PHYC90045 Introduction to Quantum Computing

Assignment 2

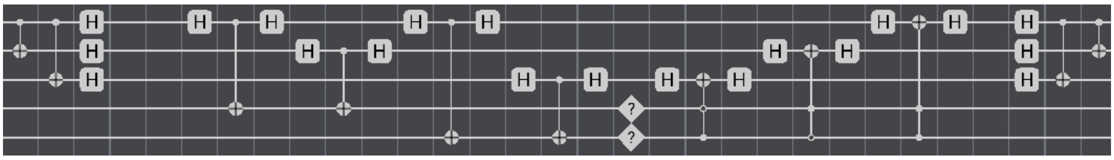
Name: Xinnan Shen

Student Number: 1051380

1.

I have introduced some single qubit gates in the syndrome stage and modified the encoding process. The measuring result for no Z error is 00; the result for Z error at first qubit is 11; the result for Z error at second qubit is 10; the result for Z error at third qubit is 01. Therefore, I use some Toffoli gates to correct the error. After correction, I decode the qubits and get the original $|q\rangle$ in the end.

Note that the first X-rotation gate is to generate the initial state $|q\rangle$.



The stabilisers are:

XXI

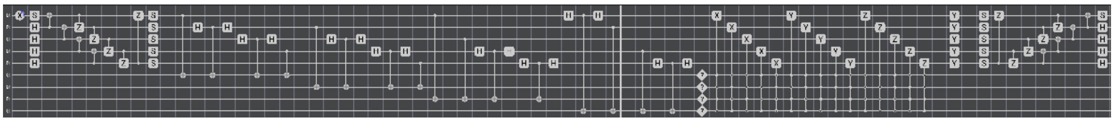
XIX

The circuit link is:

https://qui.science.unimelb.edu.au/circuits/5d9a75a02b3f4e0053f2b90d

2.

The circuit is shown as follows:



Note that the first X-rotation gate is to generate the initial state $|\psi\rangle$.

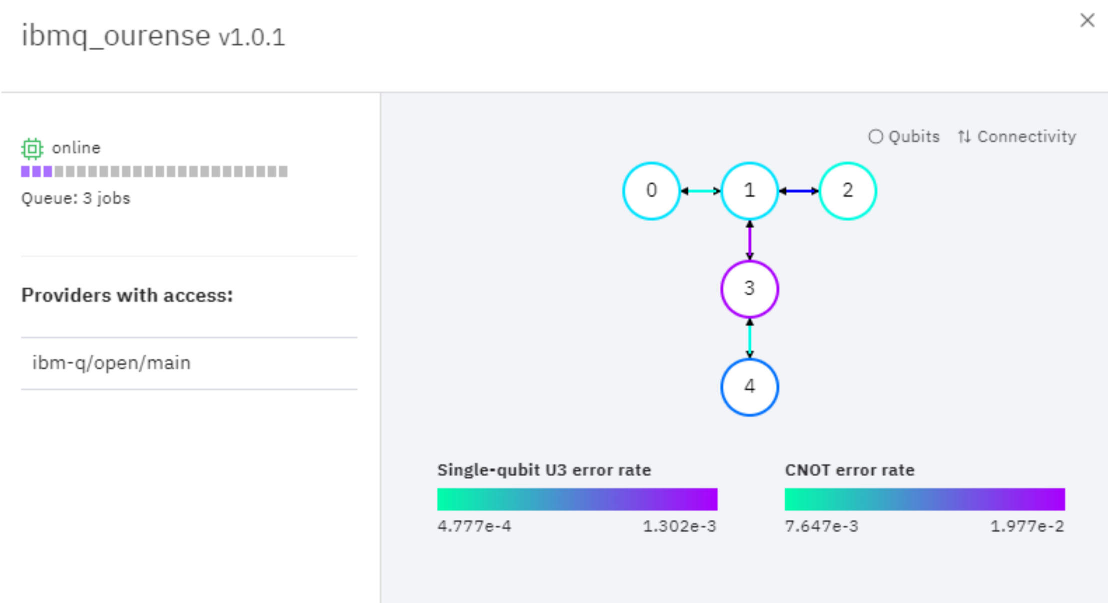
The circuit link is:

https://qui.science.unimelb.edu.au/circuits/5d9b02aa58861400a2e119e5

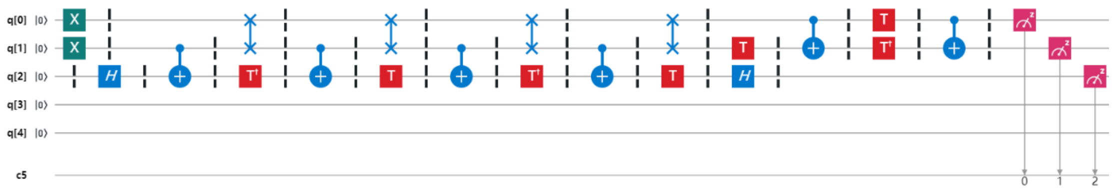
3.

(a)

I have chosen the qubit 0, 1 and 2 because these three qubits have the smallest error rates. (see the below screenshot)



The circuit is shown below:



Text code of the circuit:

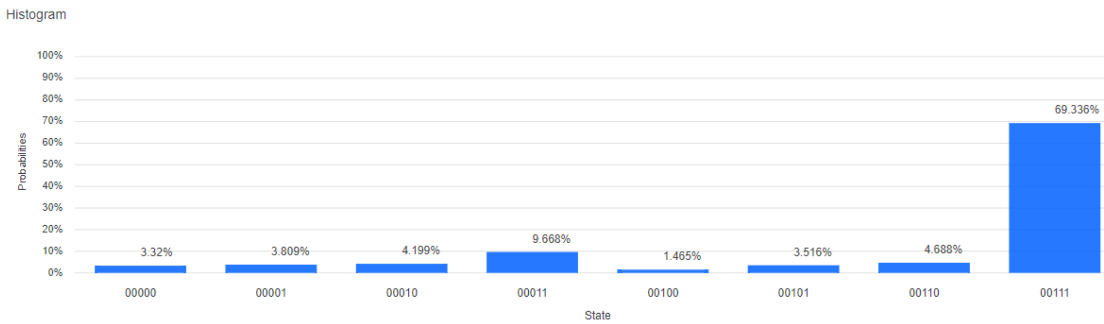
```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[5];
creg c[5];

x q[0];
x q[1];
barrier q[2];
barrier q[0];
barrier q[1];
h q[2];
barrier q[2];
cx q[1],q[2];
barrier q[1];
barrier q[2];
swap q[0],q[1];
```

```
tdg q[2];
barrier q[0];
barrier q[1];
barrier q[2];
cx q[1],q[2];
barrier q[1];
barrier q[2];
swap q[0],q[1];
t q[2];
barrier q[0];
barrier q[1];
barrier q[2];
cx q[1],q[2];
barrier q[1];
barrier q[2];
swap q[0],q[1];
tdg q[2];
barrier q[0];
barrier q[1];
barrier q[2];
cx q[1],q[2];
barrier q[1];
barrier q[2];
swap q[0],q[1];
t q[2];
barrier q[0];
barrier q[1];
barrier q[2];
t q[1];
h q[2];
barrier q[1];
barrier q[2];
cx q[0],q[1];
barrier q[0];
```

barrier q[1];
t q[0];
tdg q[1];
barrier q[0];
barrier q[1];
cx q[0],q[1];
barrier q[0];
barrier q[1];
measure q[0] -> c[0];
measure q[1] -> c[1];
measure q[2] -> c[2];

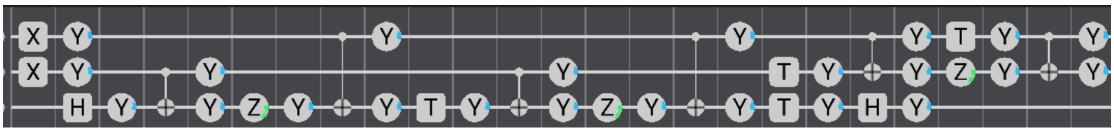
(b)

The result of circuit in (a) is shown below:



Note that in this circuit I have measured q[0] as c[0], q[1] as c[1] and q[2] as c[2], so the result states are inverse of the states in the QUI.

From the graph we can see that the probability of $|00111\rangle$ is 69%. In the QUI, I have used Y-rotation to simulate the error (the rotation angle is 0.098π), and the circuit can be transformed as below:



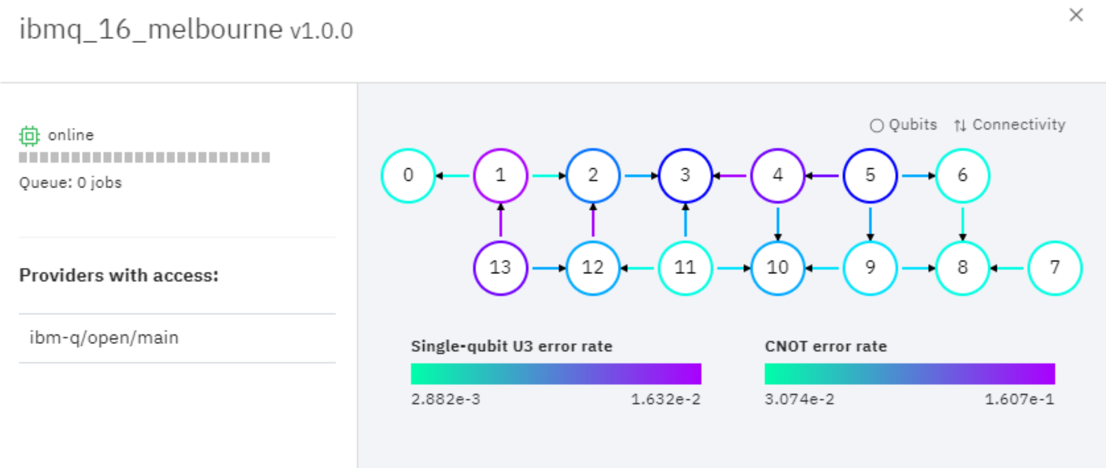
After running the circuit in the QUI, the probability of $|111\rangle$ is 69.4%, which is very close to the result in the real machine.

The circuit link is:

https://qui.science.unimelb.edu.au/circuits/5da151058d1519008abd116d

(c)

I have chosen the qubit 6, 7 and 8 of the machine as the error rates of them are the best among all the qubits. (see the screenshot below)



The program code:

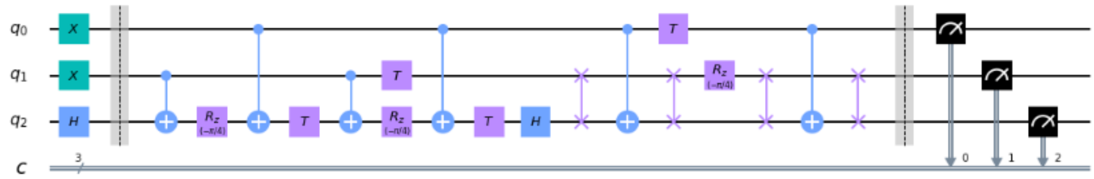
```
%matplotlib inline

# Importing standard Qiskit libraries and configuring account
import numpy as np
from qiskit import *
from qiskit.compiler import transpile, assemble
from qiskit.tools.jupyter import *
from qiskit.visualization import *

# Loading your IBM Q account(s)
provider = IBMQ.load_account()

# Create a Quantum Register with 3 qubits.
q = QuantumRegister(3, 'q')
c = ClassicalRegister(3, 'c')

# Create a Quantum Circuit acting on the q register
circ = QuantumCircuit(q, c)

circ.x(q[0])
circ.x(q[1])
circ.h(q[2])
```

```
circ.barrier()
circ.cx(q[1], q[2])
circ.rz(-np.pi/4,q[2])
circ.cx(q[0], q[2])
circ.t(q[2])
circ.cx(q[1], q[2])
circ.rz(-np.pi/4,q[2])
circ.cx(q[0], q[2])
circ.t(q[1])
circ.t(q[2])
circ.h(q[2])
circ.swap(q[1],q[2])
circ.cx(q[0], q[2])
circ.swap(q[1],q[2])
circ.t(q[0])
circ.rz(-np.pi/4,q[1])
circ.swap(q[1],q[2])
circ.cx(q[0], q[2])
circ.swap(q[1],q[2])
circ.barrier()
# Measure the results
circ.measure(q, c)

circ.draw()

backend = provider.backends()[2]
print("The backend is " + backend.name())

from qiskit.tools.monitor import job_monitor
shots = 1024
# Number of shots to run the program (experiment); maximum is 8192 shots.
job_exp = execute(circ, backend=backend, initial_layout={q[0]:6, q[1]:7,
q[2]:8},shots=shots)
job_monitor(job_exp)
```

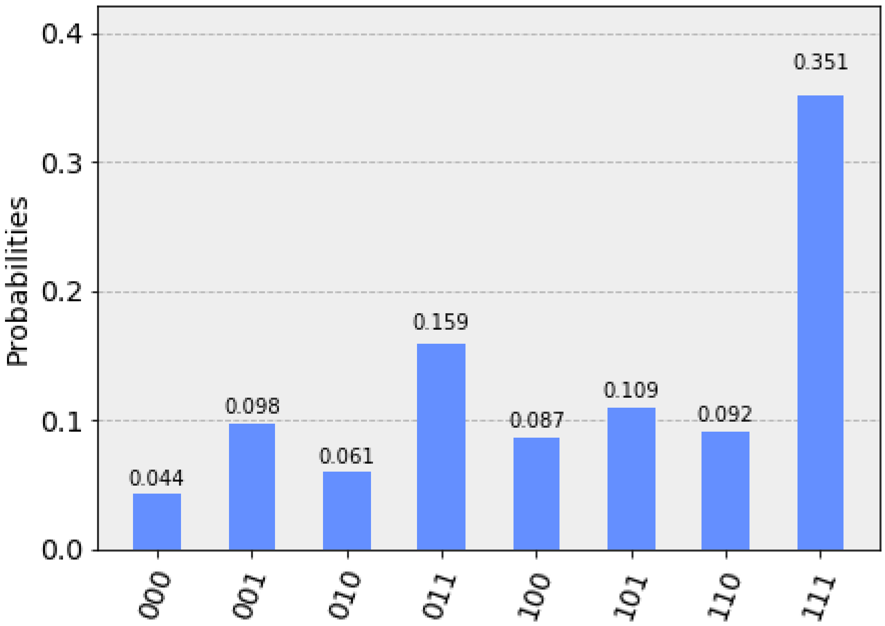
result_exp = job_exp.result()

counts_exp = result_exp.get_counts(circ)
plot_histogram([counts_exp])

The compiled circuit image is shown below:



The result of the circuit is shown below:



From the graph, we can see that the probability of $|111\rangle$ is the biggest among all the result, which shows that the circuit to simulate Toffoli gate is successful to some extent. This is because the circuit is very likely to turn the input state of $|110\rangle$ into the output state $|111\rangle$, which satisfies the function of Toffoli gate. However, the result is quite different from the result in (a), as the error is much more than that in (a). The main reason is that the error rate of IBM 5-qubit machine is much lower than that of IBM 14-qubit machine. (We can get it from the screenshot in (a) and (c).)

4.

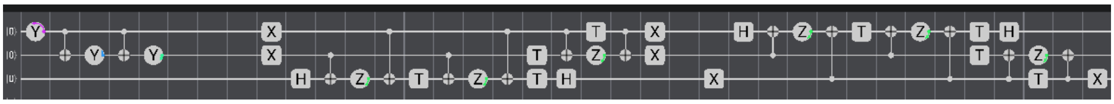
(a)

To get the target state, the circuit should be first a rotation gate on first qubit, and the result is $a|000\rangle + b|111\rangle$. Then, the controlled rotation gate on the second qubit should operate and the result is $a|000\rangle + bc|100\rangle + bd|110\rangle$. In order to get the target state, the third qubit should flip if the other two qubits are $|0\rangle$ and $|0\rangle$ respectively, so that the $|000\rangle$ state can become $|001\rangle$. Furthermore, the first qubit should also flip if the other two qubits are $|1\rangle$ and $|0\rangle$ respectively. So the result should finally become $a|001\rangle + bc|100\rangle + bd|010\rangle$. To get the $|\mathrm{K}\rangle$ state, we need to set $a = \sqrt{\frac{1}{3}}, b = \sqrt{\frac{2}{3}}, c = \sqrt{\frac{11}{14}}, d = -\sqrt{\frac{3}{14}}$.

The first rotation on first qubit is to change $|0\rangle$ into $\sqrt{\frac{1}{3}}|0\rangle + \sqrt{\frac{2}{3}}|1\rangle$. As $\begin{bmatrix} \sqrt{\frac{1}{3}} & -\sqrt{\frac{2}{3}} \\ \sqrt{\frac{2}{3}} & \sqrt{\frac{1}{3}} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{1}{3}} \\ \sqrt{\frac{2}{3}} \end{bmatrix}$, the rotation is around the Y-axis and the global phase $\theta_g = 0$, rotation angle $\theta_R = 0.608\pi$.

The second rotation on the second qubit is to change $|0\rangle$ into $\sqrt{\frac{11}{14}}|0\rangle - \sqrt{\frac{3}{14}}|1\rangle$. As $\begin{bmatrix} \sqrt{\frac{11}{14}} & \sqrt{\frac{3}{14}} \\ -\sqrt{\frac{3}{14}} & \sqrt{\frac{11}{14}} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -\sqrt{\frac{3}{14}} \\ \sqrt{\frac{11}{14}} \end{bmatrix}$, the rotation is around the Y-axis and the global phase $\theta_g = 0$, rotation angle $\theta_R = -0.306\pi$.

In this problem, the controlled rotation gate and the flip operation can be represented by single qubit gates and CNOTs. The circuit is shown below:

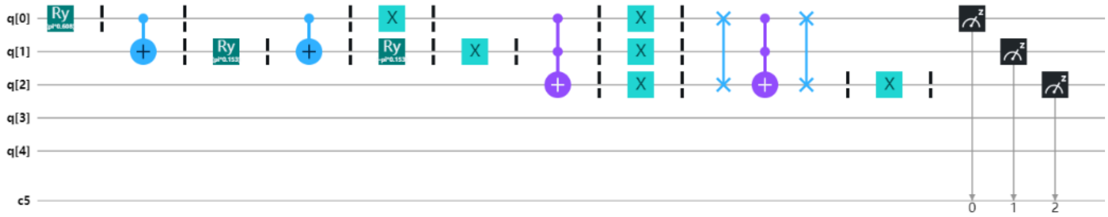


The circuit link is:

https://qui.science.unimelb.edu.au/circuits/5d9c5c9118b21a00ad5f0510

(b)

In this part, I have altered the circuit to try to reduce the error. In (a), I have mentioned that I have used some single qubit gates and CNOTs to replace the flip operations. In this circuit, these flip operations can be replaced by X-gates and Toffoli gates. The circuit that will be run on the IBM-Q device is shown below:

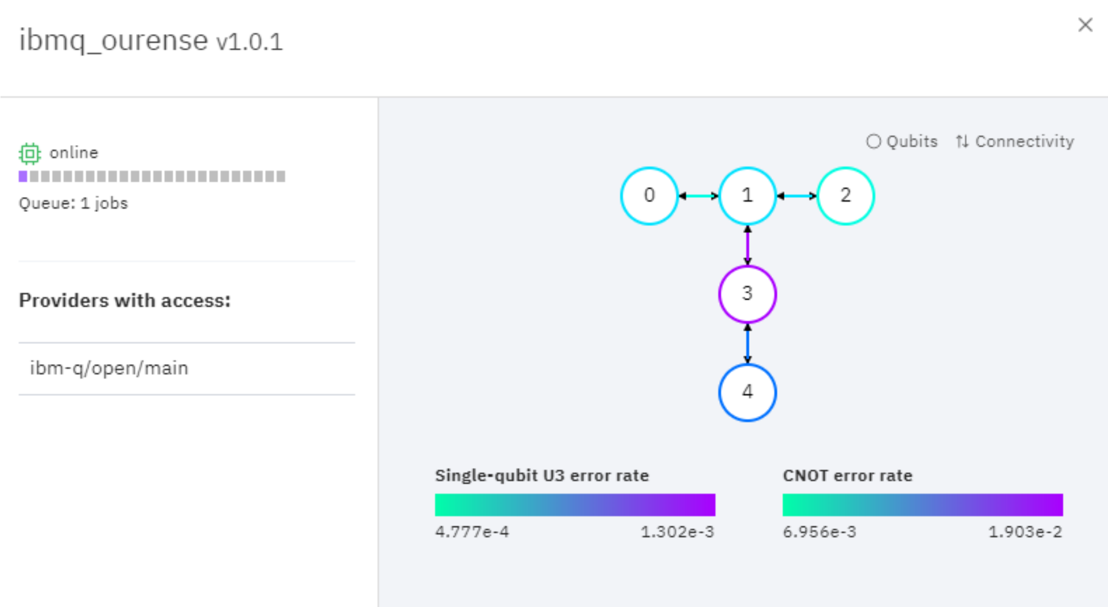


The text code of this circuit:

```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[5];
creg c[5];

ry(pi*0.608) q[0];
barrier q[0];
cx q[0],q[1];
barrier q[0];
barrier q[1];
ry(pi*0.153) q[1];
barrier q[1];
cx q[0],q[1];
barrier q[0];
barrier q[1];
x q[0];
ry(-pi*0.153) q[1];
barrier q[0];
barrier q[1];
x q[1];
barrier q[1];
ccx q[0],q[1],q[2];
```

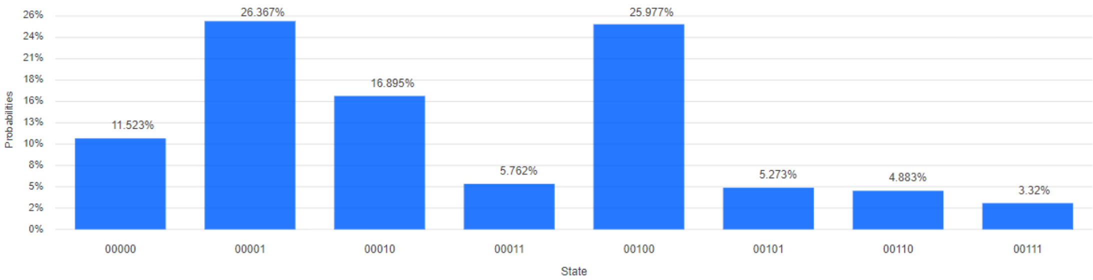
```
barrier q[0];
barrier q[1];
barrier q[2];
x q[0];
x q[1];
x q[2];
barrier q[0];
barrier q[1];
barrier q[2];
swap q[0],q[2];
ccx q[0],q[1],q[2];
swap q[0],q[2];
barrier q[2];
x q[2];
barrier q[2];
measure q[0] -> c[0];
measure q[1] -> c[1];
measure q[2] -> c[2];
```

I have run it on the ibmq_ourense device. As qubit 0, 1 and 2 has the smallest error rates, I have used these three qubits. (see the below screenshot)



(c)

The measurement result is:



Note that in this circuit I have measured q[0] as c[0], q[1] as c[1] and q[2] as c[2], so the result states are inverse of the states in the QUI. Thus, $|00001\rangle$ state is actually $|100\rangle$ state in QUI, $|00010\rangle$ state is actually $|010\rangle$ state in QUI, and $|00100\rangle$ state is actually $|001\rangle$ state in QUI.

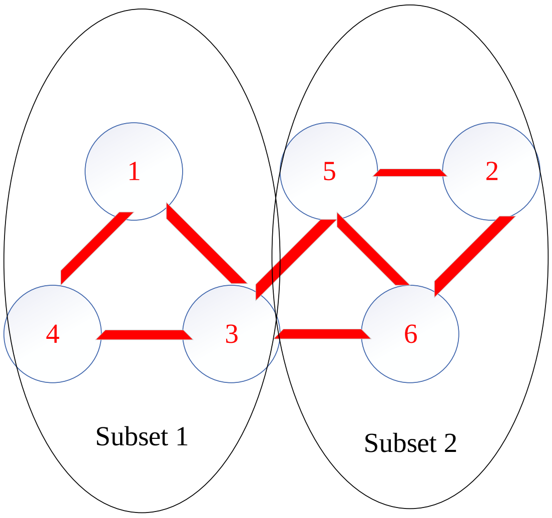
From the result I can see that the probability of the states $|001\rangle$, $|010\rangle$ and $|100\rangle$ are the biggest among all the states. In addition, the probability of state $|010\rangle$ is the smallest among the three states, which is the same as what we have seen in the QUI. The reason why other states occur in the result is that there exists some errors in the real machine, and the probability of other states are below 12%, which is acceptable to some extent.

5.

(a)

The partition solution is $\{1,3,4\}$ and $\{2,5,6\}$. The quantum version is $|101100\rangle$, $|010011\rangle$.

The possible graph partition solutions are shown below:



(b)

$H_A = (\sum_i Z_i)^2 = 2\sum_{i<j} Z_i Z_j + cI;$

$H_B = \sum_{i,j\in E} \frac{I - Z_i Z_j}{2}.$

$$H = AH_A + BH_B = H_A + H_B$$
$$= \left(\sum_i Z_i\right)^2$$
$$+ \sum_{i,j\in E} \frac{I - Z_i Z_j}{2}$$
$$= \frac{3}{2}(Z_1Z_3 + Z_1Z_4 + Z_2Z_5 + Z_2Z_6 + Z_3Z_4 + Z_3Z_5 + Z_3Z_6 + Z_5Z_6)$$
$$+ 2(Z_1Z_2 + Z_1Z_5 + Z_1Z_6 + Z_2Z_3 + Z_2Z_4 + Z_4Z_5 + Z_4Z_6)$$
$$+ constant.$$

(c)

Basis State, i	$E[H_A]$	$E[H_B]$	Basis State Energy
000000	36	0	36
000001	16	3	19
000010	16	3	19
000011	4	4	8
000100	16	2	18
000101	4	5	9
000110	4	5	9
000111	0	6	6
001000	16	4	20

001001	4	5	9
001010	4	5	9
001011	0	4	4
001100	4	4	8
001101	0	5	5
001110	0	5	5
001111	4	4	8
010000	16	2	18
010001	4	3	7
010010	4	3	7
010011	**0**	**2**	**2**
010100	4	4	8
010101	0	5	5
010110	0	5	5
010111	4	4	8
011000	4	6	10
011001	0	5	5
011010	0	5	5
011011	4	2	6
011100	0	6	6
011101	4	5	9
011110	4	5	9
011111	16	2	18
100000	16	2	18
100001	4	5	9
100010	4	5	9
100011	0	6	6
100100	4	2	6
100101	0	5	5
100110	0	5	5
100111	4	6	10
101000	4	4	8
101001	0	5	5
101010	0	5	5
101011	4	4	8
101100	**0**	**2**	**2**
101101	4	3	7
101110	4	3	7
101111	16	2	18
110000	4	4	8
110001	0	5	5
110010	0	5	5
110011	4	4	8

110100	0	4	4
110101	4	5	9
110110	4	5	9
110111	16	4	20
111000	0	6	6
111001	4	5	9
111010	4	5	9
111011	16	2	18
111100	4	4	8
111101	16	3	19
111110	16	3	19
111111	36	0	36

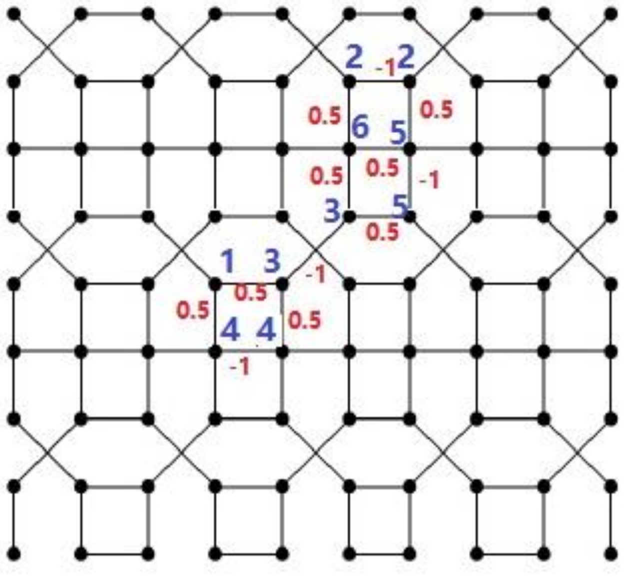
(d)

The minor embedding details are:

First, assign 0.5 to the edges that exist in the graph;

Then, assign -1 to the edges that connect two qubits that represent the same qubit in the graph.

The qubits and couplings are labelled below:



$H(s) = (1 - s)H_x + sH_p, 0 \leq s \leq 1$

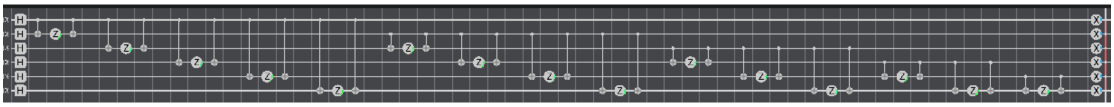
$s = \frac{t}{T}$

When we use AQC to solve our problem, we will start at the known ground state of a simple Hamiltonian. We will slowly change the simple Hamiltonian to our problem Hamiltonian. If we move slowly enough, the system will remain in the ground state.

Then we will find the ground state of the problem Hamiltonian so that we can solve the original problem.

(e)

The circuit is shown below:

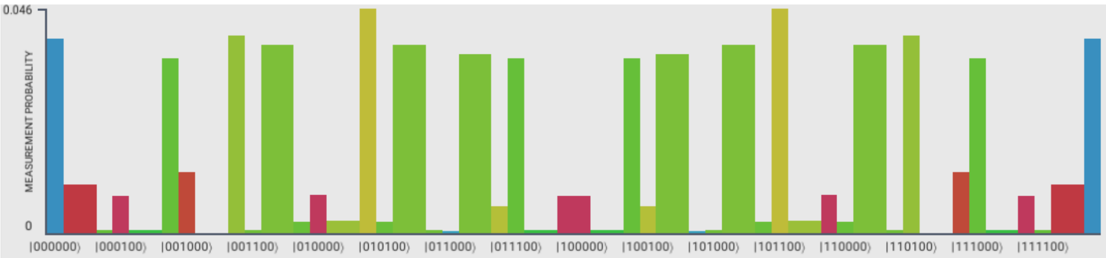


The circuit link is:

https://qui.science.unimelb.edu.au/circuits/5da19bee3b2158007fdd5192

(f)

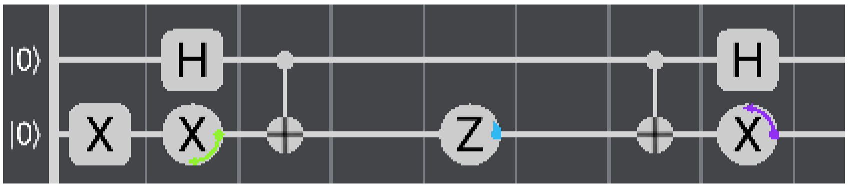
In the circuit in (e), I have set α=-0.2π and β=0.125π, so that the states |101100⟩ and |010011⟩ can achieve the highest probability among all states. The result can be shown below:



6.

(a)

In this circuit, I first construct the initial state of $|01\rangle$ by applying X-gate on the second qubit. Then, I use the theories of constructing a Z-Z coupling to construct a X-Y coupling. As $X = HZH$ and $Y = \left(\sqrt{X}\right)^{+} Z\sqrt{X}$, I have applied an H gate on the first qubit before the circuit and another H gate on the first qubit after the circuit. Similarly, I have applied a $\left(\sqrt{X}\right)^{+}$ gate on the second qubit before the circuit and a $\sqrt{X}$ gate on the second qubit after the circuit. So, the $|\varphi(\theta)\rangle$ state can be constructed. The circuit is shown below:



The circuit link is:

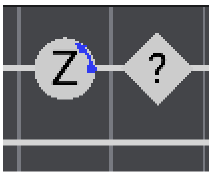
https://qui.science.unimelb.edu.au/circuits/5daab16a5ec7440097ed2b98

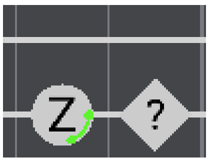
(b)

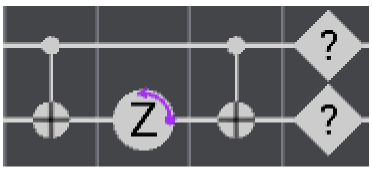
If we want to measure in X-basis, we will first measure in Z-basis and then apply an H gate to that qubit. Having done that, we will finally get the measurement in X-basis.

If we want to measure in Y-basis, we will first measure in Z-basis and then apply a $(\sqrt{X})^{+}$ gate to that qubit. Having done that, we will finally get the measurement in Y-basis.

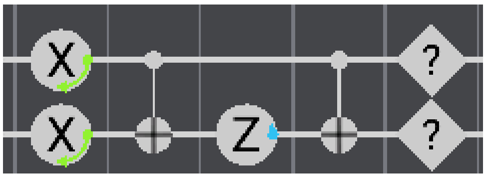
For each term in $\langle H \rangle$, I will construct the circuit element based on the Hamiltonian coupling. After that, I will measure the qubits in the given basis many times and then calculate the expectation value. If we measure $|0\rangle$, it should be the +1 value; If we measure $|1\rangle$, it should be the -1 value. Therefore, by measuring for many times, we can calculate the expectation value.

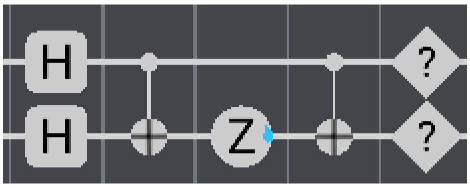
The first item is a constant. The second term in $\langle H \rangle$ is $\langle Z_1 \rangle$, and we should construct the circuit like: 

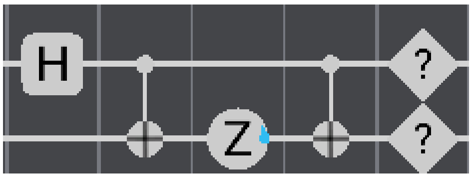
. Then we will measure the first qubit in the Z-basis. Similarly, we should construct the circuit of 

 for the third term. Then we will measure the second qubit in the Z-basis. For the fourth term, it is a ZZ coupling, so the circuit should be 

. Then we will measure the first

and the second qubit in the Z-basis. Similarly, we should construct 

for the YY coupling, and measure the first and the second qubit in the Y-basis. The circuit 

is for the XX coupling. We will then measure the first and second qubit in the X-basis. For the last term, the circuit should be 

. Then we will measure the first qubit in X-basis and the second qubit in Z-basis.

After constructing all the circuits, we need to measure in all circuits for many times to get the expectation value (the expectation value of each term in $\langle H \rangle$ is the probability multiplied by the state(which should be changed to either +1 or -1 in the calculation process)), so that we can calculate $\langle H \rangle$ for a given value of θ.

(c)

When I construct all the circuit in (b) and do the measurement many times, I can get the expectation value for each term in $\langle H \rangle$ for a given value of θ. When I try to change the value of θ, the value of $\langle H \rangle$ will also get changed. Therefore, I have changed the value of θ many times so that I can get the best value of θ that can produce the biggest value of $\langle H \rangle$.

In practice, the best value of θ is 0.9277π. On this occasion, $\langle Z_1 \rangle = -0.9743$, $\langle Z_2 \rangle = 0.9743$, $\langle Z_1 Z_2 \rangle = -1$, $\langle Y_1 Y_2 \rangle = -0.2252$, $\langle X_1 X_2 \rangle = -0.2252$, $\langle X_1 Z_2 \rangle = 0$.

So, $\langle H \rangle = -0.8517$ and it is the minimized value.