

RELAZIONE TELETRASPORTO QUANTISTICO

Lavoro svolto da:

-Magno Alessandro : 4478234

Funzionamento

Il teletrasporto quantistico è il trasferimento degli stati quantistici da un qubit ad un altro. Questo non significa il trasporto fisico di un qubit da un posto ad un altro, ma si intende un trasferimento dell'informazione quantistica da un qubit ad un altro.

In un computer classico, copiare qualcosa e trasferirla da una parte ad un'altra non è un problema. Nei computer quantistici, l'azione di provare a trasferire qualcosa copiandola non è possibile perché nel momento in cui si copia, in realtà implicitamente si sta facendo una misura, la quale distrugge lo stato che si sta cercando di trasferire da un punto A ad un punto B. Per aggirare questo problema, si utilizza l'entanglement.

Implementazione

Passo a) costruzione dello stato $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ con $|\alpha|^2 = 0.9$ e $|\beta|^2 = 0.1$

```
1. from qiskit import *
2. import math
3.
4. vector = [
5.     math.sqrt(0.9),
6.     math.sqrt(0.1)
7. ]
8.
9. q = QuantumRegister(3)
10. c = ClassicalRegister(3)
11.
12. #costruzione circuito quantistico costituito da 3 quantum bits e 3 bits classici
13. qc = QuantumCircuit(q,c)
14.
15. #inizializzazione del vettore nel qubit 0
16. qc.initialize(vector, q[0])
17. qc.barrier()
18.
```

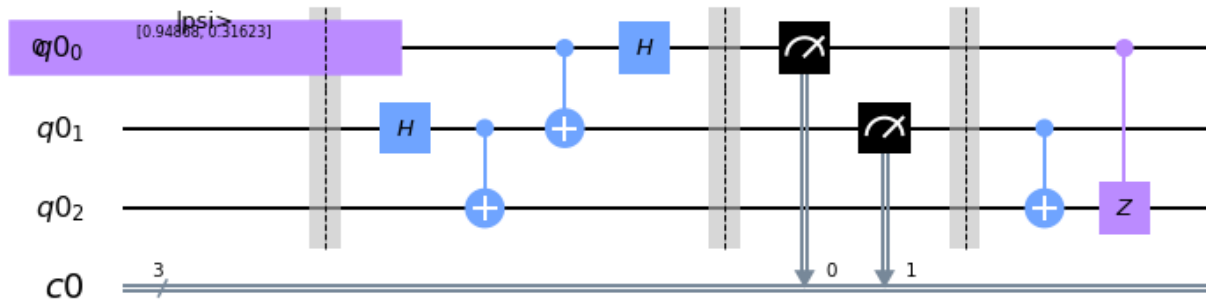
Passo b) si implementi un esperimento di teletrasporto fra Alice(qubit 0) e Bob(qubit 2)

```
1. #inizio del teletrasporto: trasporto lo stato del qubit 0 al qubit 2
2. #il qubit 1 viene utilizzato come risorsa per questo processo
3.
4. #creazione entanglement tra q1 e q2
5. qc.h(1)
6.
7. #applico CNOT x a q1 e q2
8. qc.cx(1,2)
9.
10. #teletrasporto dice di applicare 2 trasformazioni
11. #CNOT x a q0 e q1
12. qc.cx(0,1)
13.
14. #HADMART a q0
15. qc.h(0)
16.
17. #2 misure su q0 e q1 --> li registro nei bit classici
18. qc.barrier()
19. qc.measure([0,1], [0,1])
20.
```

```

21. #applico ancora 2 trasform. CNOT x e CNOT z
22. qc.barrier()
23. qc.cx(1,2)
24. qc.cz(0,2)
25.
26. #stampa circuito
27. qc.draw(output='mpl')

```

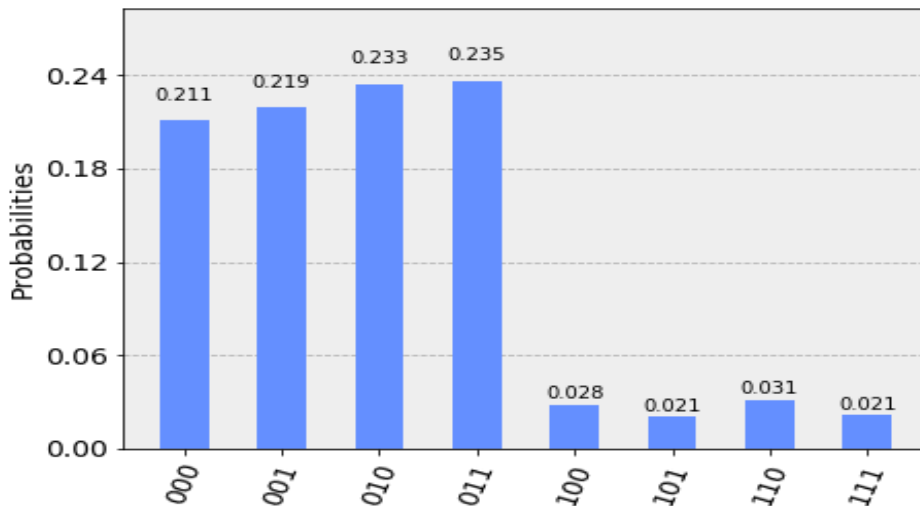


Per controllare che lo stato finale di Bob (qubit 2) sia quello corretto, effettuo una misura statistica dello stato finale teletrasportato. A priori, mi aspetto che dopo la misura, il sistema collasserà il più delle volte nello stato 0 (probabilità 0.9) e meno volte nello stato 1 (probabilità 0.1).

```

1. #misuro q2 e lo metto nel bit classico 2
2. qc.measure(2,2)
3. simulator = Aer.get_backend('qasm_simulator')
4. result = execute(qc, backend = simulator, shots = 1024).result()
5. counts = result.get_counts()
6.
7. from qiskit.tools.visualization import plot_histogram
8. plot_histogram(counts)

```



Si nota che i risultati sull'asse delle x nell'istogramma sono ordinati come $c_2c_1c_0$. Si può vedere che i risultati dove $c_2 = 0$ sono maggiori rispetto ai risultati dove $c_2 = 1$.

Passo c) si ripetano i punti precedenti con un altro stato in cui $|\alpha|^2 = 0.3$ e $|\beta|^2 = 0.7$.

```

1. vector = [
2.     math.sqrt(0.3),
3.     math.sqrt(0.7)

```

Il resto del codice è uguale al precedente. Ciò che cambia è il risultato, confermato dall'istogramma seguente. Si può notare che, dopo la misura, il sistema collassa il più delle volte nello stato 1 (probabilità 0.7) e meno volte nello stato 0 (probabilità 0.3).

