JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

Quantum Computer Programming

Qiskit Section

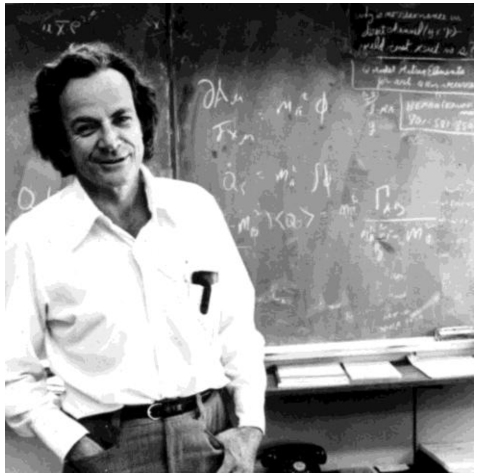
Andrés Muñoz-Moller

Quantum Information and Computing Team

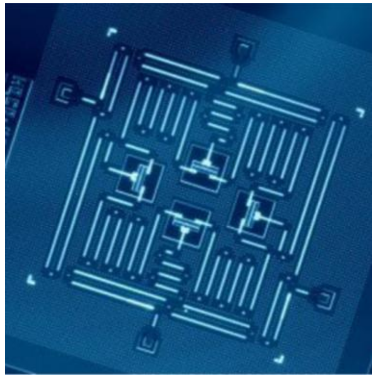
Contents of Qiskit Section

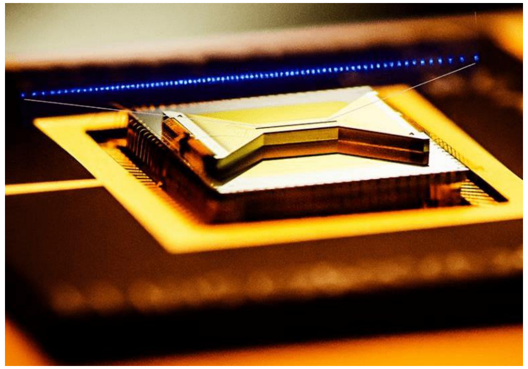
- Quantum Information and Quantum Computing Basics
- Quantum Circuits and Noisy Simulators
- Quantum Circuit Transpilation
- Fault-Tolerant Quantum Computing: QPE and Shor’s Algorithm
- Qiskit Primitives and Qiskit Runtime
- Variational Algorithms: VQE and QAOA

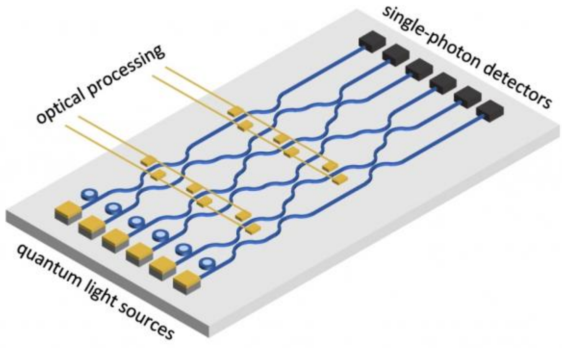
What is Quantum Computing (QC)?



- Proposed by Richard Feynman in 1982 for simulating quantum systems.
- QC use the principles of quantum mechanics to perform computations, exploiting quantum phenomena like superpositions, entanglement and decoherence.
- Several QC architectures:
  - Superconductors
  - Atom and Ion traps
  - Photonic







Classical Bits vs Quantum Bits

- Classical **Bit** → System that has classical states 0 or 1.
- **Quantum States** → Unit (column) vectors in a Hilbert space.
- **Qubit** (Quantum bit) → Quantum state in a two-dimensional Hilbert space.

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\end{pmatrix}$$

$\{|0\rangle, |1\rangle\} \longrightarrow$ Computational Basis

$$|\alpha|^2 + |\beta|^2 = 1, \quad \alpha, \beta \in \mathbb{C}$$

$\longrightarrow$ Probability Amplitudes

- **Superposition** of states = Linear combination of the basis states.

Dirac Notation

- **Bra** → N-dimensional column vector

$$|\psi\rangle = \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{N-1} \end{pmatrix}$$

- **Ket** → Conjugate transpose → Row vector

$$\langle\psi| = \begin{pmatrix} \alpha_0^* & \dots & \alpha_{N-1}^* \end{pmatrix}$$

- **Braket** → Inner product

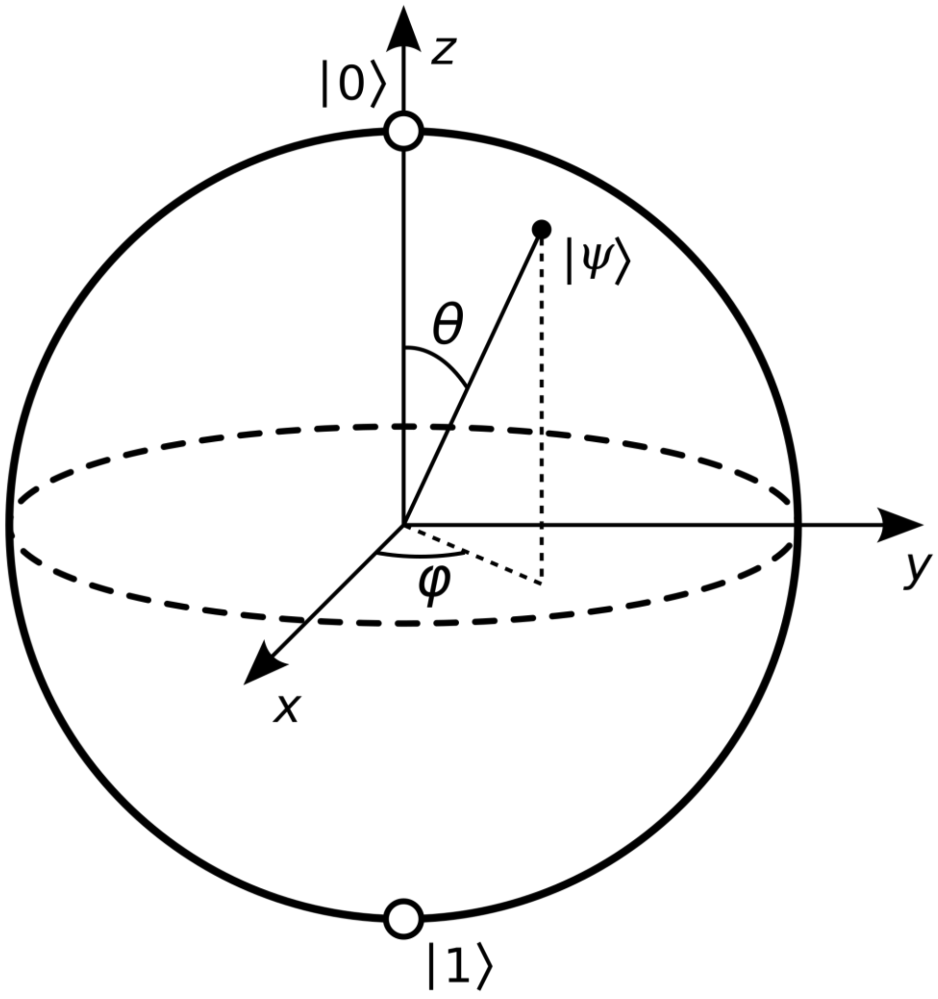
$$\langle\psi|\phi\rangle = \langle\psi| \cdot |\phi\rangle$$

Bloch Sphere

Single qubit quantum states can also be represented using the **Bloch sphere representation**.

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle,$$

$$\phi \in [0, 2\pi], \quad \theta \in [0, \frac{\pi}{2}]$$



Measuring a quantum state

- We cannot observe the state vector of a qubit directly.
- To obtain information from a qubit we must perform a **measurement** (usually in the computational basis) → Probabilistic!

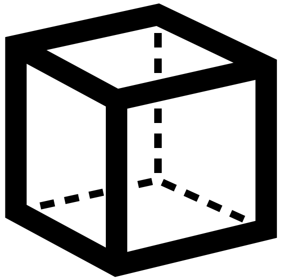
$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

$$|\psi\rangle \xrightarrow{measure} \begin{cases} 0, \text{with probability}\, p_0 = |\langle\psi|0\rangle|^2 = |\alpha|^2, \\ 1, \text{with probability}\, p_1 = |\langle\psi|1\rangle|^2 = |\beta|^2. \end{cases}$$

Qubit

$$|\psi\rangle \longrightarrow$$

Measurement



$$\longrightarrow$$

Bit

$$\{0, 1\}$$

Global and relative phases

- Vectors that differ by a **global phase** represent the same quantum state: the probability amplitudes upon measuring are equal for both vectors.

Global Phase

$$|\psi\rangle = e^{i\theta}|\phi\rangle$$

$$|\langle a|\psi\rangle|^2 = \left|e^{i\theta}\langle a|\phi\rangle\right|^2 = |e^{i\theta}|^2|\langle a|\phi\rangle|^2 = |\langle a|\phi\rangle|^2$$

- On the other hand, a **relative phase** can affect measurement outcomes (e.g. if measured in a basis other than the computational basis.)

Relative Phase

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle,$$

Multiple Bits/Qubits

- Classical state of multiple bits (**Bitstring**) → **Cartesian product.**

$$0 \times 0 = 00$$
$$0 \times 1 \times 0 = 010$$

- Quantum states of multiple qubits → **Tensor product.**

$$|0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = |01\rangle$$

Entanglement

- **Product States** → States that can be written as a tensor product.

$$|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$$

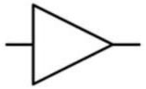
- **Entangled States** → States that **cannot** be written as a tensor product (e.g. Bell states).

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \neq |\psi_A\rangle \otimes |\psi_B\rangle$$

Classical Gates

- In a classical circuit, we perform operations on bits using **classical (logical) gates**.
- For single bits, the only gates available are the **buffer** and the **NOT** (inverter) gate.

Buffer

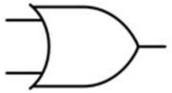


Input	Output
0	0
1	1

AND

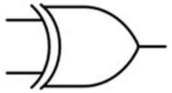
A	B	Output
0	0	0
1	0	0
0	1	0
1	1	1

OR



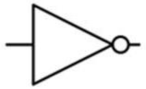
A	B	Output
0	0	0
1	0	1
0	1	1
1	1	1

XOR



A	B	Output
0	0	0
1	0	1
0	1	1
1	1	0

Inverter

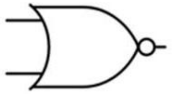


Input	Output
0	1
1	0

NAND

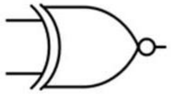
A	B	Output
0	0	1
1	0	1
0	1	1
1	1	0

NOR



A	B	Output
0	0	1
1	0	0
0	1	0
1	1	0

XNOR



A	B	Output
0	0	1
1	0	0
0	1	0
1	1	1

Quantum Gates

Conjugate transpose

$$UU^\dagger = U^\dagger U = I,$$

$$U^\dagger = U^{-1}$$

- As qubits correspond to quantum states, our quantum gates correspond to **unitary operations**.
- Quantum gates are **reversible**, whereas most classical logical gates are not (e.g. AND gate).
- In general, gates **do not** commute!

$$U_a U_b \neq U_b U_a$$

$$S = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{bmatrix}, \quad S^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & e^{-i\frac{\pi}{2}} \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Examples Single Qubit Quantum Gates

Operator	Gate(s)		Matrix
Pauli-X (X)	**X**	$\oplus$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)	**Y**		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)	**Z**		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)	**H**		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)	**S**		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\boldsymbol{\pi/8}$ **(T)**	**T**		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$

X Gate (NOT Gate)

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

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

Y Gate and Z Gate

$$Y|0\rangle = i|1\rangle, \quad Y|1\rangle = -i|0\rangle$$

Global phases

$$Z|0\rangle = |0\rangle, \quad Z|1\rangle = -|1\rangle$$

Eigenvectors

Hadamard Gate

The **Hadamard Gate** creates an equal superposition state if given a computational basis state.

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$$

Relative Phase

$$H|1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$$

Pauli X basis

$$\{|+\rangle, |-\rangle\} \qquad X|+\rangle = |+\rangle, \quad X|-\rangle = -|-\rangle$$

Phase Gate

The **Phase Gate (S)** induces a pi-half phase.

$$S|0\rangle = |0\rangle\,, \quad S|1\rangle = i|1\rangle\,, \quad S = \sqrt{Z}$$

$$S|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm i|1\rangle) = |\pm i\rangle$$

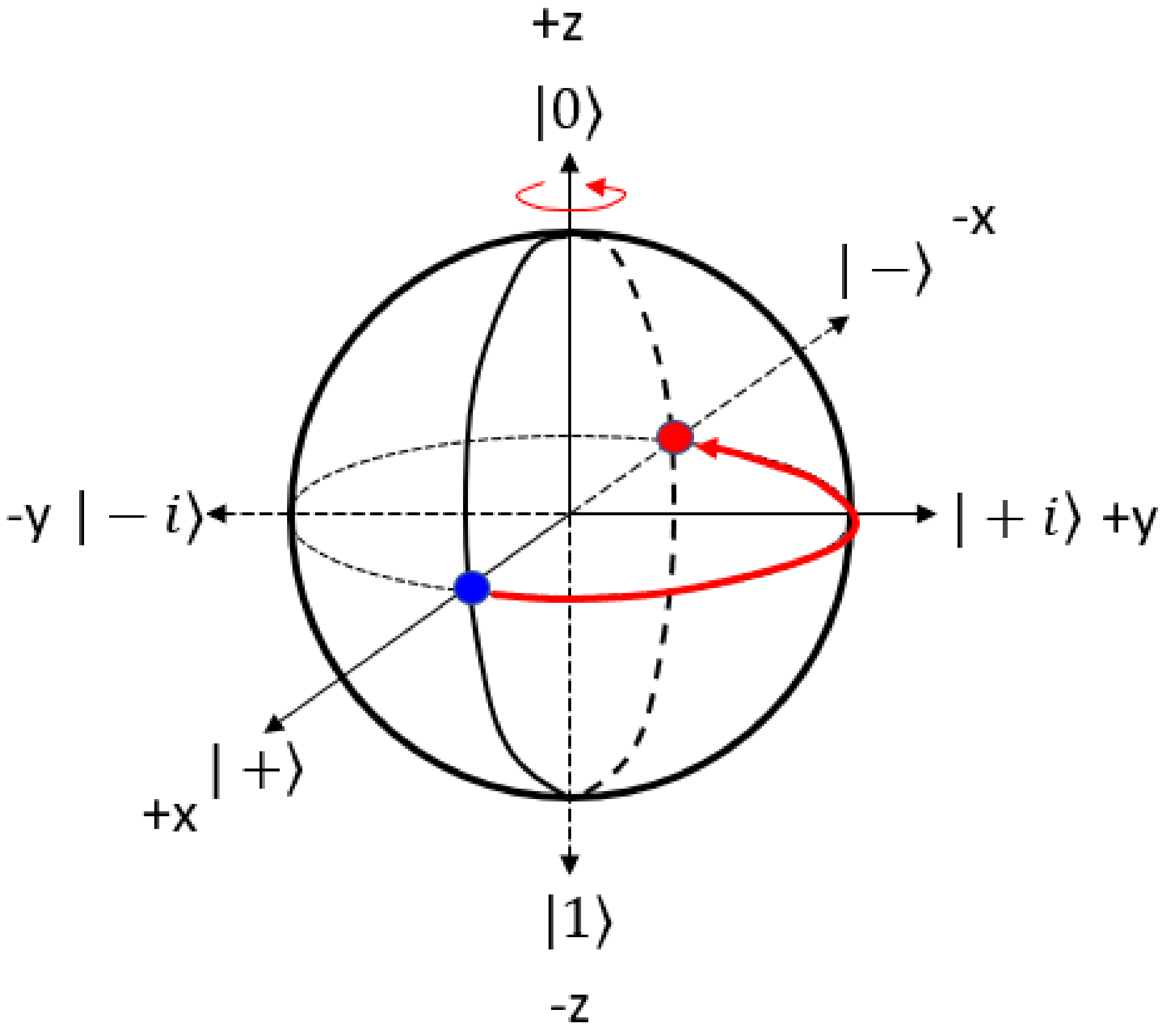
Pauli Y basis

$$\{|+i\rangle\,, |-i\rangle\} \quad Y|+i\rangle = |+i\rangle\,, \quad Y|-i\rangle = -|-i\rangle$$

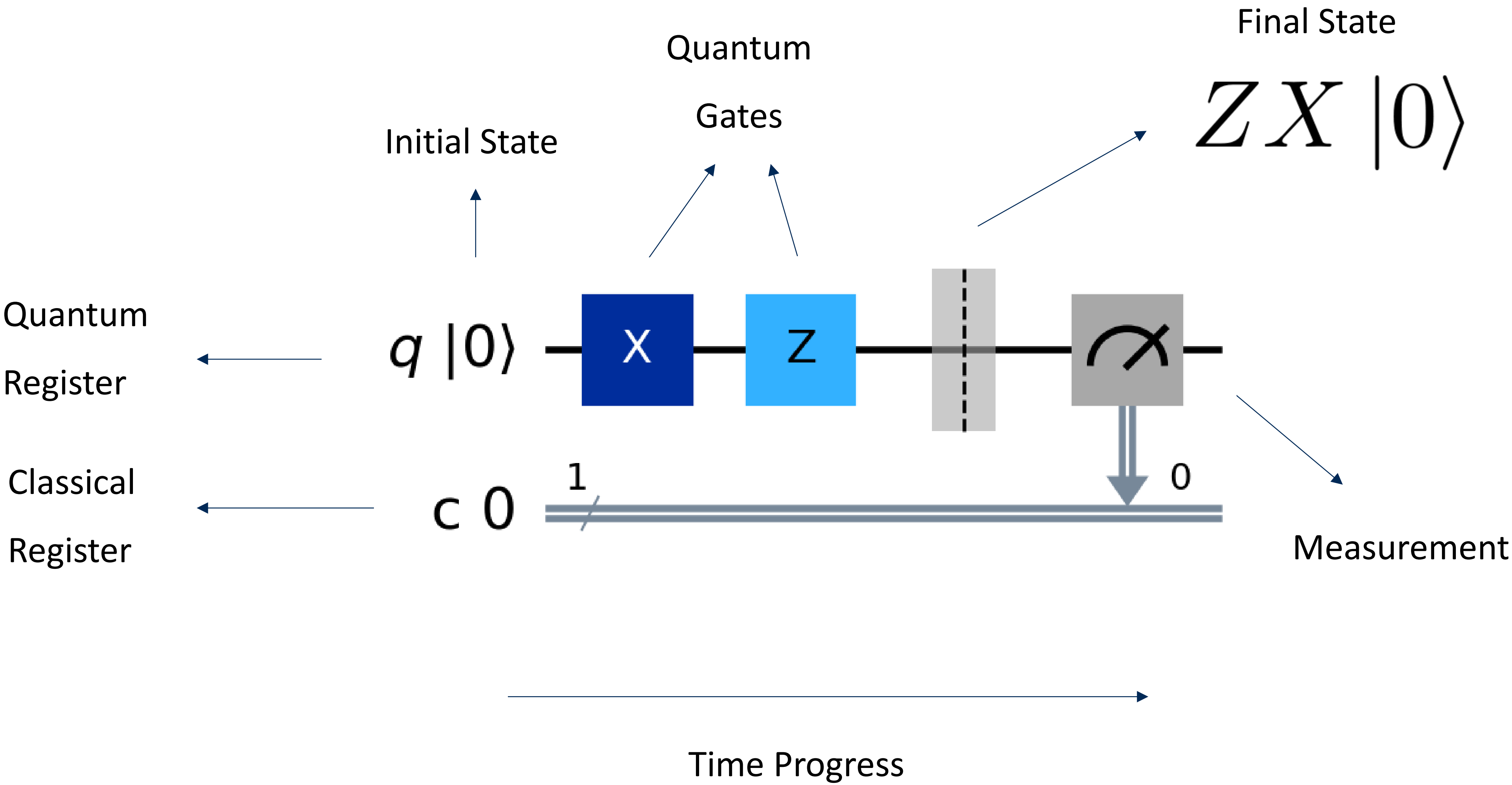
Rotations in the Bloch sphere

Single qubit gates are rotations around the Bloch sphere.

- (X ,Y, Z) Gate → 180° rotation over the (X, Y, Z) Axis.
- H Gate → 90° rotation over Y-Axis + 180° rotation over the X-Axis.
- S Gate → 90° rotation over the Z-Axis.

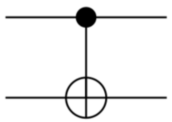


Quantum Circuit Model



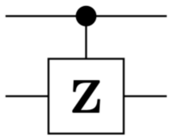
Multiqubit Gates

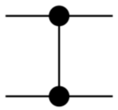
Controlled Not (CNOT, CX)



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

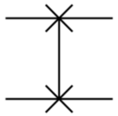
Controlled Z (CZ)





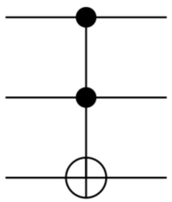
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

SWAP



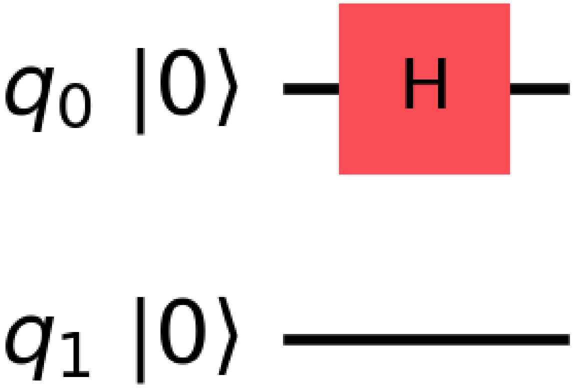
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Toffoli (CCNOT, CCX, TOFF)



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

Local Gates



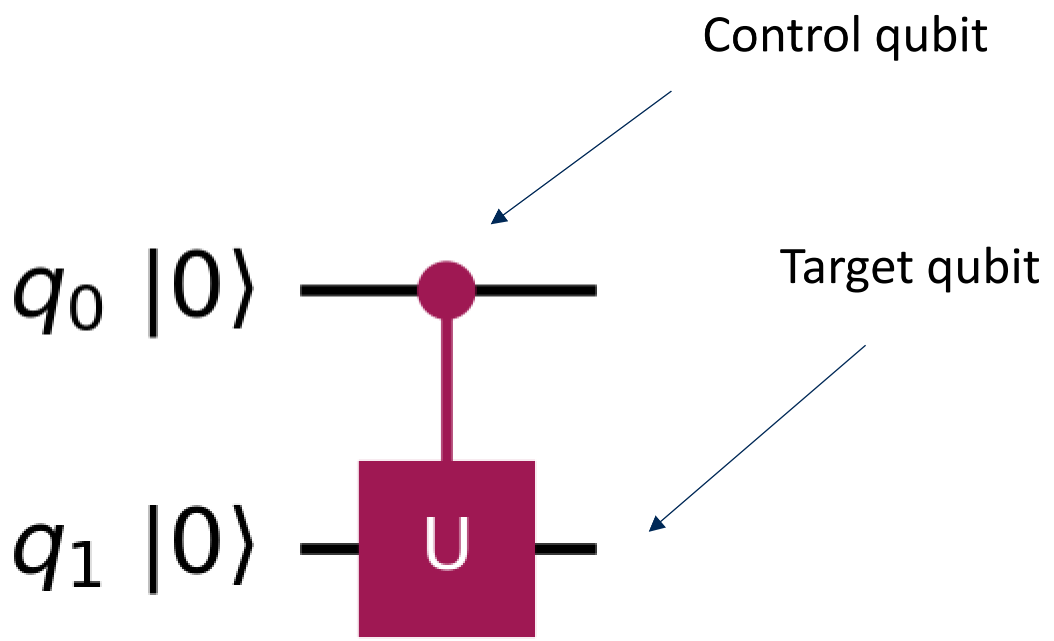
- **Local gates** → Gates that act non-trivially only on a single qubit.

$$(H \otimes I)\,|00\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \otimes \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}|00\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 0 & 1 & 0\\ 0 & 1 & 0 & 1\\ 1 & 0 & -1 & 0\\ 0 & 1 & 0 & -1\end{pmatrix}\begin{pmatrix}1\\0\\0\\0\end{pmatrix}$$

$$= \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = |+\rangle\,|0\rangle$$

Controlled Gates

- Control gates apply a unitary gate to the **target** qubit(s) if the **control** qubit(s) is 1.
- Common examples: **CNOT** (CX), **CZ** and **Toffoli** gate.



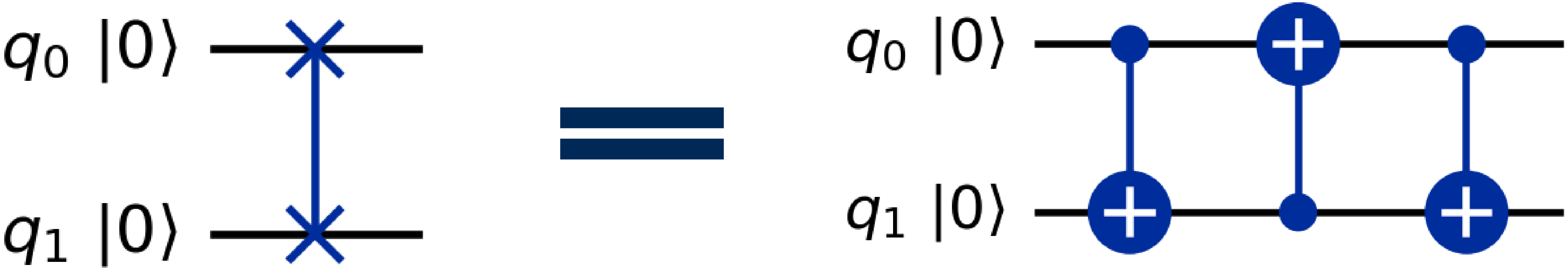
$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\begin{aligned}
\mathrm{CNOT}\,|0\rangle\,|0\rangle &= |0\rangle\,|0\rangle \\
\mathrm{CNOT}\,|0\rangle\,|1\rangle &= |0\rangle\,|1\rangle \\
\mathrm{CNOT}\,|1\rangle\,|0\rangle &= |1\rangle\,X\,|0\rangle = |1\rangle\,|1\rangle \\
\mathrm{CNOT}\,|1\rangle\,|1\rangle &= |1\rangle\,X\,|1\rangle = |1\rangle\,|0\rangle
\end{aligned}$$

SWAP Gate

- The **SWAP gate** is a gate that swaps the states of two qubits.

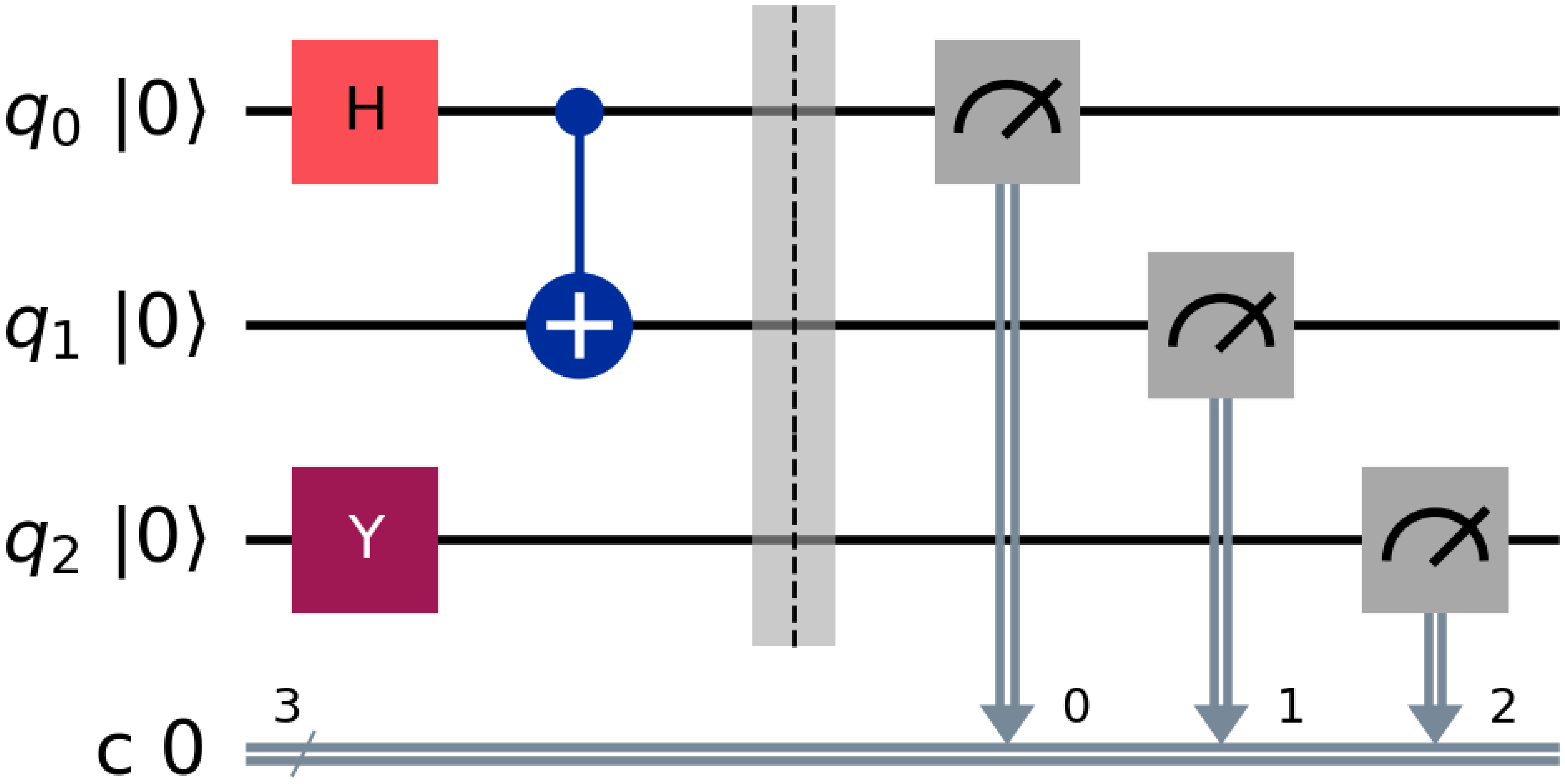
$$\mathrm{SWAP}\,|\psi\rangle\,|\phi\rangle = |\phi\rangle\,|\psi\rangle$$



Quantum Circuit Model

$$(\mathrm{CNOT}_{01} \otimes I)(H \otimes I \otimes Y)\,|000\rangle$$

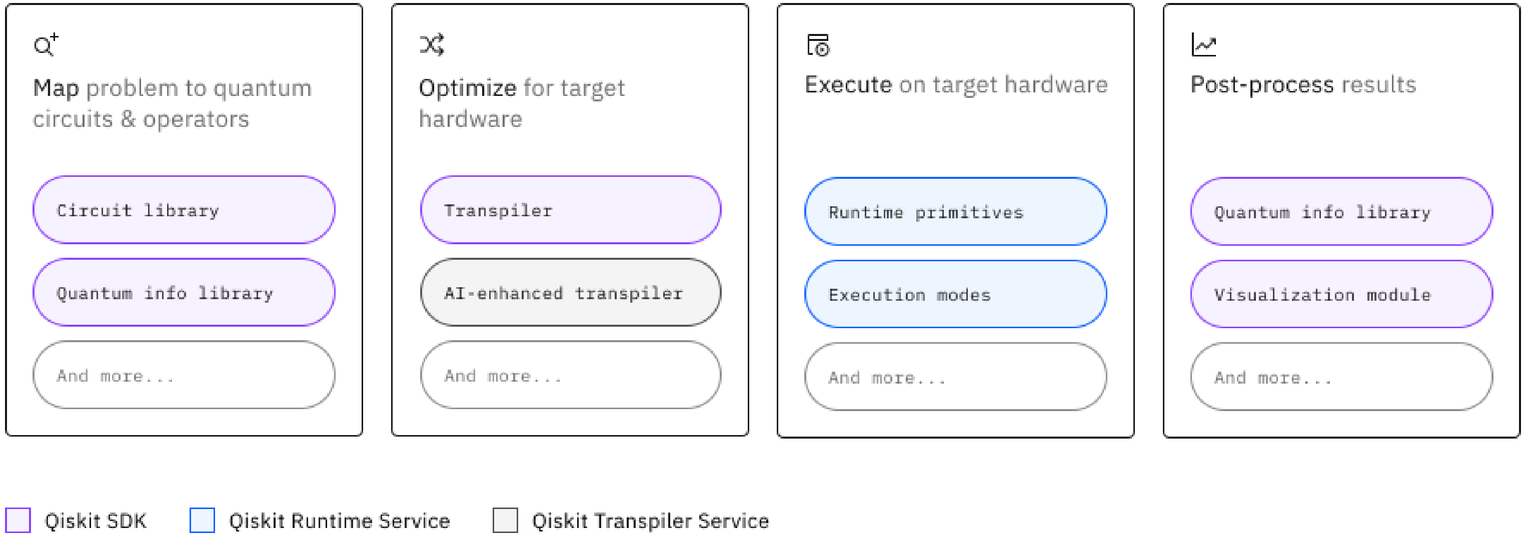
$$\mathrm{CNOT}_{01} Y_2 H_0\,|000\rangle$$



What is Qiskit?

Qiskit is a software stack for quantum computing.

- **Qiskit SDK**: Open-source SDK for working with quantum computers (or simulators) at the level of quantum circuits, quantum operators and primitives functions.
- **Qiskit Runtime**: Environment for executing workloads in IBM Quantum Computers.
- **Qiskit Ecosystem**: Collection of tools created by the community (researchers and developers).



Things to do before starting

- Install Python 3.12 (Preferably using Anaconda)
- Optional: Create a minimal environment.
- Install Qiskit and Qiskit-related packages:
  - pip install qiskit
  - pip install qiskit-ibm-runtime
  - pip install qiskit[visualization]
  - pip install qiskit-aer
  - pip install rustworkx
- Optional: Install VS Code (or use Jupyter Notebook/Google Colab) and Graphviz.

ANACONDA®

```
(base) C:\>conda create -n qiskit-course python=3.12
```

```
(base) C:\>conda activate qiskit-course
```

```
(qiskit-course) C:\>pip install qiskit
```

JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ