

# Super Dense Coding

Kevin Cattaneo

Il protocollo di super-dense coding afferma che, attraverso lo sfruttamento dell'entanglement, è possibile trasferire più informazione tramite i qubit rispetto ai bit classici. L'idea di partenza coinvolge Alice e Bob: Alice vuole inviare due bit di informazione a Bob. Quantisticamente la stessa mole di informazione si dimostra altresì trasportabile attraverso un solo qubit.

I sistemi entangled sono sistemi in cui una misura su uno stato influenza un altro stato inducendo un collasso anche in quest'ultimo.

Brevemente il super dense coding è descritto secondo i seguenti passi:

- Una terza parte genera uno stato quantistico entangled (di Bell) che è condiviso da Alice e Bob
- Alice sceglie la stringa da inviare: 00, 01, 10, 11
- A seconda della stringa scelta Alice applica le seguenti porte logiche sullo stato entangled:
  - Identità, se vuole inviare 00
  - Z, se vuole inviare 10
  - X, se vuole inviare 01
  - iY (o ZX), se vuole inviare 11
- Alice invia il suo qubit trasformato a Bob attraverso un canale classico
- Bob riceve il qubit di Alice ed applica una porta CNOT, che distrugge l'entanglement, e una porta di Hadamard
- Infine, Bob effettua una misura, ottenendo con una probabilità del 100% (\*) la stringa inviata da Alice.

Dall'ultimo passo (\*) si denota come il protocollo descritto non è probabilistico, bensì deterministico.

Premetto che il canale di comunicazione non presenta rumore, né è stato aggiunto artificialmente.

Nella mia implementazione, parto dunque dalla generazione dello stato di Bell, entangled:

$$|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Tale stato è condiviso fra Alice e Bob, dove il primo qubit è di Alice e il secondo è di Bob.

Alice sceglie dunque la stringa che vuole inviare, applicando di conseguenza una porta logica al suo qubit, in particolare:

Alice sceglie	Alice applica
00	I
01	X
10	Z
11	iY = ZX

ottenendo così:

$$|00\rangle \rightarrow \frac{|00\rangle + |11\rangle}{\sqrt{2}} \text{ applicando I}$$

$$|01\rangle \rightarrow \frac{|01\rangle + |10\rangle}{\sqrt{2}} \text{ applicando X}$$

$$|10\rangle \rightarrow \frac{|00\rangle - |11\rangle}{\sqrt{2}} \text{ applicando Z}$$

$$|11\rangle \rightarrow \frac{|01\rangle - |10\rangle}{\sqrt{2}} \text{ applicando ZX}$$

In codice la creazione dello stato di Bell si traduce in:

```
def bell_creation():
    bell_circuit = cirq.Circuit()
    a, b = cirq.LineQubit.range(2)
    bell_circuit.append(cirq.H(a))
    bell_circuit.append(cirq.CNOT(a,b))
    return a, b, bell_circuit
```

Mentre l'applicazione delle porte viene codificata nel seguente modo:

```
print('-- Alice sends 00 -- ')
a, b, bell_circuit = bell_creation()
myprint(s, bell_circuit)
```

```
print('-- Alice sends 01 -- ')
a, b, bell_circuit = bell_creation()
bell_circuit.append(cirq.X(a))
myprint(s, bell_circuit)
```

```
print('-- Alice sends 10 -- ')
a, b, bell_circuit = bell_creation()
bell_circuit.append(cirq.Z(a))
myprint(s, bell_circuit) # global phase
```

```
print('-- Alice sends 11 -- ')
a, b, bell_circuit = bell_creation()
bell_circuit.append(cirq.Z(a))
bell_circuit.append(cirq.X(a))
myprint(s, bell_circuit) # global phase
```

Dopodiché Alice invia il suo qubit a Bob attraverso un canale classico.

Bob riceve il qubit di Alice, possedendo ora entrambi i qubit dello stato entangled (sia il primo che il secondo), applica quindi una porta CNOT e una porta di Hadamard sul primo qubit, come segue:

$$\begin{aligned} \frac{|00\rangle + |11\rangle}{\sqrt{2}} &\rightarrow \frac{|00\rangle + |10\rangle}{\sqrt{2}} \rightarrow |00\rangle \\ \frac{|10\rangle + |01\rangle}{\sqrt{2}} &\rightarrow \frac{|11\rangle + |01\rangle}{\sqrt{2}} \rightarrow |01\rangle \\ \frac{|00\rangle - |11\rangle}{\sqrt{2}} &\rightarrow \frac{|00\rangle - |10\rangle}{\sqrt{2}} \rightarrow |10\rangle \end{aligned}$$

$$\frac{|10\rangle - |01\rangle}{\sqrt{2}} \rightarrow \frac{|01\rangle - |11\rangle}{\sqrt{2}} \rightarrow |11\rangle$$

Si osserva che la porta CNOT distrugge l'entanglement.

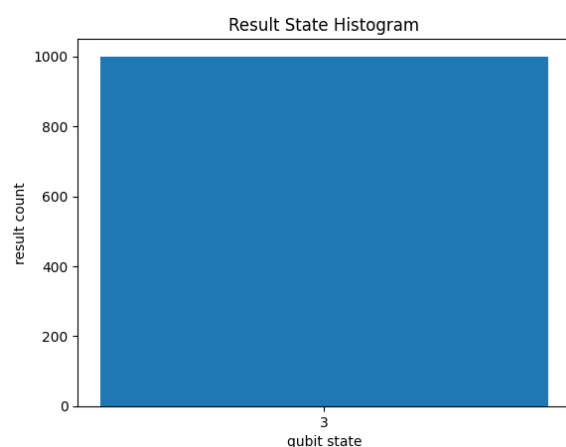
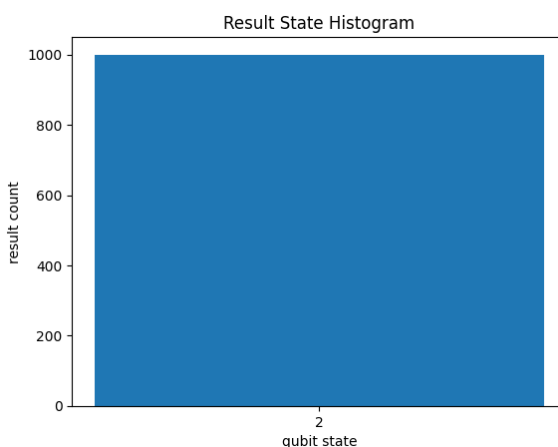
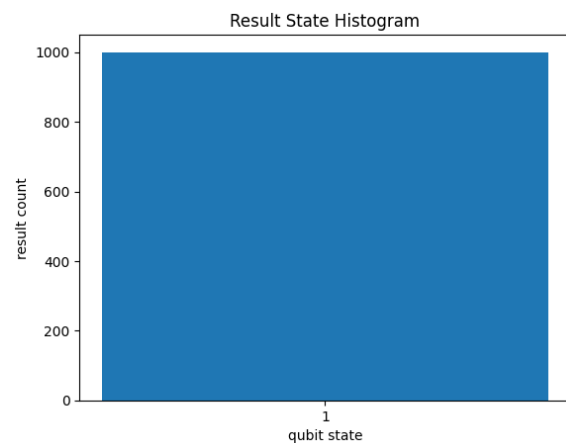
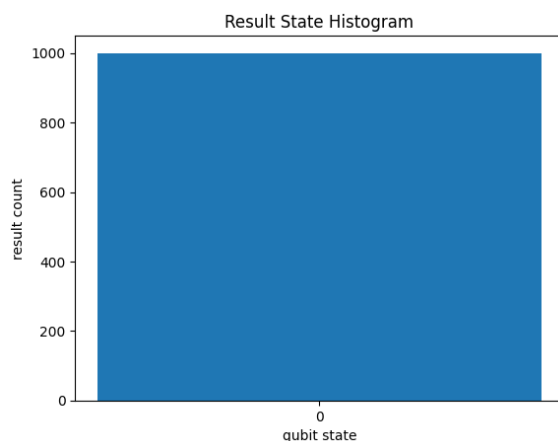
Infine, per verificare quale stringa Alice ha inviato, Bob effettua una misura sullo stato risultante, ottenendo con una probabilità del 100% le stringhe inviate, tale misura non distrugge lo stato in quanto già autostato nella base Z.

Le operazioni effettuate da Bob sono state implementate come segue:

```
bell_circuit.append(cirq.CNOT(a,b))
bell_circuit.append(cirq.H(a))
myprint(s, bell_circuit)
measureAndHist(s, a, b, bell_circuit)
```

Vista anche l'assenza di rumore come da premessa, si osserva un determinismo, e non una base probabilistica che influisce nella misura, nella trasmissione delle stringhe. Inoltre, la distinzione netta data dalla misura è dovuta anche all'ortogonalità degli stati di Bell.

A riprova della caratteristica deterministica del protocollo, e per verificare effettivamente il 100% di probabilità di ottenere correttamente la stringa attesa, sono state effettuate mille ripetizioni del simulatore del circuito per la misura, e il risultato è stato tracciato su un istogramma, uno per ogni possibile scelta di stringa da parte di Alice (00,01,10,11).



Si osserva che su tutte le ripetizioni, la misura non sbaglia, producendo lo stato atteso.

Si conclude quindi che la trasmissione di un solo qubit condensa due bit di informazione logica.