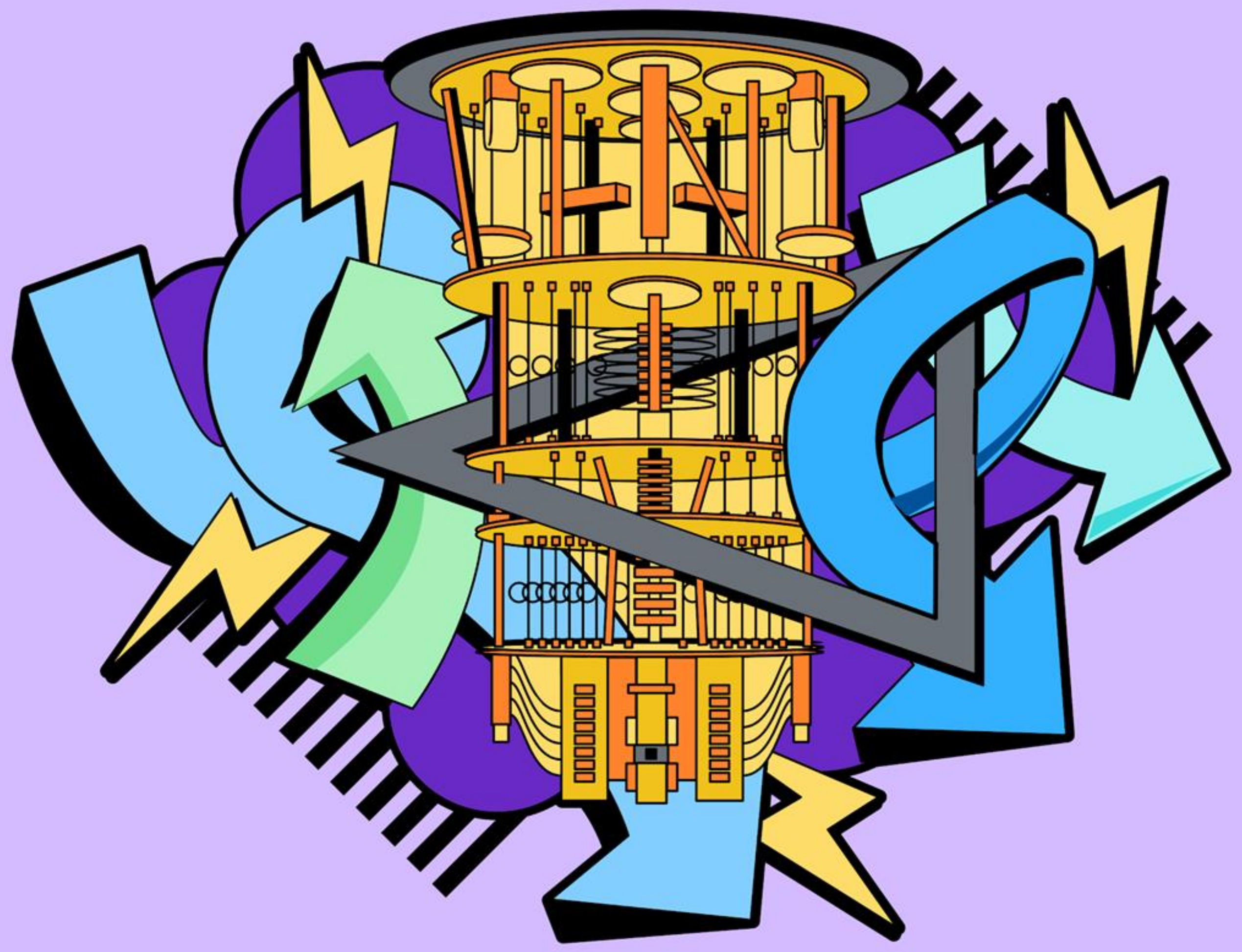


Qiskit

Qiskit | Fall Fest 2022

Qiskit | Fall Fest
Morocco 2022

Qiskit Introduction

Hinde Adnani

Master degree in Physics, Faculty of Sciences,

Mohammed V University in Rabat.

Qiskit Advocate with Qiskit community.

Co-founder of QMorocco with Qworld community.

AGENDA

Qiskit

- Creating a Quantum Circuit
- Single & two Gates
- Drawing a Quantum Circuit
- Ressources to learn Qiskit

Quantum Circuit

- A quantum circuit is a model for quantum computation, where the steps to solve the problem are quantum gates performed on one or more qubits.
- A quantum gate is an operation applied to a qubit that changes the quantum state of the qubit.
- A quantum circuit is composed of quantum and classical registers.

```
qc = QuantumCircuit(qreg,creg)
```

```
qreg = QuantumRegister(num_qubits)
```

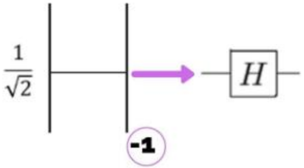
```
creg = ClassicalRegister(num_qubits)
```

Hadamard Gate

- The Hadmard Gate is particulary important
- It can be used to create a superposition of the |0> and |1> states.

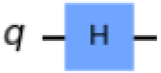
$$|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \qquad |1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

- One of the most important gates for quantum computing



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

```
qc = QuantumCircuit(1)
qc.h(0)
qc.draw()
```

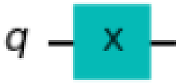


Pauli-X Gate

- The X-Gate is directly analogous to the classical NOT gate.
- It transforms $|0\rangle$ to $|1\rangle$ and $|1\rangle$ to $|0\rangle$.

$$-\boxed{X}- = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = |1\rangle\langle 0| + |0\rangle\langle 1|$$

```python
qc = QuantumCircuit(1)
qc.x(0)
qc.draw()
```



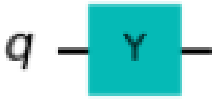
Pauli-Y gate

- Similary to Pauli-X Gate, the Y Gate represents a rotation of around the y axis by radians

- It transforms $|0\rangle$ to i $|1\rangle$ and $|1\rangle$ to -i $|0\rangle$.

$$-\boxed{Y}- = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = i\,|1\rangle\,\langle 0| - i\,|0\rangle\,\langle 1|$$

```
qc = QuantumCircuit(1)
qc.y(0)
qc.draw()
```

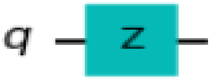


Pauli-Z gate

- The Z gate is actually a special case of the phase shift gate where $\phi = \pi = 180°$.
- It has no effect on $|0\rangle$ but transforms $|1\rangle$ to - $|1\rangle$.

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

```
qc = QuantumCircuit(1)
qc.z(0)
qc.draw()
```



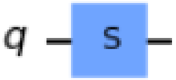
S gate

- The phase gate (S gate) is a single-qubit operation defined by:

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

- The S gate is also known as the phase gate or Z90 gate, because it represents a 90-degree rotation around the z-axis.

```
qc = QuantumCircuit(1)
qc.s(0)
qc.draw()
```



T gate

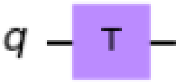
- The T gate is a single-qubit operation defined by:

$$T = \begin{pmatrix} 1 & 0 \\ 0 & \exp\left(\frac{i\pi}{4}\right) \end{pmatrix}$$

- The T gate is related to the S gate by the relationship:

$$S = T^2$$

```
qc = QuantumCircuit(1)
qc.t(0)
qc.draw()
```

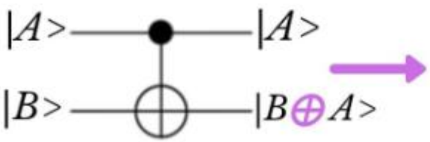


CNOT Gate

- Controlled NOT gate is a generalisation of the classical XOR:

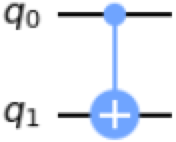
$$CNOT\,|A,B> = |A,B \oplus A>$$

- Acts on two qubits- one the contrôle and the other the target



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

```
qc = QuantumCircuit(2)
qc.cx(0,1)
qc.draw()
```

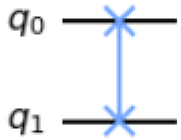


Swap

- The SWAP gate is two-qubit operation. Expressed in basis states, the SWAP gate swaps the state of the two qubits involved in the operation:

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

```
qc = QuantumCircuit(2)
qc.swap(0,1)
qc.draw()
```

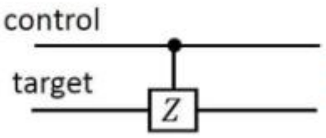


CZ-Gate

- Controlled Z gate is the controlled version of the Z gate.
- The effect on a two qubit state from the computational basis can be summed up as

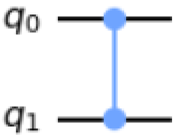
$$CZ\,|A,B> = (-1)^{A.B}\,|A,B>$$

- Acts on two qubits- one the control and the other the target



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

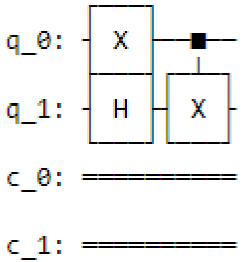
```
qc = QuantumCircuit(2)
qc.cz(0,1)
qc.draw()
```



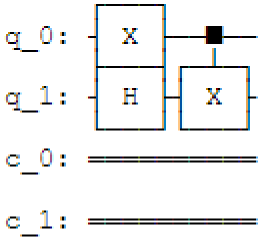
Drawing a Quantum circuit

- When building a Quantum circuit, it often helps to draw the circuit. This is supported natively by a QuantumCircuit object.
- You can either call print() on the circuit or draw() methode on the object.

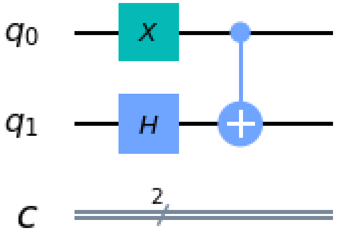
```
print(qc)
```



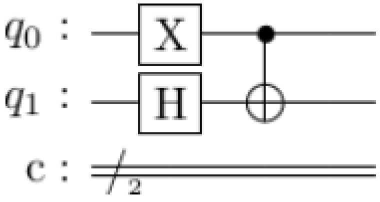

```
qc.draw()
```




```
qc.draw(output='mpl')
```




```
qc.draw(output='latex')
```



Ressources

- Qiskit textbook : https://qiskit.org/textbook/preface.html
- Qmorocco Workshops : Qbronze, Qsilver and Qnickel https://qworld.net/qmorocco/
- IBM Global Summer School: https://qiskit.org/events/summer-school/
- IBM challenges: https://challenges.quantum-computing.ibm.com/fall-2022

Thank you!