Quantum Computing Assignment 4 - Group 18

Rallabhandi, Anand Krishna   Mustafa, Syed Husain   , Mohammed Kamran

December 10, 2020

Exercise 5.1
Bell Inequality

(Bell states and superdense coding)

(a)

Given : $|0\rangle = \alpha|a\rangle + \beta|b\rangle$ & $|1\rangle = \gamma|a\rangle + \delta|b\rangle$, where $\{|a\rangle, |b\rangle\}$ is the orthonormal basis of $\mathbb{C}^2$

Consider $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$

$$|01\rangle = |0\rangle \otimes |1\rangle \implies |01\rangle = \begin{pmatrix} \alpha a \begin{pmatrix} \gamma a \\ \delta b \end{pmatrix} \\ \beta b \begin{pmatrix} \gamma a \\ \delta b \end{pmatrix} \end{pmatrix} \implies |01\rangle = \Big(\alpha\gamma|aa\rangle + \alpha\delta|ab\rangle + \beta\gamma|ba\rangle + \beta\delta|bb\rangle\Big) \quad \textbf{(I)}$$

$$|10\rangle = |1\rangle \otimes |0\rangle \implies |10\rangle = \begin{pmatrix} \gamma a \begin{pmatrix} \alpha a \\ \beta b \end{pmatrix} \\ \delta b \begin{pmatrix} \alpha a \\ \beta b \end{pmatrix} \end{pmatrix} \implies |10\rangle = \Big(\alpha\gamma|aa\rangle + \beta\gamma|ab\rangle + \alpha\delta|ba\rangle + \beta\delta|bb\rangle\Big) \quad \textbf{(II)}$$

$$\frac{1}{\sqrt{2}}\Big(\textbf{(I)} - \textbf{(II)}\Big) = \frac{1}{\sqrt{2}}\begin{pmatrix} \alpha\gamma aa - \alpha\gamma aa \\ \alpha\delta ab - \beta\gamma \\ \beta\gamma ba - \alpha\delta ba \\ \beta\delta bb - \beta\delta bb \end{pmatrix} = \frac{1}{\sqrt{2}}(\alpha\delta - \beta\gamma)\frac{|ab\rangle - |ba\rangle}{\sqrt{2}}$$

$$\therefore \frac{|01\rangle\,|10\rangle}{\sqrt{2}} = (\alpha\delta - \beta\gamma)\frac{|ab\rangle - |ba\rangle}{\sqrt{2}}$$

(b)

$$\text{Given, } |\psi\rangle = \tfrac{1}{\sqrt{2}}\big(\,|01\rangle - |10\rangle\,\big) \equiv \frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix} = |\beta_{11}\rangle$$

$$Q = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad R = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$S = \frac{1}{\sqrt{2}}(-Z - X) = \frac{1}{\sqrt{2}}\left(-\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix} - \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\right) = \frac{1}{\sqrt{2}}\begin{pmatrix}-1 & -1\\ -1 & 1\end{pmatrix}$$

$$T = \frac{1}{\sqrt{2}}(Z - X) = \frac{1}{\sqrt{2}}\left(\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix} - \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\right) = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & -1\\ -1 & -1\end{pmatrix}$$

$$\text{(i)}\ \ \langle\psi|\, Q \otimes S\, |\psi\rangle = \frac{1}{2\sqrt{2}}\begin{pmatrix}0 & 1 & -1 & 0\end{pmatrix}\begin{pmatrix}-1 & -1 & 0 & 0\\ -1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\\ 0 & 0 & 1 & -1\end{pmatrix}\begin{pmatrix}0\\ 1\\ -1\\ 0\end{pmatrix} = \frac{1}{2\sqrt{2}}\begin{pmatrix}-1 & 1 & -1 & -1\end{pmatrix}\begin{pmatrix}0\\ 1\\ -1\\ 0\end{pmatrix} = \frac{1}{\sqrt{2}}$$

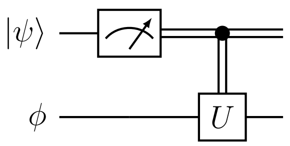
$$\text{(ii)}\ \ \langle\psi|\, R \otimes S\, |\psi\rangle = \frac{1}{2\sqrt{2}}\begin{pmatrix}0 & 1 & -1 & 0\end{pmatrix}\begin{pmatrix}0 & 0 & -1 & -1\\ 0 & 0 & -1 & 1\\ -1 & -1 & 0 & 0\\ -1 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}0\\ 1\\ -1\\ 0\end{pmatrix} = \frac{1}{2\sqrt{2}}\begin{pmatrix}1 & 1 & -1 & 1\end{pmatrix}\begin{pmatrix}0\\ 1\\ -1\\ 0\end{pmatrix} = \frac{1}{\sqrt{2}}$$

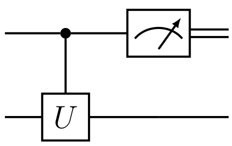
$$\text{(iii)}\ \ \langle\psi|\, R \otimes T\, |\psi\rangle = \frac{1}{2\sqrt{2}}\begin{pmatrix}0 & 1 & -1 & 0\end{pmatrix}\begin{pmatrix}0 & 0 & 1 & -1\\ 0 & 0 & -1 & -1\\ 1 & -1 & 0 & 0\\ -1 & -1 & 0 & 0\end{pmatrix}\begin{pmatrix}0\\ 1\\ -1\\ 0\end{pmatrix} = \frac{1}{2\sqrt{2}}\begin{pmatrix}-1 & 1 & -1 & -1\end{pmatrix}\begin{pmatrix}0\\ 1\\ -1\\ 0\end{pmatrix} = \frac{1}{\sqrt{2}}$$

$$\text{(iv)}\ \ \langle\psi|\, Q \otimes T\, |\psi\rangle = \frac{1}{2\sqrt{2}}\begin{pmatrix}0 & 1 & -1 & 0\end{pmatrix}\begin{pmatrix}1 & -1 & 0 & 0\\ -1 & -1 & 0 & 0\\ 0 & 0 & -1 & 0\\ 0 & 0 & 1 & 1\end{pmatrix}\begin{pmatrix}0\\ 1\\ -1\\ 0\end{pmatrix} = \frac{1}{2\sqrt{2}}\begin{pmatrix}-1 & -1 & 1 & -1\end{pmatrix}\begin{pmatrix}0\\ 1\\ -1\\ 0\end{pmatrix} = -\frac{1}{\sqrt{2}}$$

Exercise 5.2
Quantum Teleportation Circuit using IBM Q & Qiskit

(a)





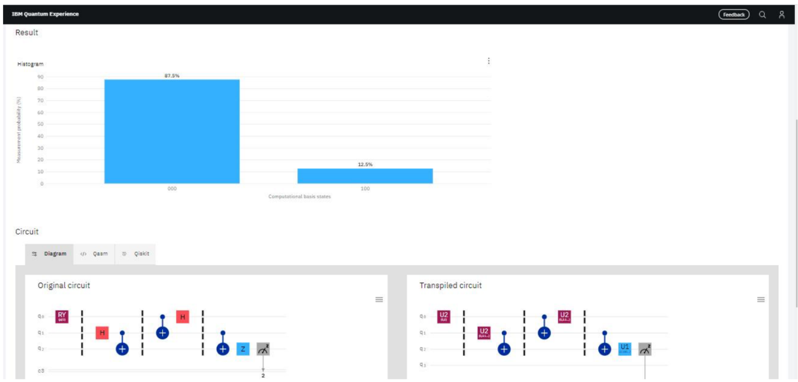
Left Figure:
$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ (Given)
Consider state $|\Phi_1\rangle = (\alpha\,|0\rangle + \beta\,|1\rangle) \otimes |\phi\rangle$ (State after meter)
Here the probability of $|0\rangle\,|\phi\rangle$ is $|\alpha|^2$ & probability of $|1\rangle\,|\phi\rangle$ is $|\beta|^2$, where $|\alpha|^2 + |\beta|^2 = 1$.
Consider state $|\Phi_2\rangle = (\alpha\,|0\rangle\,|\phi\rangle + \beta U\,|1\rangle\,|\phi\rangle)$ (State after Control-U Gate)
Here probability of $|0\rangle\,|\phi\rangle$ is $|\alpha|^2$ & the probability of $U\,|1\rangle\,|\phi\rangle$ is $|\beta|^2$.

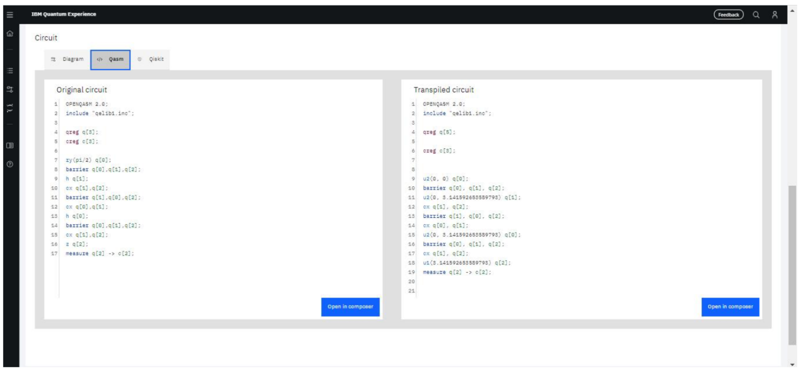
Right Figure:
Consider $|\Phi_3\rangle = (\alpha\,|0\rangle\,|\phi\rangle + \beta U\,|1\rangle\,|\phi\rangle)$ (State after Control-U Gate)
Here probability of $|0\rangle\,|\phi\rangle$ is $|\alpha|^2$ & the probability of $U\,|1\rangle\,|\phi\rangle$ is $|\beta|^2$.
Consider $|\Phi_4\rangle = (\alpha\,|0\rangle\,|\phi\rangle + \beta U\,|1\rangle\,|\phi\rangle)$. (State after meter)
Here probability of $|0\rangle\,|\phi\rangle$ is $|\alpha|^2$ & the probability of $U\,|1\rangle\,|\phi\rangle$ is $|\beta|^2$.

Conclusion: We can see in both the figures that at the end both $|\Phi_2\rangle$ & $|\Phi_4\rangle$ have the same probability distribution. Hence we can conclude that the placement of the meter does not infact change our end result.

$\therefore$ The *deferred measurement* assertion is true.

(b)





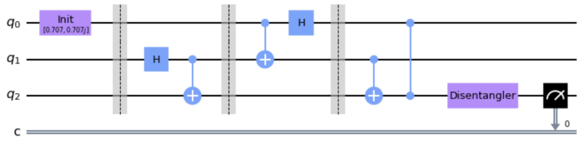
(c)

```
In [72]: %matplotlib inline
         # Importing standard Qiskit libraries
         from qiskit import QuantumCircuit, execute, Aer, IBMQ
         from qiskit.compiler import transpile, assemble
         from qiskit.tools.jupyter import *
         from qiskit.visualization import *
         from iqx import *
         from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister, execute, BasicAer, IBMQ
         from qiskit.visualization import plot_histogram, plot_bloch_multivector
         from qiskit.extensions import Initialize
         from qiskit_textbook.tools import random_state, array_to_latex
         import math as m
         # Loading your IBM Q account(s)
         provider = IBMQ.load_account()

          ibmqfactory.load_account:WARNING:2020-12-08 21:17:35,496: Credentials are already in use. The existing account in the session will be re
          placed.
```

```
In [73]: from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
         from numpy import pi

         def create_bell_pair(qc, a, b):
             """Creates a bell pair in qc using qubits a & b"""
             #qc.h(a) # Put qubit a into state |+>
             qc.h(a)
             qc.cx(a,b) # CNOT with a as control and b as targetdef create_bell_pair(qc, a, b):
         def alice_gates(qc, psi, a):
             qc.cx(psi, a)
             qc.h(psi)
         def measure_and_send(qc, a, b):
             """Measures qubits a & b and 'sends' the results to Bob"""
             qc.barrier()
             qc.measure(a,0)
             qc.measure(b,1)
         def bob_gates(qc, qubit, crz, crx):
             # Here we use c_if to control our gates with a classical
             # bit instead of a qubit
             qc.x(qubit).c_if(crx, 1) # Apply gates if the registers
             qc.z(qubit).c_if(crz, 1) # are in the state '1'
         def new_bob_gates(qc, a, b, c):
             qc.cx(b, c)
             qc.cz(a, c)
```

In [94]:

```
teleportation_circuit= QuantumCircuit(3, 1)
#psi = [
#    m.sqrt(3) / 2 * complex(0,1),
#    1 / 2 * complex(1,0)]
initial_state = [m.sqrt(3)/2, 1/(2)]  # Define state |q_0>

init_gate = Initialize(psi)
init_gate.label = "init"
inverse_init_gate = init_gate.gates_to_uncompute()
## STEP 0
# First, let's initialize Alice's q0
teleportation_circuit.append(init_gate, [0])
teleportation_circuit.barrier()

## STEP 1
create_bell_pair(teleportation_circuit, 1, 2)

## STEP 2
teleportation_circuit.barrier() # Use barrier to separate steps
alice_gates(teleportation_circuit, 0, 1)

## STEP 3
#measure_and_send(teleportation_circuit, 0, 1)

## STEP 4
teleportation_circuit.barrier() # Use barrier to separate steps
new_bob_gates(teleportation_circuit, 0,1,2)
## STEP 5
# reverse the initialization process
teleportation_circuit.append(inverse_init_gate, [2])
teleportation_circuit.measure(2,0)
teleportation_circuit.draw()
```



In [84]:

```
IBMQ.load_account()
provider = IBMQ.get_provider(hub='ibm-q')
```

```
ibmqfactory.load_account:WARNING:2020-12-08 21:27:40,591: Credentials are already in use. The existing account in the session will be re
placed.
```

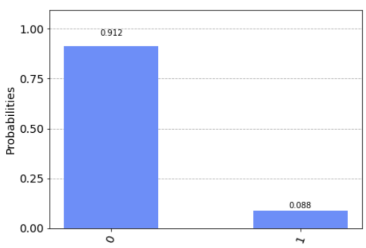
In [95]:

```
# get the least-busy backend at IBM and run the quantum circuit there
from qiskit.providers.ibmq import least_busy
backend = least_busy(provider.backends(filters=lambda b: b.configuration().n_qubits >= 3 and
                                   not b.configuration().simulator and b.status().operational==True))
job_exp = execute(teleportation_circuit, backend=backend, shots=1024)
```

In [96]:

```
from qiskit.tools.monitor import job_monitor
job_monitor(job_exp)  # displays job status under cell

# Get the results and display them
exp_result = job_exp.result()
exp_measurement_result = exp_result.get_counts(teleportation_circuit)
print(exp_measurement_result)
plot_histogram(exp_measurement_result)
```

```
Job Status: job has successfully run
{'0': 934, '1': 90}
```



In [98]:

```
error_rate_percent = sum([exp_measurement_result[result] for result in exp_measurement_result.keys() if result[0]=='1']) \
                    * 100./ sum(list(exp_measurement_result.values()))
print("The experimental error rate : ", error_rate_percent, "%")
```

```
The experimental error rate :  8.7890625 %
```

In [99]:

```
sim = Aer.get_backend('statevector_simulator')
job = execute(teleportation_circuit, sim)
print(job.result().get_statevector(teleportation_circuit))
```

```
[0.5-1.5308085e-17j 0.5-1.5308085e-17j 0.5-1.5308085e-17j
 0.5-1.5308085e-17j 0. -0.0000000e+00j 0. +0.0000000e+00j
 0. +0.0000000e+00j 0. +0.0000000e+00j]
```