陳志偉 B06901126 電機五

Electrical Engineering Lab（topics on Communication System）
Lab1 Report

All source will push to my github repo: https://github.com/finalwee/CommLab

1. a)

```
alice send: [0 0 0 0] || bob get: {'0000': 1024} ==> Accuracy = 100.00%
alice send: [0 0 0 1] || bob get: {'1000': 1024} ==> Accuracy = 100.00%
alice send: [0 0 1 0] || bob get: {'0100': 1024} ==> Accuracy = 100.00%
alice send: [0 1 0 0] || bob get: {'0010': 1024} ==> Accuracy = 100.00%
alice send: [1 0 0 0] || bob get: {'0001': 1024} ==> Accuracy = 100.00%
alice send: [0 0 1 1] || bob get: {'1100': 1024} ==> Accuracy = 100.00%
alice send: [0 1 1 0] || bob get: {'0110': 1024} ==> Accuracy = 100.00%
alice send: [1 1 0 0] || bob get: {'0011': 1024} ==> Accuracy = 100.00%
alice send: [1 0 0 1] || bob get: {'1001': 1024} ==> Accuracy = 100.00%
alice send: [0 1 1 1] || bob get: {'1110': 1024} ==> Accuracy = 100.00%
alice send: [1 1 1 0] || bob get: {'0111': 1024} ==> Accuracy = 100.00%
alice send: [1 1 0 1] || bob get: {'1011': 1024} ==> Accuracy = 100.00%
alice send: [1 0 1 1] || bob get: {'1101': 1024} ==> Accuracy = 100.00%
alice send: [1 1 1 1] || bob get: {'1111': 1024} ==> Accuracy = 100.00%
```

b)

```
least busy backend:  ibmq_lima
Job Status: job has successfully run
alice send: [0 0 0 0] || bob get: {'0000': 353, '0001': 11, '0010': 3, '0011': 3, '0100': 189, '0101': 5, '0110': 2, '0111': 
1, '1000': 213, '1001': 4, '1010': 3, '1011': 1, '1100': 233, '1101': 1, '1110': 1, '1111': 1} ==> Accuracy: 34.47%
SER: 33.69% BER: 22.75%
================================================================================
Job Status: job has successfully run
alice send: [0 0 0 1] || bob get: {'0000': 156, '0010': 2, '0100': 203, '0101': 1, '0110': 1, '0111': 1, '1000': 376, '100
1': 3, '1010': 6, '1011': 2, '1100': 266, '1101': 2, '1110': 2, '1111': 3} ==> Accuracy: 36.72%
SER: 32.23% BER: 21.29%
================================================================================
Job Status: job has successfully run
alice send: [0 0 1 0] || bob get: {'0000': 193, '0001': 2, '0010': 2, '0011': 1, '0100': 379, '0101': 3, '0110': 4, '0111': 
1, '1000': 168, '1001': 2, '1010': 4, '1011': 2, '1100': 253, '1101': 4, '1110': 2, '1111': 4} ==> Accuracy: 37.01%
SER: 32.62% BER: 20.80%
================================================================================
Job Status: job has successfully run
alice send: [0 1 0 0] || bob get: {'0000': 9, '0001': 1, '0010': 344, '0100': 4, '0110': 201, '0111': 2, '1000': 4, '1010': 
186, '1011': 2, '1100': 10, '1110': 260, '1111': 1} ==> Accuracy: 33.59%
SER: 34.33% BER: 23.80%
================================================================================
Job Status: job has successfully run
alice send: [1 0 0 0] || bob get: {'0000': 11, '0001': 360, '0010': 1, '0011': 5, '0100': 1, '0101': 194, '0111': 4, '1000': 
5, '1001': 171, '1011': 4, '1100': 5, '1101': 261, '1111': 2} ==> Accuracy: 35.16%
SER: 33.45% BER: 23.29%
================================================================================
Job Status: job has successfully run
alice send: [0 0 1 1] || bob get: {'0000': 153, '0001': 1, '0010': 1, '0100': 245, '0101': 2, '0110': 3, '1000': 254, '100
1': 3, '1010': 6, '1100': 346, '1101': 3, '1110': 4, '1111': 3} ==> Accuracy: 33.79%
SER: 33.89% BER: 20.80%
================================================================================
Job Status: job has successfully run
alice send: [0 1 1 0] || bob get: {'0000': 6, '0010': 210, '0011': 1, '0100': 13, '0101': 3, '0110': 344, '0111': 8, '1000': 
1, '1001': 1, '1010': 185, '1011': 1, '1100': 10, '1101': 1, '1110': 238, '1111': 2} ==> Accuracy: 33.59%
SER: 34.33% BER: 21.88%
================================================================================
Job Status: job has successfully run
alice send: [1 1 0 0] || bob get: {'0000': 1, '0001': 4, '0010': 9, '0011': 343, '0101': 7, '0110': 2, '0111': 200, '1001': 
6, '1010': 3, '1011': 185, '1100': 3, '1101': 7, '1110': 8, '1111': 246} ==> Accuracy: 33.50%
SER: 35.01% BER: 24.05%
================================================================================
Job Status: job has successfully run
alice send: [1 0 0 1] || bob get: {'0000': 1, '0001': 171, '0010': 1, '0011': 3, '0100': 6, '0101': 194, '0111': 6, '1000': 
13, '1001': 343, '1011': 10, '1100': 8, '1101': 259, '1110': 1, '1111': 8} ==> Accuracy: 33.50%
SER: 34.91% BER: 22.53%
================================================================================
Job Status: job has successfully run
alice send: [0 1 1 1] || bob get: {'0000': 3, '0010': 139, '0011': 4, '0100': 2, '0110': 223, '0111': 3, '1000': 8, '1010': 
227, '1011': 3, '1100': 7, '1101': 1, '1110': 398, '1111': 6} ==> Accuracy: 38.87%
SER: 31.69% BER: 19.43%
================================================================================
Job Status: job has successfully run
alice send: [1 1 1 0] || bob get: {'0000': 2, '0001': 7, '0010': 3, '0011': 194, '0100': 1, '0101': 11, '0110': 12, '0111': 
325, '1000': 3, '1001': 7, '1010': 4, '1011': 174, '1101': 6, '1110': 3, '1111': 272} ==> Accuracy: 31.74%
SER: 35.84% BER: 22.66%
================================================================================
```

```
Job Status: job has successfully run
alice send: [1 1 0 1] || bob get: {'0000': 2, '0001': 3, '0010': 1, '0011': 172, '0100': 1, '0101': 5, '0110': 3, '0111': 21
5, '1000': 2, '1001': 15, '1010': 12, '1011': 346, '1101': 6, '1110': 8, '1111': 233} ==> Accuracy: 33.79%
SER: 34.52% BER: 22.85%
================================================================================
Job Status: job has successfully run
alice send: [1 0 1 1] || bob get: {'0000': 6, '0001': 165, '0010': 3, '0011': 5, '0100': 7, '0101': 225, '0110': 2, '0111':
9, '1000': 9, '1001': 232, '1011': 2, '1100': 10, '1101': 340, '1110': 1, '1111': 8} ==> Accuracy: 33.20%
SER: 35.50% BER: 22.27%
================================================================================
Job Status: job has successfully run
alice send: [1 1 1 1] || bob get: {'0000': 1, '0001': 3, '0010': 7, '0011': 146, '0100': 1, '0101': 5, '0110': 4, '0111': 22
9, '1000': 1, '1001': 6, '1010': 2, '1011': 230, '1100': 1, '1101': 8, '1110': 10, '1111': 370} ==> Accuracy: 36.13%
SER: 33.40% BER: 20.63%
================================================================================
```

Code:

```python
def callBellPair(circuit, h_qubit, x_qubit):
    circuit.h(q[h_qubit])
    circuit.cx(q[h_qubit], q[x_qubit])

    return circuit

def encodeMsg(circuit, qubit, msg):
    if len(msg) > 6:
        raise ValueError(f"message '{msg}' is invalid")

    if msg[0] == 1:
        circuit.z(q[qubit])

    if msg[1] == 1:
        circuit.x(q[qubit])

    if msg[2] == 1:
        circuit.z(q[qubit+2])

    if msg[3] == 1:
        circuit.x(q[qubit+2])

    circuit.id(q[qubit])
    circuit.id(q[qubit+2])

    return circuit

def decodeMsg(circuit, h_qubit, x_qubit):
    circuit.cx(q[h_qubit], q[x_qubit])
    circuit.h(q[h_qubit])

    return circuit
```

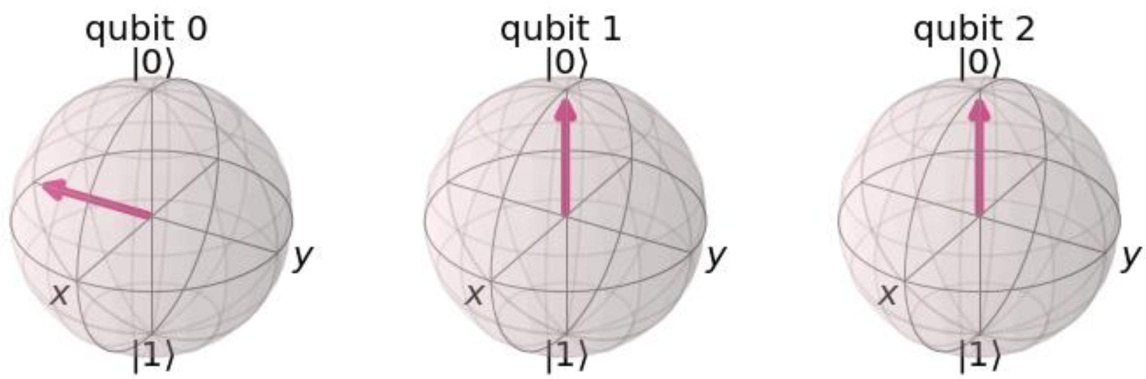
```
alice_bits = np.array([
    [0, 0, 0, 0],
    [0, 0, 0, 1],
    [0, 0, 1, 0],
    [0, 1, 0, 0],
    [1, 0, 0, 0],
    [0, 0, 1, 1],
    [0, 1, 1, 0],
    [1, 1, 0, 0],
    [1, 0, 0, 1],
    [0, 1, 1, 1],
    [1, 1, 1, 0],
    [1, 1, 0, 1],
    [1, 0, 1, 1],
    [1, 1, 1, 1],
])

# For IBMQ
from qiskit.tools.monitor import job_monitor
from qiskit.providers.ibmq import least_busy
# Load local account information
IBMQ.load_account()
# Get the least busy backend
provider = IBMQ.get_provider(hub='ibm-q')
backend = least_busy(provider.backends(filters=lambda x:
x.configuration().n_qubits >= 2
and not x.configuration().simulator
and x.status().operational==True))
print("least busy backend: ", backend)

shots = 1024

for alice in alice_bits:
    q = QuantumRegister(4)
    c = ClassicalRegister(4)
    circuit = QuantumCircuit(q, c)

    circuit = callBellPair(circuit, 1, 0)
    circuit = callBellPair(circuit, 3, 2)
    circuit.barrier()
    circuit = encodeMsg(circuit, 0, alice)
    circuit.barrier()
    circuit = decodeMsg(circuit, 0, 1)
    circuit = decodeMsg(circuit, 2, 3)

    circuit.barrier()
    circuit.measure(q, c)

    # For qsam_simulator
#     simulator = Aer.get_backend('qasm_simulator')
#     job = execute(circuit, simulator, shots = shots)

    # For IBMQ
    job = execute(circuit, backend = qcomp, shots = shots)
    job_monitor(job)

    result = job.result()
    counts = result.get_counts(circuit)

    # Flip alice to check accuracy
    message = np.flip(alice)
    correct_result = counts[''.join(map(str, message))]
    accuracy = (correct_result/shots)*100

    message = np.flip(alice)
    message = ''.join(map(str, message))
```

```
first_qubit = message[:2]
second_qubit = message[2:4]

BER = 0
SER = 0

for state, count in counts.items():
    correct_bit = 0
    correct_symbol = 0
    for i in range(len(state)):
        if state[i] == message[i]:
            correct_bit += 1
    BER += (correct_bit/len(state)*count)

    if state[:2] == first_qubit:
        correct_symbol += 1

    if state[2:4] == second_qubit:
        correct_symbol += 1

    SER += (correct_symbol/(len(state)/2)*count)

print(f"alice send: {alice} ||", f"bob get: {counts}", "==>", f"Accuracy: {accuracy:.2f}%")
print(f"SER: {(1-SER/shots)*100:.2f}%", f"BER: {(1-BER/shots)*100:.2f}%")
print("================================================================================")
```

2. a)

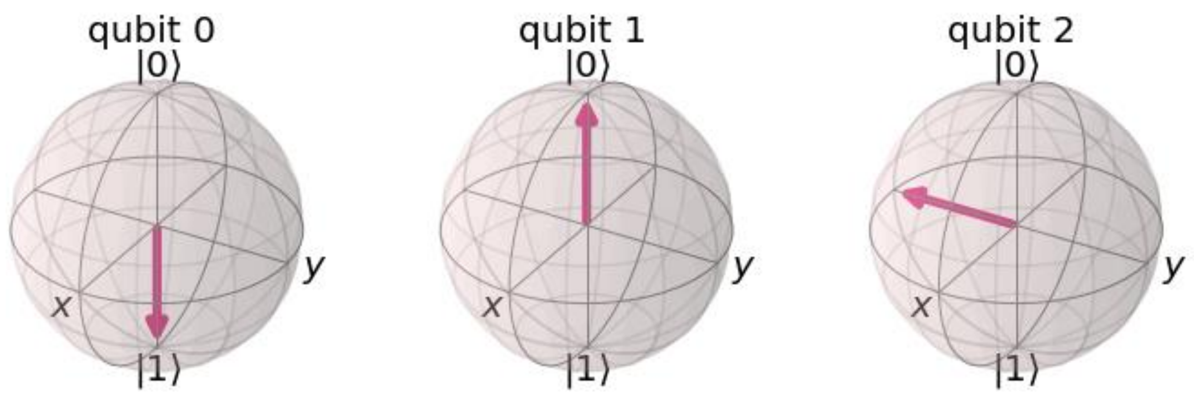
i) $|\psi\rangle_B$:

```
[0.65328148+0.27059805j 0.27059805-0.65328148j 0.        +0.j
 0.        +0.j         0.        +0.j         0.        +0.j
 0.        +0.j         0.        +0.j        ]
```



$|\psi'\rangle_B$:

```
[ 0.        +0.j          0.65328148+0.27059805j  0.        +0.j
 -0.        +0.j         -0.        +0.j          0.27059805-0.65328148j
 -0.        +0.j         -0.        +0.j        ]
```



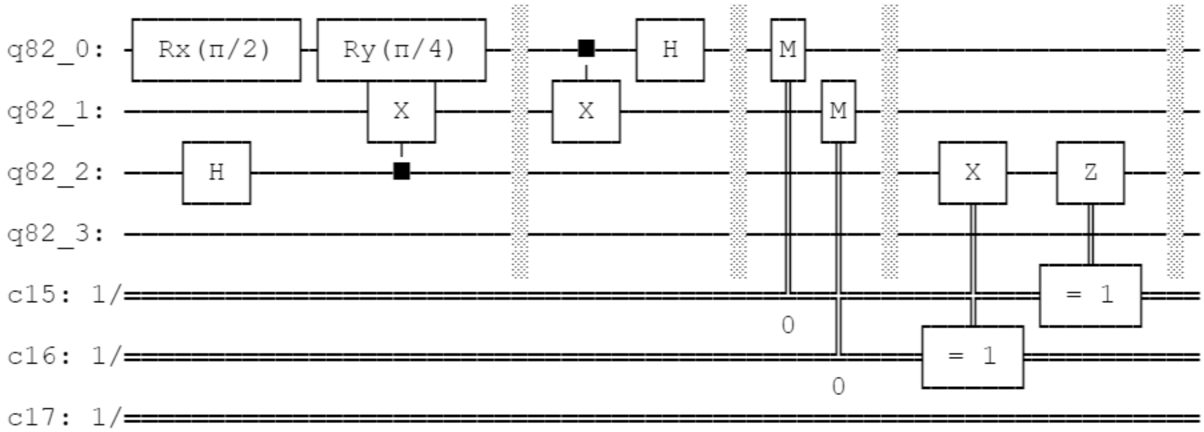


Figure 1
Figure 1 show the circuit for quantum teleportation. From the figure of i), quantum teleportation work well.

Code:

```
def callBellPair(circuit, h_qubit, x_qubit):
    circuit.h(h_qubit)
    circuit.cx(h_qubit, x_qubit)

    return circuit

def callInvBellPair(circuit, h_qubit, x_qubit):
    circuit.cx(h_qubit, x_qubit)
    circuit.h(h_qubit)

    return circuit

def measureSend(circuit, a, b):
    circuit.measure(a, 0)
    circuit.measure(b, 1)

    return circuit

def bobGates(circuit, qubit, cz, cx):
    circuit.x(qubit).c_if(cx, 1)
    circuit.z(qubit).c_if(cz, 1)

    return circuit
```

```python
q = QuantumRegister(4)
cz = ClassicalRegister(1)
cx = ClassicalRegister(1)
c = ClassicalRegister(1)
circuit = QuantumCircuit(q, cz, cx, c)

circuit.rx(pi/2, q[0])
circuit.ry(pi/4, q[0])
circuit = callBellPair(circuit, 2, 1)

# circuit.id(q[3]) # Perfect initial state for fourth qubit
circuit.barrier()

circuit = callInvBellPair(circuit, 0, 1)
circuit.barrier()
circuit = measureSend(circuit, 0, 1)
circuit.barrier()
circuit = bobGates(circuit, 2, cz, cx)
circuit.barrier()

# circuit.cz(q[3], q[2])
# circuit.x(q[2])
# circuit.measure(q[2], c)

circuit.draw()
```

ii)