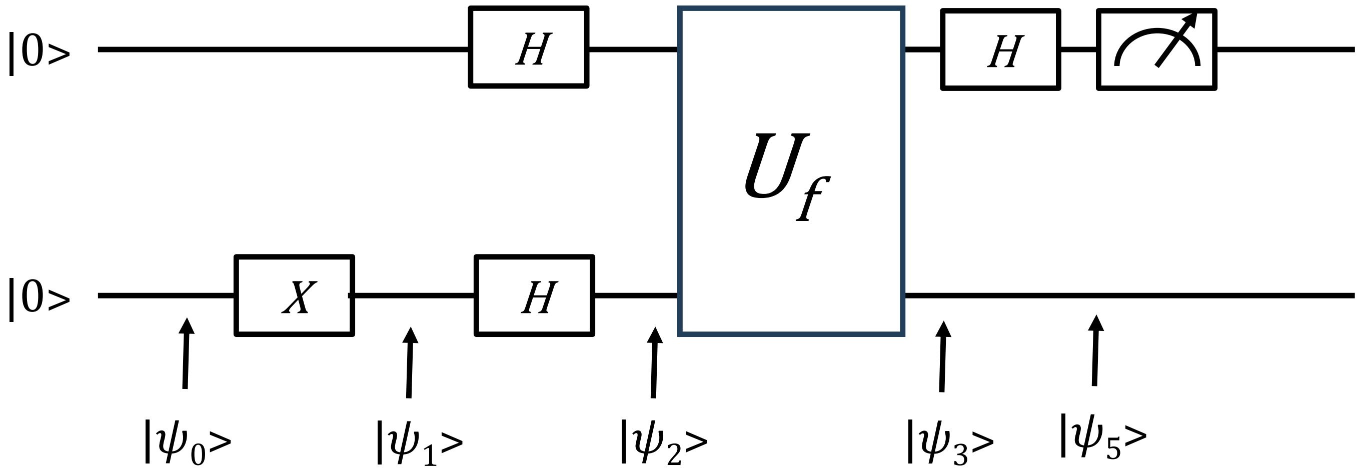
Balance (Identity)	
X	f(x)
0	0
1	1

Deutch's Algorithm

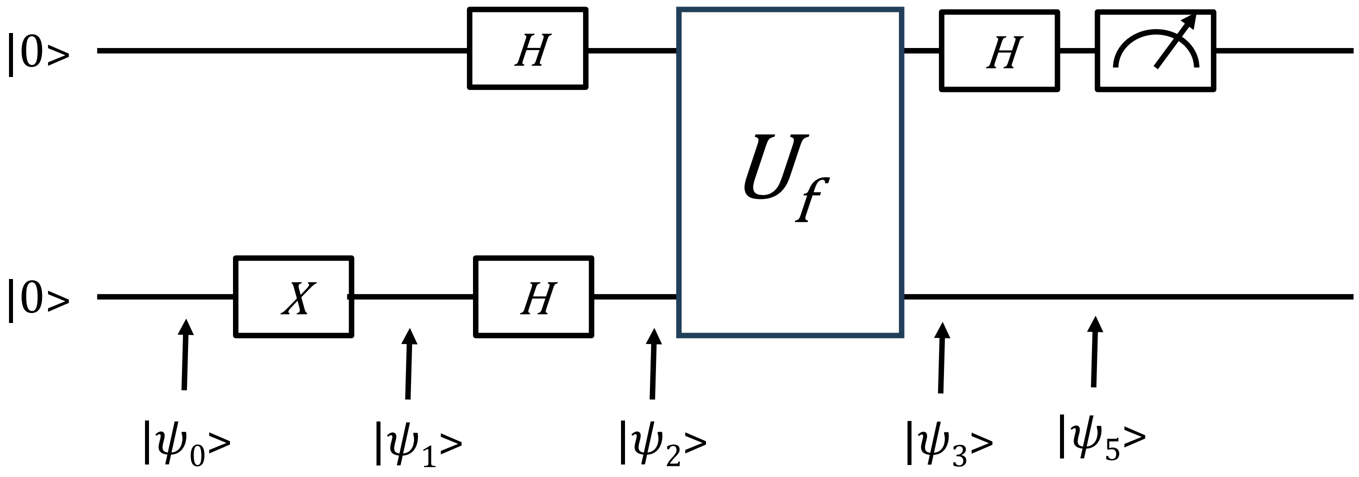
To check whether the function is constant or balance, we need to query the oracle twice. Once with 0 as input and once with 1 as input.

If f(0) = f(1), the function is constant
If f(0) ≠ f(1), the function is balance

With quantum computers, and with Deutch's algorithm, we only need one query of the function to find out whether the function is constant or balance!



Deutch's Algorithm



$$|\psi_0> = |00>$$

$$|\psi_1> = |01>$$

$$|\psi_2> = |+->$$

$$|\psi_2> = \frac{1}{\sqrt{2}}(|0> - |1>)|->$$

$$|\psi_2> = \frac{1}{\sqrt{2}}(|0>|-> - |1>|->)$$

$$|\psi_3> = U_f\frac{1}{\sqrt{2}}(|0>|-> - |1>|->)$$

$$|\psi_3> = \frac{1}{\sqrt{2}}(U_f|0>|-> - U_f|1>|->)$$

Recall the Phase Oracle, $U_f\,|x>|-> = (-1)^{f(x)}|x>|->$

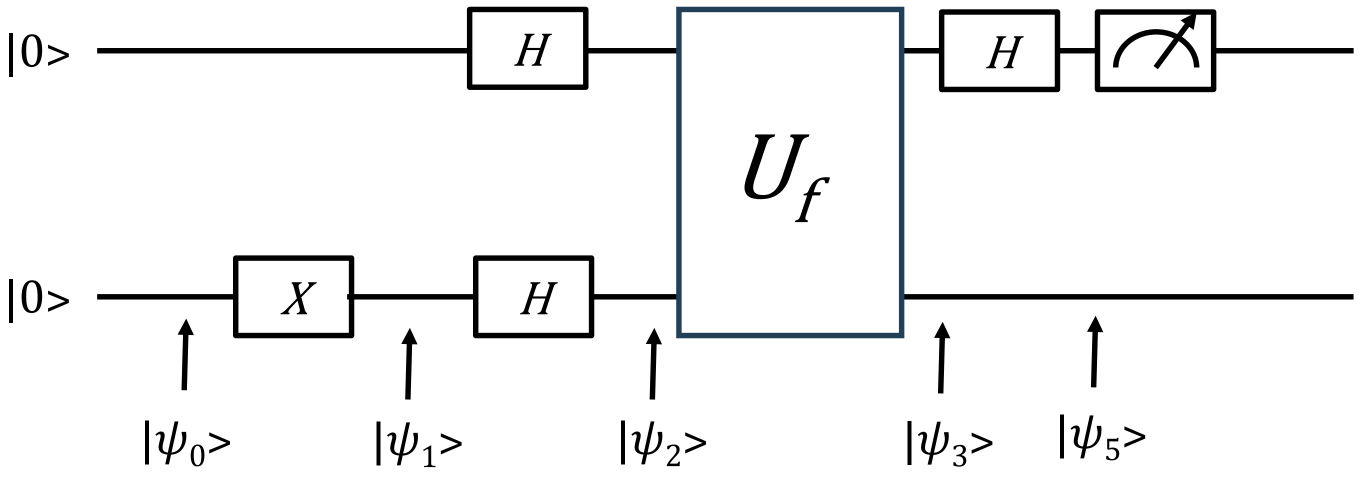
$$|\psi_3> = \frac{1}{\sqrt{2}}((-1)^{f(0)}|0>|-> + (-1)^{f(1)}|1>|->)$$

We may ignore the |->

$$|\psi_3> = \frac{1}{\sqrt{2}}((-1)^{f(0)}|0> + (-1)^{f(1)}|1>)$$

$|\psi_3> = \frac{1}{\sqrt{2}}((-1)^{f(0)}|0> + (-1)^{f(1)}|1>)$

Deutch's Algorithm



$Scenario\ 1: f(0) = f(1)$

$Scenario\ 1\ (Case\ 1): f(0) = f(1) = 0$

$|\psi_3> = \frac{1}{\sqrt{2}}(|0> + |1>)$

$Scenario\ 1\ (Case\ 2): f(0) = f(1) = 1$

$|\psi_3> = \frac{1}{\sqrt{2}}(-|0> - |1>)$

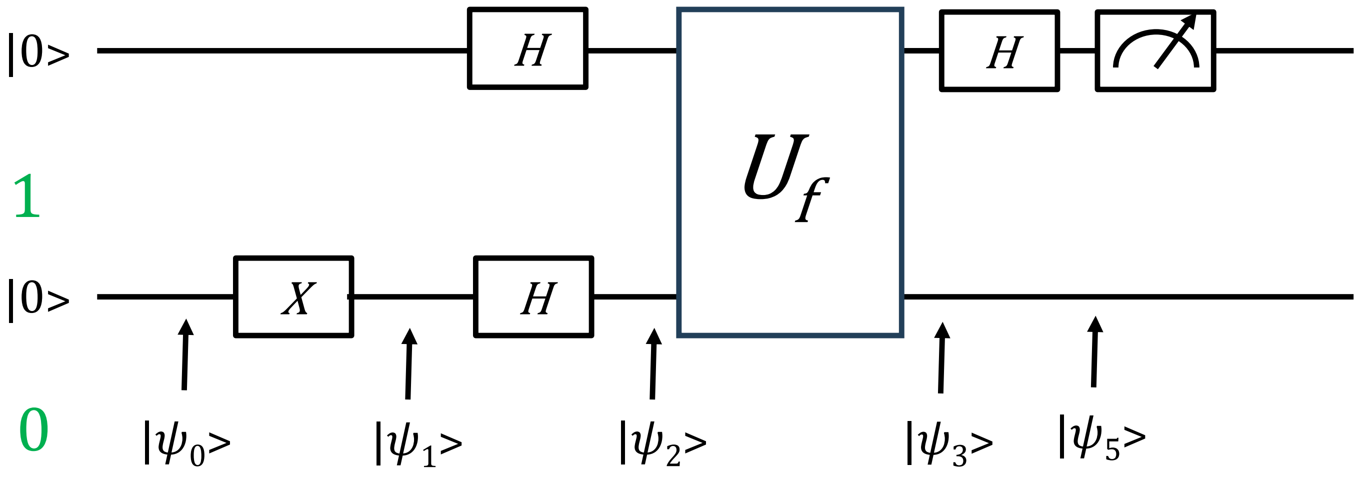
$We\ can\ factor\ out\ the\ global\ phase\ of\ |->$

$|\psi_3> = \frac{1}{\sqrt{2}}(|0> + |1>)$

$That\ means, if\ f(0) = f(1), the\ state\ becomes \frac{1}{\sqrt{2}}(|0> + |1>)$ which is $|+>$

$|\psi_3> = \frac{1}{\sqrt{2}}((-1)^{f(0)}|0> + (-1)^{f(1)}|1>)$

Deutch's Algorithm



$Scenario\ 2: f(0) \neq f(1)$

$Scenario\ 2\ (Case\ 1): f(0) = 0, f(1) = 1$

$$|\psi_3> = \frac{1}{\sqrt{2}}(|0> - |1>)$$

$Scenario\ 2\ (Case\ 2): f(0) = 1, f(1) = 0$

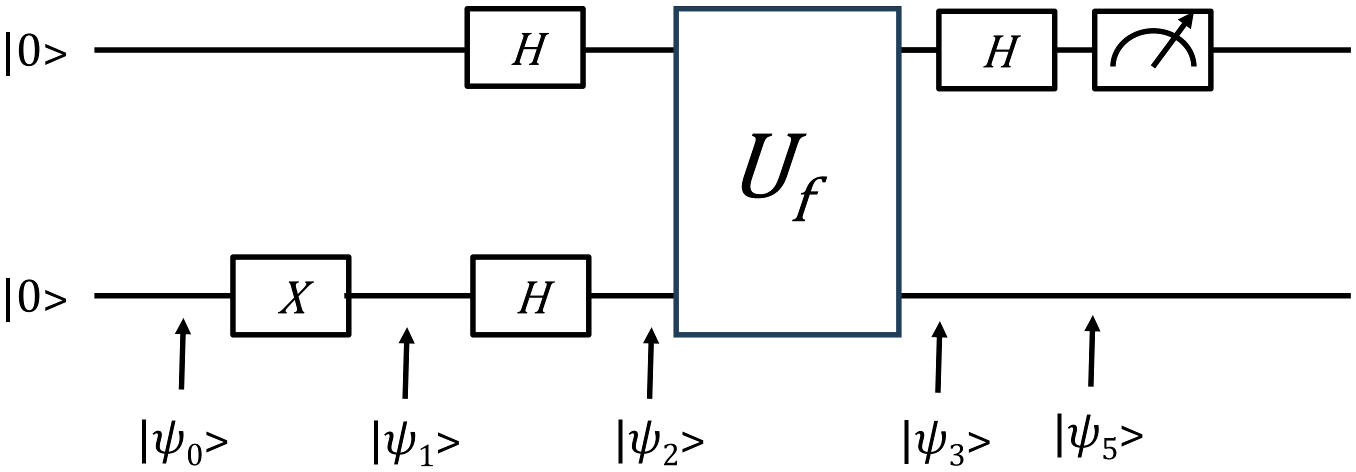
$$|\psi_3> = \frac{1}{\sqrt{2}}(-|0> + |1>)$$

$We\ can\ factor\ out\ the\ global\ phase\ of\ |->$

$$|\psi_3> = \frac{1}{\sqrt{2}}(|0> - |1>)$$

$That\ means, if\ f(0) \neq f(1), the\ state\ becomes \frac{1}{\sqrt{2}}(|0> - |1>)\ \text{which is}\ |->$

Deutch's Algorithm



$$f(0) = f(1)$$

$$|\psi_3> = \frac{1}{\sqrt{2}}(|0> + |1>) = |+>$$

$$|\psi_4> = \mathrm{H}|\psi_3> = \mathrm{H}|+> = |0>$$

That means, if we measure ($|\psi_4>$) as 0, the function is constant

$$f(0) \neq f(1)$$

$$|\psi_3> = \frac{1}{\sqrt{2}}(|0> - |1>) = |->$$

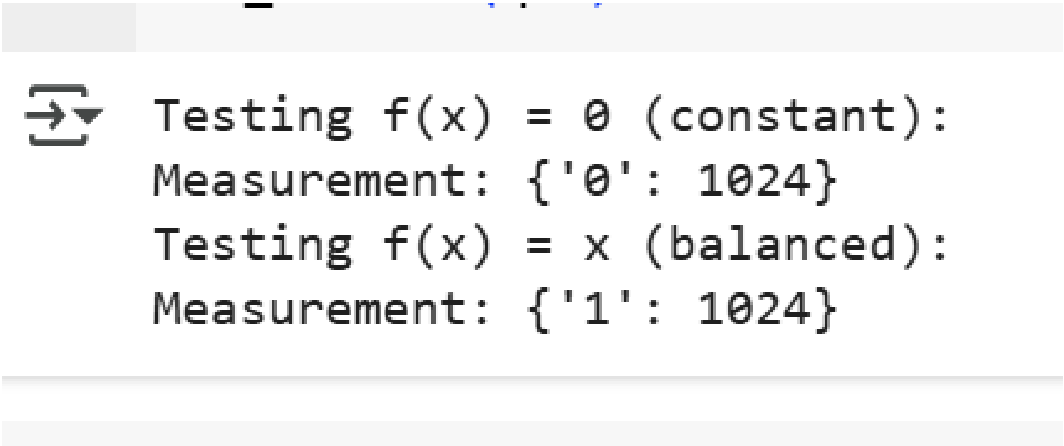
$$|\psi_4> = \mathrm{H}|\psi_3> = \mathrm{H}|-> = |1>$$

That means, if we measure ($|\psi_4>$) as 0, the function is balance.

```
from qiskit import QuantumCircuit
from qiskit_aer import AerSimulator
from qiskit.compiler import transpile
import matplotlib.pyplot as plt
from qiskit.visualization import plot_histogram

# === Oracles ===
def oracle_constant_0(qc):
    # f(x) = 0 → do nothing
    pass

def oracle_constant_1(qc):
    # f(x) = 1 → apply X to output qubit
    qc.x(1)

def oracle_balanced_x(qc):
    # f(x) = x → apply CNOT
    qc.cx(0, 1)

def oracle_balanced_not_x(qc):
    # f(x) = ¬x → apply X-CNOT-X
    qc.x(0)
    qc.cx(0, 1)
    qc.x(0)
```

```
# === Deutsch's Algorithm Circuit ===

def deutsch_algorithm(oracle_function):
    qc = QuantumCircuit(2, 1)

    # Initialize input: |0⟩|1⟩
    qc.x(1)

    # Apply Hadamard to both qubits
    qc.h(0)
    qc.h(1)

    # Apply oracle
    oracle_function(qc)

    # Hadamard again on input qubit
    qc.h(0)

    # Measure input qubit
    qc.measure(0, 0)

    return qc
```

```python
# === Run the circuit using AerSimulator (no execute) ===

def run_circuit(qc):
    backend = AerSimulator()
    tqc = transpile(qc, backend)
    result = backend.run(tqc, shots=1024).result()
    counts = result.get_counts()
    print("Measurement:", counts)
    plot_histogram(counts)
    plt.show()

# === Test ===

print("Testing f(x) = 0 (constant):")
qc1 = deutsch_algorithm(oracle_constant_0)
run_circuit(qc1)

print("Testing f(x) = x (balanced):")
qc2 = deutsch_algorithm(oracle_balanced_x)
run_circuit(qc2)
```



```
Testing f(x) = 0 (constant):
Measurement: {'0': 1024}
Testing f(x) = x (balanced):
Measurement: {'1': 1024}
```

Deutch-Jozsa Algorithm

Deutch-Jozsa Algorithm

Deutch-Jozsa algorithm does the same thing that Deutch's Algorithm does, that is to check whether the function is constant or balance

Except, Deutch-Jozsa algorithm works on any number of qubits.

Deutch’s Algorithm: $f : \{0, 1\} \rightarrow \{0, 1\}$
Deutch-Jozsa Algorithm: $f : \{0, 1\}^n \rightarrow \{0, 1\}$

Deutch-Jozsa Algorithm

Constant and Balance Function Revisited:

Constant ZERO	
x	f(x)
000	0
001	0
010	0
011	0
100	0
101	0
110	0
111	0

Constant ZERO	
x	f(x)
000	1
001	1
010	1
011	1
100	1
101	1
110	1
111	1

Balance	
x	f(x)
000	0
001	1
010	1
011	0
100	0
101	1
110	1
111	0

To determine whether f is constant or balanced takes a classical computer at worst $2^{n-1} + 1$ queries of f, where n is number of bits.

With a quantum computer, we just need one query, to determine whether the function is constant or balance.

Deutch-Jozsa Algorithm

$$|\psi_0> = |0>^{\otimes n} |->$$

$$|\psi_1> = H^{\otimes n}|0>^{\otimes n} |->$$

$H^{\otimes n}|0>^{\otimes n} = H|0> H|0> H|0>$......n times

$H^{\otimes n}|0>^{\otimes n} = |+> |+> |+>$......n times

$= \frac{1}{\sqrt{2}}(|0> + |1>) \frac{1}{\sqrt{2}}(|0> + |1>)$......n times

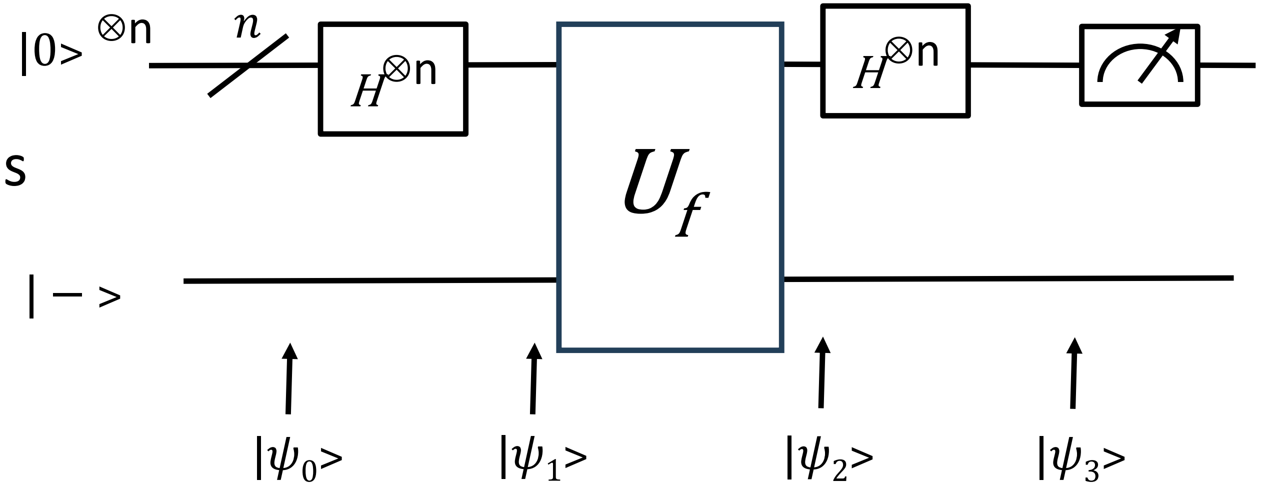
$= \frac{1}{\sqrt{2}}(|0> + |1>) \frac{1}{\sqrt{2}}(|0> + |1>)$......n times

For n=2

$$H^{\otimes 2}|0>^{\otimes 2} = \frac{1}{\sqrt{2}}(|0> + |1>) \frac{1}{\sqrt{2}}(|0> + |1>)$$

$$= \frac{1}{\sqrt{2^2}}(|00> + |01> + |10> + |11>)$$

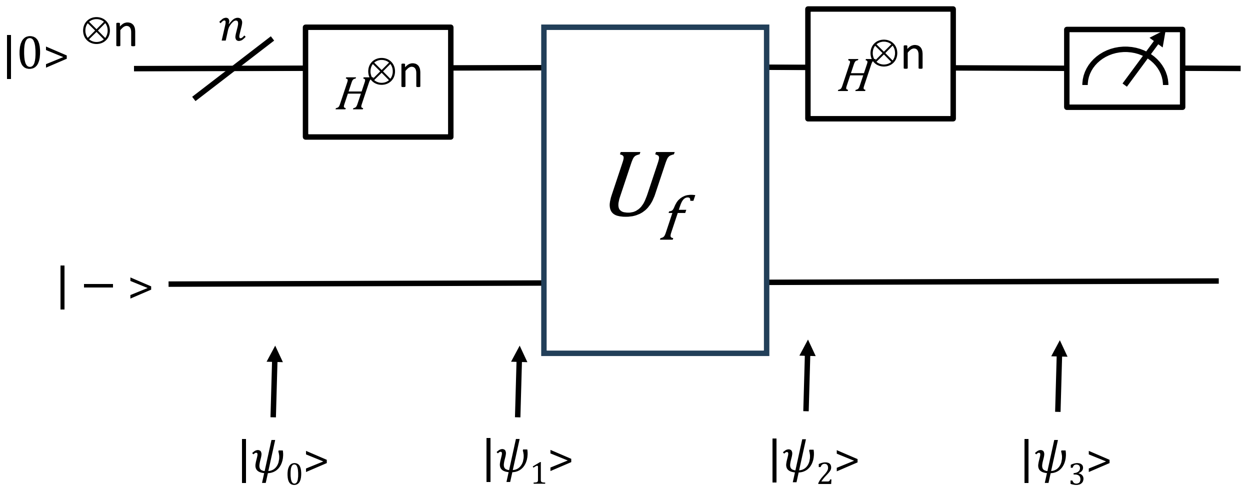
$$H^{\otimes 2}|0>^{\otimes 2} = \frac{1}{\sqrt{2^2}} \sum_{x \in \{0,1\}^2} |x>$$



For n=3 $\quad H^{\otimes 3}|0>^{\otimes 3} = \frac{1}{\sqrt{2^3}} \sum_{x \in \{0,1\}^3} |x>$

For any n $\quad \boxed{H^{\otimes n}|0>^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x>}$

Deutch-Jozsa Algorithm



$$|\psi_1> = H^{\otimes n}|0>^{\otimes n} | - >$$

$$|\psi_1> = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x> | - >$$

$$|\psi_2> = \mathrm{U_f}\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x> | - >$$

$$|\psi_2> = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} \mathrm{U_f}|x> | - >$$

Recall the Phase Oracle, $\mathrm{U_f}$ |x> |-> = $(-1)^{\mathrm{f(x)}}|\mathrm{x}>|->$

$$|\psi_2> = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{\mathrm{f(x)}}|\mathrm{x}>|->$$

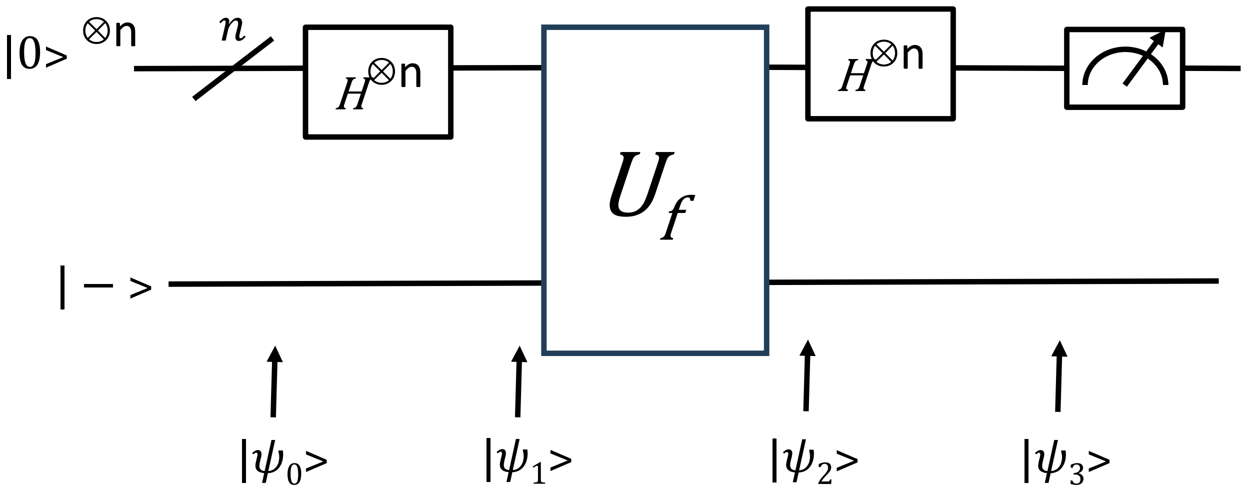
We may ignore the |->

$$|\psi_2> = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{\mathrm{f(x)}}|\mathrm{x}>$$

Deutch-Jozsa Algorithm

$$|\psi_3> = H^{\otimes n}\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x>$$

$$|\psi_3> = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}H^{\otimes n}|x>$$



$$H^{\otimes n}|x> = \mathrm{H}|x_0x_1x_2 \ldots .xn_{-1}>$$
$$= \mathrm{H}|x_0>\mathrm{H}|x_1>\mathrm{H}|x_2>\cdots.\mathrm{H}|x_{n-1}>$$

$H|xi> = \frac{1}{\sqrt{2}}(|0> + (-1)^{xi}|1), where\ xi$
$\in \{0, 1\}$

$$H|0> = \frac{1}{\sqrt{2}}(|0> + |1) = |+>$$

$$H|1> = \frac{1}{\sqrt{2}}(|0> - |1) = |->$$

$$\boxed{H^{\otimes n}|x> = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{x.z)}|z>}$$

(Derivation skipped)

Deutch-Jozsa Algorithm

$$|\psi_3> = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n}(-1)^{f(x)} H^{\otimes\ n}|x>$$

$$|\psi_3> = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n}(-1)^{f(x)} \frac{1}{\sqrt{2^n}}\ \sum_{x\in\{0,1\}^n}(-1)^{x.z)}|z>$$

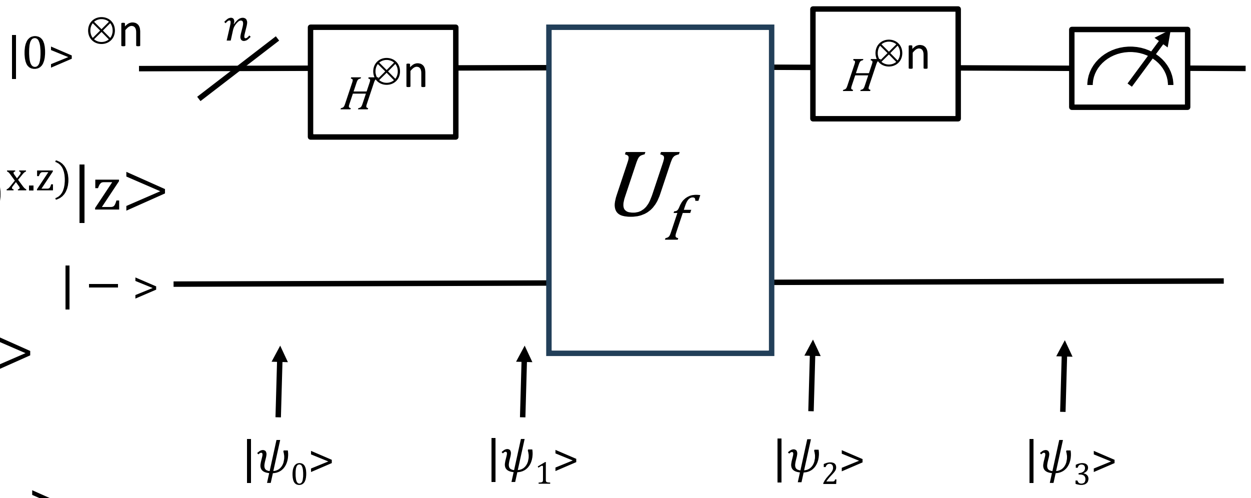
$$|\psi_3> = \sum_{x\in\{0,1\}^n} \frac{1}{2^n} \sum_{x\in\{0,1\}^n}(-1)^{f(x)}(-1)^{x.z)}|z>$$

$$|\psi_3> = \sum_{x\in\{0,1\}^n} \frac{1}{2^n} \sum_{x\in\{0,1\}^n}(-1)^{f(x)}(-1)^{x.z)}\ |z>$$

$$|\psi_3> = \sum_{x\in\{0,1\}^n} \frac{1}{2^n} \sum_{x\in\{0,1\}^n}(-1)^{f(x)\ +\ x.z}\ |z>$$

$$|\psi_3> = \sum_{x\in\{0,1\}^n} \frac{1}{2^n} \sum_{x\in\{0,1\}^n}(-1)^{f(x)\ +\ x.(0.0.0.0.0\ \cdots\ 0)}\ |z>$$

$$|\psi_3> = \sum_{x\in\{0,1\}^n} \frac{1}{2^n} \sum_{x\in\{0,1\}^n}(-1)^{f(x)}\ |z>$$



If the function f is constant, the value of f(x) would be same, irrespective of input

Deutch-Jozsa Algorithm

If the function f is constant, the value of f(x) would be same, irrespective of input

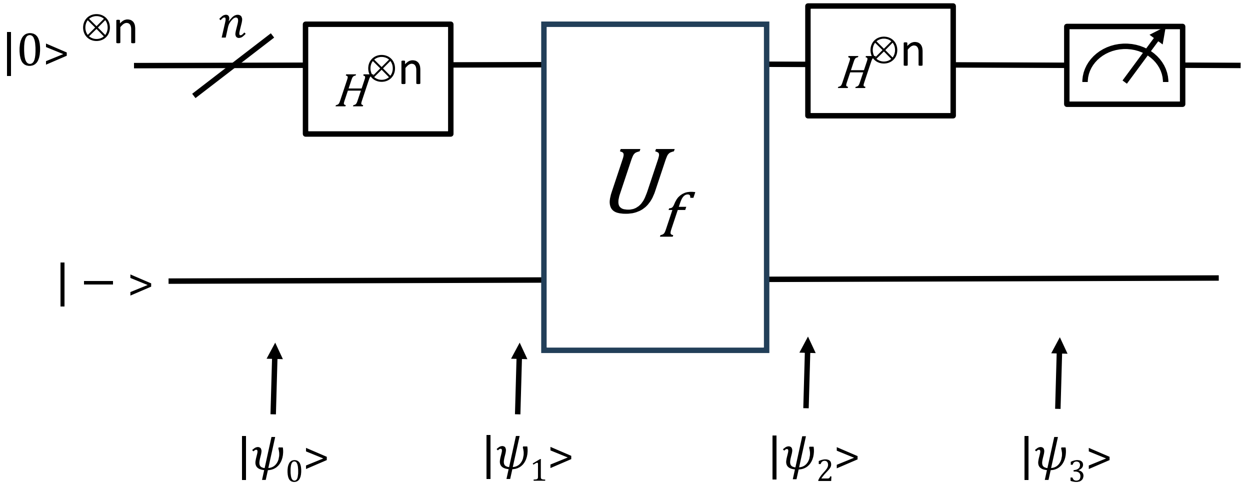
$$\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{f(x)} = \frac{1}{2^n}\sum_{x\in\{0,1\}^n} 1 \text{ ,if f(x) = 0 for all x}$$

$$\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{f(x)} = \frac{1}{2^n} 2^n\text{,if f(x) = 0 for all x}$$

$$\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{f(x)} = 1\text{, if f(x) = 0 for all x}$$

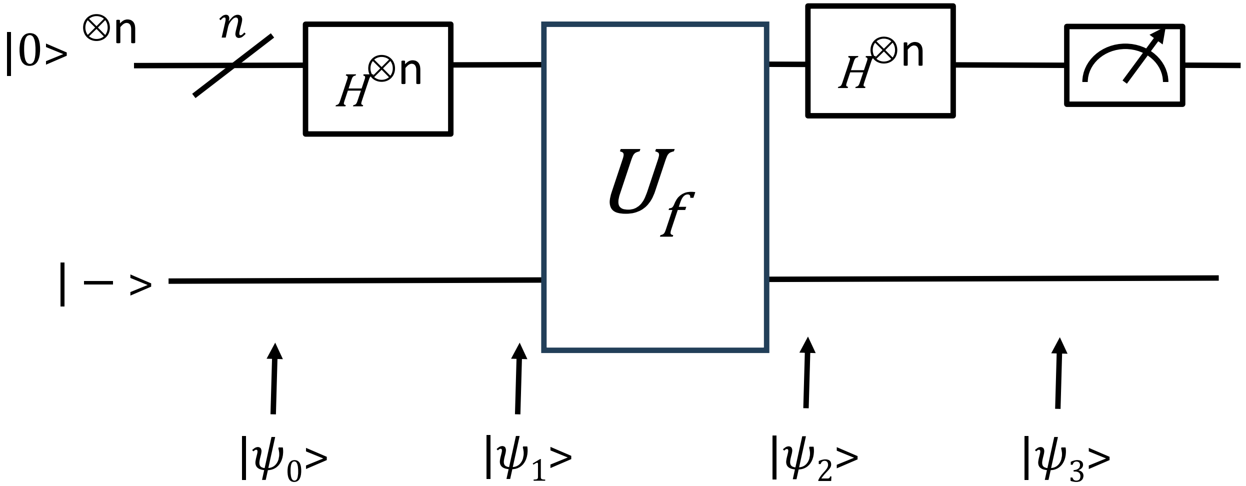
$$\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{f(x)} = -1\text{, if f(x) = 1 for all x}$$

If the function f is constant, Amplitude is ± 1



Deutch-Jozsa Algorithm

If the function f is balance, half the output would result into 0 and other in 1.



$$\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{\text{f(x)}} = \frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{\text{f(x)}}$$

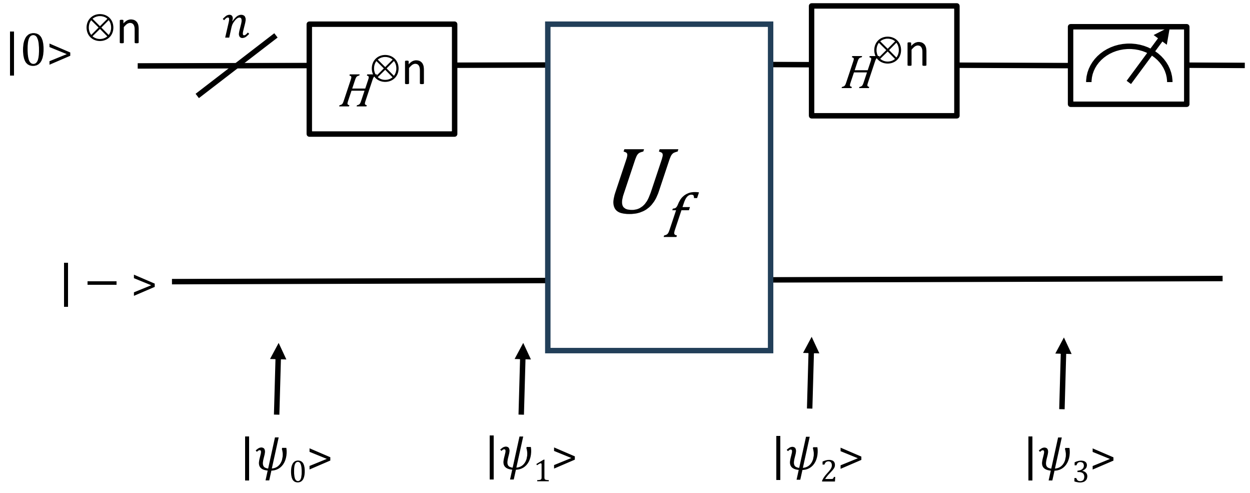
$$\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{\text{f(x)}} = \frac{1}{2^n}\left((-1)^0+(-1)^1++(-1)^0+(-1)^1+\cdots(-1)^1\right)$$

$$\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{\text{f(x)}} = \frac{1}{2^n}(0)$$

$$\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{\text{f(x)}} = 0$$

If the function f is balance, Amplitude is 0

Deutch-Jozsa Algorithm



If the function is constant, Amplitude = ± 1

That is probability of measuring |0000…0> is 1 (100%)

If we measure |0000…0> state then f is constant.

If the function is constant, Amplitude = 0

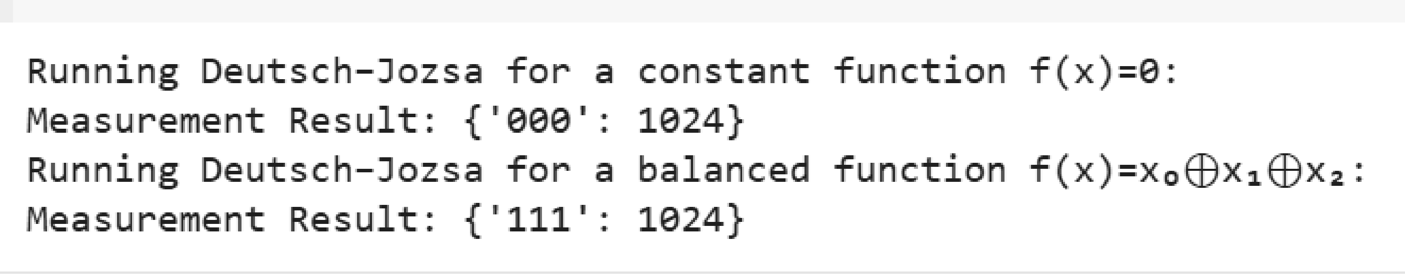
That is probability of measuring |0000…0> is 0 (0%)

If we measure any other state then f is balanced.

```
from qiskit_aer import Aer
from qiskit import QuantumCircuit
from qiskit.compiler import transpile
from qiskit_aer import AerSimulator
from qiskit.visualization import plot_histogram
import matplotlib.pyplot as plt

# === Oracles ===

def constant_oracle(qc, n, output=0):
    if output == 1:
        qc.x(n)  # Flip the ancilla if f(x)=1 always

def balanced_oracle(qc, n):
    # f(x) = $x_0$ XOR $x_1$ XOR ... XOR $x_{n-1}$
    for qubit in range(n):
        qc.cx(qubit, n)
```

```
# === Deutsch-Jozsa Algorithm ===

def deutsch_jozsa(n, oracle_func, oracle_type="balanced"):
    qc = QuantumCircuit(n + 1, n)

    # Initialize input |0⟩^n and ancilla |1⟩
    qc.x(n)
    qc.h(range(n + 1))

    # Apply oracle
    oracle_func(qc, n)

    # Apply Hadamard on first n qubits
    qc.h(range(n))

    # Measure first n qubits
    qc.measure(range(n), range(n))

    return qc
```

```python
# === Run the circuit ===

def run_circuit(qc):
    backend = AerSimulator()
    tqc = transpile(qc, backend)
    result = backend.run(tqc, shots=1024).result()
    counts = result.get_counts()
    print("Measurement Result:", counts)
    plot_histogram(counts)
    plt.show()

# === Run for both cases ===

print("Running Deutsch-Jozsa for a constant function f(x)=0:")
qc_const = deutsch_jozsa(n=3, oracle_func=lambda qc, n: constant_oracle(qc, n,
output=0))
run_circuit(qc_const)

print("Running Deutsch-Jozsa for a balanced function f(x)=x₀⊕x₁⊕x₂:")
qc_bal = deutsch_jozsa(n=3, oracle_func=balanced_oracle)
run_circuit(qc_bal)
```



```
Running Deutsch-Jozsa for a constant function f(x)=0:
Measurement Result: {'000': 1024}
Running Deutsch-Jozsa for a balanced function f(x)=x₀⊕x₁⊕x₂:
Measurement Result: {'111': 1024}
```

Bernstein-Vazirani Algorithm

Bernstein-Vazirani Algorithm

Imagine we have a function f: $\{0, 1\}^n \rightarrow \{0, 1\}$

f (x) = x . s (mod2)

x is a bit string of length n and s is a secret string of same size. This function returns either 0 or 1. Our task is to find out secret string s.

(mod2) mean we divide the answer of x . s by 2 and take the reminder
For instance, if x . s = 5, x . s (mod2) = 1

Bernstein-Vazirani Algorithm

In Classical Computing,

$f\ (0000....01) = s_0(0) + s_1(0) + s_2\ (0) + s_3\ (0) + ... + s_{n-2}\ (0) + s_{n-1}\ (1) = s_{n-1}$
$f\ (0000....10) = s_0(0) + s_1(0) + s_2\ (0) + s_3\ (0) + ... + s_{n-2}\ (1) + s_{n-1}\ (0) = s_{n-2}$

.....

.....

$f\ (0001....00) = s_0(0) + s_1(0) + s_2\ (0) + s_3\ (1) + ... + s_{n-2}\ (0) + s_{n-1}\ (0) = s_3$
$f\ (0010....00) = s_0(0) + s_1(0) + s_2\ (1) + s_3\ (0) + ... + s_{n-2}\ (0) + s_{n-1}\ (0) = s_2$
$f\ (0100....00) = s_0(0) + s_1(1) + s_2\ (0) + s_3\ (0) + ... + s_{n-2}\ (0) + s_{n-1}\ (0) = s_1$
$f\ (1000....00) = s_0(1) + s_1(0) + s_2\ (0) + s_3\ (0) + ... + s_{n-2}\ (0) + s_{n-1}\ (0) = s_0$

where s is, $s_0s_1s_2s_3..............s_{n-2}s_{n-1}$

This means, we need to query the function n times.

Bernstein-Vazirani Algorithm

$$|\psi_0> = |0>^{\otimes n} \, |->$$

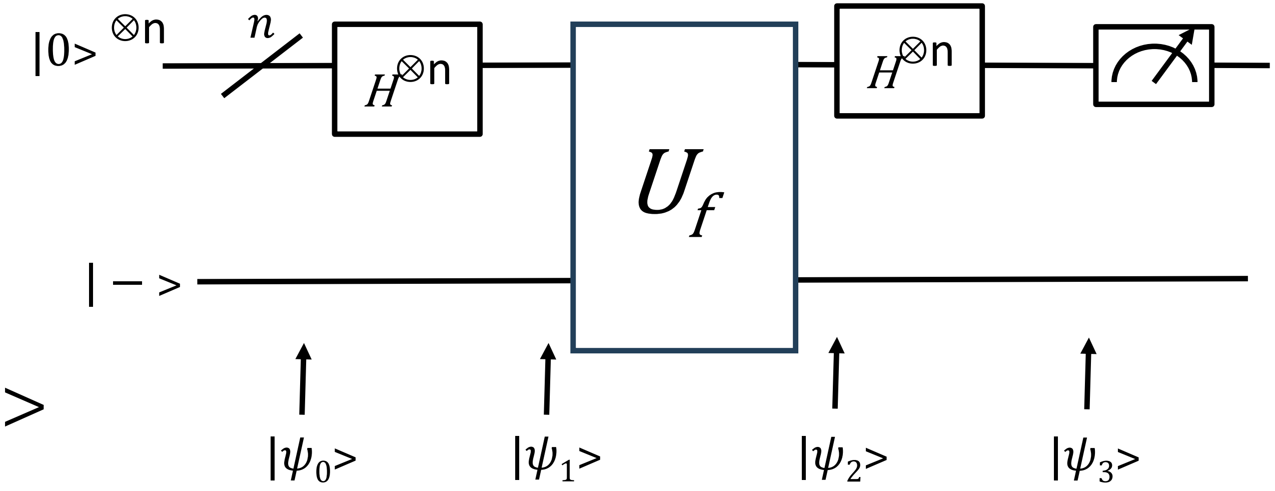
$$|\psi_1> = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x> \, |->$$

$$|\psi_2> = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{x.s} |x>$$

$$|\psi_3> = H^{\otimes n} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{x.s} |x>$$

$$|\psi_3> = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} \sum_{x\in\{0,1\}^n} (-1)^{x.s} (-1)^{x.z)} \, |z>$$

$$|\psi_3> = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} \sum_{x\in\{0,1\}^n} (-1)^{x(s\,+\,z)} \, |z>$$



+ is bitwise XOR

Same circuit as Deutch-Jozsa Algorithm

$$Amplitude\ of\ the\ |s>state:$$

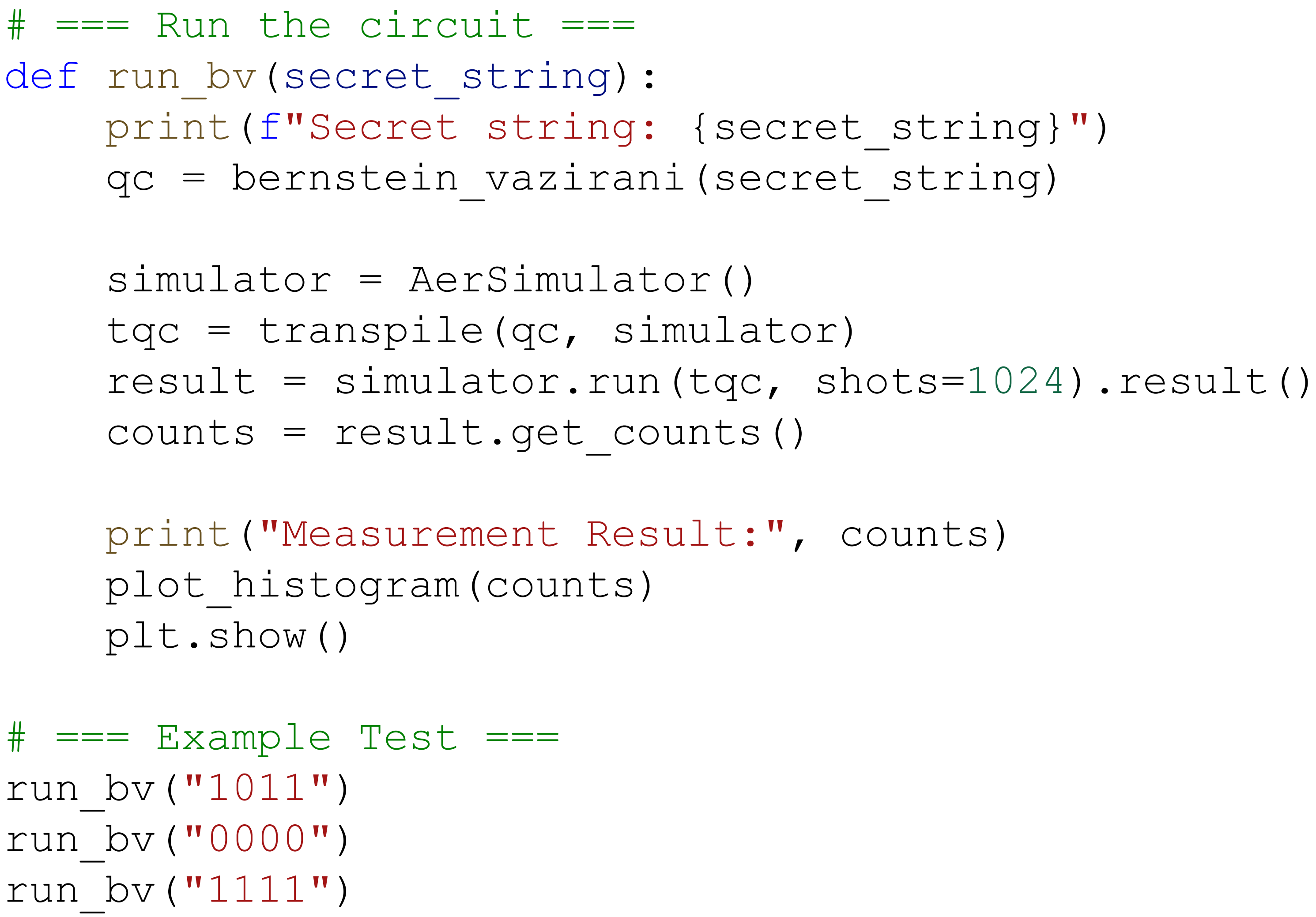
$$= \frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{(s+s).x} = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{(00..0).x} = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} 1 = \frac{1}{2^n} 2^n = \mathbf{1}$$

$$Amplitude\ of\ the\ |s>state = \mathbf{1}$$

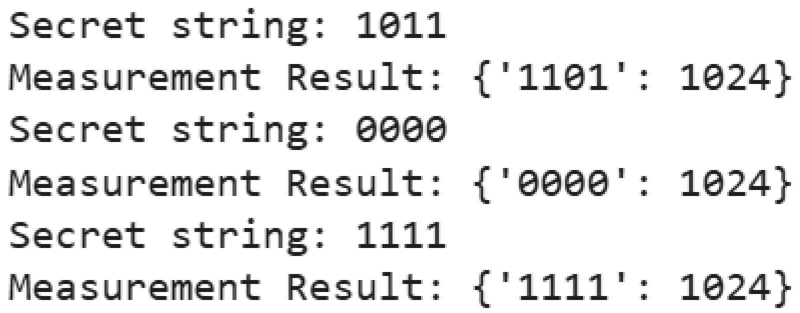
This means, the probability of measuring s after applying the algorithm is 1. That is it. After one query of the function, we can find s by measuring the qubits

```python
from qiskit_aer import Aer
from qiskit import QuantumCircuit
from qiskit_aer import AerSimulator
from qiskit.compiler import transpile
from qiskit.visualization import plot_histogram
import matplotlib.pyplot as plt

# === Oracle for Bernstein-Vazirani ===
def bv_oracle(qc, secret_string):
    n = len(secret_string)
    for i, bit in enumerate(secret_string):
        if bit == '1':
            qc.cx(i, n)  # Apply CNOT from each qubit to ancilla based on secret bit
```

```
# === Main BV Algorithm ===
def bernstein_vazirani(secret_string):
    n = len(secret_string)
    qc = QuantumCircuit(n + 1, n)

    # Step 1: Initialize last qubit (ancilla) to |1⟩
    qc.x(n)

    # Step 2: Hadamard on all qubits
    qc.h(range(n + 1))

    # Step 3: Apply Oracle U_f
    bv_oracle(qc, secret_string)

    # Step 4: Hadamard on first n qubits
    qc.h(range(n))

    # Step 5: Measure first n qubits
    qc.measure(range(n), range(n))

    return qc
```



```python
# === Run the circuit ===
def run_bv(secret_string):
    print(f"Secret string: {secret_string}")
    qc = bernstein_vazirani(secret_string)

    simulator = AerSimulator()
    tqc = transpile(qc, simulator)
    result = simulator.run(tqc, shots=1024).result()
    counts = result.get_counts()

    print("Measurement Result:", counts)
    plot_histogram(counts)
    plt.show()

# === Example Test ===
run_bv("1011")
run_bv("0000")
run_bv("1111")
```



```
Secret string: 1011
Measurement Result: {'1101': 1024}
Secret string: 0000
Measurement Result: {'0000': 1024}
Secret string: 1111
Measurement Result: {'1111': 1024}
```