

Introducere in Qiskit

Bianca Necula



Fig.1

Qiskit este un framework open-source care ofera instrumente pentru crearea programelor cuantice si rulara acestora pe dispozitive cuantice cu ajutorul platformei IBM Q Experience sau pe computerul personal pe simulatoare. A fost creat de catre IBM Research in 2017, iar in prezent are sute de contribuitori din intreaga lume. Varianta principala de Qiskit foloseste Python ca limbaj de programare, dar exista si versiuni mai putin folosite de Swift si JavaScript.

Acest tutorial nu va face referire la conceptul de calcul cuantic, ci va fi de ajutor pentru dezvoltarea unui simplu program in Qiskit. Este necesara cunoasterea limbajului Python.

Qiskit se poate folosi pentru dezvoltarea de software cuantic atat la nivelul limbajului de asamblare OpenQASM (Open Quantum Assembly Language), cat si la un nivel abstractizat high-level. Astfel, exista 4 componente:

- **Terra** instrumente pentru circuite cuantice cu ajutorul portilor logice cuantice
- **Aqua** functii folositoare in domenii precum chimie, inteligenta artificiala, finante
- **Aer** simulari pe dispozitive cuantice mici
- **Ignis** instrumente pentru optimizari si detectare a "zgomotului"

Pentru inceput este necesara instalarea locala a Qiskit si Jupyter Notebook. Odata ce mediul este pregatit, vom incepe prin a importa toate functionalitatile modulului Qiskit.

```
from qiskit import *
```

Cea mai importanta parte in calculul cuantic este folosirea de qubiti in loc de bitii standard. Un bit pe calculatorul clasic este reprezentat de valorile 0 sau 1, in timp ce pe calculatorul cuantic starea un qubit, conform mecanicii cuantice, poate fi 1, 0, sau oricare superpozitie cuantica a acestora. Vom acoperi aceste notiuni in tutorialele viitoare.

Vom continua prin a contrui un registru cuantic cu 2 qubiti si unul clasic pentru a efectua masuratorile finale.

```
q = QuantumRegister(2)
c = ClassicalRegister(2)
```

Acum construim circuitul cu ajutorul celor doua. In orice moment dupa, putem vizualiza circuitul si schimbarile care apar pe acesta (ne folosim de biblioteca matplotlib).

```

circ = QuantumCircuit(q, c)
%matplotlib inline
circ.draw()

```

Output:

q0_0 :

q0_1 :

c0_0 :

c0_1 :

Putem observa ca circuitul este gol momentan. Putem sa aplicam acum pe unul din qubiti o poarta logica cuantica. Exista multe astfel de porti: Hadamard (H), Pauli-X (X), Pauli-Y (Y), Pauli-Z (Z), CNOT (CX) etc. Vom aplica Hadamard pe qubitul 0 si vom afisa circuitul cu parametrul 'mpl' pentru o afisare grafica.

```

circ.h(q[0])
circ.draw('mpl')

```

Output:

