HW5

Xichen Li, EE521 - Group 5

A) Xichen Li: I did HW5 independently.

Interpreting Quantum Mechanics

We've seen several examples of the "spooky action at a distance" so characteristic to quantum mechanics. In the early 20th century, as physicists were beginning to realize all the physical consequences of the quantum formalism they were developing, many were very reluctant to accept when they were theorizing.

Albert Einstein, who won a Nobel prize in 1921 for this work in quantum mechanics, was famously skeptical of some of the philosophical implications of quantum mechanics. At the time, many of the greatest physicists in history were having passionate debates, not just of the mechanics itself, but also how we should interpret this new formalism. Einstein for one, was not particularly happy with the apparent non-determinism we observe in quantum mechanics.

All of classical mechanics (and even relativistic physics) is fully deterministic, so it's only in the quantum world when things get more complicated. Einstein argued "god does not play dice," that only quantum mechanics looks non-deterministic because we don't have the full picture - some piece of the puzzle is missing, and if we had it, everything would be nice and deterministic again. We might think of that missing, unknown piece of information as a hidden state of our system, and it's only because we don't know that hidden state that we think quantum mechanics is non-deterministic.

An alternative to this "hidden state" theory, was the Copenhagen interpretation proposed by Niels Bohr and Werner Heisenberg among others. Simply put, the Copenhagen interpretation proposes that the properties of a quantum system are not necessarily well defined in general. Instead, it is the act of measuring the system that causes the state it to "collapse" into having whatever property that was measured. This "collapse" can essentially be treated as irreversible and non-deterministic. As a result, the Copenhagen interpretation in many ways diametrically opposes hidden state theories.

So far we have focused on the Copenhagen interpretation, which is traditionally the most popular interpretation of quantum mechanics. However, as good scientists we can't just blindly trust the ramblings of some old professors almost a century ago. Now we will discuss the famous EPR paradox and Bell's theorem, a brilliant experiment designed to test the hidden state theory.

EPR Paradox

The Einstein–Podolsky–Rosen paradox was developed by Albert Einstein, Boris Podolsky, and Nathan Rosen to challenge the Copenhagen interpretation in favor of a hidden state theory.

The original presentation of the EPR paradox uses observables like position and momentum, which were already shown by Heisenberg not to commute, however here we'll discuss a simplified and more common rendition of the paradox using spin. Remember that spin (of an electron, for example) can be along a certain axis, and that the spin operators along different axes don't commute. This means the spin of an electron can only be known along one axis at a time.

Suppose we have a source which creates an electron-positron pair. The Standard Model (contemporary physics) tells us when an electron-positron pair is produced the spin of the electron will always be exactly opposite to that of the positron, no matter what direction the spin actually has. The electron is sent to Alice's lab and the positron to Bob's. Let's imagine Alice could either measure the spin along the X or Z axis, either of which have two possible outcomes: 1 or -1. From the Standard Model, we know that if she were to measure the X axis, and get an outcome of 1 then if Bob measures the spin of his positron along X he must measure -1 (and vice versa). However, if Alice measures Z and Bob measures X, then Bob sometimes gets 1 and sometimes -1 in an unpredictable manner. Furthermore, this observation doesn't change if Alice and Bob make their measurements at the exact same time. This suggests that Alice's choice of axis is somehow transmitted instantaneously from her electron to Bob's positron, which may be arbitrarily far away. However, special relativity, and by extension causality itself, does not allow any information to travel faster than the unflinchingly finite speed of light - sending information instantaneously is not possible!

The solution to this paradox presented by EPR argues quantum mechanics as we know it must not be complete. They suggest since Bob's positron cannot get the information from Alice's measurement instantaneously, instead both Alice and Bob's outcomes are dependent on some underlying process, or hidden variable, not treated by the current formalism of quantum mechanics. If we possessed knowledge of this hidden variable then Bob's outcome of measuring X after Alice measured Z would no longer be unpredictable.

Bell's Inequality

Starting with some ideas from the de Broglie–Bohm theory (an alternative to the Copenhagen interpretation), John Stewart Bell developed Bell's theorem in 1964, which suggested a relatively simple statistical experiment to test hidden variable theory. Since it's conception, experimental physicists have tried to empirically verify Bell's theorem, however reliably producing and measuring entangled particles (in this case photons) is easier said than done. In 2015, several experiments finally eliminated all major loopholes to the experimental procedure and successfully violated Bell's inequality, as predicted by quantum mechanics.

Here we will simulate several quantum circuits to violate the CHSH inequality, which is an equivalent alternative to Bell's original inequality that is easier to simulate. The CHSH inequality, just like Bell's inequality, is a statistical law that follows from several (seemingly) reasonable assumptions such as locality and realism. Locality requires that an event at one point in space and time cannot affect an event at another point in space at the same time, while realism posits observables have some definite outcome independent of measurement. Quantum mechanics violating the CHSH inequality, or equivalently Bell's inequality, means either locality or realism (or both) do not apply in a quantum world.

Just as in the EPR paradox, we begin with a source that can produce pairs of particles that have interacted in some way. In quantum computing terms, that means a pair of qubits that are entangled. For each pair, we'll give Alice one of the qubits, and Bob the other one. Then we'll have two sets of two unique observables called $\mathbf{A}/\mathbf{A}'$ that Alice can measure, and $\mathbf{B}/\mathbf{B}'$ for Bob. We'll have to be a little careful how we measure our qubits, because remember that in quantum computing the outcome of any measurement is always either 0 or 1. Alice and Bob coordinate beforehand so they split the total number of qubit pairs they have shared into four groups, and use a different possible combination of observables for each group ($\mathbf{AB}$, $\mathbf{AB}'$, $\mathbf{A}'\mathbf{B}$, and $\mathbf{A}'\mathbf{B}'$). The outcome of each measurement is recorded as +1 if you get 0 and -1 if you get 1, so the outcome of measuring an entangled pair is interpreted as +1 if both Alice an Bob get the same value (they measure 00 or 11), and -1 if they get different values (01 or 10). Finally these outcomes are averaged for each combination of observables are combined to estimate the total correlation between qubits in the following way (the derivation of which follows directly from locality and realism):

$$C = \langle \mathbf{AB} \rangle - \langle \mathbf{AB}' \rangle + \langle \mathbf{A}'\mathbf{B} \rangle + \langle \mathbf{A}'\mathbf{B}' \rangle$$

where $\langle \mathbf{M} \rangle$ refers to the average outcome of measuring the combination $\mathbf{M}$.

Let's first think about what kind of result we would expect assuming both locality and realism were true, which is what classical mechanics (and our intuition) would suggest. Measuring the outcome of $\mathbf{A}$ (or $\mathbf{A'}$) should not affect the outcome of $\mathbf{B}$ (or $\mathbf{B'}$), consequently we can use the linearity of expectation to group terms leaving:

$$C_{\text{classical}} = \langle \mathbf{A}(\mathbf{B} - \mathbf{B'}) \rangle + \langle \mathbf{A'}(\mathbf{B} + \mathbf{B'}) \rangle$$

The outcomes of measuring $\mathbf{B}$ and $\mathbf{B'}$ must either be different (one is +1, one is -1) or the same (both are +1 or -1). If there is no correlation between Alice's and Bob's measurements the average outcomes should be close to zero, so $|C_{\text{classical}}| \approx 0$. If there is some correlation, the largest we can imagine is if the outcomes given a certain measurement of Alice are always either the same or different but always in the same way. This would mean one of the two terms approaches 2 while the other approaches 0. In any case, $|C_{\text{classical}}|$ cannot exceed a value of 2, because that would mean either the outcomes of $\mathbf{B}$ and $\mathbf{B'}$ are not defined before measurement (breaking realism), or the outcomes depend on whether $\mathbf{A}$ or $\mathbf{A'}$ is measured, which means the information of which measurement Alice used would have to be transmitted to Bob's qubit instantaneously (breaking locality).

In [1]:

```
import numpy as np
%matplotlib notebook
import matplotlib.pyplot as plt
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister
from qiskit import BasicAer # for simulating circuits
from qiskit.tools.visualization import plot_histogram
from qiskit import execute
```

Problem 1: Violating the CHSH Inequality

In our experimental design, we'll try to violate the CHSH inequality as much as is allowed by quantum mechanics. This means we'll start with a maximally entangled state - a cat state $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, and that we have to be careful about what measurements to choose. It turns out choosing the measurements:

$$\mathbf{A} = \mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\mathbf{A'} = \mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$\mathbf{B} = \mathbf{W} = \frac{1}{\sqrt{2}}(\mathbf{Z} + \mathbf{X})$$

$$\mathbf{B'} = \mathbf{V} = \frac{1}{\sqrt{2}}(\mathbf{Z} - \mathbf{X})$$

will maximize the expected violation. To simulate measuring all these different observables using just the computational basis ($\mathbf{Z}$), we do a some linear algebra to find the right transformation. For this purpose, we might imagine there to be a measurement operator $\mathbf{M_B}$ for basis $\mathbf{B}$. Now we can write each of our measurement "operators" in terms of $\mathbf{M_Z}$:

$$\mathbf{M_Z} = \mathbf{M_Z}$$

$$\mathbf{M_X} = \mathbf{M_Z H}$$

$$\mathbf{M_W} = \mathbf{M_Z HTHS}$$

$$\mathbf{M_V} = \mathbf{M_Z HT^\dagger HS}$$

where $\mathbf{H}$ is the hadamard, $\mathbf{T}$ is the T gate, and $\mathbf{S}$ is the S gate.

With this setup, you can compute $|C_{\text{quantum}}|$ using the following equation:

$$C_{\text{quantum}} = \langle \mathbf{XW} \rangle - \langle \mathbf{XV} \rangle + \langle \mathbf{ZW} \rangle + \langle \mathbf{ZV} \rangle$$

where $\langle \mathbf{M_1 M_2} \rangle$ refers to the average outcome of measuring observable $\mathbf{M_1}$ on the first qubit and $\mathbf{M_2}$ on the second qubit.

Below there is some code that will create a quantum circuit that will produce our initial entangled cat state $|\psi\rangle$ (see `create_cat`), and code that will, given a `QuantumCircuit` `circ`, and two qubits (from a `QuantumRegister`) `qbit0` and `qbit1`, transform the state to measure different combinations of observables (including `XW`, `XV`, `ZW`, and `ZV`). Finally, there's also a utility function to let you measure your qubits and store them in a classical register (see `measure_and_save`).

Your job is to (1) build the circuits necessary to compute each of the averages $\langle \mathbf{AB} \rangle$, $\langle \mathbf{AB'} \rangle$, $\langle \mathbf{A'B} \rangle$, and $\langle \mathbf{A'B'} \rangle$, (2) simulate each of the circuits using the `qasm_simulator` simulator with 1000 shots (see `execute` function) per circuit, and (3) to compute $|C_{\text{quantum}}|$ as predicted by quantum mechanics.

Hint 1: If you get stuck, I suggest working through the circuit on paper, and simulating your circuits using the `statevector_simulator` and comparing the state vector your circuit produces to the one you would expect.

Hint 2: Problem the hardest part of this problem is not building or simulating the circuits, but interpreting and analyzing the results to compute $|C_{\text{quantum}}|$. When building the circuits you should not have to add any gates to circuits yourself (instead use the provided `XW`, `XV`, `ZW`, and `ZV` functions).

In [2]:

```
def create_cat():
    q = QuantumRegister(2, 'q') # specify the number of qubits in the register and a name

    circ = QuantumCircuit(q)

    circ.h(q[0])
    circ.cx(q[0], q[1])
    return circ, q

def example(circ, qbit0, qbit1):
    circ.ch(qbit1, qbit0)

def XW(circ, qbit0, qbit1):
    # qubit 0: X measurement -> Z measurement
    circ.h(qbit0)

    # qubit 1: W measurement -> Z measurement
    circ.s(qbit1)
    circ.h(qbit1)
    circ.t(qbit1)
    circ.h(qbit1)

def XV(circ, qbit0, qbit1):
    # qubit 0: X measurement -> Z measurement
    circ.h(qreg[0])

    # qubit 1: V measurement -> Z measurement
    circ.s(qbit1)
    circ.h(qbit1)
    circ.tdg(qbit1)
    circ.h(qbit1)

def ZW(circ, qbit0, qbit1):
    # qubit 0: Z measurement (default)

    # qubit 1: W measurement -> Z measurement
    circ.s(qbit1)
    circ.h(qbit1)
    circ.t(qbit1)
    circ.h(qbit1)

def ZV(circ, qbit0, qbit1):
    # qubit 0: Z measurement (default)

    # qubit 1: V measurement -> Z measurement
    circ.s(qbit1)
    circ.h(qbit1)
    circ.tdg(qbit1)
    circ.h(qbit1)
```

```
def measure_and_save(circ, qreg, name='c'):
    c = ClassicalRegister(len(qreg), name)
    circ.add_register(c)
    circ.measure(qreg, c)
```

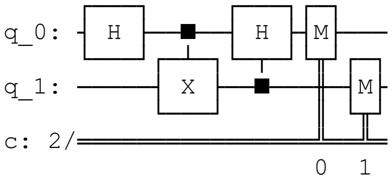
Hint 3: An example of building and simulating a circuit is shown below (the results of which unfortunately isn't useful for this experiment)

In [3]:

```
circ, qreg = create_cat()
qbit0, qbit1 = qreg[0], qreg[1]
example(circ, qbit0, qbit1)
#circ.draw()
measure_and_save(circ, qreg)
backend = BasicAer.get_backend('qasm_simulator')
counts = execute(circ, backend, shots=1000).result().get_counts()
print('Result:', counts)
circ.draw()
```

```
Result: {'00': 492, '10': 245, '11': 263}
```

Out[3]:



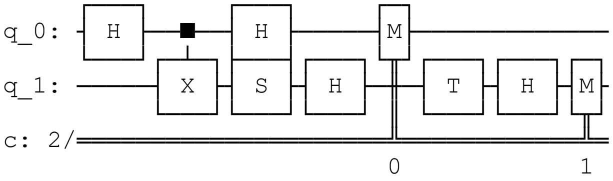
Answer:

In [4]:

```
#<AB>
circ, qreg = create_cat()
qbit0, qbit1 = qreg[0], qreg[1]
XW(circ, qbit0, qbit1)
measure_and_save(circ, qreg)
backend = BasicAer.get_backend('qasm_simulator')
counts = execute(circ, backend, shots=1000).result().get_counts()
print('Result:', counts)
circ.draw()
```

```
Result: {'00': 430, '11': 450, '10': 61, '01': 59}
```

Out[4]:

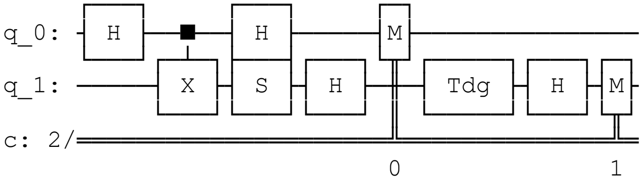


In [5]:

```
#<AB'>
circ1, qreg1 = create_cat()
qbit1_0, qbit1_1 = qreg1[0], qreg1[1]
XV(circ1, qbit1_0, qbit1_1)
measure_and_save(circ1, qreg1)
backend1 = BasicAer.get_backend('qasm_simulator')
counts1 = execute(circ1, backend1, shots=1000).result().get_counts()
print('Result:', counts1)
circ1.draw()
```

```
Result: {'10': 433, '01': 435, '11': 59, '00': 73}
```

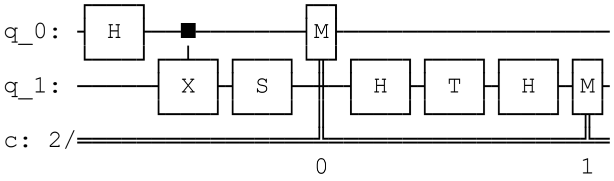
Out[5]:



In [6]:
```
#<A'B>
circ2, qreg2 = create_cat()
qbit2_0, qbit2_1 = qreg2[0], qreg2[1]
ZW(circ2, qbit2_0, qbit2_1)
measure_and_save(circ2, qreg2)
backend2 = BasicAer.get_backend('qasm_simulator')
counts2 = execute(circ2, backend2, shots=1000).result().get_counts()
print('Result:', counts2)
circ2.draw()
```

```
Result: {'00': 428, '11': 425, '01': 73, '10': 74}
```

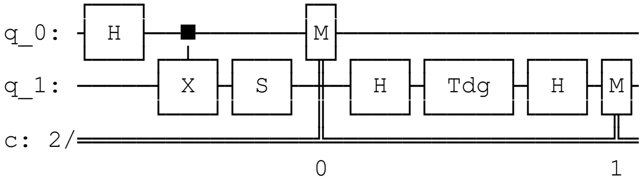
Out[6]:



In [7]:
```
#<A'B'>
circ3, qreg3 = create_cat()
qbit3_0, qbit3_1 = qreg3[0], qreg3[1]
ZV(circ3, qbit3_0, qbit3_1)
measure_and_save(circ3, qreg3)
backend3 = BasicAer.get_backend('qasm_simulator')
counts3 = execute(circ3, backend3, shots=1000).result().get_counts()
print('Result:', counts3)
circ3.draw()
```

```
Result: {'10': 88, '11': 422, '00': 417, '01': 73}
```

Out[7]:



In [8]:

```
ave=(1*(counts['11']+counts['00'])-1*(counts['10']+counts['01']))/1000
ave1=(1*(counts1['11']+counts1['00'])-1*(counts1['10']+counts1['01']))/1000
ave2=(1*(counts2['11']+counts2['00'])-1*(counts2['10']+counts2['01']))/1000
ave3=(1*(counts3['11']+counts3['00'])-1*(counts3['10']+counts3['01']))/1000
print (" <AB>: {:.3f}, <AB'>: {:.3f}, <A'B>: {:.3f}, <A'B'>: {:.3f}".format(ave,ave1,ave2,ave3))
C_abs=ave-ave1+ave2+ave3
print (" |C_quantum|={:.3f}".format(C_abs))
```

```
 <AB>: 0.760, <AB'>: -0.736, <A'B>: 0.706, <A'B'>: 0.678
 |C_quantum|=2.880
```

In the previous analysis, wegot $|C_{\text{classical}}|$ cannot exceed a value of 2 in any case. Now, $|C_{\text{quantum}}|$ is larger than 2, which means the results violate the CHSH inequality as much as is allowed by quantum mechanics.

Challenging Local Realism

As you should see, $|C_{\text{quantum}}| \approx 2\sqrt{2}$, since the ideal experiment would produce $\langle \mathbf{XW} \rangle = \frac{1}{\sqrt{2}}$, $\langle \mathbf{XV} \rangle = -\frac{1}{\sqrt{2}}$, $\langle \mathbf{ZW} \rangle = \frac{1}{\sqrt{2}}$, and $\langle \mathbf{ZV} \rangle = \frac{1}{\sqrt{2}}$.

What does this mean for locality and realism? If locality and realism are both to be true, $|C| \leq 2$.

Suppose the outcome of our measurements for our pair of entangled qubits was somehow decided in advance, perhaps by some hidden variables or details related to the process that entangled them in the first place. Bell's theorem proves there is no way to set the outcomes in such a way that $|C| > 2$, yet we observed $|C| \approx 2\sqrt{2}$. (Provided we can trust our simulator - if you don't, you can actually try running this experiment on a real quantum computer.) This shows hidden variable theories are not powerful enough to reproduce our observations - so what is going on?

Coming from the perspective of the Copenhagen interpretation, we can make sense of the CHSH inequality violation by being more careful about what we mean by locality and realism. First of all, Bell's theorem leaves us little choice but to conclude that somehow Alice measuring her qubit instantaneously affects Bob's qubit. That might sound like information is traveling faster than the speed of light, which would essentially mean we have to throw out all of special relativity.

Thankfully, there is a rather elegant solution which allows us to keep weaker form of locality that is sufficient to explain Bell's theorem without abandoning special relativity and information theory. The trick is eventhough Bob's qubit is affected the instant Alice measures her qubit, Bob can't make any meaningful use of this change in his qubit. That is to say, Bob can't know that he violated Bell's inequality until Alice tells him what kind of measurement she used. To further illustrate this subtlety, imagine a simpler experiment where Alice and Bob each have one of the qubits that are entangled in a cat state. Now suppose Alice and Bob both measure their qubits in the regular computational basis. The instant Alice measures her qubit she will know with 100% certainty the outcome for Bob if he measures his qubit, however, there's no way for her to communicate her knowledge faster than light speed to Bob. So in some sense, eventhough the state of Bob's qubit changes instantaneously, Bob can't know how it changed, or even that it changed, until after Alice has told him, which can't happen at any rate faster than light speed.

This perspective also forces us to relax our notion of realism. While both Bob and Alice can be certain of the outcome of their measurements after they are done (provided there are no further interactions with the qubits), it may, and in many cases is, impossible for either Alice or Bob to be able to predict the outcomes of their measurements before they have been completed. As the Copenhagen interpretation argues, in general the observables of a quantum system need not be defined at all up to the point at which they are measured and they "collapse" into having a certain property.

Neither the Copenhagen interpretation nor a hidden state theory, nor any other interpretation for that matter, provides a fully satisfying explanation for us to interpret the results of quantum experiments such as violating the CHSH inequality above. Perhaps an even more popular approach to answering these sorts of questions today is the instrumentalist philsophy which can be summed up with "Shut up and calculate!"

Problem 2: Simple Grover's Algorithm

We define four phone number $p_1 = 00, p_1 = 01, p_3 = 10$ and $p_4 = 11$ and the names corresponding to the phone numbers are $n_1 = 00, n_2 = 01, n_3 = 10$ and $n_4 = 11$ respectively. Recall that in Grover's algorithm, the phone numbers are stored in two qubits and the name are stored in two other qubits. In terms of qubit states, the phone numbers are $|p_1\rangle = |00\rangle, |p_2\rangle = |01\rangle, |p_3\rangle = |10\rangle$ and $|p_4\rangle = |11\rangle$ and the corresponding names stored on two other qubits are $|n_1\rangle = |00\rangle, |n_2\rangle = |01\rangle, |n_3\rangle = |10\rangle$ and $|n_4\rangle = |11\rangle$. The phone numbers are identical to the names which are stored on two other qubits. Starting out with the state $|0000\rangle = |00\rangle|00\rangle$, find the gate operations to reach the following states in the Grover's algorithm sequentially:

(a) $|init\rangle = \frac{1}{2}[|00\rangle|p_1\rangle + |01\rangle|p_2\rangle + |10\rangle|p_3\rangle + |11\rangle|p_4\rangle]$
(b) $\hat{V}|init\rangle = -\frac{1}{2}|10, p_3\rangle + \frac{\sqrt{3}}{2}[\frac{1}{\sqrt{3}}(|00, p_1\rangle + |01, p_2\rangle + |11, p_4\rangle)]$
(c) $\hat{W}\hat{V}|init\rangle = |10\rangle|p_3\rangle$

Implement the above gates in qiskit and confirm that your realization is correct.

Hint: You can refer to Lecture note 7.5.1 section

In [9]:
```
n=2
gc = QuantumRegister(n, 'q')
circ = QuantumCircuit(gc)
circ.draw()
```

Out[9]:
```
q_0:

q_1:
```

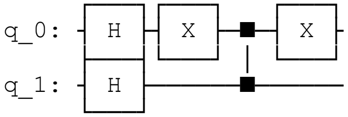
In [10]:
```
circ.h(gc[0])
circ.h(gc[1])
```

Out[10]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x1f18d4e51c0>
```

In [11]:
```
circ.x(gc[0])
#circ.x(gc[1])
circ.cz(0,1)
circ.x(gc[0])

circ.draw() # This is after step-2 which generates V|init>
```

Out[11]:

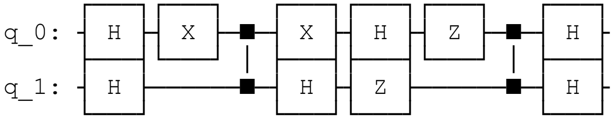


In [12]:
```
backend = BasicAer.get_backend('statevector_simulator')
job = execute(circ, backend)
result = job.result()
gcstate = result.get_statevector(circ)
print('Statevector:', gcstate)
```

```
Statevector: [ 0.5-6.123234e-17j  0.5+0.000000e+00j -0.5+6.123234e-17j
  0.5+0.000000e+00j]
```

In [13]:
```
circ.h([0,1])
circ.z([0,1])
circ.cz(0,1)
circ.h([0,1])
circ.draw() # This is after step-3 which generates WV|init>
```

Out[13]:



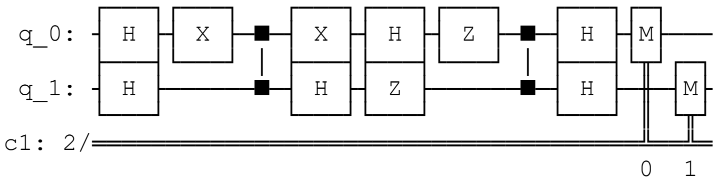
In [14]:
```
backend = BasicAer.get_backend('statevector_simulator')
job = execute(circ, backend)
result = job.result()
gcstate = result.get_statevector(circ)
print('Statevector:', gcstate)
```

```
Statevector: [0.+0.00000000e+00j 0.+0.00000000e+00j 1.-2.22044605e-16j
 0.+0.00000000e+00j]
```

In [15]:

```
#Confirm the results by taking a measurement
c1 = ClassicalRegister(n, 'c1')
circ.add_register(c1)
circ.measure(gc[0], c1[0])
circ.measure(gc[1], c1[1])
circ.draw()
```

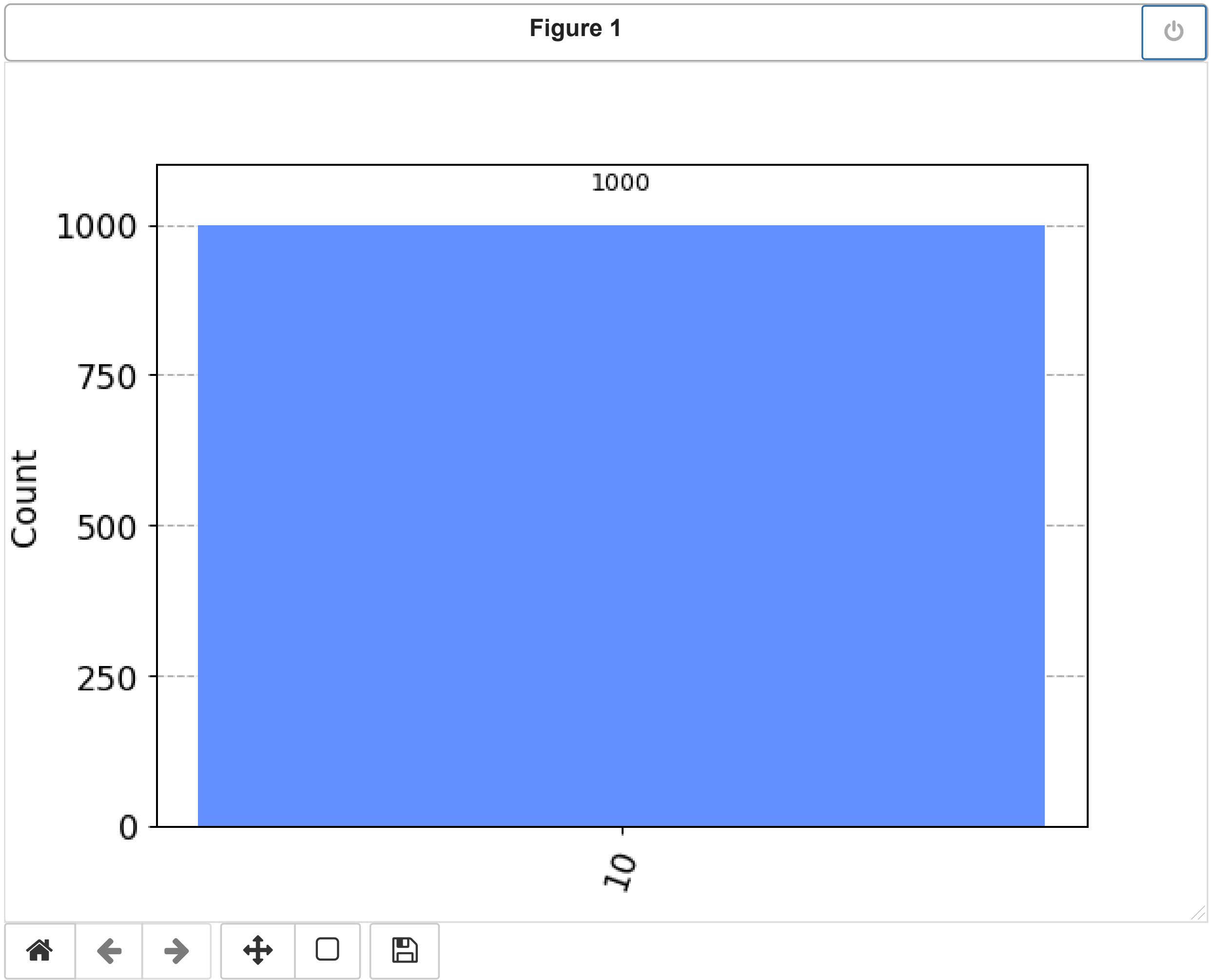
Out[15]:



In [16]:

```
backend_gc = BasicAer.get_backend('qasm_simulator')
job_gc = execute(circ, backend_gc, shots=1000)
result_gc = job_gc.result()
counts_gc = result_gc.get_counts()
print(counts_gc)
plot_histogram(counts_gc)
```

```
{'10': 1000}
```



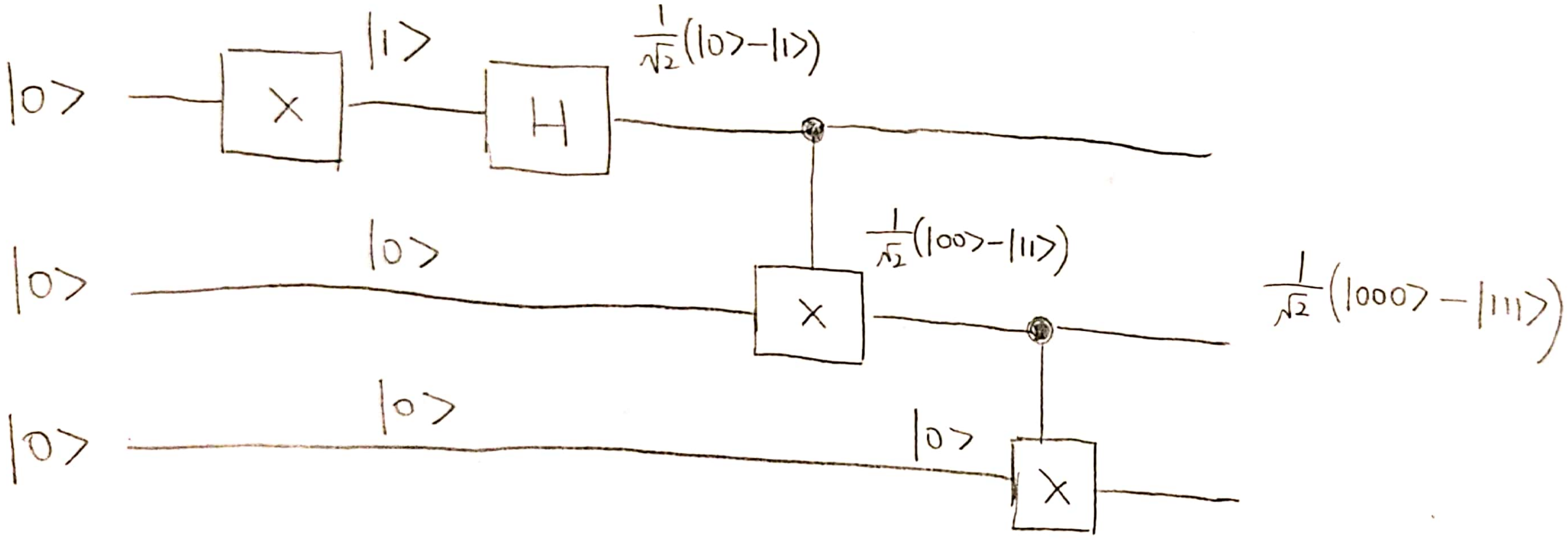
From the histogram above, we can find the measured results match with the result after $\hat{W}\hat{V}|init\rangle$.

End of Chapter Problems

Chapter8 Problem-1:

(a) Draw the circuit to realize $|\psi >= \frac{1}{\sqrt{2}}|000 > -|111 >$ from the initial state |000>.

Answer



|0>
|0>
|0>
X
|1>
H
$\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)$
|0>
|0>
X
$\frac{1}{\sqrt{2}}(|00\rangle-|11\rangle)$
|0>
X
$\frac{1}{\sqrt{2}}(|000\rangle-|111\rangle)$

(b) Draw the circuit to realize (a) using Qiskit. What states do you measure at the end of the gate operations in the real experiment?

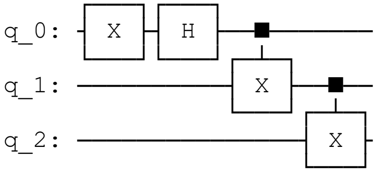
In [17]:

```
n=3
pc1 = QuantumRegister(n, 'q')
circ = QuantumCircuit(pc1)
circ.x(pc1[0])

circ.h(pc1[0])
circ.cx(pc1[0],pc1[1])
circ.cx(pc1[1],pc1[2])

circ.draw()
```

Out[17]:



In [18]:

```
backend_p1 = BasicAer.get_backend('statevector_simulator')
job_p1 = execute(circ, backend_p1)
result_p1 = job_p1.result()
state_p1 = result_p1.get_statevector(circ)
print('Statevector:', state_p1)
```

```
Statevector: [ 0.70710678+0.00000000e+00j  0.        +0.00000000e+00j
  0.        +0.00000000e+00j  0.        +0.00000000e+00j
  0.        +0.00000000e+00j  0.        +0.00000000e+00j
  0.        +0.00000000e+00j -0.70710678-8.65956056e-17j]
```

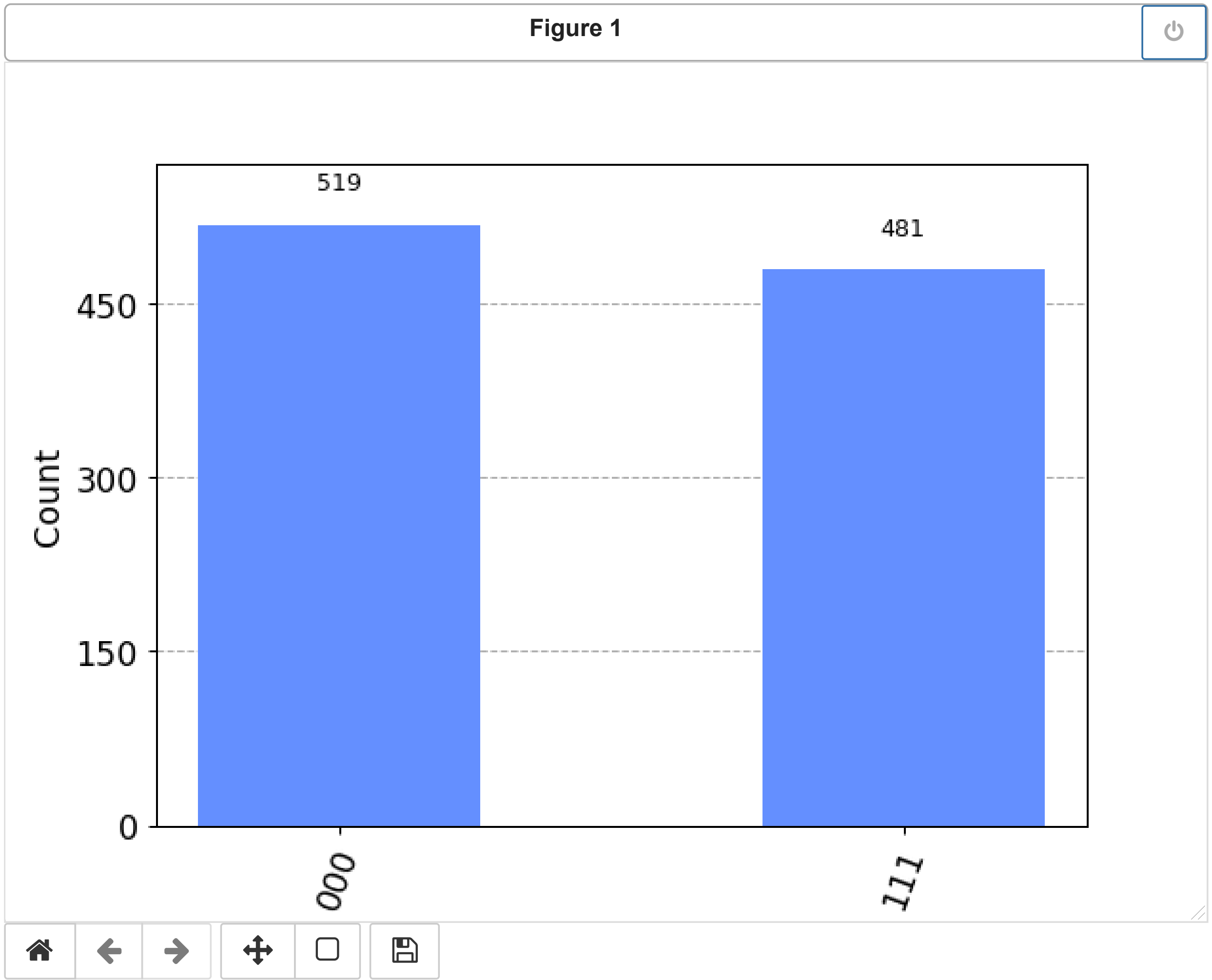
In the real measurement, we will measure the state either in $|000 >$ and $|111 >$ with eqaul prabablities. The simulation results in Qiskit is shown below.

In [19]:

```
cp1 = ClassicalRegister(n, 'c1')
circ.add_register(cp1)
circ.measure(pc1[0], cp1[0])
circ.measure(pc1[1], cp1[1])
circ.measure(pc1[2], cp1[2])

circ.draw()
backend_p1 = BasicAer.get_backend('qasm_simulator')
job_p1 = execute(circ, backend_p1, shots=1000)
result_p1 = job_p1.result()
counts_p1 = result_p1.get_counts()
print(counts_p1)
plot_histogram(counts_p1)
```

```
{'000': 519, '111': 481}
```



Chapter8 Problem-2:

A two-qubit Hadamard gate acts on the state $|00 >$. Determine if the resultsing state is an entangled or product state.

Answer:

$$H \otimes H = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & +\frac{1}{2} \end{pmatrix}$$

Apply this operator on the state $|00>$ will result in a state:

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$$

This represents the state $\frac{1}{2}\{|00>+|01>|10>+|11>\}$. Also we can see $ad=\frac{1}{4}=bc$. Thus, the resulting state is a product state.

Chapter8 Problem-3:

Design a gate to transform $\frac{1}{\sqrt{8}}\{|000>+|001>+|010>+|011>+|100>|101>+|110>+|111>\}$ to $\frac{1}{\sqrt{8}}\{-|000>+|001>+|010>+|011>-|100>|101>+|110>+|111>\}$

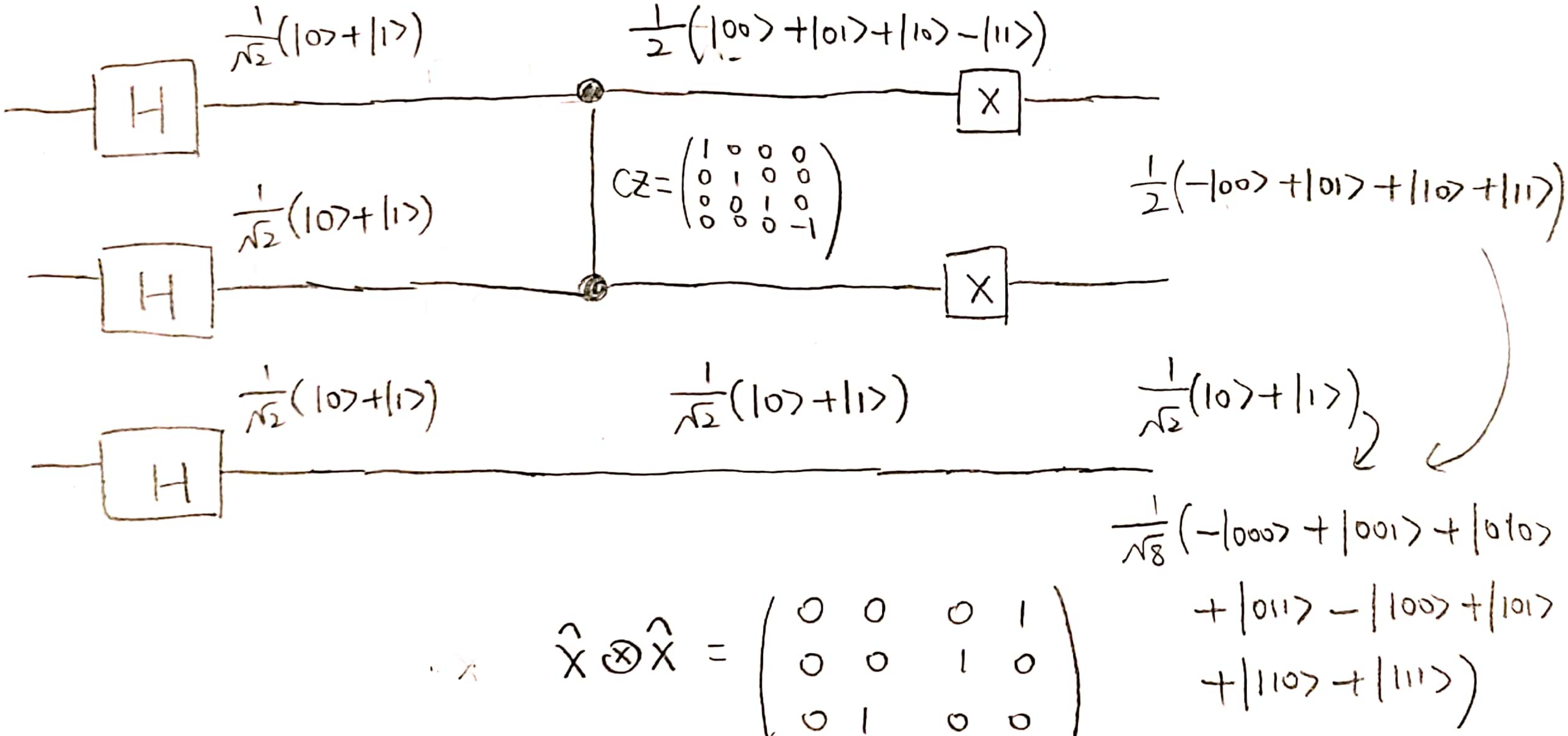
Answer

First use three H gates to generate the state $\frac{1}{\sqrt{8}}|000>+|001>+|010>+|011>+|100>|101>+|110>+|111>$.

Then, apply a CZ gate to bit-1 and bit-0, which will create a minus sign for state $|11>$.

Last, apply two X gates to bit-1 and bit-0 which will move the minus sign from the state $|11>$ to $|00>$. This effectively will introduce minus signa for the state $|000>$ and $|100>$.

The detailed circuit is shown below.


H
H
H
$\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$
$\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$
$\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$
$\frac{1}{2}(|00\rangle+|01\rangle+|10\rangle-|11\rangle)$
$CZ=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&-1\end{pmatrix}$
$\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$
X
X
$\frac{1}{2}(-|00\rangle+|01\rangle+|10\rangle+|11\rangle)$
$\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$
$\frac{1}{\sqrt{8}}(-|000\rangle+|001\rangle+|010\rangle+|011\rangle-|100\rangle+|101\rangle+|110\rangle+|111\rangle)$

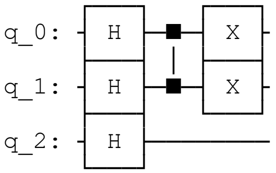
$$\hat{X}\otimes\hat{X}=\begin{pmatrix}0&0&0&1\\0&0&1&0\\0&1&0&0\\1&0&0&0\end{pmatrix}$$

The quantum circuit is verified in Qiskit which is shown below.

In [20]:
```
n=3
pc3 = QuantumRegister(n, 'q')
circ = QuantumCircuit(pc3)
circ.h([0,1,2])
#circ.x([0,1])
circ.cz(0,1)
circ.x([0,1])


circ.draw()
```

Out[20]:



In [21]:
```
backend_p3 = BasicAer.get_backend('statevector_simulator')
job_p3 = execute(circ, backend_p3)
result_p3 = job_p3.result()
state_p3 = result_p3.get_statevector(circ)
print('Statevector:', state_p3)
```

```
Statevector: [-0.35355339+0.j  0.35355339+0.j  0.35355339+0.j  0.35355339+0.j
 -0.35355339+0.j  0.35355339+0.j  0.35355339+0.j  0.35355339+0.j]
```

In [22]:
```
#cp3 = ClassicalRegister(n, 'c1')
#circ.add_register(cp3)
#circ.measure(pc3[0], cp3[0])
#circ.measure(pc3[1], cp3[1])
#circ.measure(pc3[2], cp3[2])

#circ.draw()
#backend_p3 = BasicAer.get_backend('qasm_simulator')
#job_p3 = execute(circ, backend_p3, shots=1000)
#result_p3 = job_p3.result()
#counts_p3 = result_p3.get_counts()
#print(counts_p3)
#plot_histogram(counts_p3)
```

Chapter8 Problem-4:

Show that if Bob apploes $Z_B X_B$ to $a|1_B > -b|0_B >$, Bob obtains $a|0_B > +b|1_B >$

Answer:

$Z_B X_B$ means apply a gate operation X_B and a gate operation Z_B to a single qubit in sequence.
After applying gate operation X_B to a single qubit $a|1_B> -b|0_B>$:

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

Then apply gate operation Z_B to the qubit state $-b|1_B> +a|0_B>$:

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

Thus, Bob obtains $a|0_B> +b|1_B>$.

In [ ]:

```
```

In [ ]:

```
```