

Superdense coding

Fondamenti di computazione quantistica

Giacomo Carloni <S4816683@studenti.unige.it>

Valeria Rebagliati <S4852454@studenti.unige.it>

Marco Zucca <S4828628@studenti.unige.it>

May 10, 2023

1 Introduzione: Cosa è il super dense coding?

È un protocollo quantistico che permette di inviare un certo numero di bit logici codificandoli con un numero inferiore di qubit.

2 Funzionamento

Supponiamo di avere due attori che chiameremo Alice (A) e Bob (B); Alice vuole mandare dei bit, nel nostro esempio sono 2, di informazione logica a Bob utilizzando un solo qubit. La precondizione per cui questo avvenga è che i due attori condividano uno stato *entangled* composto nel seguente modo:

$$\begin{aligned} |\beta_{00}\rangle &= \frac{1}{\sqrt{2}}(|0\rangle_A |0\rangle_B + |1\rangle_A |1\rangle_B) = \\ &= \frac{|00\rangle + |11\rangle}{\sqrt{2}} \end{aligned}$$

il qubit con pedice A appartiene ad Alice e quello con pedice B a Bob.

Vediamo di seguito le combinazioni possibili:

1. Alice invia a Bob il messaggio **00**:
Alice non dovrà applicare nessuna porta (per chiarezza viene applicata la porta *Identità*), quindi lo stato ricevuto da Bob sarà lo stesso.
2. Alice invia a Bob il messaggio **10** Alice applicherà la porta la porta Z, ottenendo lo stato

$$|\beta_{00}\rangle \xrightarrow{Z} \frac{1}{\sqrt{2}}(|0\rangle_B \otimes |0\rangle_B - |1\rangle_B \otimes |1\rangle_B)$$

3. Alice invia a Bob il messaggio **01** Alice applicherà la porta X, ottenendo lo stato

$$|\beta_{00}\rangle \xrightarrow{X} \frac{1}{\sqrt{2}}(|1\rangle_B \otimes |0\rangle_B + |0\rangle_B \otimes |1\rangle_B)$$

4. Alice invia a Bob il messaggio **11** Alice applicherà la porta iY , ottenendo lo stato

$$|\beta_{00}\rangle \xrightarrow{iY} \frac{1}{\sqrt{2}}(|0\rangle_B \otimes |1\rangle_B - |1\rangle_B \otimes |0\rangle_B)$$

Adesso Bob è in possesso del suo qubit e di quello di Alice, ma avendo quattro possibili stati inviati da Alice, Bob deve misurare con certezza lo stato che gli arriva per poter conoscere il messaggio originario.

Si nota che le quattro possibilità sono **stati ortogonali di Bell**, quindi Bob è in grado di ottenere misure con probabilità 100% (come vedremo in seguito nella simulazione). A questo punto Bob è in grado di "decodificare" il messaggio inviato da Alice applicando in sequenza una porta CNOT su i suoi due qubit e una porta di Hadamard al primo qubit.

$$\xrightarrow{I} \frac{1}{\sqrt{2}}(|0\rangle_B \otimes |0\rangle_B + |1\rangle_B \otimes |1\rangle_B) \xrightarrow{CNOT} \frac{1}{\sqrt{2}}(|0\rangle_B + |1\rangle_B) \otimes |0\rangle_B \xrightarrow{H} |00\rangle_B$$

$$\xrightarrow{X} \frac{1}{\sqrt{2}}(|1\rangle_B \otimes |0\rangle_B + |0\rangle_B \otimes |1\rangle_B) \xrightarrow{CNOT} \frac{1}{\sqrt{2}}(|1\rangle_B + |0\rangle_B) \otimes |1\rangle_B \xrightarrow{H} |01\rangle_B$$

$$\xrightarrow{Z} \frac{1}{\sqrt{2}}(|0\rangle_B \otimes |0\rangle_B - |1\rangle_B \otimes |1\rangle_B) \xrightarrow{CNOT} \frac{1}{\sqrt{2}}(|0\rangle_B - |1\rangle_B) \otimes |0\rangle_B \xrightarrow{H} |10\rangle_B$$

$$\xrightarrow{iY} \frac{1}{\sqrt{2}}(|0\rangle_B \otimes |1\rangle_B - |1\rangle_B \otimes |0\rangle_B) \xrightarrow{CNOT} \frac{1}{\sqrt{2}}(|0\rangle_B - |1\rangle_B) \otimes |1\rangle_B \xrightarrow{H} |11\rangle_B$$

Misurando in base canonica Bob ottiene i bit logici inviati originariamente da Alice.

3 Confronto con il Teletrasporto quantistico

La differenza tra questi due protocolli, spiegata in breve, è la seguente:

il Teletrasporto quantistico è un processo in cui lo stato di un qubit può essere trasmesso mediante l'uso di bit logici e stato *entangled*, mentre il Superdense coding è una procedura che trasmette due bit logici con l'uso di un singolo qubit.

4 Codice

Abbiamo scelto di suddividere le funzioni nel modo seguente per evidenziare i passaggi logici della creazione, della trasmissione e della decodifica. Per comprendere meglio i vari casi (i messaggi che Alice può inviare a Bob) abbiamo deciso di ripetere l'esperimento per ognuno di essi.

4.1 Funzioni d'appoggio

Le seguenti funzioni serviranno a supporto dei vari esperimenti. La prima `createEntangled()` serve appunto a creare uno stato *entangled* condiviso tra A e B. Per fare ciò creiamo un circuito con 2 qubit, gli applichiamo una porta di Hadamard e in seguito una porta CNOT.

```
def createEntangled():
    es = QuantumCircuit(2)
    es.h(1)
    es.cx(1, 0)
    return es
```

La funzione `decode()` fa l'esatto opposto della funzione precedente, infatti applica una porta CNOT e in seguito una porta di Hadamard per distruggere l'*entanglement*.

```
def decode(de):
    de.cx(1,0)
    de.h(1)
    return de
```

La funzione `encode()` decide quale porta applicare, a seconda del messaggio inviato da Alice.

```
def encode(sdc,msg):
    if msg == "00":
        sdc.i(1)
        return sdc
    elif msg == "01":
        sdc.x(1)
        return sdc
    elif msg == "10":
        sdc.z(1)
        return sdc
    else :
        sdc.y(1)
        return sdc
```

La funzione `misurazioniSim()` ci servirà alla fine per effettuare le misurazioni sul simulatore e raccogliere i dati dell'esperimento.

```
def misurazioniSim(sdc):  
    job = execute(sdc, backend = deviceSim)  
    device_result = job.result().get_counts()  
    return device_result;
```

La funzione `misurazioniReal()` serve, invece, per le misurazioni su un computer quantistico reale e raccogliere i dati dell'esperimento.

```
def misurazioniReal(sdc):  
    job = execute(sdc, backend = deviceReal, shots = 1024)  
    print(job.job_id())  
    job_monitor(job)  
    device_result = job.result().get_counts()  
    return device_result
```

4.2 Misurazioni Reali vs Simulate

Per ogni caso descritto precedentemente abbiamo effettuato misurazioni sia sul simulatore che su un vero computer quantistico fornito da IBM, nelle pagine successive sono presenti due tipi di istogrammi, uno verde che rappresenta i dati ricevuti dalla simulazione e uno blu che rappresenta i dati del computer quantistico.

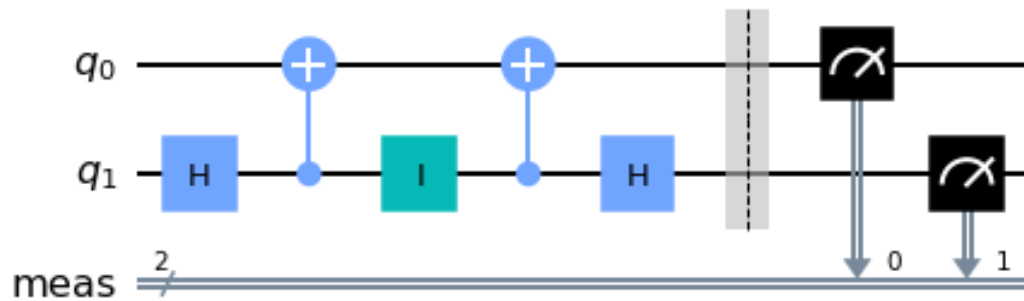
Come si può notare i vari istogrammi blu sono diversi dai rispettivi istogrammi verdi; infatti sugli istogrammi blu anche se c'è un picco sulla decodifica corretta (l'unica presente negli istogrammi verdi), in alcuni casi vengono generate decodifiche errate, questo è dato dal fatto che il computer quantistico è soggetto a interferenze e non sempre è in grado di fornire risultati esatti; per questo motivo dobbiamo affidarci alla probabilità per poter dire qual'è il risultato, eseguendo il programma per più volte (nel nostro esperimento 1024).

4.3 Porta Identità

Questo è il caso 1 della sezione "Funzionamento" in questo caso, dopo aver creato lo stato entangled, Alice invia il messaggio "00" quindi la porta applicata da `encode()` sarà quella **Identità** con il comando `sdc.i(1)`.

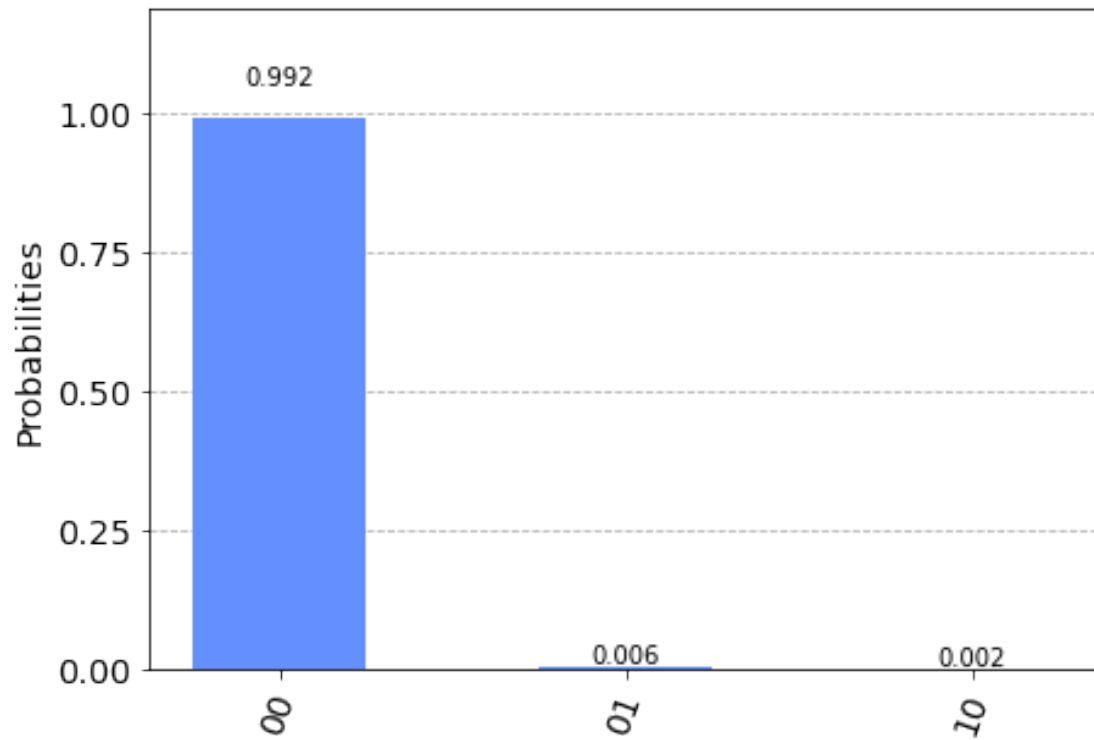
```
#Alice vuole mandare i bit 00 a Bob
sdc = createEntangled()
sdc = encode(sdc, "00") #Alice invia
sdc = decode(sdc) #Bob riceve
sdc.measure_all()
sdc.draw('mpl')
```

Il disegno del circuito:



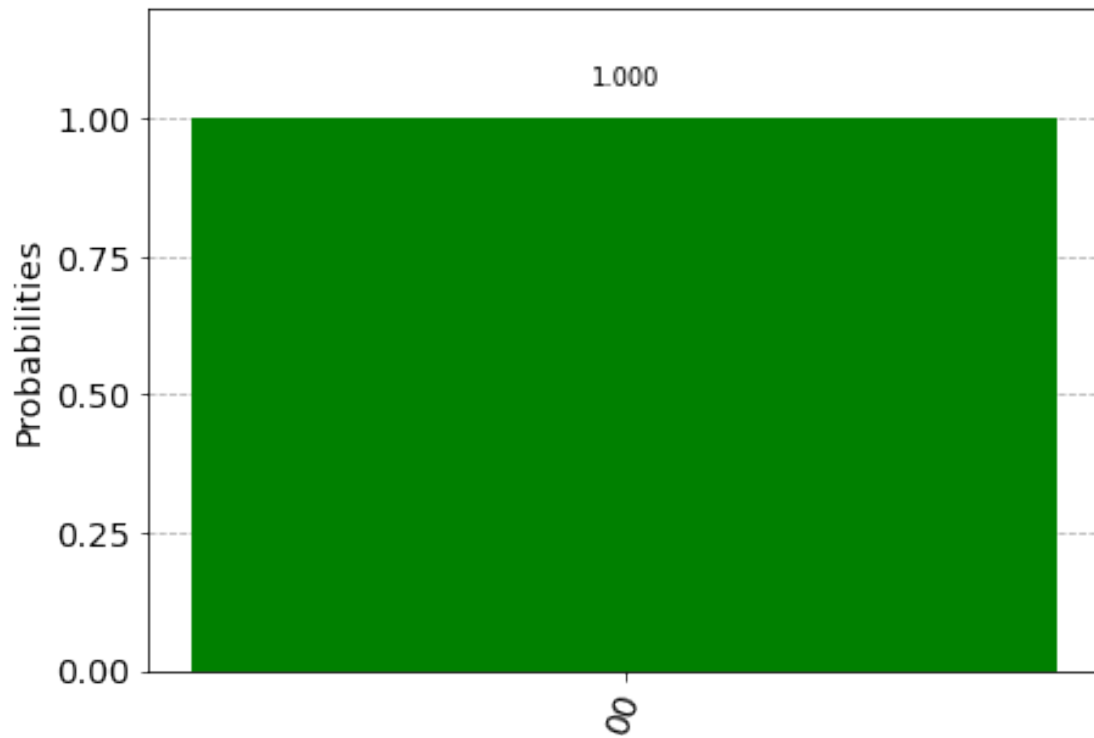
Istogramma derivato dall'esecuzione su computer quantistico

```
device_result = misurazioniReal(sdc)  
plot_histogram(device_result)
```



Istogramma derivato dall'esecuzione sul simulatore

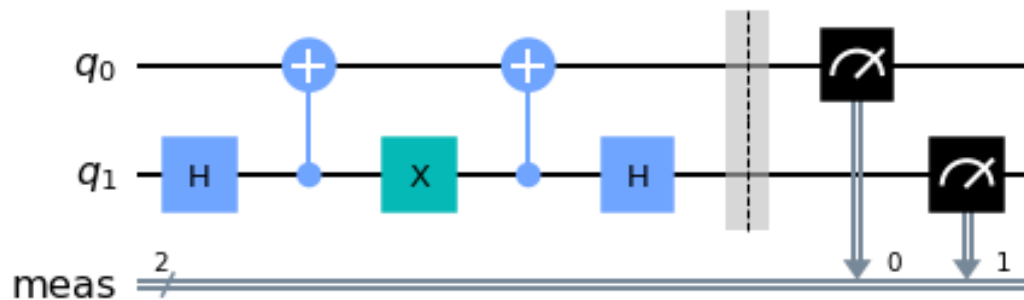
```
device_result = misurazioniSim(sdc)
plot_histogram(device_result, color='green')
```



4.4 Porta X

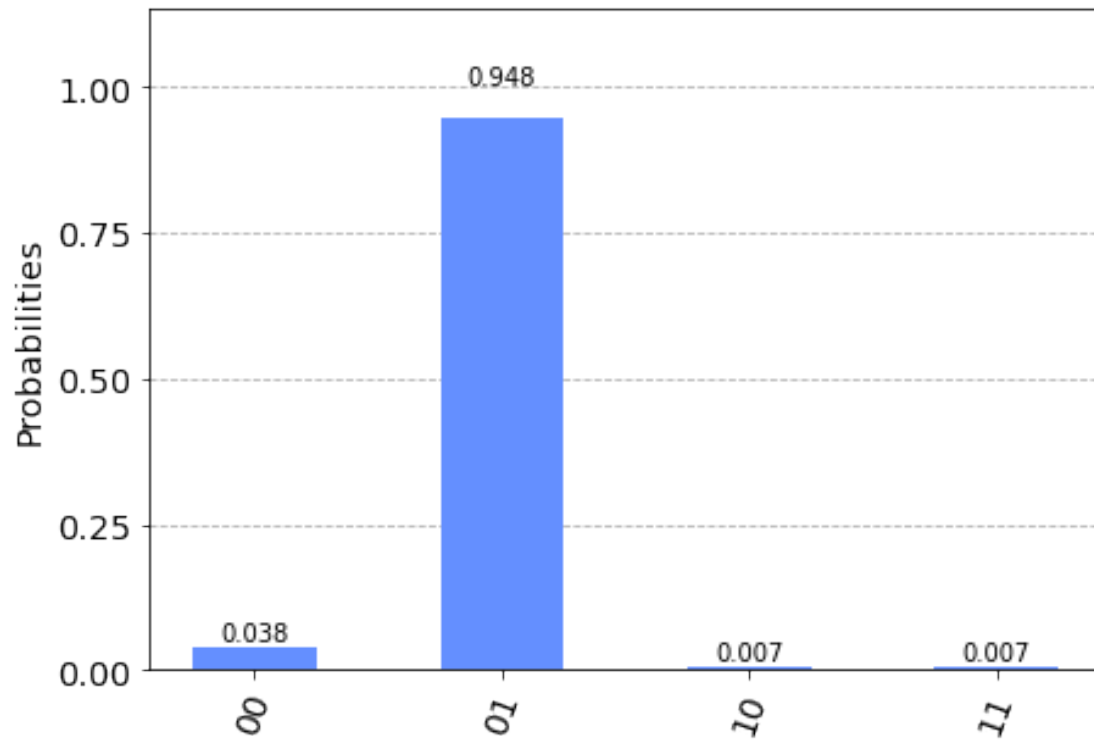
Questo è il caso 2 della sezione "Funzionamento" in questo caso dopo aver creato lo stato entangled, Alice invia il messaggio "01" e per questo sarà applicata da encode() la porta X tramite il comando `sdc.x(1)`.

```
#Alice vuole mandare i bit 01 a Bob
sdc = createEntangled()
sdc = encode(sdc, "01") #Alice invia
sdc = decode(sdc) #Bob riceve
sdc.measure_all()
sdc.draw('mpl')
```



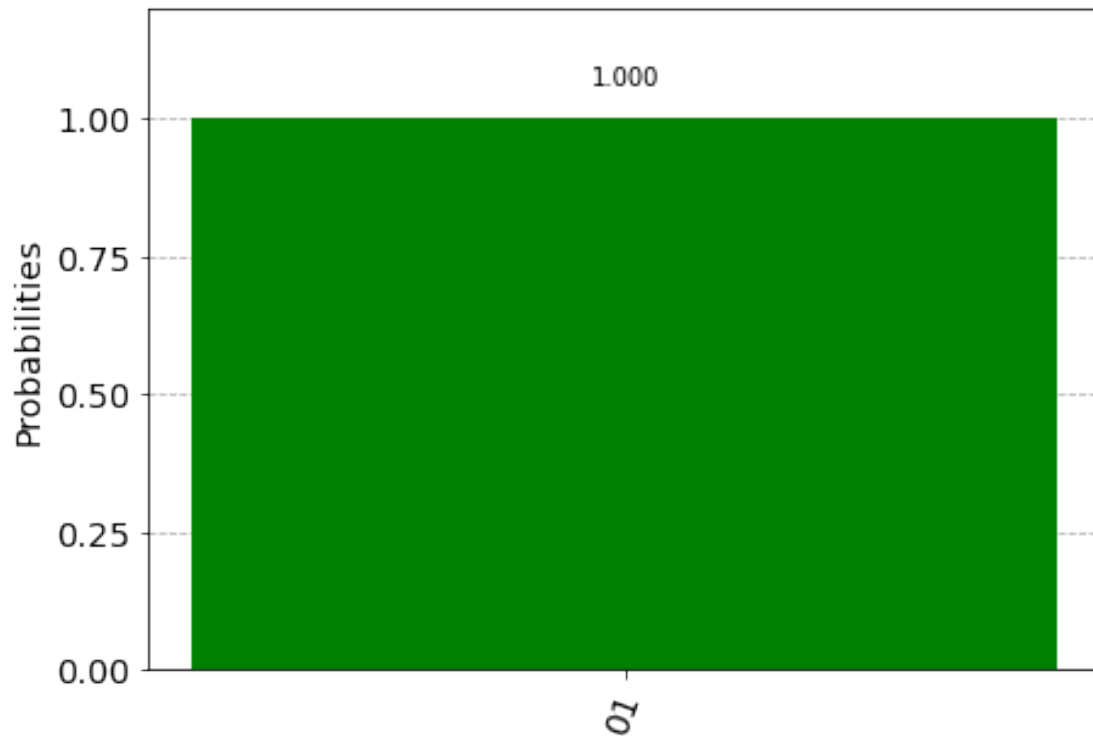
Istogramma derivato dall'esecuzione su computer quantistico

```
device_result = misurazioniReal(sdc)  
plot_histogram(device_result)
```



Istogramma derivato dall'esecuzione sul simulatore

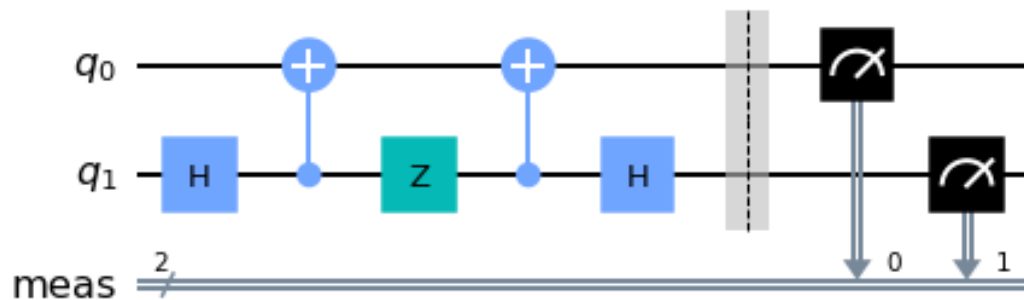
```
device_result = misurazioniSim(sdc)
plot_histogram(device_result, color='green')
```



4.5 Porta Z

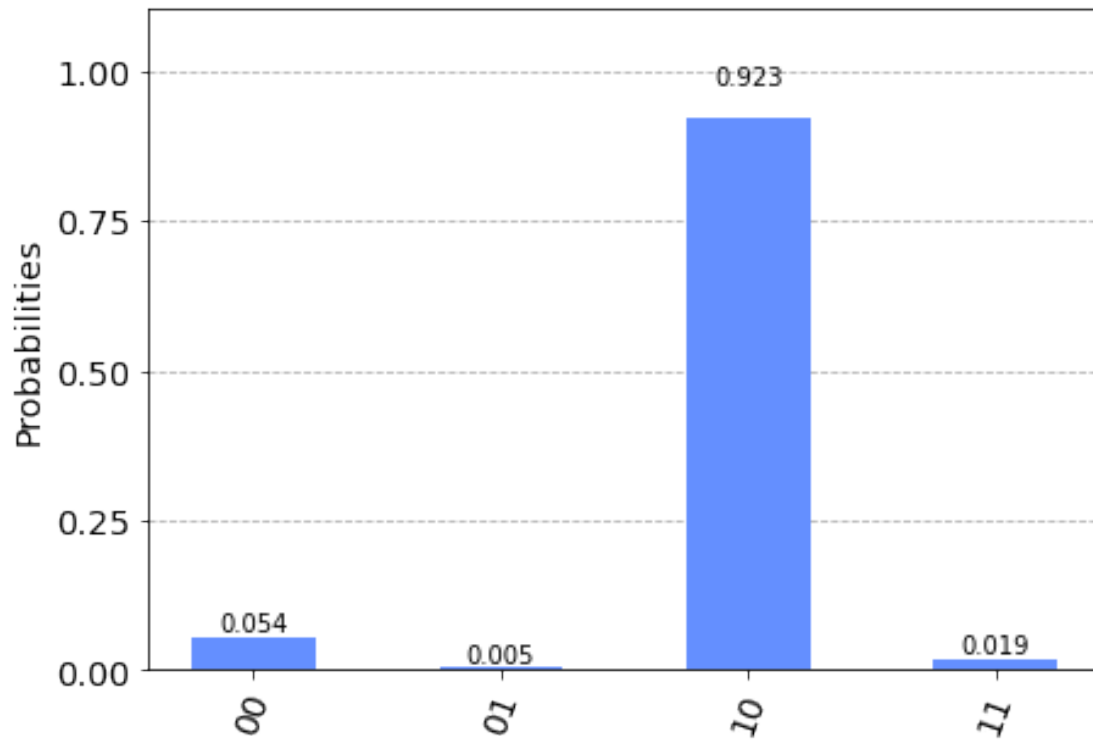
Questo è il caso 3 della sezione "Funzionamento" in questo caso dopo aver creato l'entangled, Alice invia il messaggio "10", quindi verrà applicata da `encode()` la porta **Z** tramite il comando `sdc.z(1)`.

```
#Alice vuole mandare i bit 10 a Bob
sdc = createEntangled()
sdc = encode(sdc, "10") #Alice invia
sdc = decode(sdc) #Bob riceve
sdc.measure_all()
sdc.draw('mpl')
```



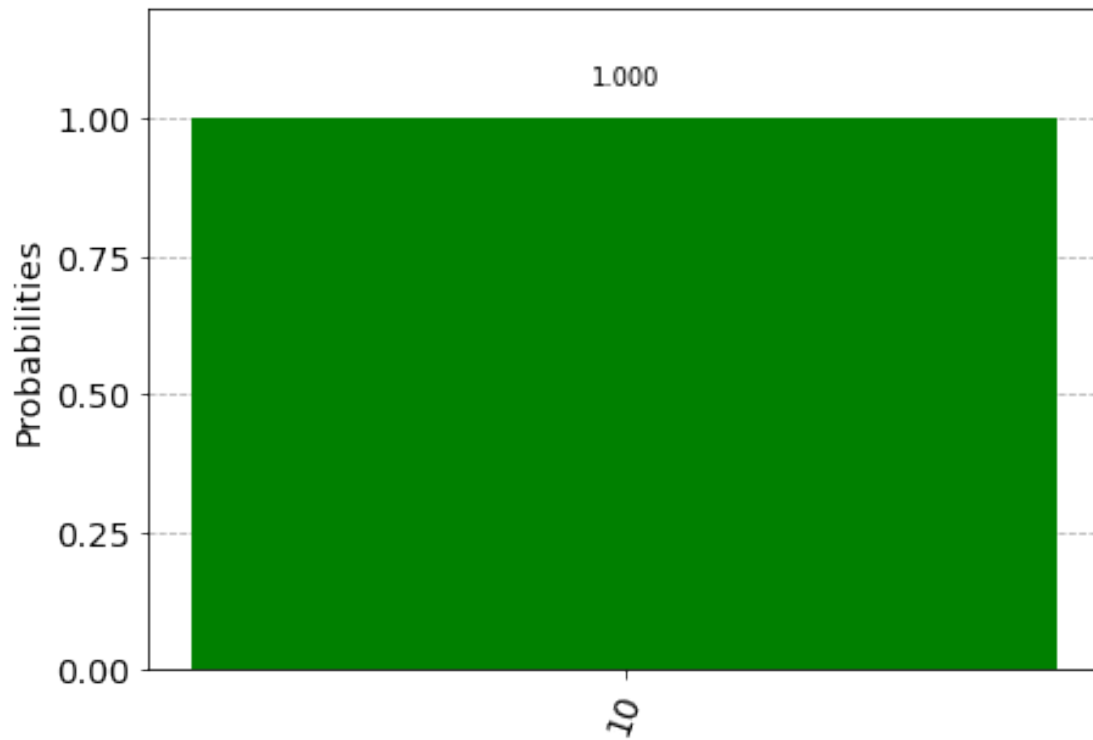
Istogramma derivato dall'esecuzione su computer quantistico

```
device_result = misurazioniReal(sdc)
plot_histogram(device_result)
```



Istogramma derivato dall'esecuzione sul simulatore

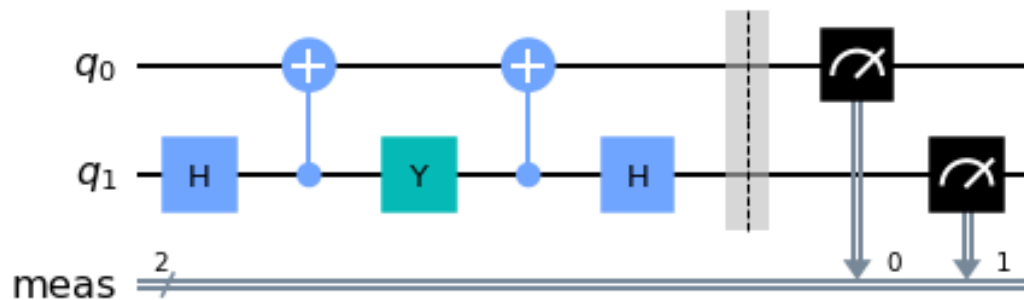
```
device_result = misurazioniSim(sdc)
plot_histogram(device_result, color='green')
```



4.6 Porta iY

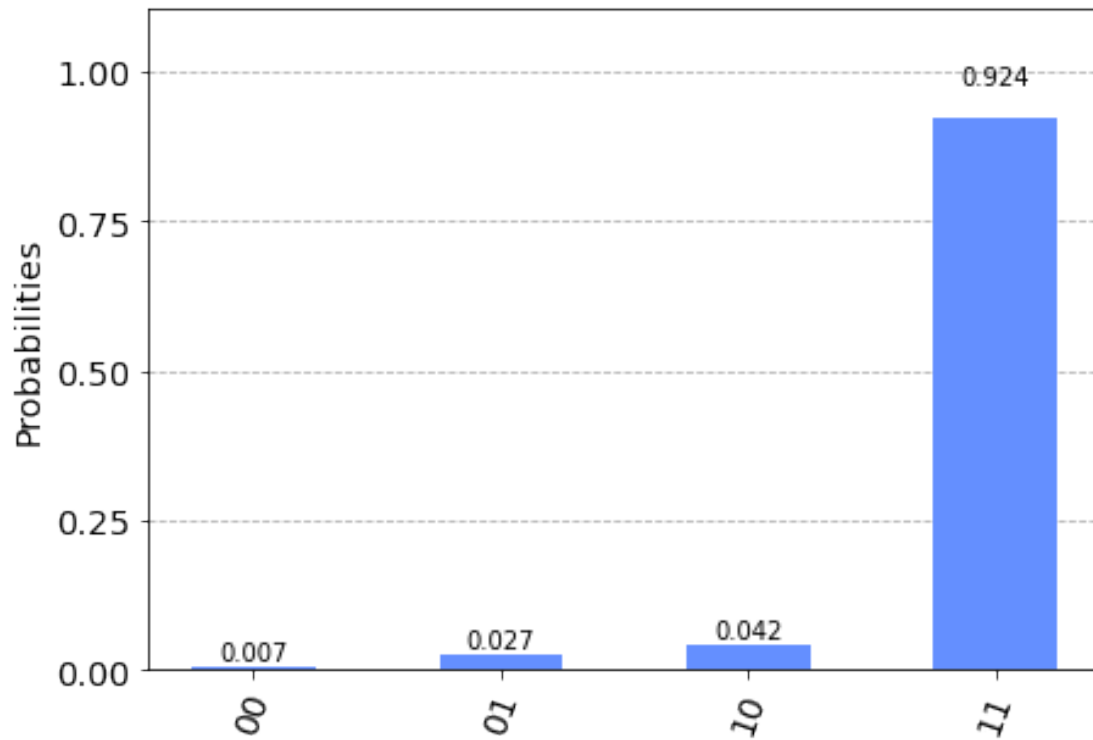
Questo è il caso 4 della sezione "Funzionamento" in questo caso, dopo aver applicato lo stato entangled, Alice invia il messaggio "11", la porta che applica encode() in questo caso è iY tramite il comando `sdc.y(1)`.

```
#Alice vuole mandare i bit 11 a Bob  
sdc = createEntangled()  
sdc = encode(sdc, "11") #Alice invia  
sdc = decode(sdc) #Bob riceve  
sdc.measure_all()  
sdc.draw('mpl')
```



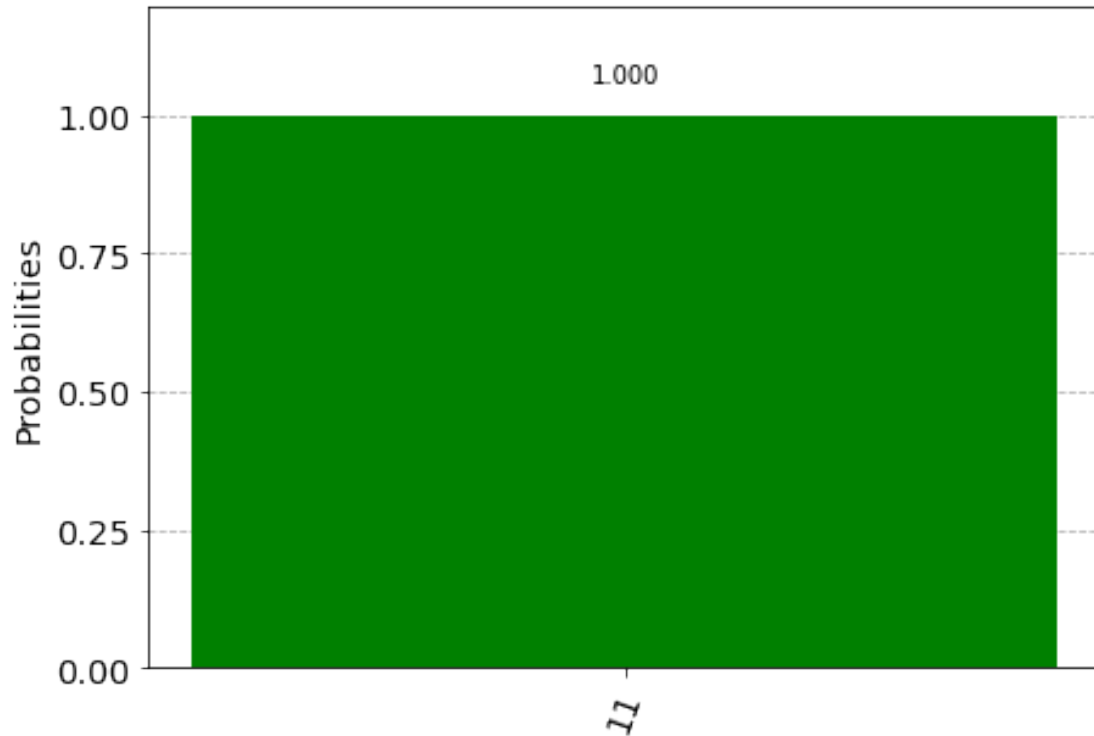
Istogramma derivato dall'esecuzione su computer quantistico

```
device_result = misurazioniReal(sdc)
plot_histogram(device_result)
```



Istogramma derivato dall'esecuzione sul simulatore

```
device_result = misurazioniSim(sdc)
plot_histogram(device_result, color='green')
```



5 Considerazioni finali

Gli istogrammi dei dati raccolti coincidono con i dati attesi dal protocollo. Gli istogrammi dei dati simulati hanno dato al 100% risultati corretti e conformi alla teoria. Mentre come ci aspettavamo i dati raccolti dal computer quantistico sono stati soggetti a interferenze e non hanno dato sempre risultati corretti: lo sono in più del 90% dei casi, quindi è un'ottima riuscita.