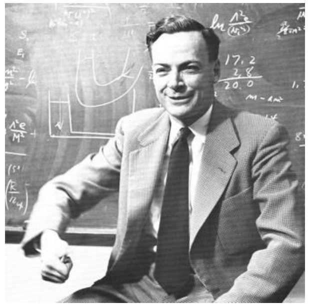
# Introduction to Quantum computing and quantum simulation

Chiara Capecci
Università degli studi di Trento
chiara.capecci@unitn.it

# Quantum computing: what is it?



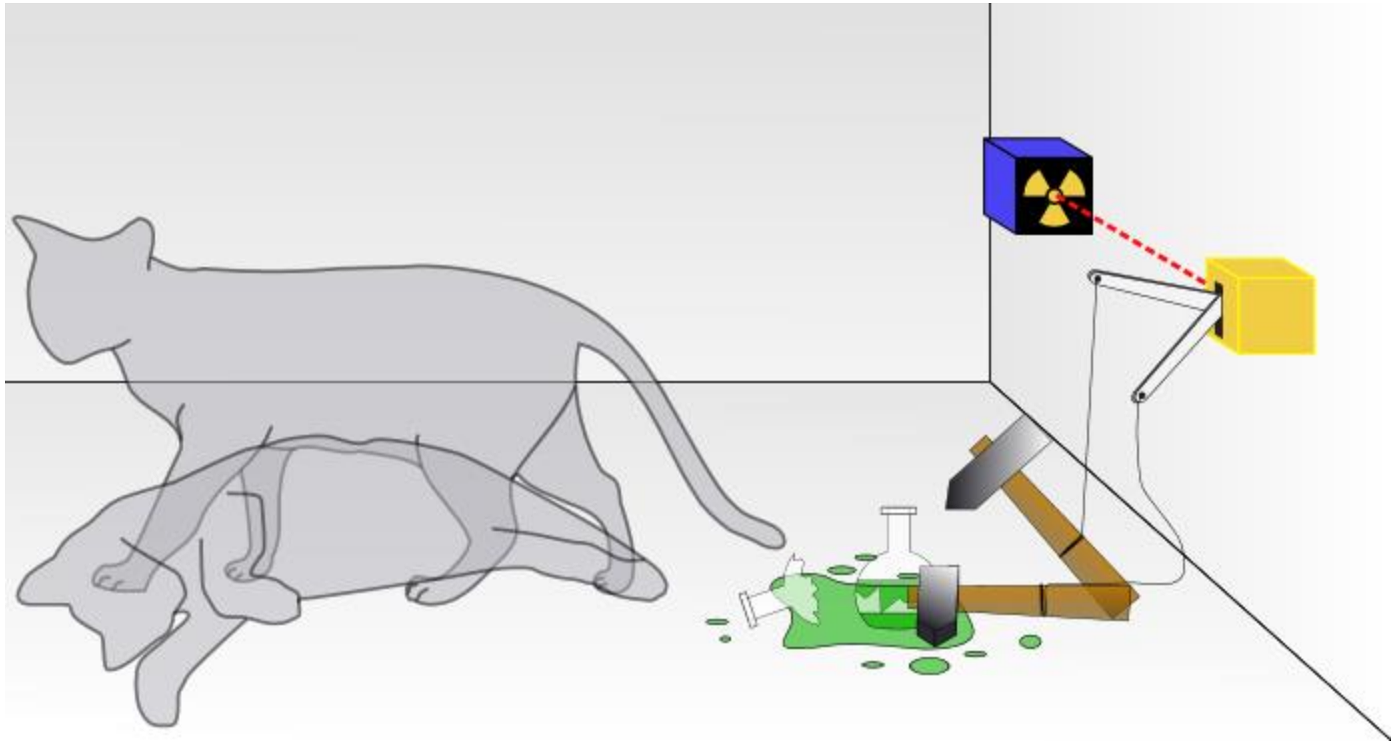
Richard Feynman’s famous quote[1]:
'Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical.'

A **quantum computer** is a device that can manipulate quantum systems to perform calculations, in the same way that a traditional computer manipulates bits.

Quantum computing leverages **quantum mechanics principles** to solve complex problems faster/ more efficiently than classical computers.

[1] **Feynman, Richard P.** *International Journal of Theoretical Physics*, vol. 21, no. 6-7, 1982, pp. 467-488.

# Quantum computing: what is it?



**Qubits**

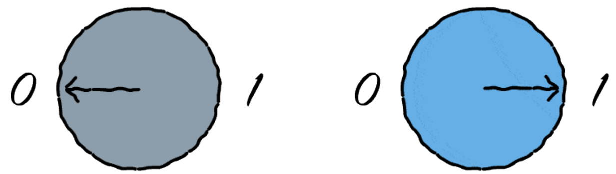
**Superposition**

**Entanglement**

**Measurement**

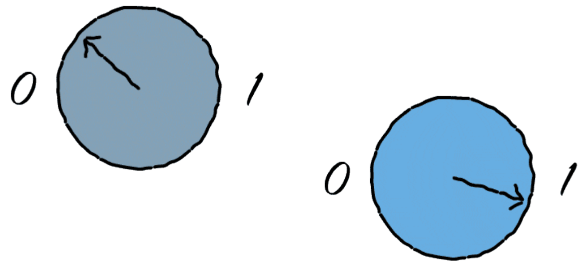
# Qubits and quantum properties

BIT




Possible values: 0 or 1

QUANTUM BIT or QUBIT

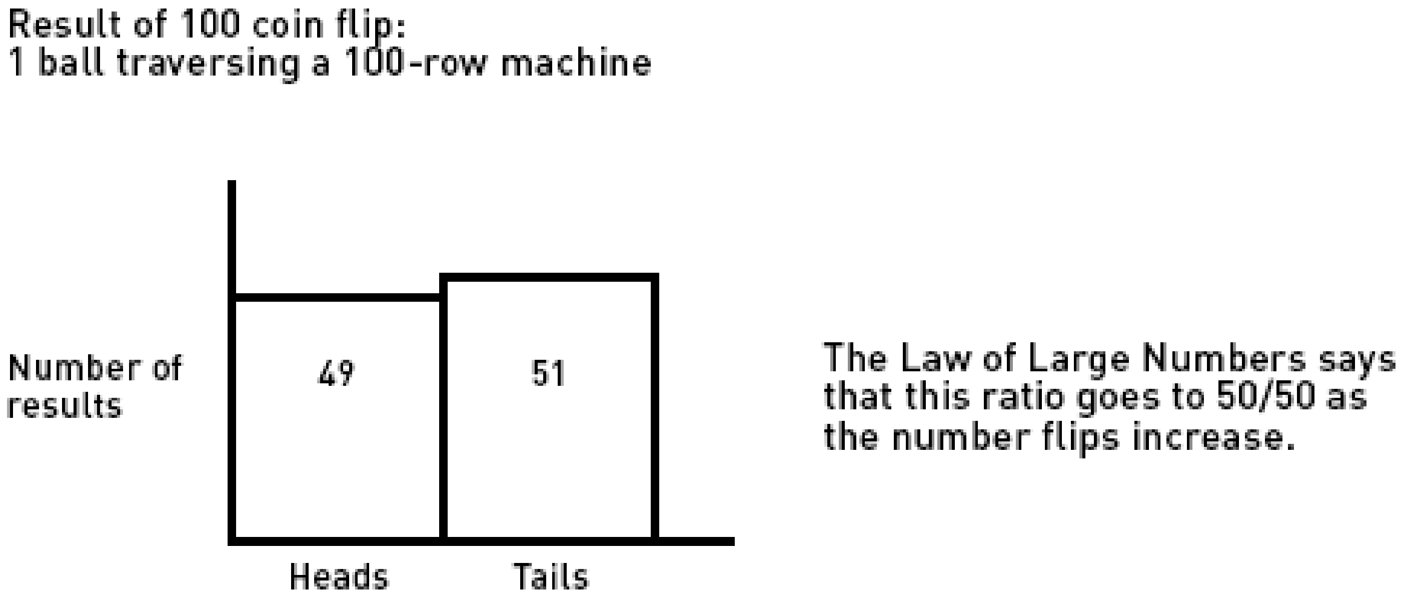



It can be in a **SUPERPOSITION** of 0 and 1

LATER IN DETAILS ...

# Qubits and quantum properties

**MEASUREMENTS**:
Flipping a coin to show heads and tails at once, with each side representing a state. Only when 'measured' does it show heads or tails.

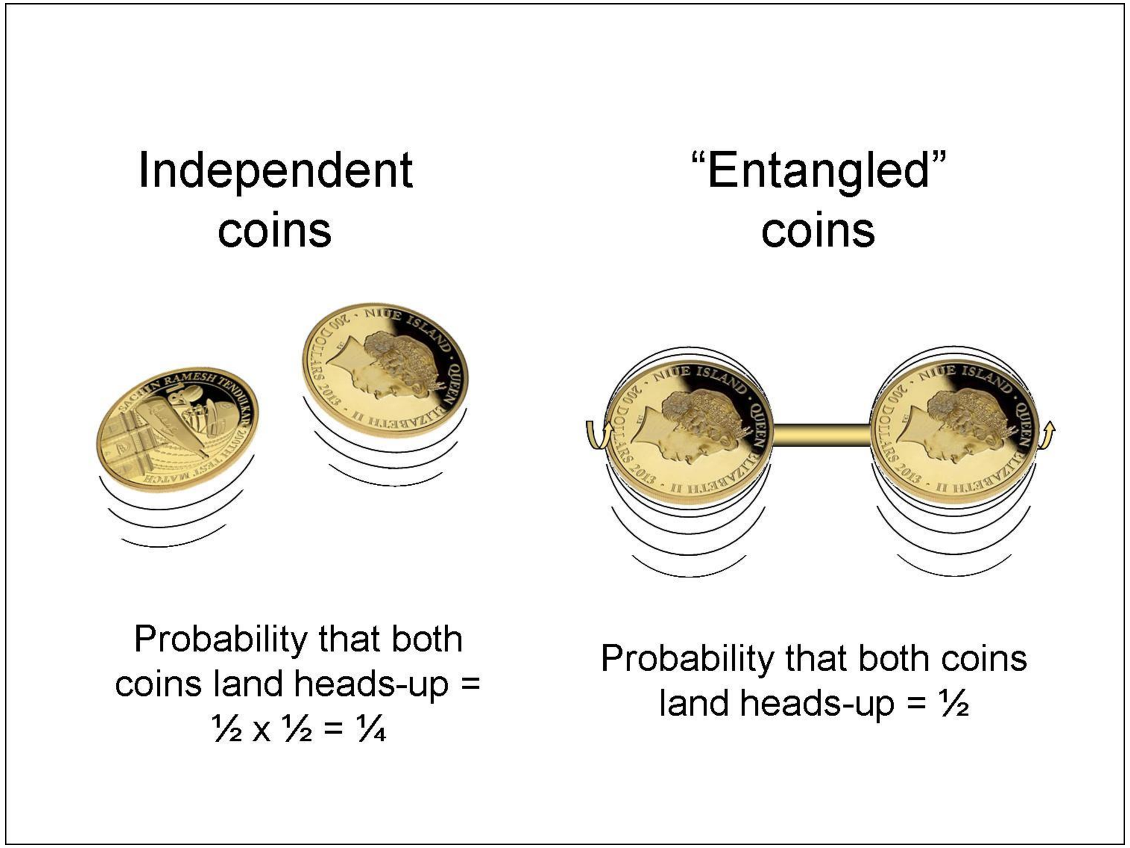


# Qubits and quantum properties

**ENTANGLEMENT:**

It is a phenomenon where two qubits are correlated, so the state of one qubit instantly affects the state of the other.

An entangled pair of coins, where the result of one determines the result of the other, even if separated by distance.



# Quantum computation: why?

## Classical Computing Limitations

A classical N-bit computer can only exist in one of $2^N$ possible states at a time.

0 1

exponential growth in the complexity of many computational problems
→limited processing power and scalability of classical devices.

## Quantum Computing Advantage

A quantum computer with N qubits can exist in a superposition of all $2^N$ states simultaneously.

0 0 + 0 1 + 1 0 + 1 1

it leverages superposition to perform operations on all $2^N$ states at once, enabling exponential speedup for certain problems.

**Applications of Quantum Computing**:
Quantum computing can address complex tasks in molecular simulations, optimizations, cryptography and machine learning

# Quantum computation: how?

Quantum computers based on **QUBITS** → require robust physical implementations:

desired evolution

prepared in specified initial states

ability to measure the final output

# Quantum computation: how?

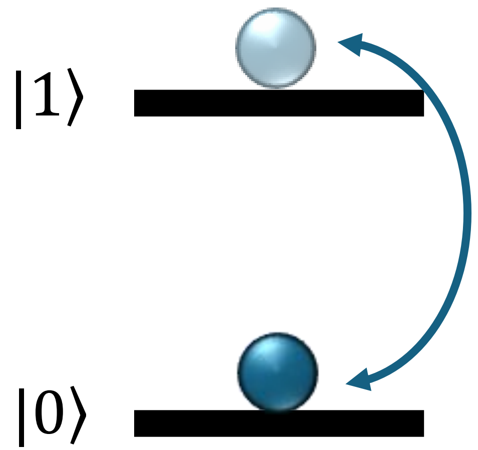
Quantum computers based on **QUBITS** → require robust physical implementations:

desired evolution

prepared in specified initial states

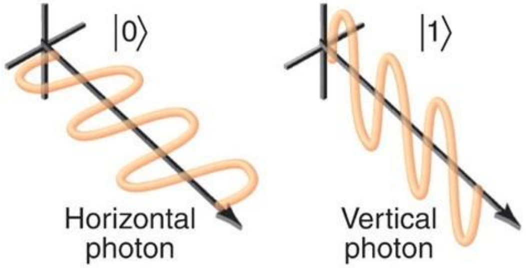
ability to measure the final output

two-level quantum systems

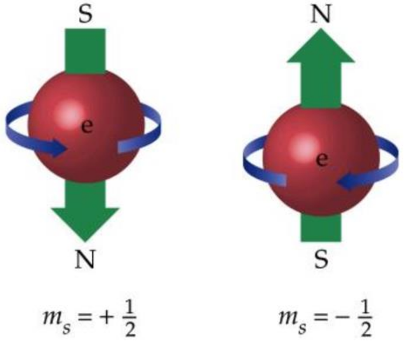


# Qubit: two-level sistem

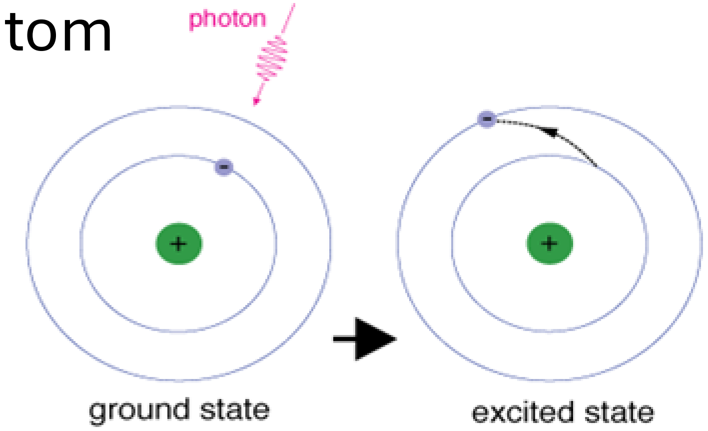
Photon polarization



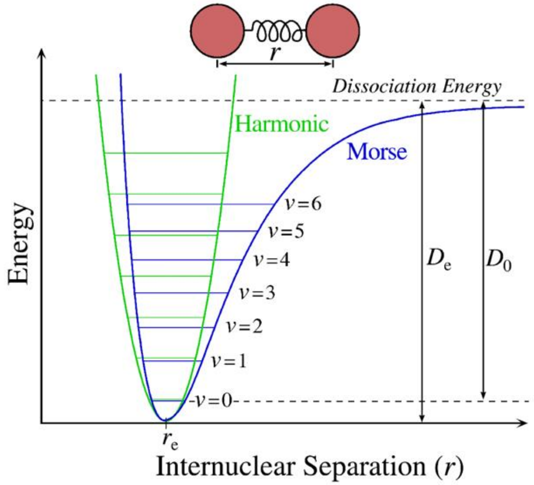

Electronic spin




Hydrogen atom

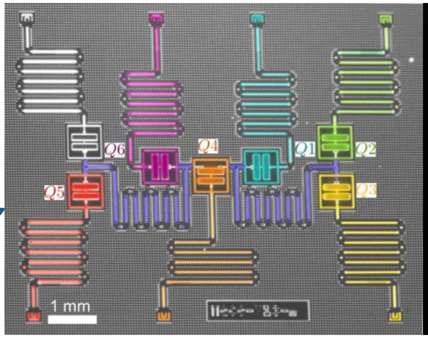


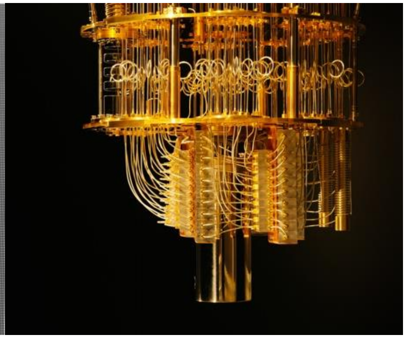

Harmonic oscillator



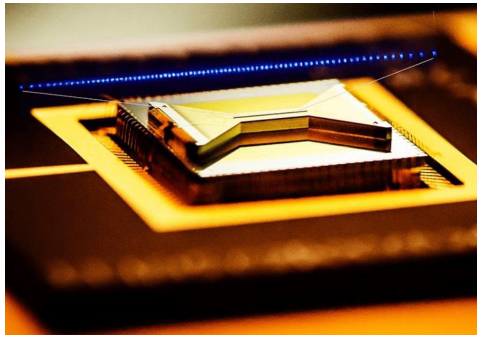
# Qubit: how to realize it?

Different types of quantum devices:



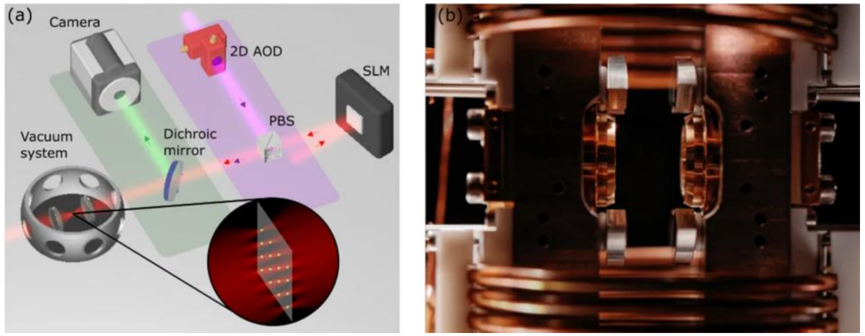



Superconductive e.g.: IBM, GOOGLE



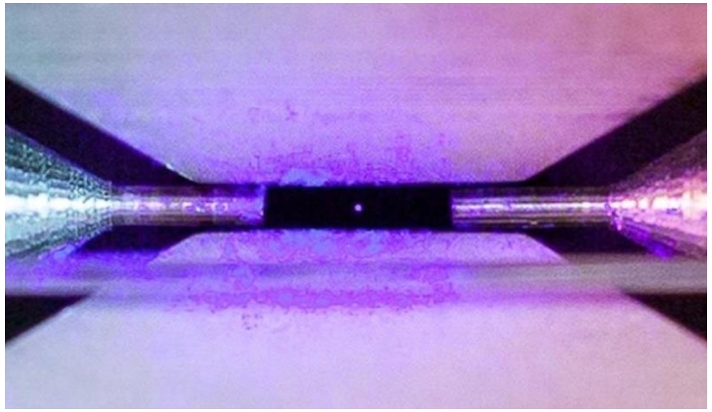
Ionic e.g.: IONQ

Neutral atoms e.g.: PASQAL




Photonic, logic qubits ...

# Qubit: how to realize it?
## TRAPPED ION DEVICES

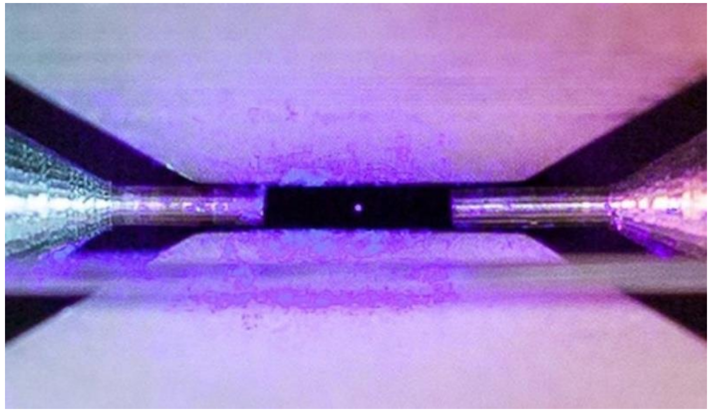


**ION = QUBIT**

IonQ: ytterbium ions ($Yb^+$)

Ions can be confined and suspended in free space using electromagnetic fields

# Qubit: how to realize it?
## TRAPPED ION DEVICES



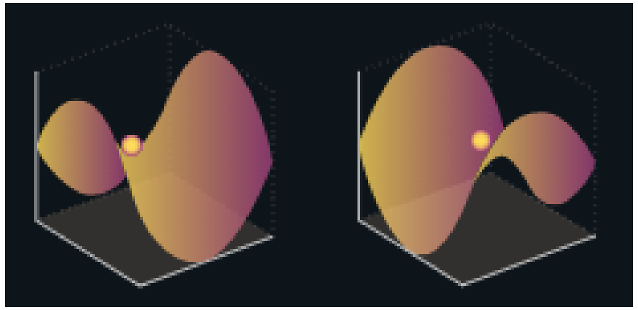
**ION = QUBIT**

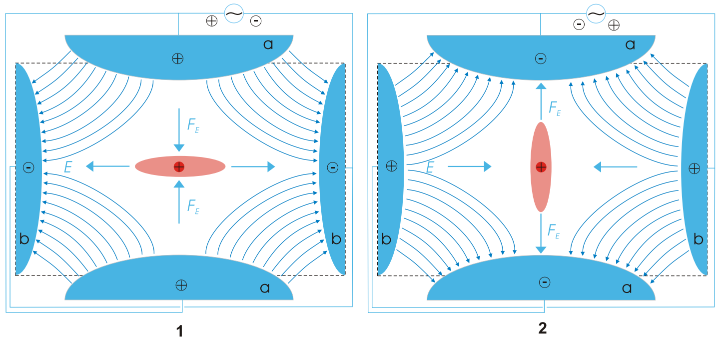
IonQ: ytterbium ions (Yb+)

Ions can be confined and suspended in free space using electromagnetic fields

**Paul Trap Mechanism:**

- Quadrupole electric fields combine static (DC) and oscillating at radio frequency (RF) fields to trap ions (at the saddle point ).
- RF fields provide radial confinement, and DC fields provide axial confinement.
- The oscillating RF field creates an effective 'pseudopotential' that keeps ions oscillating in the trap.
- Once trapped, the ions should be cooled using Doppler cooling.

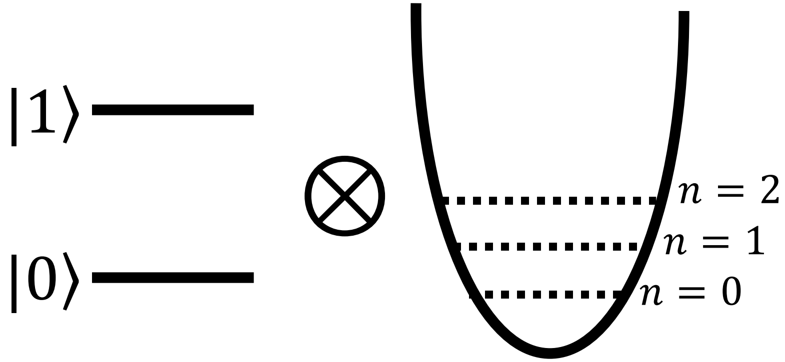




# Qubit: how to realize it?
## TRAPPED ION DEVICES

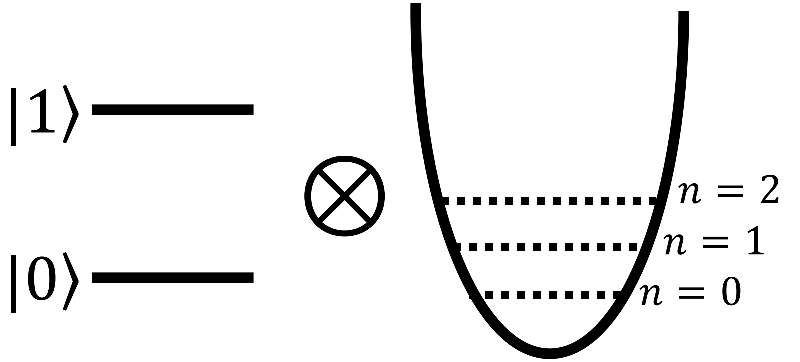
Qubit state = the electronic states of the ion ($|0\rangle$ and $|1\rangle$ )
(two ground state hyperfine levels)

Operations to manipulate qubits' state:
individual laser beams plus one global beam.



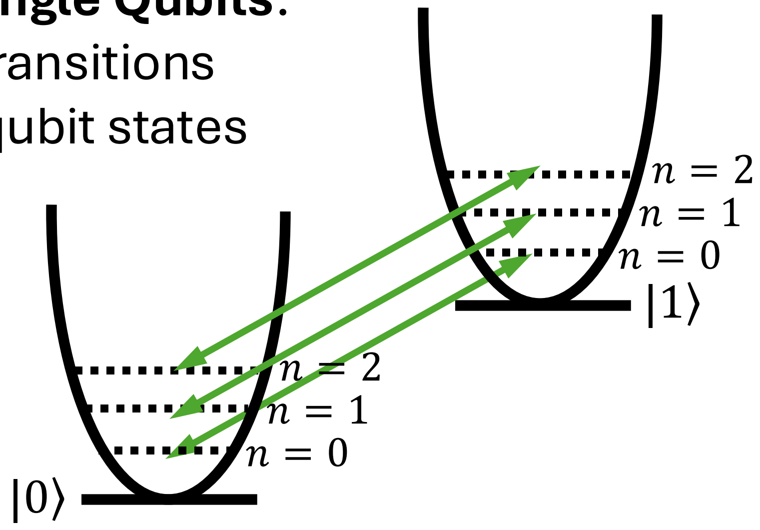
# Qubit: how to realize it?
## TRAPPED ION DEVICES

Qubit state = the electronic states of the ion (|0⟩ and |1⟩ )
(two ground state hyperfine levels)

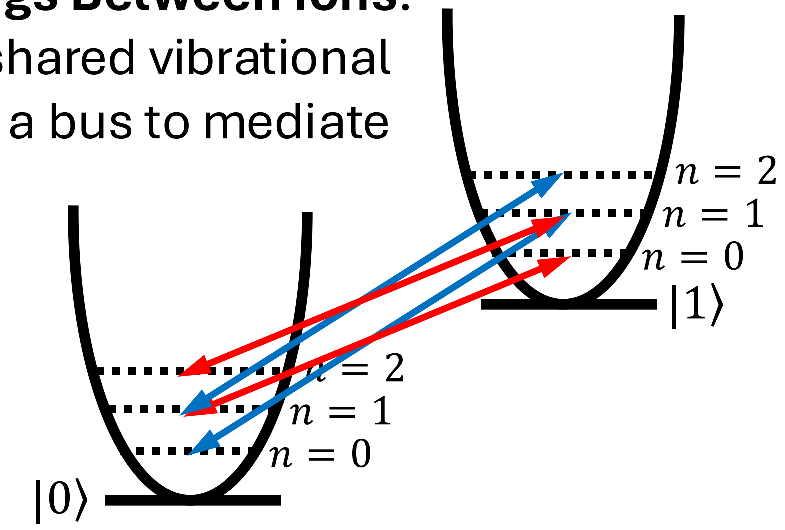



Operations to manipulate qubits' state:
individual laser beams plus one global beam.

Addressing **Single Qubits**:
Lasers drive transitions between ion qubit states (|0⟩ and |1⟩).




Creating **Couplings Between Ions**:
Ions interact via shared vibrational modes, acting as a bus to mediate interactions.



# **QUBIT**: mathematical representation

The qubit lives in a 2D ***Hilbert*** space: orthonormal basis state vectors

**Computational basis states** $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

A qubit can be in a state that is a **superposition** of the basis vector

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

# **QUBIT**: mathematical representation

The qubit lives in a 2D ***Hilbert*** space: orthonormal basis state vectors

**Computational basis states** $|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}$ $|1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$

A qubit can be in a state that is a **superposition** of the basis vector

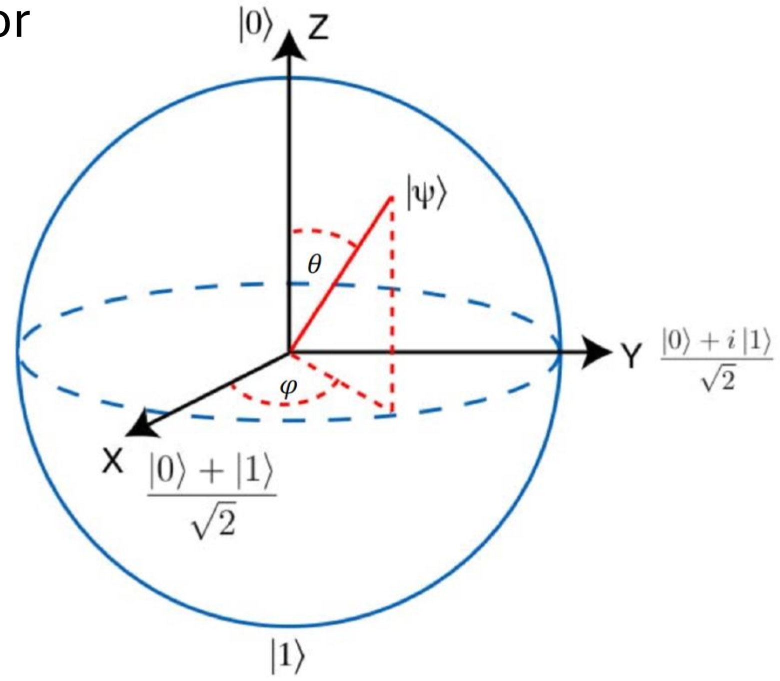
$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$|\Psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\vartheta}{2}|1\rangle$$

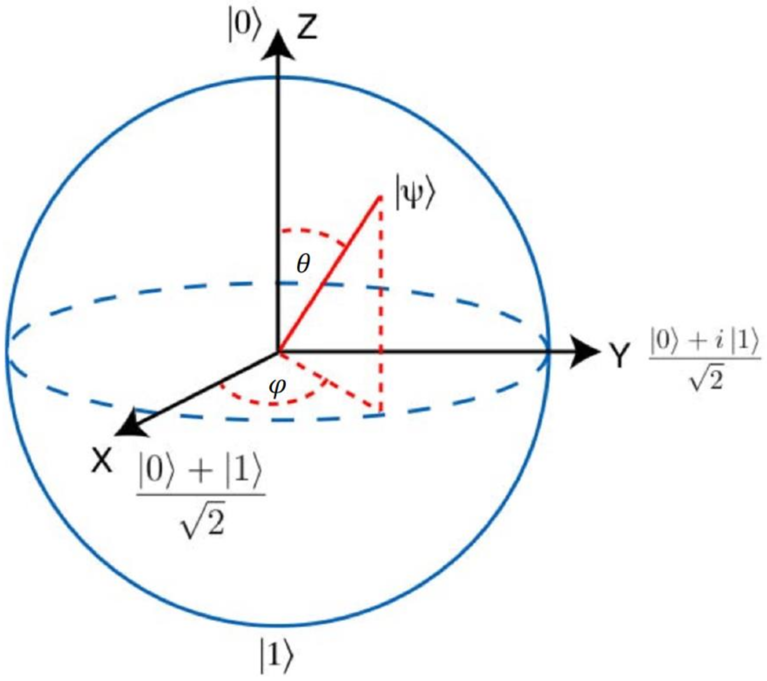
$\theta$ and $\varphi$ uniquely define a point on the surface of the unit 3D sphere → cartesian coordinates:

$$(x, y, z) = (cos\varphi sin\theta, sin\varphi sin\theta, cos\theta)$$

Computational basis States → at the opposite poles of the z axis



# **QUBIT**: mathematical representation




**Computational basis states**

**Z axes** $\quad |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

**X axes** $\quad |+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad |+\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$

**Y axes** $\quad |R\rangle = \frac{|0\rangle + i|1\rangle}{\sqrt{2}} \quad |L\rangle = \frac{|0\rangle - i|1\rangle}{\sqrt{2}}$

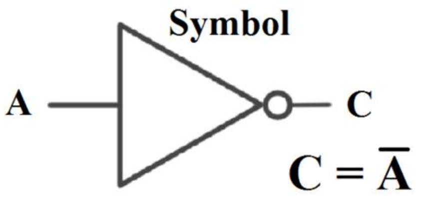
# **Quantum gates**: manipulation of qubits

Basic elements of **classical** digital computer:

Elementary unit: bit: 0,1

Elementary operations: logical gates

NOT Gate




Truth Table

| INPUT | OUTPUT |
|---|---|
| A | NOT A |
| 0 | 1 |
| 1 | 0 |

Basic elements of **quantum** digital computer:

Elementary unit: qubit: $\alpha|0\rangle + \beta|1\rangle$

Elementary operations: **quantum** gates

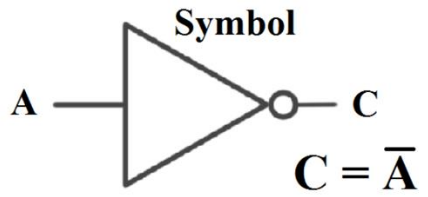
# **Quantum gates**: manipulation of qubits

Basic elements of **classical** digital computer:

Elementary unit: bit: 0,1

Elementary operations: logical gates

**NOT Gate**




**Truth Table**

| INPUT | OUTPUT |
|---|---|
| A | NOT A |
| 0 | 1 |
| 1 | 0 |

Basic elements of **quantum** digital computer:

Elementary unit: qubit: $\alpha|0\rangle + \beta|1\rangle$

Elementary operations: **quantum** gates

to change the state of the qubits

reversible and preserve probabity (UNITARY → $U^{\dagger}U = I$)

A **single-qubit quantum gate** is represented by a 2×2 unitary matrix

$\boxed{U\colon\ |\Psi'\rangle = U|\Psi\rangle}$

→The quantum state $|\Psi'\rangle$ after the gate action is obtained by multiplying the original state by the gate matrix.

# quantum gates: single qubit gates

**Gate X, o bit-flip**

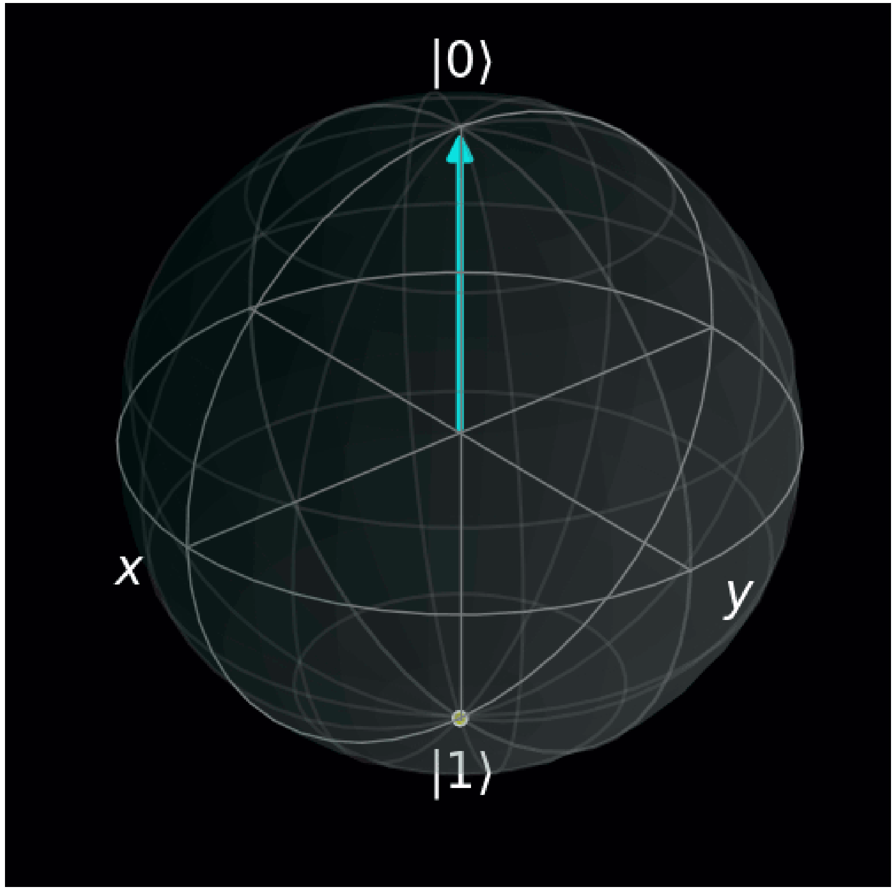
the analogue of the classical NOT gate

Basis states: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Matrix representation (Pauli X)

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

$$X|0\rangle \rightarrow |1\rangle$$
$$X|1\rangle \rightarrow |0\rangle$$



# quantum gates: single qubit gates

**Gate X, o bit-flip**

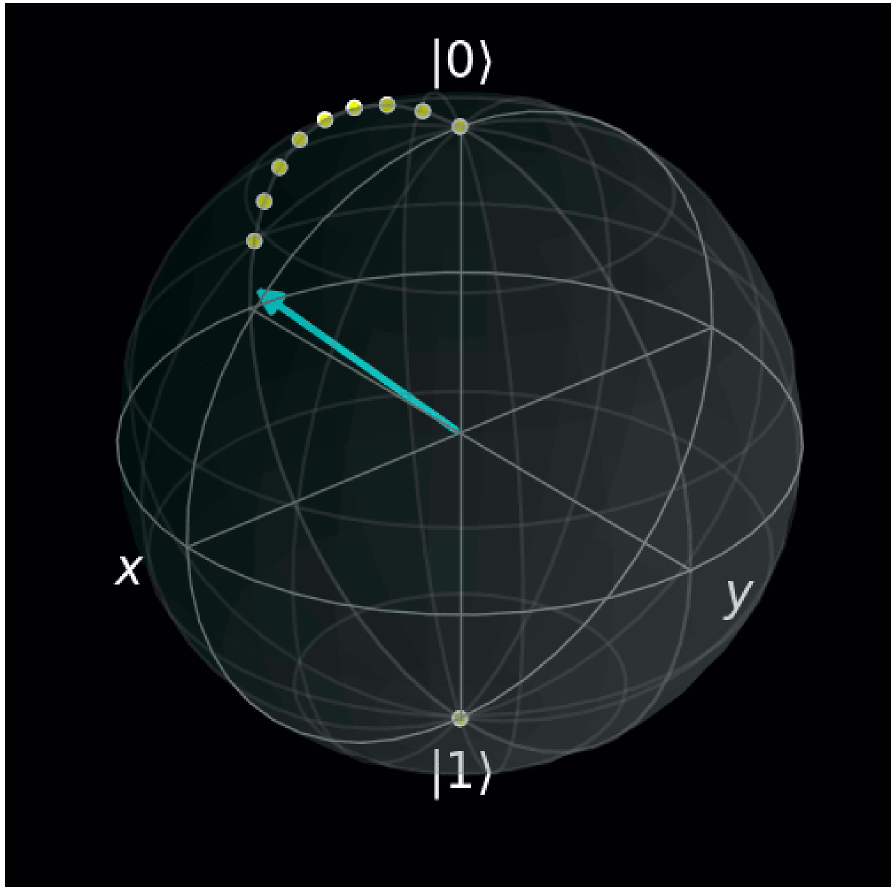
the analogue of the classical NOT gate

Basis states: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Matrix representation (Pauli X)

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

$$X|0\rangle \rightarrow |1\rangle$$
$$X|1\rangle \rightarrow |0\rangle$$



# quantum gates: single qubit gates

**Gate X, o bit-flip**

the analogue of the classical NOT gate

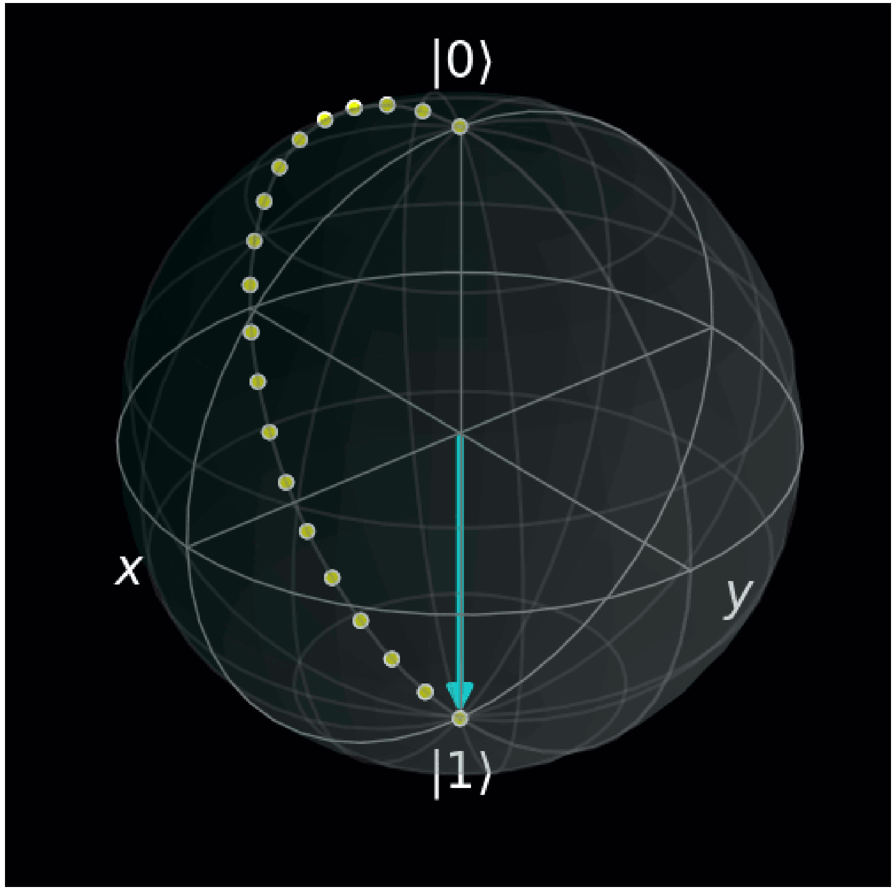
Basis states: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Matrix representation (Pauli X)

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

$$X|0\rangle \rightarrow |1\rangle$$
$$X|1\rangle \rightarrow |0\rangle$$

$$\boxed{X \;\rightarrow\; \mathrm{R_X}(\theta) = \mathrm{e}^{-\mathrm{i}\frac{\theta}{2}\sigma^X}}$$



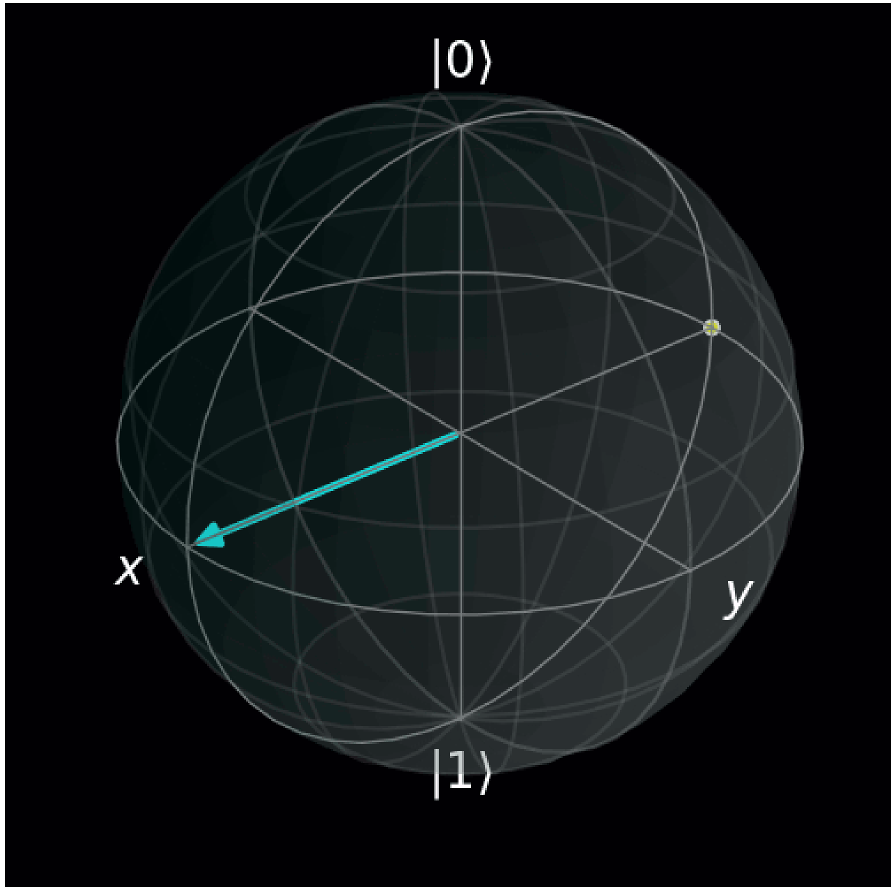
# quantum gates: single qubit gates

**Gate Z, o phase-flip**

Basis states: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Matrix representation (Pauli Z)

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




$$Z|0\rangle \rightarrow |0\rangle$$

$$Z|1\rangle \rightarrow -|1\rangle$$

# quantum gates: single qubit gates

**Gate Z, o phase-flip**

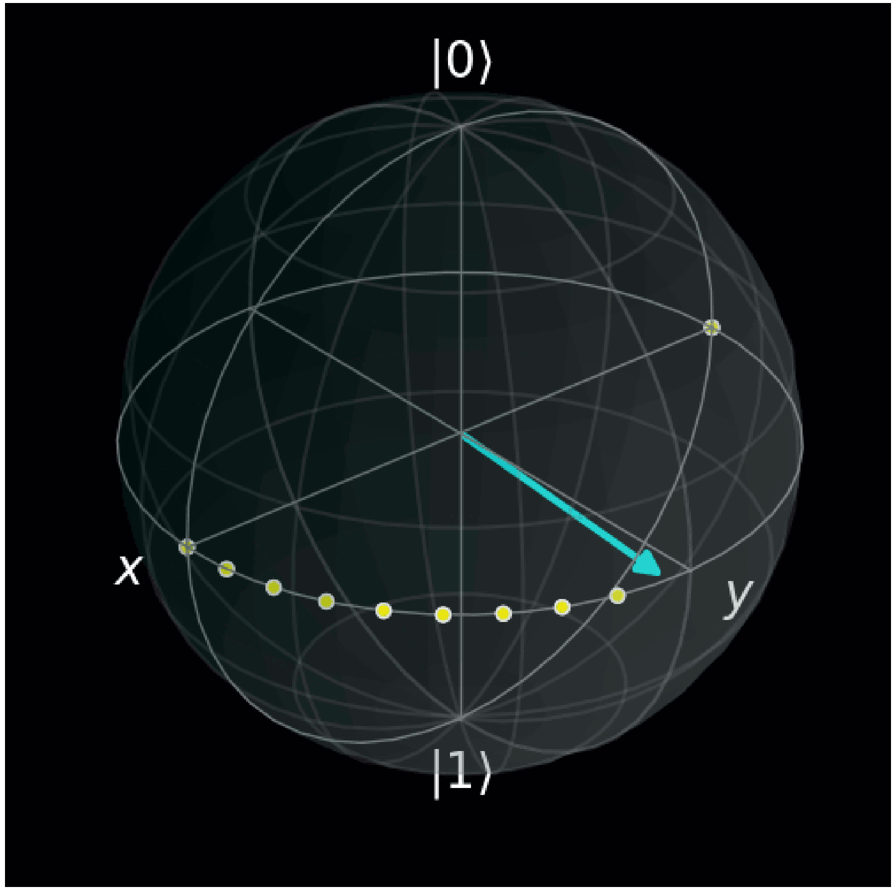
Basis states: $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$ $\quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$

Matrix representation (Pauli Z)

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

$$Z|0\rangle \rightarrow |0\rangle$$

$$Z|1\rangle \rightarrow -|1\rangle$$



# quantum gates: single qubit gates

**Gate Z, o phase-flip**

Basis states: $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$

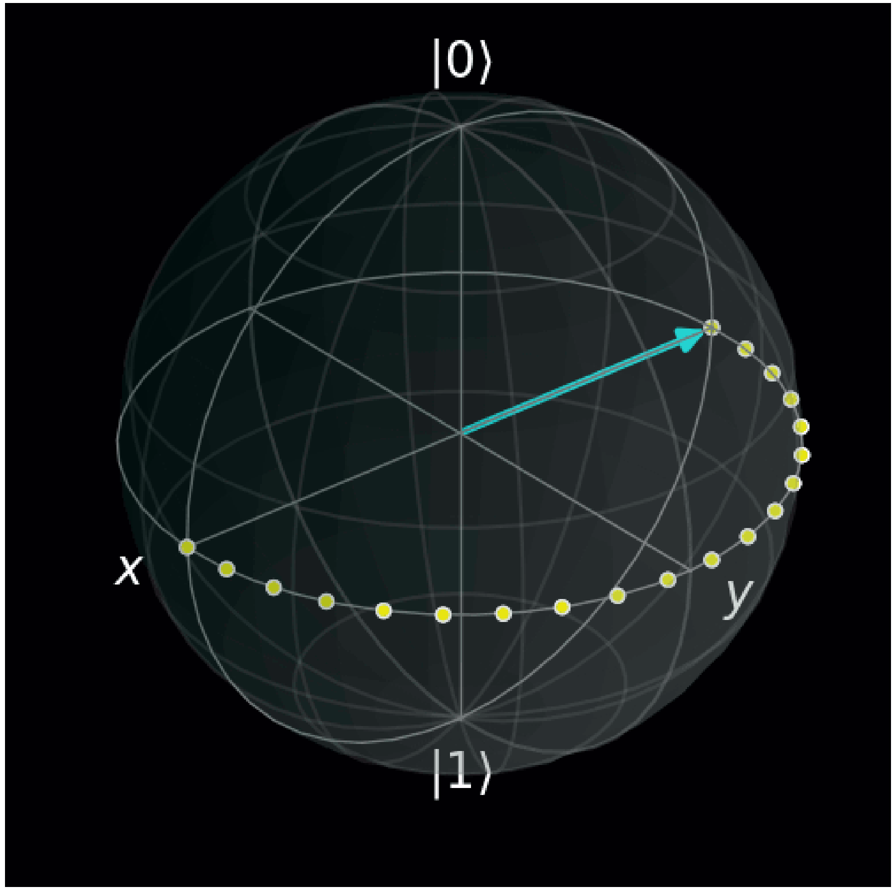
Matrix representation (Pauli Z)

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

$$Z|0\rangle \to |0\rangle$$

$$Z|1\rangle \to -|1\rangle$$

$$\boxed{Z \;\to\; \mathrm{R_Z}(\theta) = \mathrm{e}^{-\mathrm{i}\frac{\theta}{2}\sigma^Z}}$$



# quantum gates: single qubit gates

**Gate Y**

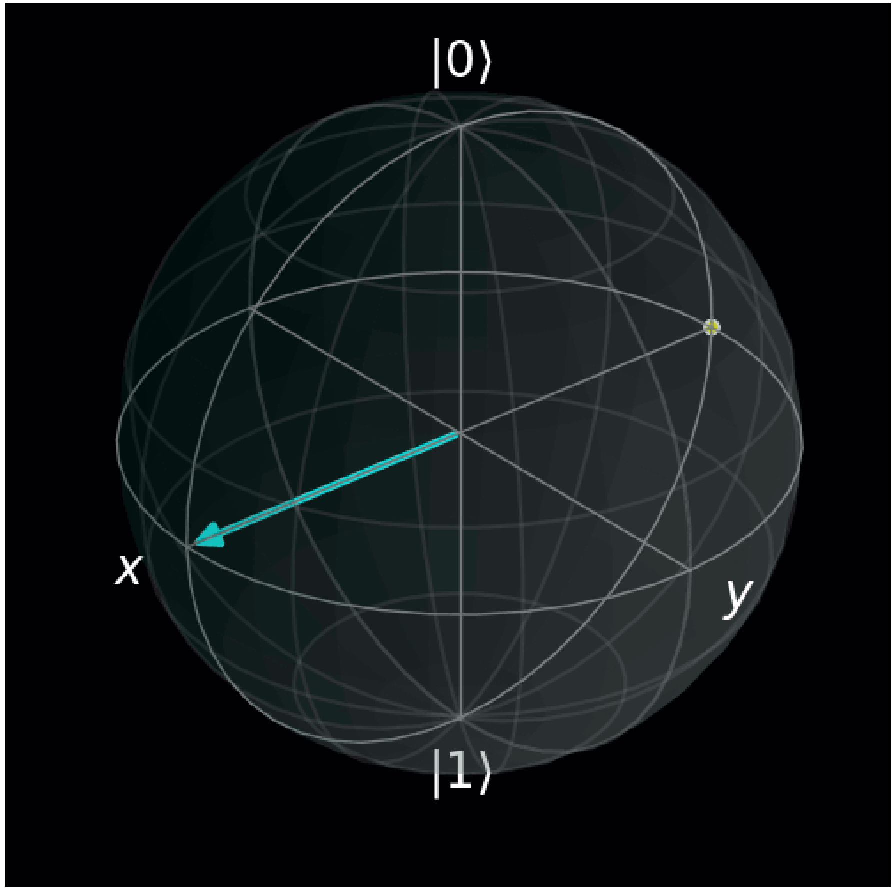
Basis states: $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$

Matrix representation (Pauli Y)

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

$$\mathrm{Y}|0\rangle \rightarrow i|1\rangle$$

$$\mathrm{Y}|1\rangle \rightarrow -i|0\rangle$$



# quantum gates: single qubit gates

**Gate Y**

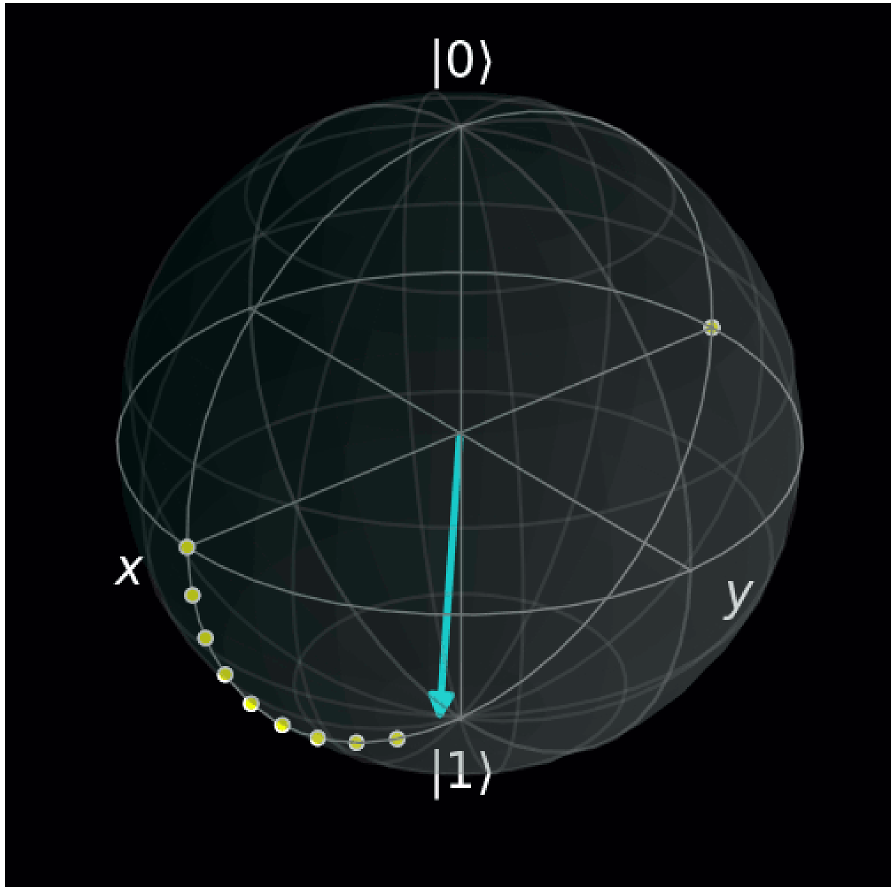
Basis states: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Matrix representation (Pauli Y)

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

$$Y|0\rangle \to i|1\rangle$$

$$Y|1\rangle \to -i|0\rangle$$



# quantum gates: single qubit gates

**Gate Y**

Basis states: $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$

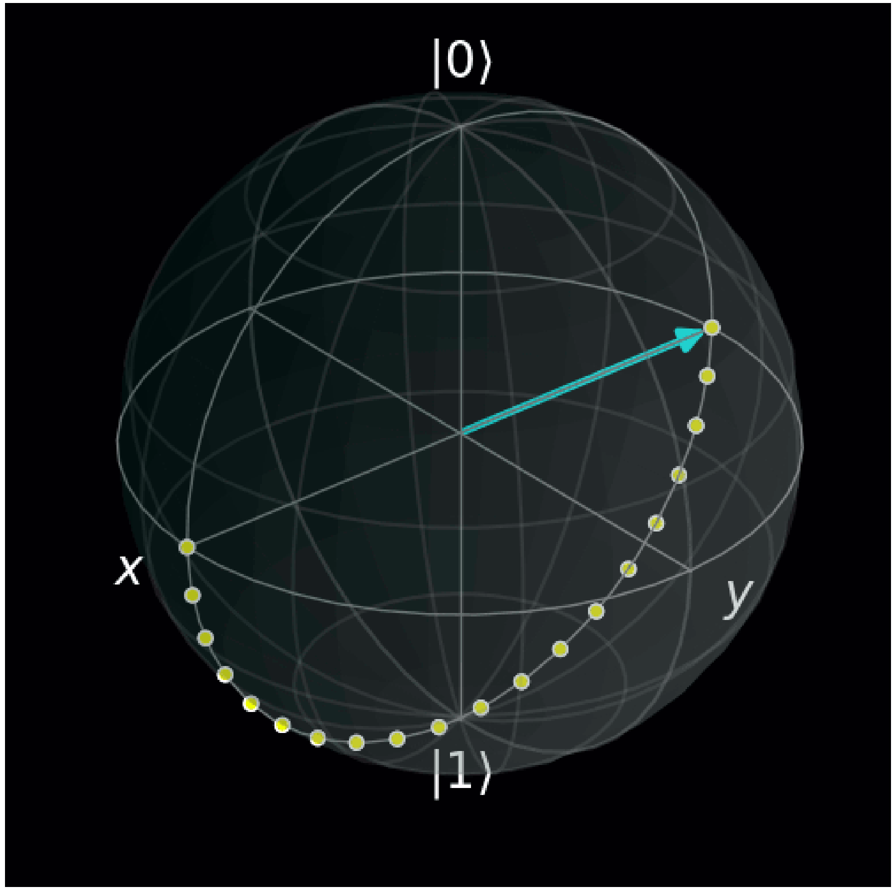
Matrix representation (Pauli Y)

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

$$Y|0\rangle \rightarrow i|1\rangle$$

$$Y|1\rangle \rightarrow -i|0\rangle$$

$$\boxed{Y \ \rightarrow \ R_Y(\theta) = e^{-i\frac{\theta}{2}\sigma^Y}}$$



# quantum gates: single qubit gates

**Hadamard Gate (H)**

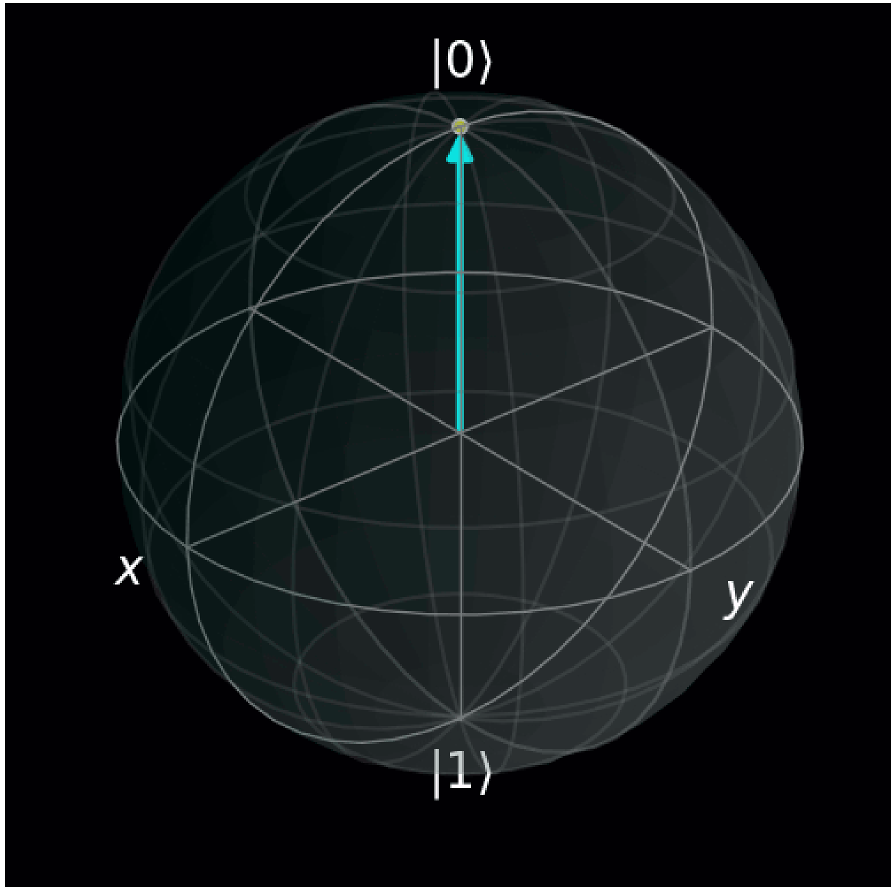
Basis states: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Matrix representation:

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

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

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



# quantum gates: single qubit gates

**Hadamard Gate (H)**

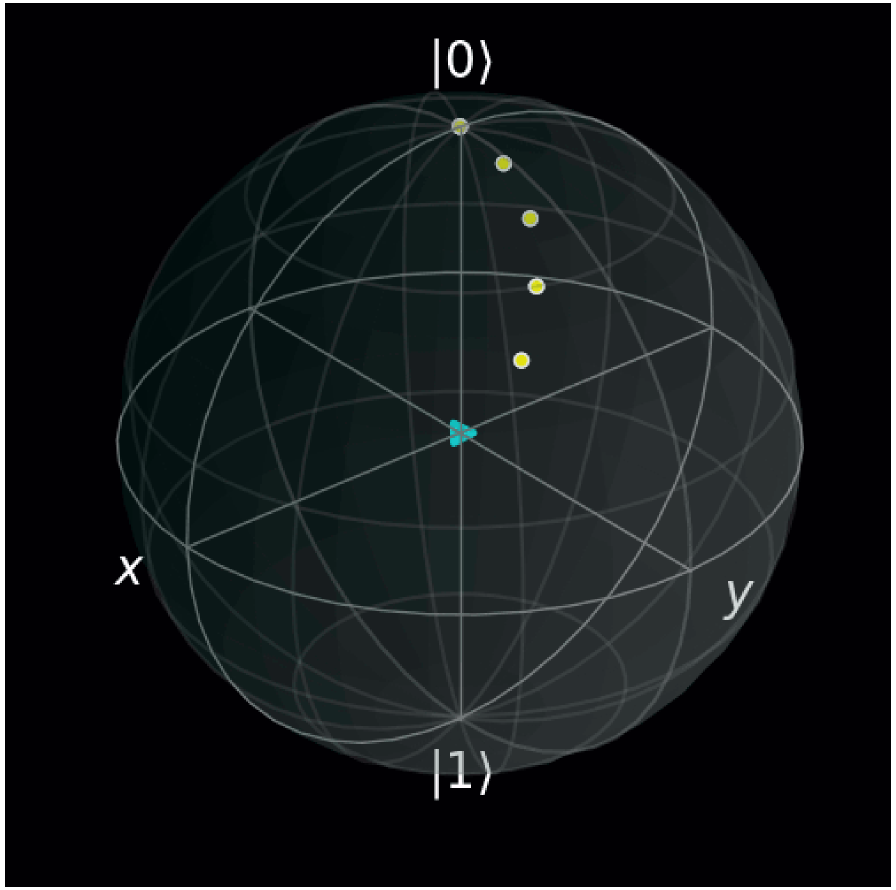
Basis states: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Matrix representation:

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

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

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



# quantum gates: single qubit gates

**Hadamard Gate (H)**

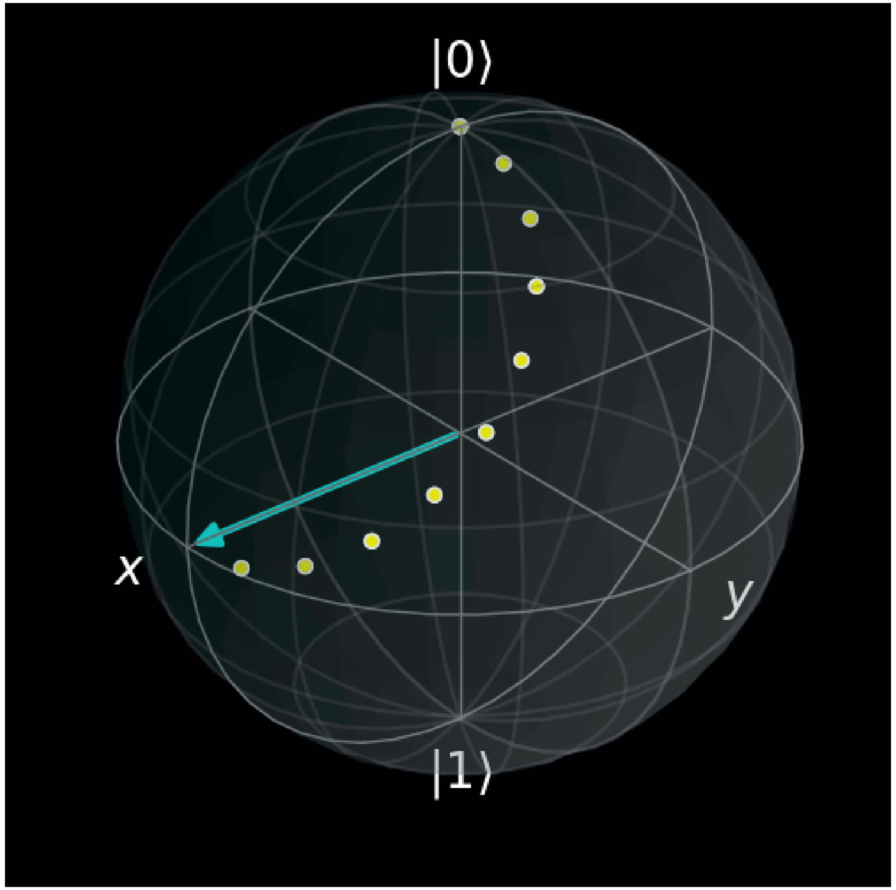
Basis states: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Matrix representation:

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

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

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



# quantum gates: single qubit gates

**Hadamard Gate (H)**

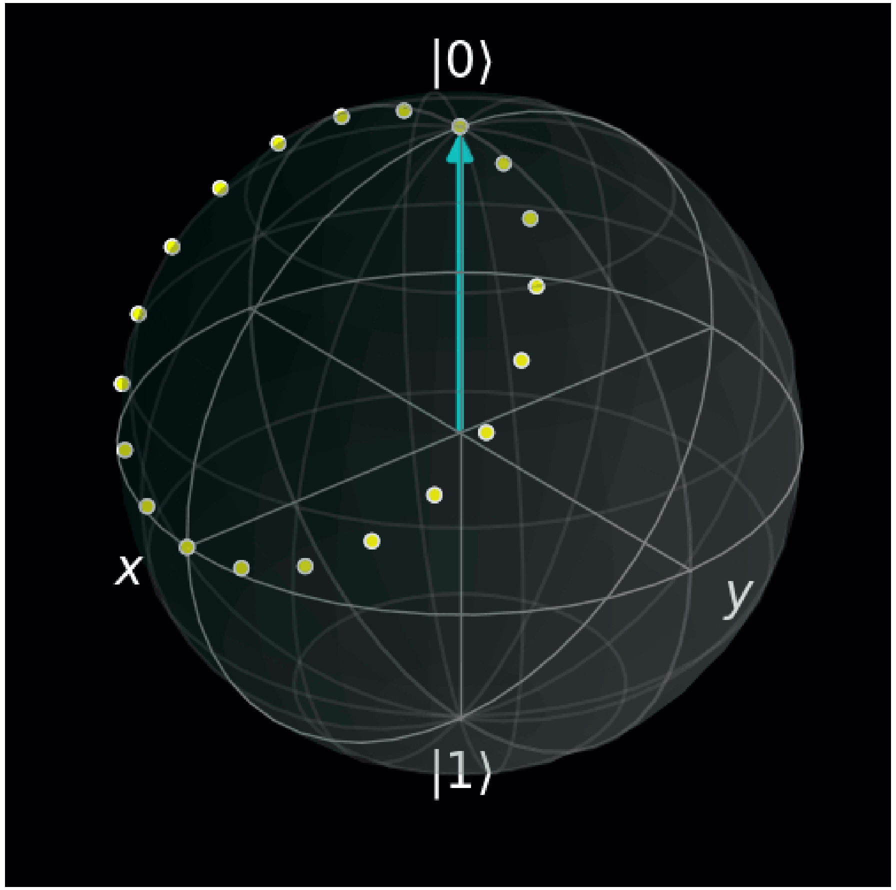
Basis states: $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$

Matrix representation:

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

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

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



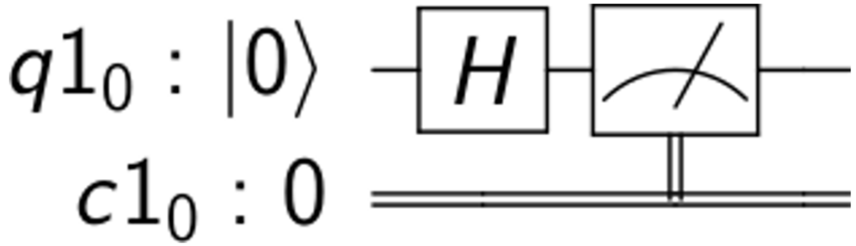
# superposition

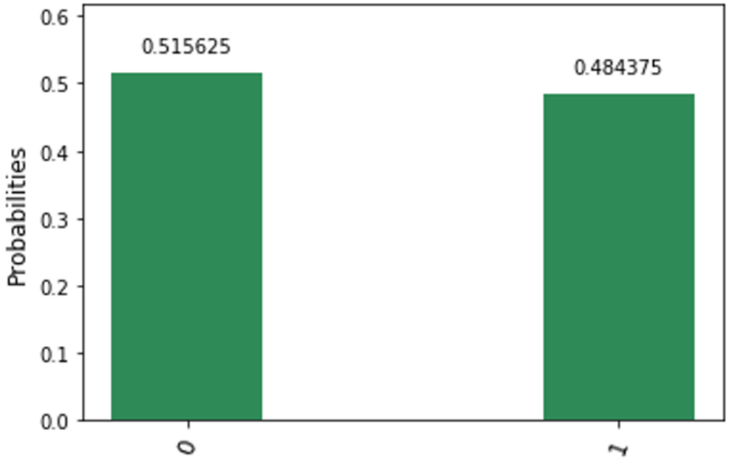
A quantum computer with N qubits can exist in a superposition of all its $2^N$ logical states

What is the difference between superposition and probability?

A row of N coins, each of which can be either heads or tails, has $2^N$ possible states, but is only in one of these states. A quantum state, however, can be in a combination of these states.

The operation we use to create superposition on a qubit is the Hadamard gate H:

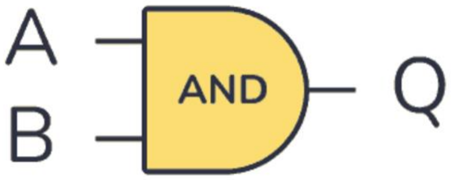


# Multiple qubit gates

Basic elements of **classical** digital computer:

| | |
|---|---|
| Elementary unit: | bit: 0,1 |
| Elementary operations: | logical gates |



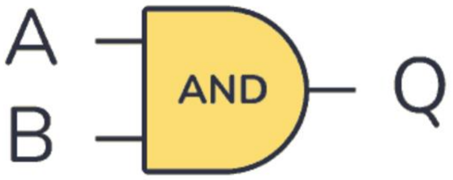

| A | B | Q |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

# Multiple qubit gates

Basic elements of **classical** digital computer:

Elementary unit: bit: 0,1

Elementary operations: logical gates




| A | B | Q |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Basic elements of **quantum** digital computer:

Elementary unit: qubit: $\alpha|0\rangle + \beta|1\rangle$ in 2D **Hilbert** space

n qubits system: $H_1 \otimes H_2 \otimes \ldots \otimes H_n$ ($2^n$-dim Hilbert space)

Elementary operations: **quantum** gates

→ To do something useful with all the possible configurations of a quantum computer, logical operations between different qubits ne e d to be considered.

# Multiple qubit gates : entanglement

**Product state:** $\frac{1}{\sqrt{2}}|0\rangle(|0\rangle + |1\rangle)$

**Entangled state:** $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

The **Entanglement:**

An entangled state cannot be written as a product states

It can be obtained only using two-qubit gates

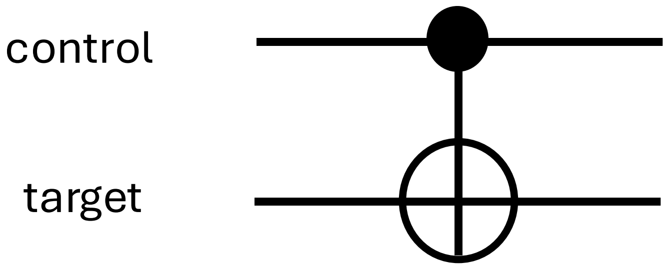
**Bell states**:
set of four maximally entangled two-qubit states.
They form an orthonormal basis for the two-qubit Hilbert space

$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \qquad |\Psi^+\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}$$

$$|\Phi^-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}} \qquad |\Psi^-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

# Multiple qubit gates : entanglement

**Controlled-NOT, or CNOT**




$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

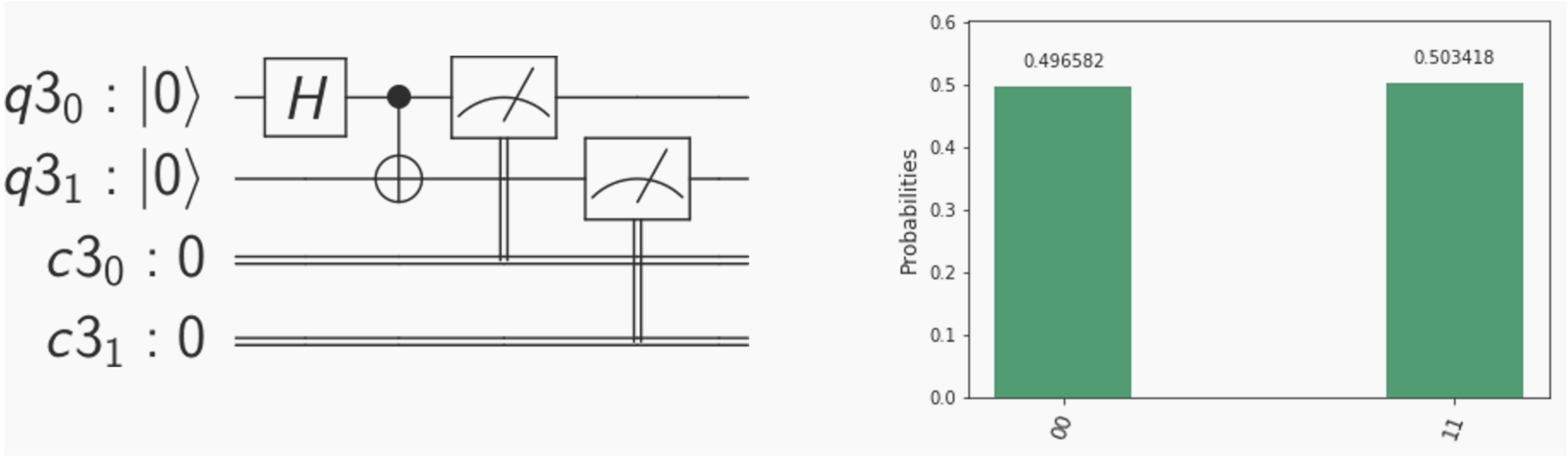
$|00\rangle \rightarrow |00\rangle$
$|01\rangle \rightarrow |01\rangle$
$|10\rangle \rightarrow |11\rangle$
$|11\rangle \rightarrow |10\rangle$

(it depends on the order of the qubits)

The CNOT gate applies a bit-flip (X) to the target qubit only if the control qubit is in $|1\rangle$, otherwise it does nothing.

# Multiple qubit gates : entanglement

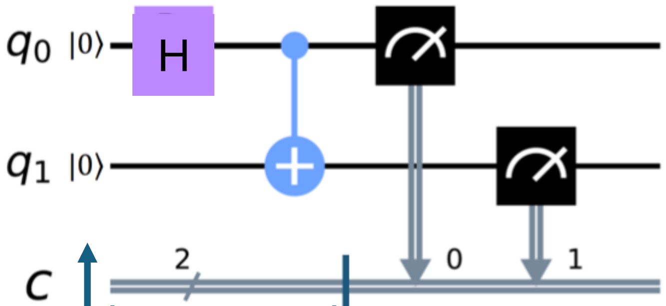
As an example, if we run this circuit that uses the CNOT gate




- We obtain an entangled state!

# Quantum measurement

Quantum measurement → for extracting information from quantum systems.
In classical systems → deterministic outcomes
In quantum systems → outcomes are probabilistic due to the nature of quantum states.

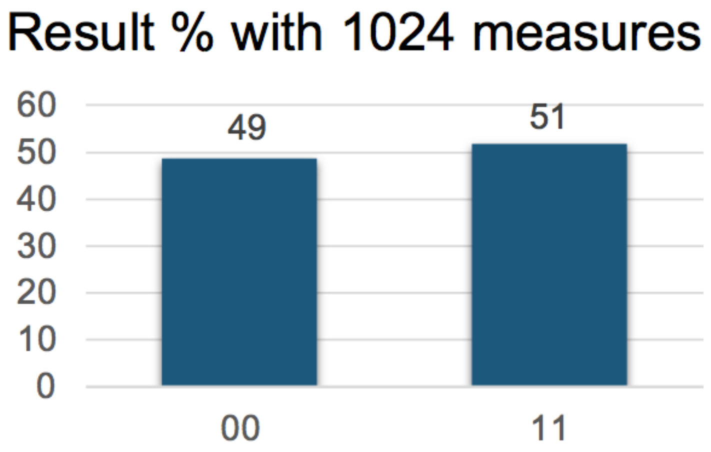



Measure of the qubit
In computational basis

Final qubits state: $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

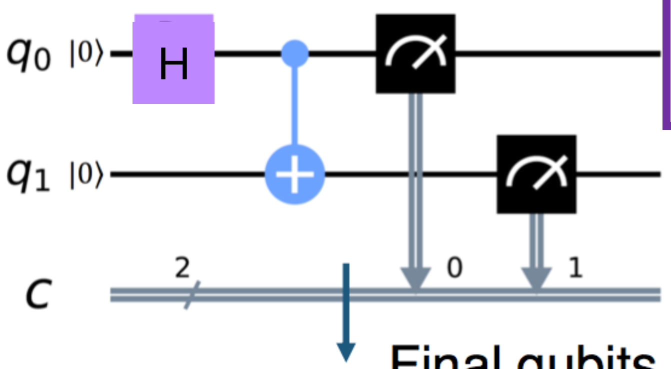
GATES → operations

Initial state of qubits



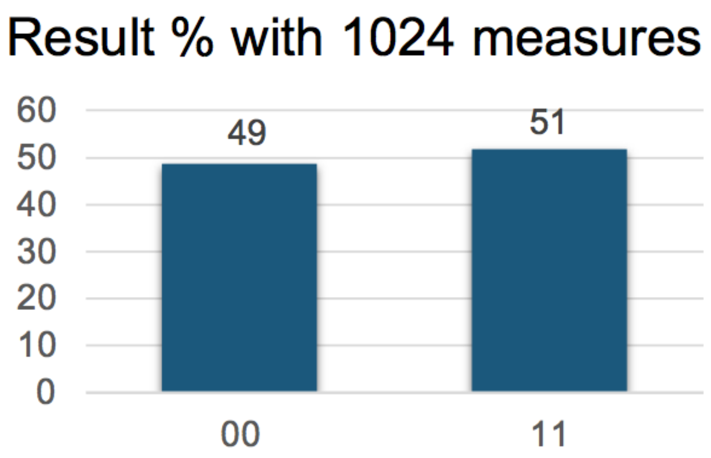
# Quantum measurement

Quantum measurement → for extracting information from quantum systems.
In classical systems → deterministic outcomes
In quantum systems → outcomes are probabilistic due to the nature of quantum states.




Measure of the qubit
In computational basis

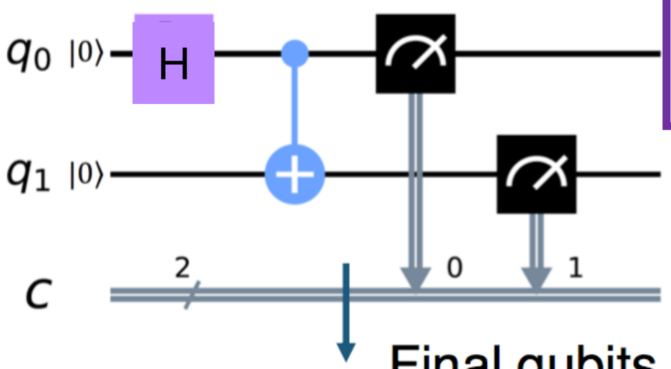
Final qubits state: $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$




The final state $|\Psi\rangle$ collapses into one of the eigenstates of the observable being measured
**Measure in the computational basis**
**= measure of $\bigotimes_i Z_i$**

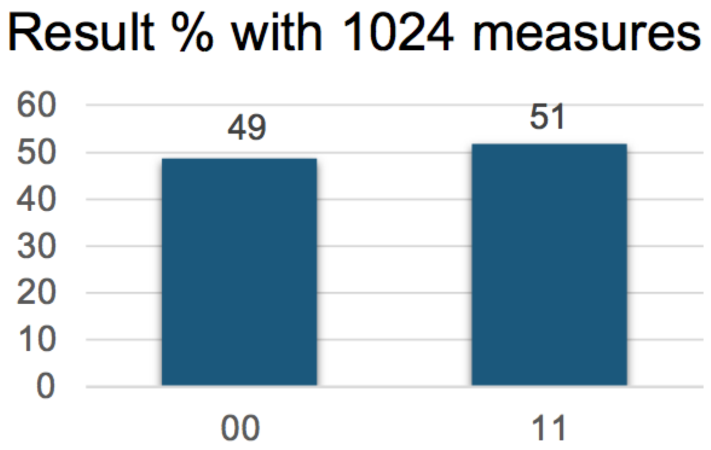
# Quantum measurement

Quantum measurement → for extracting information from quantum systems.
In classical systems → deterministic outcomes
In quantum systems → outcomes are probabilistic due to the nature of quantum states.




Measure of the qubit
In computational basis

Final qubits state: $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$




The final state $|\Psi\rangle$ collapses into one of the eigenstates of the observable being measured
**Measure in the computational basis**
**= measure of $\bigotimes_i Z_i$**

By **running the algorithm multiple times** and collecting measurements, deterministic outcomes are approximated through the statistical distribution of results.

# Quantum measurement
## Mathematical framework

**Measurement Operators**:

A set of operators $\{\mathrm{M_m}\}$ acting on the quantum state (circuit) to provide measurement outcomes.

- Probability of outcome m: $p(m) = \langle\Psi|\mathrm{M_m^\dagger}\,\mathrm{M_m}|\Psi\rangle$

- Post-Measurement State: $|\Psi'\rangle = \frac{\mathrm{M_m}\,|\Psi\rangle}{\sqrt{\langle\Psi|\mathrm{M_m^\dagger}\,\mathrm{M_m}|\Psi\rangle}}$

Completeness $\Sigma_\mathrm{m}\mathrm{M_m^\dagger}\,\mathrm{M_m} = \mathrm{I}$ ensures probabilities sum to 1: $\sum_m p(m) = 1$

# Quantum measurement
## Mathematical framework

Measurement of a single qubit in the **computational basis**:

- Measurement operators $\boldsymbol{M_0} = |\mathbf{0}\rangle\langle\mathbf{0}|$ and $\boldsymbol{M_1} = |\mathbf{1}\rangle\langle\mathbf{1}|$
- For state $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the outcome probabilities are

$$p(0) = |\alpha|^2 \text{ and } p(1) = |\beta|^2.$$

# Quantum measurement
## Mathematical framework

Measurement of a single qubit in the **computational basis**:

- Measurement operators $\boldsymbol{M_0} = |\mathbf{0}\rangle\langle\mathbf{0}|$ and $\boldsymbol{M_1} = |\mathbf{1}\rangle\langle\mathbf{1}|$
- For state $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle,$ the outcome probabilities are

$$p(0) = |\alpha|^2 \text{ and } p(1) = |\beta|^2.$$

Expectation Values:

- The expectation value of an observable $O$ is: $\langle \boldsymbol{O} \rangle = \langle \boldsymbol{\Psi} | \boldsymbol{O} | \boldsymbol{\Psi} \rangle$
- **Z operator**: $\langle Z \rangle = \langle \Psi | Z | \Psi \rangle = \langle \Psi | M_0 | \Psi \rangle - \langle \Psi | M_1 | \Psi \rangle = p(0) - p(1)$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1|$$

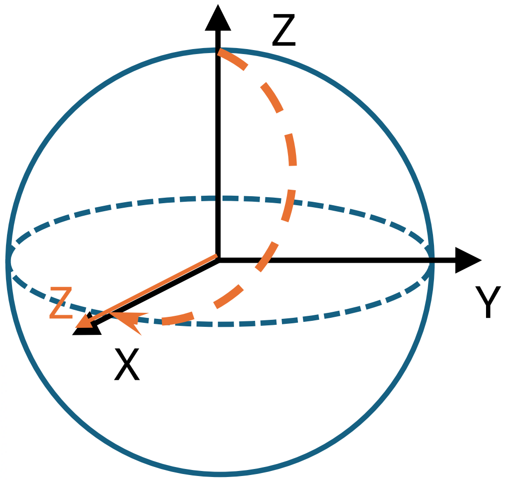
Simple to verify: $\langle \mathrm{Z} \rangle = \langle \Psi | \mathrm{Z} | \Psi \rangle = [\alpha^* \quad \beta^*] \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = |\alpha|^2 - |\beta|^2$

# Quantum measurement
## Mathematical framework

**Expectation Values of X:**

$X = H\,Z\,H$

→ Hadamard gate H effectively perform a rotation on the Bloch Sphere exchanging the x-axis with the z-axis



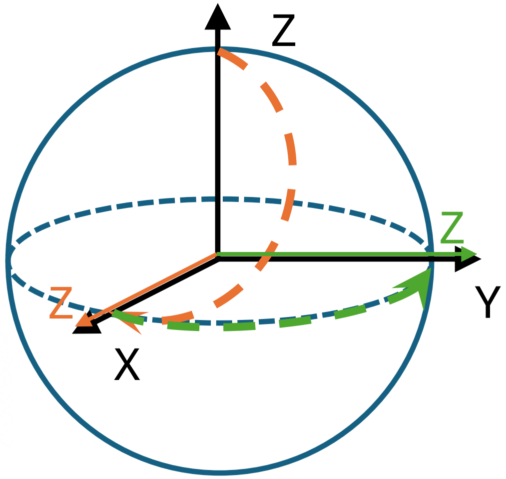

$$\langle\hat{X}\rangle = \langle\Psi|\,\hat{X}\,|\Psi\rangle = \langle\Psi|\,\hat{H}\hat{Z}\hat{H}\,|\Psi\rangle = {}_X\langle\Psi|\,\hat{Z}\,|\Psi\rangle_X = p(0) - p(1)$$

**Expectation Values of Y:**

$\mathrm{Y} = \mathrm{S\,H\,Z\,H\,S}^{\dagger}$

where $S$ represent the **Phase gate S**= $\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{pmatrix}$

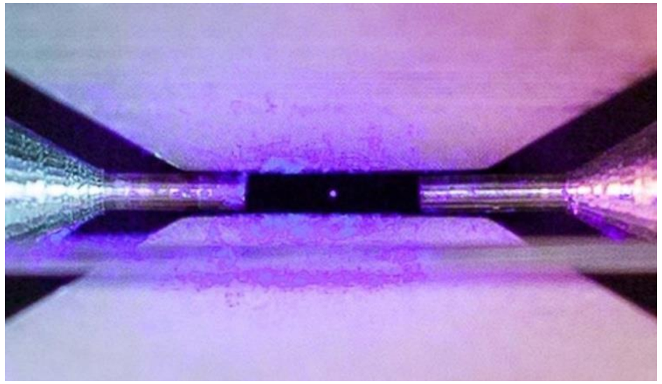
We exchange the y-axis with the z-axis




$$\langle\hat{Y}\rangle = \langle\Psi|\,\hat{Y}\,|\Psi\rangle = \langle\Psi|\,\hat{S}\hat{H}\hat{Z}\hat{H}\hat{S}^{\dagger}\,|\Psi\rangle = {}_Y\langle\Psi|\,\hat{Z}\,|\Psi\rangle_Y = p(0) - p(1)$$

# Quantum measurement

## TRAPPED IONS DEVICES



**ION = QUBIT**

IonQ: ytterbium ions (Yb+)

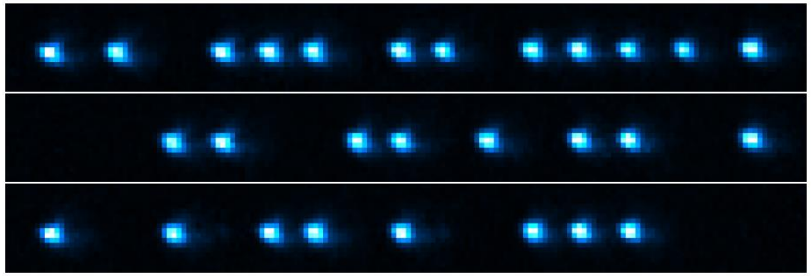
Measurement of ion's qubit state:

by shining a laser tuned to interact with **|0⟩** state
→ causing it to fluoresce.

Fluorescence detection obtained using a photon

- Fluorescing ions indicate the $|0\rangle$ state
- non-fluorescing ions indicate the $|1\rangle$ state

simultaneous measurements of multiple ions → measurement of **correlated stated** and **entanglement**



$|1\,1\,0\,1\,1\,1\,0\,1\,1\,0\,1\,1\,1\,1\,1\rangle$

$|0\,0\,1\,1\,0\,0\,1\,1\,0\,1\,0\,1\,1\,0\,1\rangle$

$|1\,0\,1\,0\,1\,1\,0\,1\,0\,0\,1\,1\,1\,0\,0\rangle$

Blatt group, Innsbruck

# Importance of quantum measurement and expectation values

Density Matrix $\rho = \sum_i p_i |\Psi_i\rangle\langle\Psi_i|$ → description of the quantum states

Quantum State Tomography → Measurement of Pauli matrices reconstructs the density matrix for a qubit.

**Bloch vector** $\boldsymbol{r} \;=\; (\langle X\rangle, \langle Y\rangle, \langle Z\rangle)$ represents the qubit state on the Bloch sphere

$$\rho = 1/2(I + \sum_{i=1}^{3} r_i\sigma_i)\ \sigma_i \qquad \rightarrow \sigma_z = Z\ ,\ \sigma_x = X\ ,\ \sigma_y = Y$$

**cost functions** and **system Hamiltonians** are often encoded using **Pauli operators**. Many algorithms, require calculating quantities like energy, which are combinations of these operators. Thus, expectation values are crucial for **extracting meaningful information** from quantum states.

# Analog vs. Digital Quantum Computing

**Analog Quantum Computing**:

- Based on the **continuous evolution** of a quantum system to find a solution. (evolution guided by a Hamiltonian, which defines the system's energy landscape)
- Example Quantum annealing, where the system is evolved from an initial Hamiltonian to a target Hamiltonian.

**Digital Quantum Computing**:

- similar to classical digital computation – it uses **quantum gates** to manipulate qubits in discrete operations.
- Example: Quantum algorithms like Shor's algorithm (for factoring) or Grover's algorithm (for searching).

# Analog vs. Digital Quantum Computing

**Analog Quantum Computing**:

- Based on the **continuous evolution** of a quantum system to find a solution. (evolution guided by a Hamiltonian, which defines the system's energy landscape)
- Example Quantum annealing, where the system is evolved from an initial Hamiltonian to a target Hamiltonian.

**Digital Quantum Computing**:

- similar to classical digital computation – it uses **quantum gates** to manipulate qubits in discrete operations.
- Example: Quantum algorithms like Shor's algorithm (for factoring) or Grover's algorithm (for searching).

A subset of quantum computing → **Quantum Simulation**:
to replicate the dynamics of a quantum system using another controllable quantum system (quantum computer)

# Analog vs. Digital Quantum Simulation

The **simulation** of a quantum system is obtained by making the system itself evolve under the action of the Hamiltonian $H_s$ that describes it, namely by determining the dynamics from an initial state of the system to a final state.

This process is obtained by solving the Shrödinger equation:

$$\boxed{i\hbar \frac{d}{dt}|\Psi(t)\rangle = H_s |\Psi(t)\rangle}$$

# Analog vs. Digital Quantum Simulation

The **simulation** of a quantum system is obtained by making the system itself evolve under the action of the Hamiltonian $H_s$ that describes it, namely by determining the dynamics from an initial state of the system to a final state.

This process is obtained by solving the Shrödinger equation:

$$\boxed{i\hbar \frac{d}{dt}|\Psi(t)\rangle = H_s|\Psi(t)\rangle}$$

Mathematical solution for time-independent Hamiltonian $H_s$

$$\boxed{|\Psi(t)\rangle = e^{-\frac{i}{\hbar}H_s t}|\Psi(0)\rangle} \qquad \boxed{U = e^{-\frac{i}{\hbar}H_s t}}$$

Mathematical solution for time-dependent Hamiltonian $H_s(t)$

$$\boxed{|\Psi(t)\rangle = \overbrace{\mathcal{T}\exp\left(\int -\frac{i}{\hbar}H_s(t')dt'\right)}^{U}|\Psi(0)\rangle}$$

# Analog vs. Digital Quantum Simulation

**Analog quantum simulations:**
(emulates a specific real system)

- The Hamiltonian describing the simulating controllable system coincides with that of the simulated system $H_S$.
- After preparing the initial state, the system evolves under the Hamiltonian.
- more robust to noise/errors.

**Digital quantum simulations:**

- exploit the formalism of quantum circuits: the target problem is mapped into a series of quantum gates that can be implemented on a quantum computer.

(A universal quantum computer can be programmed to perform many different simulations)

# Analog vs. Digital Quantum Simulation

**Analog quantum simulations:**
(emulates a specific real system)

- The Hamiltonian describing the simulating controllable system coincides with that of the simulated system $H_s$.
- After preparing the initial state, the system evolves under the Hamiltonian.
- more robust to noise/errors.

**Digital quantum simulations:**

- exploit the formalism of quantum circuits: the target problem is mapped into a series of quantum gates that can be implemented on a quantum computer.

(A universal quantum computer can be programmed to perform many different simulations)

**EXAMPLE**

Simulating a transverse field **Ising model**:

$$H = -\sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z - h\sum_i \sigma_i^x$$

$J_{ij}$ : spin-spin interactions (Z-axis)

$h$ : transverse magnetic field (X-axis).

# Analog Quantum Simulation

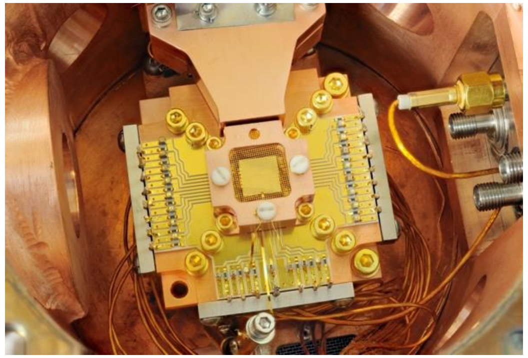
Ex: TRAPPED ION platform

**Hamiltonian engineering approach:**
Configure trapped ions with laser fields to induce spin-spin interactions matching $\boldsymbol{J_{ij}\sigma_i^z\sigma_j^z}$ **terms**.
Apply transverse field using additional laser fields for $\boldsymbol{\sigma^x}$**rotations**.

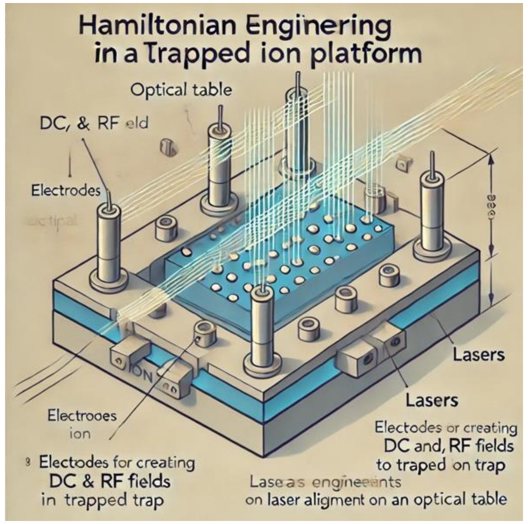
**Evolution:**
The system evolves continuously under the engineered Hamiltonian (for a time t), mimicking the Ising model dynamics.

**Outcome:**
Measure qubits to obtain ground state properties and correlation functions, study quantum phase transition.

REAL



AI-generated



# Digital Quantum Simulation: Trotterization

Trotterization → **Trotter-Suzuki decomposition:**
method to discretize the continuous time evolution operator $\mathrm{U} = \mathrm{e}^{-\mathrm{i}\mathrm{H}_\mathrm{s}\mathrm{t}}$ (of the spin system) into a sequence of quantum gates, by breaking it into small time steps:

$\boxed{\mathrm{e}^{-\mathrm{i}\mathrm{H}_\mathrm{s}\mathrm{t}} \approx \left(\mathrm{e}^{-\mathrm{i}\mathrm{H}_{\mathrm{ZZ}}\Delta\mathrm{t}}\mathrm{e}^{-\mathrm{i}\mathrm{H}_\mathrm{X}\Delta\mathrm{t}}\right)^{\mathrm{t}/\Delta\mathrm{t}}}$ where $\boxed{\mathrm{H}_{\mathrm{ZZ}} = -\sum_{\mathrm{i}<\mathrm{j}} \mathrm{J}_{\mathrm{ij}}\sigma_\mathrm{i}^\mathrm{z}}\sigma_\mathrm{j}^\mathrm{z}$ and $\boxed{\mathrm{H}_\mathrm{X} = -\mathrm{h}\sum_\mathrm{i} \sigma_\mathrm{i}^\mathrm{x}}$

The Trotter step $\Delta\mathrm{t}$ must be small enough to minimize errors from non-commuting terms.

**Quantum Circuit -** Each Trotter step corresponds to a **quantum gate sequence**:
**Ising Interaction** → Two-qubit gates (controlled-phase or controlled-Z gates) simulate the $\sigma_i^z\sigma_j^z$ interactions.
**Transverse Field Term**→ Single-qubit rotations around the X-axis represent the $\sigma_i^x$ term

**Outcome:**
measure at each time step to study time evolution and dynamic properties or excited state.

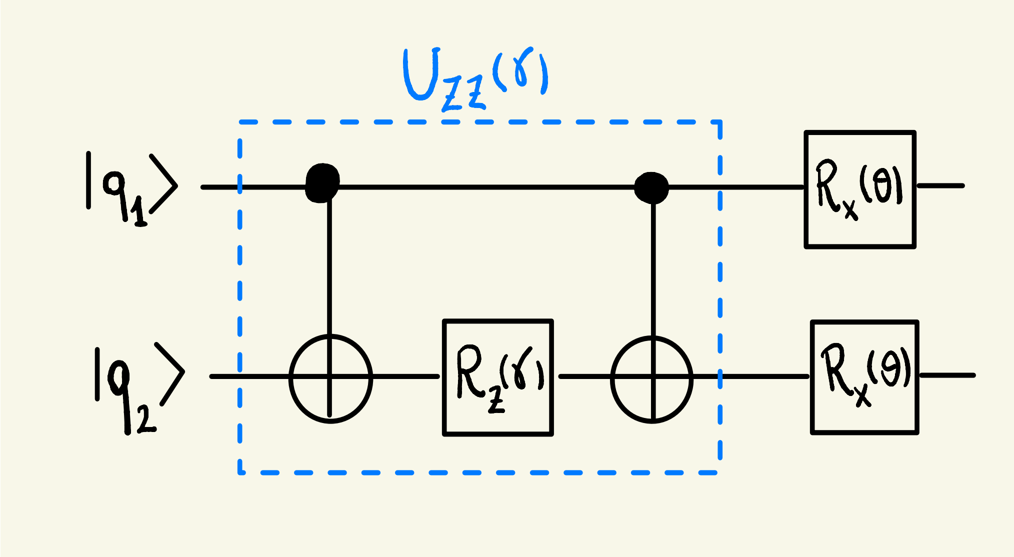
# Digital Quantum Simulation: Trotterization

$e^{-iH_{ZZ}\Delta t} = e^{i\sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z\Delta t} \rightarrow$ control-phase: $U_{ZZ}(\gamma)$ : CNOT and $R_Z(\gamma) = e^{-i\frac{\gamma}{2}\sigma^z}$

$e^{-iH_X\Delta t} = e^{ih\sum_i \sigma_i^x\Delta t} \rightarrow R_X(\theta) = e^{-i\frac{\theta}{2}\sigma^x}$

Example for the first Trotterization step:
2-qubits system
block generalized to bigger systems

$R_Z(\gamma) \rightarrow \gamma = -2J_{12}\Delta t$
$R_X(\theta) \rightarrow \theta = 2h\Delta t$

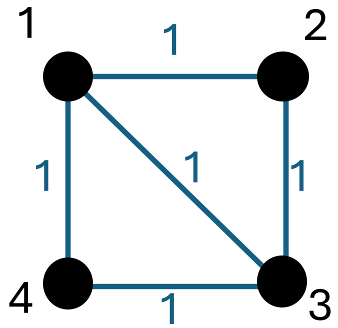


# Analog vs. Digital Quantum Computing

## **EXAMPLE:** OPTIMIZATION PROBLEM

The **Max-Cut problem** (combinatorial optimization problem):

Finding the optimal way to cut (partition) the nodes of a graph into two groups such that the number of edges between the groups is maximized.

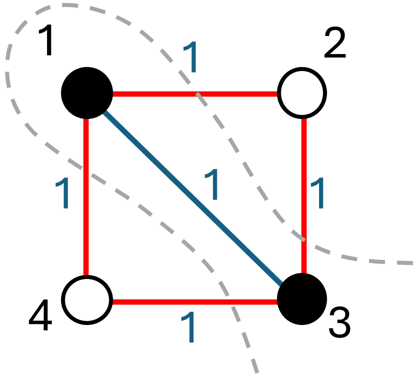



**Formulation**

graph G = (V, E) with edge weights $w_{ij}$ the **Max-Cut cost function** is:

$$C = max \sum_{i,j \in E} w_{ij} \boxed{\frac{(1 - z_i\, z_j)}{2}} \begin{cases} \textbf{=1 if } z_i\, z_j \in \textbf{different subsets} \\ \textbf{=0 if } z_i\, z_j \in \textbf{same subset} \end{cases}$$

where $z_i = \pm 1$ is the partition assignment of each node.




Cut value = 4

# Analog vs. Digital Quantum Computing

## **EXAMPLE:** OPTIMIZATION PROBLEM

**Max-Cut cost function** :

$$C = max \sum_{i,j \in E} w_{ij} \boxed{\frac{(1 - z_i z_j)}{2}} \begin{cases} =1 \text{ if } z_i\, z_j \in \text{different subsets} \\ =0 \text{ if } z_i\, z_j \in \text{same subset} \end{cases}$$

where $z_i = \pm 1$ .

The **Max-Cut problem** can be mapped onto an **Ising model Hamiltonian**:

$$H = -\sum_{(i,j) \in E} J_{ij} \sigma_i^z \sigma_j^z \text{ with } J_{ij} = w_{ij}$$

Goal: reach the ground state = the optimal cut.

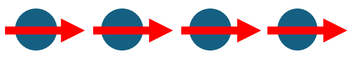
**Applications**: Useful in areas such as network optimization, clustering, and statistical physics.

# Analog quantum computing

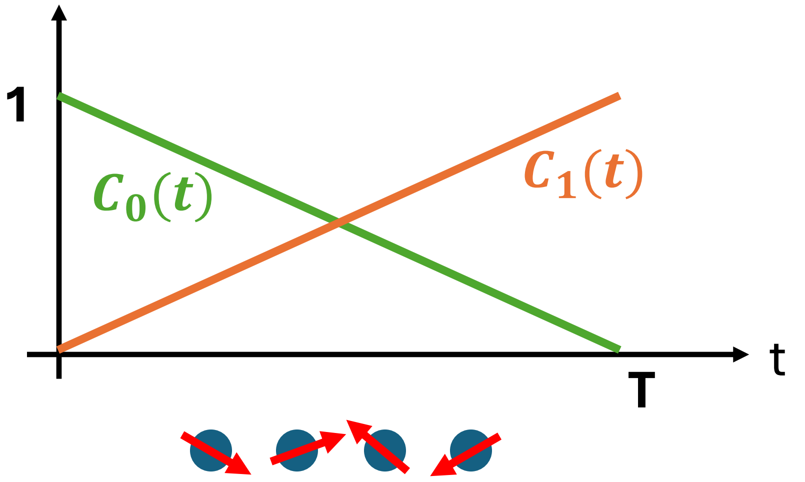
## MAX-CUT with quantum annealing

**Quantum annealing protocol:**

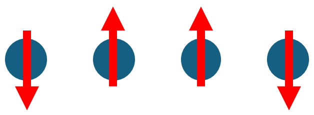
Start with an initial Hamiltonian
(known ground state)
$\mathrm{H}_0 = -\sum_i \sigma_i^x$



gradually transform to $H$ :
$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} H$




Evolution to final Hamiltonian:
$H = -\sum_{(i,j)\in E} J_{ij}\sigma_i^z \sigma_j^z$
(ground state)



If done slowly (adiabatic), the system remains in the ground state, solving the Max-Cut problem.

# Digital quantum computing:

## MAX-CUT with Quantum Approximate Optimization Algorithm

**Quantum Approximate Optimization Algorithm (QAOA)**

Hybrid quantum-classical variational approach

**Cost funtion**

$H_C = -\sum_{(i,j)\in E} J_{ij}\sigma_i^z\sigma_j^z$

**Mixing Hamiltonian:**

$\mathrm{H_M} = -\sum_i \sigma_i^x$

Emploing both $H_C$ and $\mathrm{H_M}$,
we define the **wavefunction**:

$$|\Psi(\gamma,\beta)\rangle = \prod_{i=1}^{p} e^{-i\beta_i H_M}\, e^{-i\gamma_i H_C}|\Psi(0)\rangle$$

With the initial state: $|\Psi(0)\rangle = |+\rangle^{\otimes n}$

Measure of
the cost function
$\langle\Psi(\gamma,\beta)|H_C|\Psi(\gamma,\beta)\rangle$

# Digital quantum computing:

## MAX-CUT with Quantum Approximate Optimization Algorithm

**Quantum Approximate Optimization Algorithm (QAOA)**

Hybrid quantum-classical variational approach

**Cost funtion**

$H_C = -\sum_{(i,j)\in E} J_{ij}\sigma_i^z\sigma_j^z$

**Mixing Hamiltonian:**

$\mathrm{H_M} = -\sum_i \sigma_i^x$

Emploing both $H_C$ and $\mathrm{H_M}$,
we define the **wavefunction**:

$$|\Psi(\gamma,\beta)\rangle = \prod_{i=1}^{p} \boxed{e^{-i\beta_i H_M}\, e^{-i\gamma_i H_C}} |\Psi(0)\rangle$$

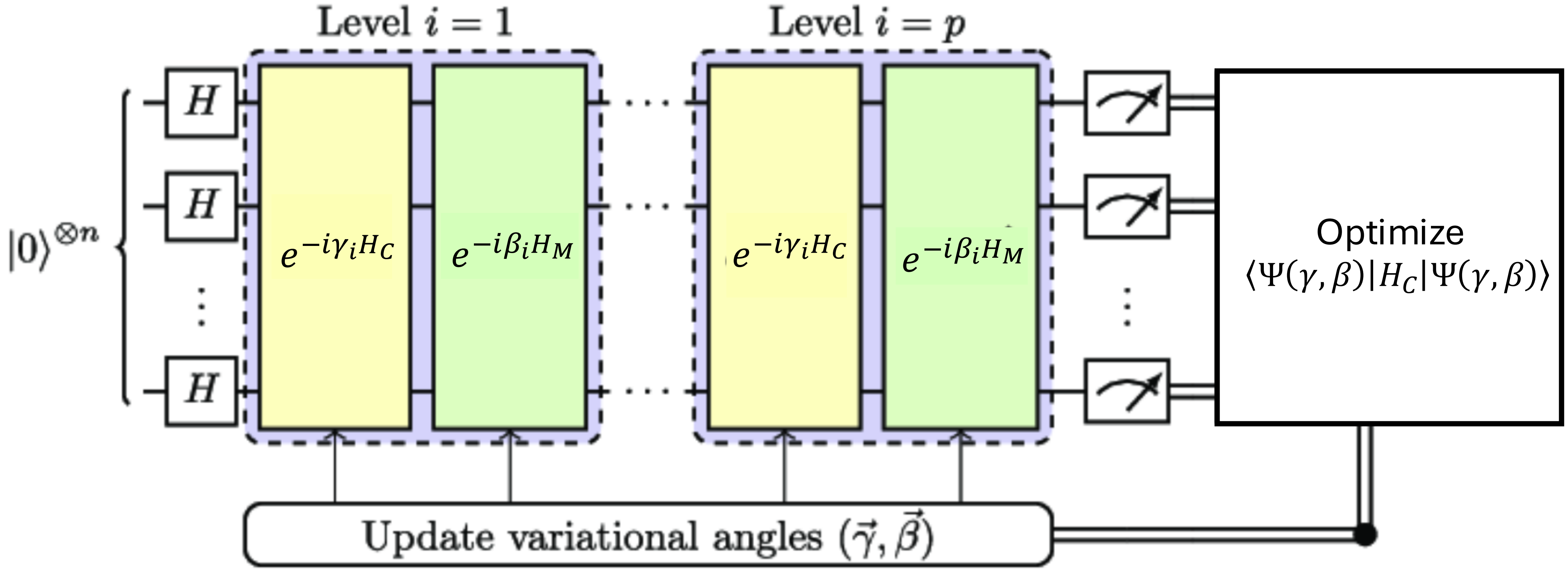
With the initial state: $|\Psi(0)\rangle = |+\rangle^{\otimes n}$

Measure of
the cost function
$\langle\Psi(\gamma,\beta)|H_C|\Psi(\gamma,\beta)\rangle$

Each term : $e^{-i\beta_i H_M}\, e^{-i\gamma_i H_C}$ → converted in **quantum gates**
Repeated $p$ times to approximate the solution. $p \dashrightarrow \infty$: QAOA can reach GS
Parameters $(\beta,\gamma)$→ optimized for maximum cut.

# Digital quantum computing:

MAX-CUT with Quantum Approximate Optimization Algorithm



# Universality in quantum computing

**Universality:**

*When a set of quantum gates can approximate any unitary transformation on a quantum system to arbitrary precision.*

A universal gate set allows for the construction of any quantum circuits and can be employed in the execution of any quantum algorithm.

Example of a **Universal Gate Set**: Hadamard ($H$), $T$ ($\pi/8$) gate and the $CNOT$ gate.

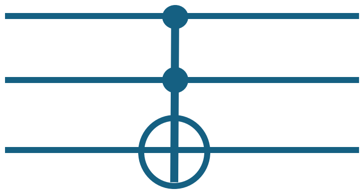
**Clifford Group**: includes gates that map Pauli operators to other Pauli operators.

Clifford gates: Pauli ($X, Y, Z$), Hadamard ($H$), Phase ($S$), and $CNOT$.

not universal - limited to states that can be classically simulated.

Universal Gate Set. **Clifford + T gate** → application in quantum error correction

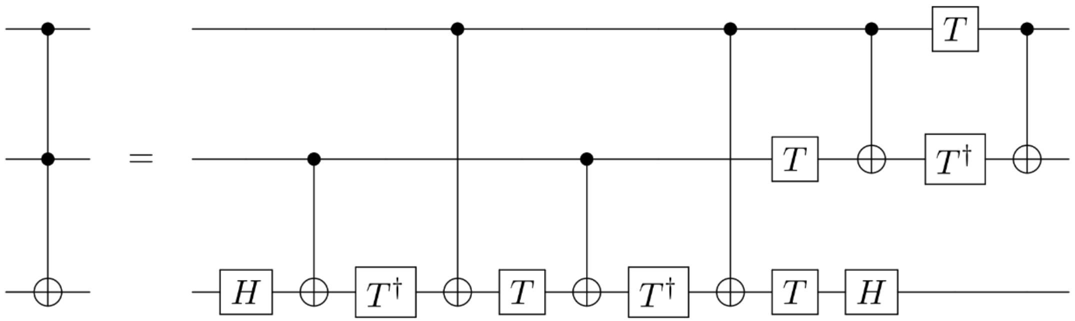
# Universality in quantum computing: Toffoli



**Toffoli gate** (CCNOT):

- flips the target qubit if both control qubits are in the state $|1\rangle$.
- represented as a controlled-controlled-NOT operation.
- universal in classical computation → employed for classical logic using quantum circuits
- useful for quantum error correction.

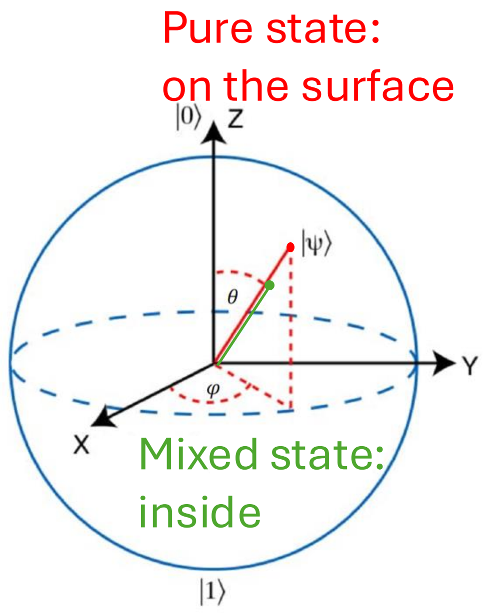
The Toffoli gate: can be decomposed using Clifford and T gates. (UNIVERSALITY)



# Quantum Computing: Quantum Errors

Origin of quantum errors

- **Decoherence**: The loss of quantum information due to interactions with the environment.
  - **Dephasing** affects the phase relationship between states → from a superposition of states to mixed states
  - **Decay** from the excited state |1⟩ to the ground state |0⟩, which reduces the quantum system's ability to maintain superpositions.
- **Gate Errors**: when quantum gates are applied imperfectly, to small deviations from the desired quantum operations arise.
  - **Bit-flip Errors**: The qubit state flips unintentionally between |0⟩ and |1⟩.
  - **Phase-flip Errors**: The relative phase of a qubit changes, altering its superposition state.
- **Measurement Errors**: Wrong measurement outcomes due to hardware limitations.
- **State Leakage**: Transition of qubits into non-computational states during measurement.



# Quantum Computing: Quantum Errors

**Quantum Error Mitigation**:

- Techniques reduce error impact on results without fully correcting errors.

**Quantum Error Correction**:

- Encodes **logical qubits** using multiple physical qubits to detect and correct errors.
- Requires significant qubit overhead

# Quantum Computing: Quantum Errors

**Quantum Error Mitigation**:

- Techniques reduce error impact on results without fully correcting errors.

**Quantum Error Correction**:

- Encodes **logical qubits** using multiple physical qubits to detect and correct errors.
- Requires significant qubit overhead

Noisy Intermediate-Scale Quantum (**NISQ**) Devices:

- limited qubits, operate with noise.
- Suitable for near-term algorithms like QAOA that tolerate certain errors.

**Fault-Tolerant** Quantum Computing:

- Enables reliable computations by detecting and correcting errors continuously.
- Uses logical qubits encoded with multiple physical qubits to correct errors in real-time.

# References


- *M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information. Cambridge: Cambridge University Press, 2010.*
- *G. Benenti, G. Casati, and G. Strini, Principles of Quantum Computation and Information. WORLD SCIENTIFIC, Apr 2004.*