MULT90063 Assignment 2

Geng Liu 1035248

1 Q1 Quantum Volume

1.1 (a)

QUI link: **https://qui.science.unimelb.edu.au/circuits/60a011698453bc0033fd7892**

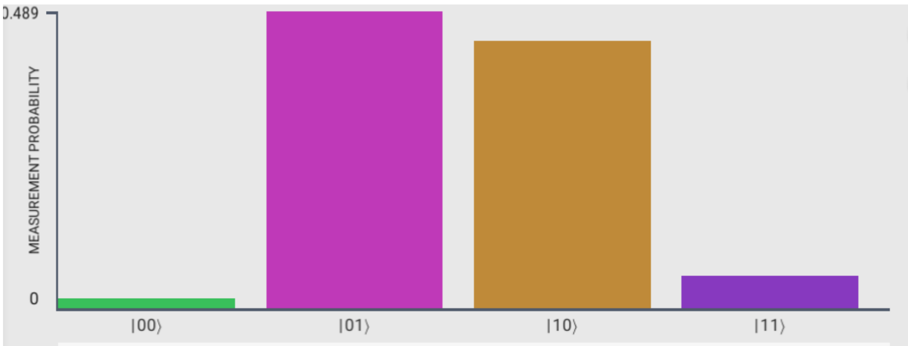


Figure 1: Output of circuit

The output is shown in figure 1. The states with the two largest probabilities are $|01\rangle$ and $|10\rangle$, with probability 0.489 and 0.441 respectively.

1.2 (b)

The probability of these four states from left to right are 0.016, 0.489, 0.441, 0.053 respectively. Total combined probability of measuring either of these states in Q1(a) is

$$h_2 = 0.489 + 0.441 = 0.930$$

1.3 (c)

Implementation on IBM quantum device (ibmq_16_melbourne) is shown below.

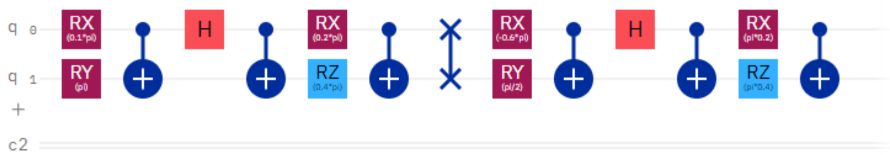


Figure 2: IBM implementation

We simulated 1000 shots and output is shown in figure 3. The probability of measure $|00\rangle$ is 0.133, $|10\rangle$ is 0.35, $|01\rangle$ is 0.413, $|11\rangle$ is 0.104.

(**Note**:Since the IBM has different read order of the states campared to QUI, all the result are transfered to the same order as QUI, and we did the same thing in all the following questions too.)

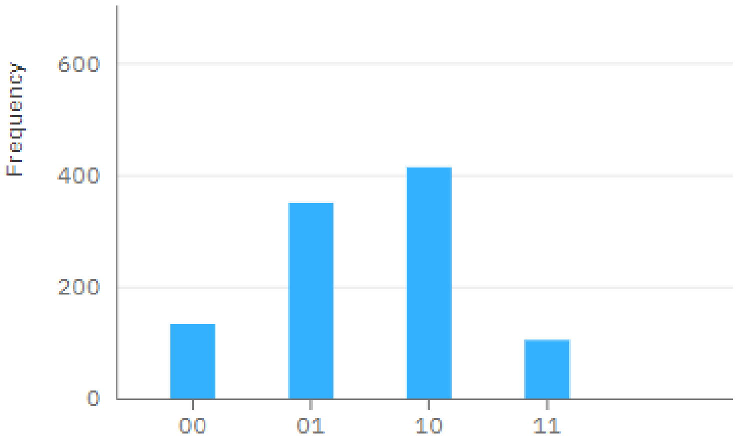


Figure 3: IBM Measurement output

1.4 (d)

Using the probability in (c), We know the median probability is (0.133+0.35)/2 = 0.2415. So the total combined probability of heavy outputs is

$$h_2 = 0.35 + 0.413 = 0.763 > \frac{2}{3}$$

So the h_2 is greater than $\frac{2}{3}$.

2 Q2 Toffoli gate on IBM Quantum device

2.1 (a)

The error rate mainly influenced by three factors: different devices, different architechtures and different gate numbers.

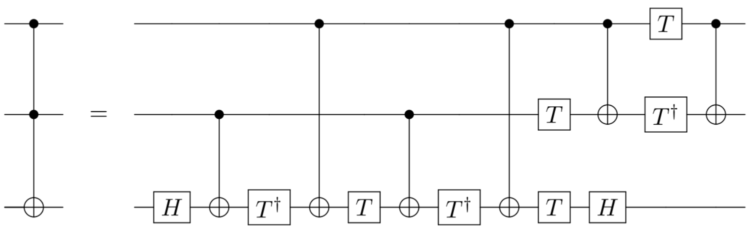


Figure 4: Toffoli gate decomposition

Because Toffoli gate will be transpiled into some basic gates in quantum computer, the target qubit should have most connections between the other two qubits so that the total number of gates could be reduced. Therefore, we made an comparasion of setting different qubit as target. The comparison result is shown in **Appendix A**. The result indicates that using the middle qubit as target can effectively reduce the total number of gates.

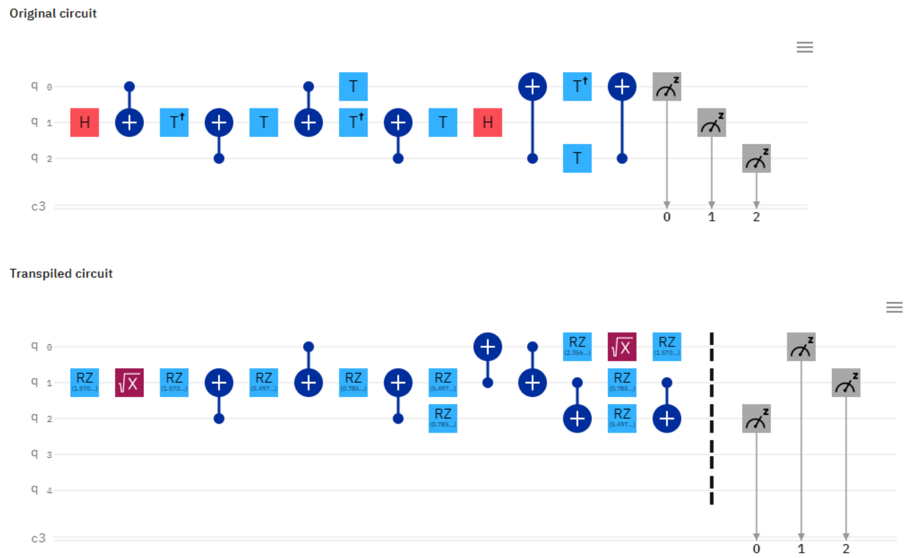


Figure 5: Toffoli gate decomposition

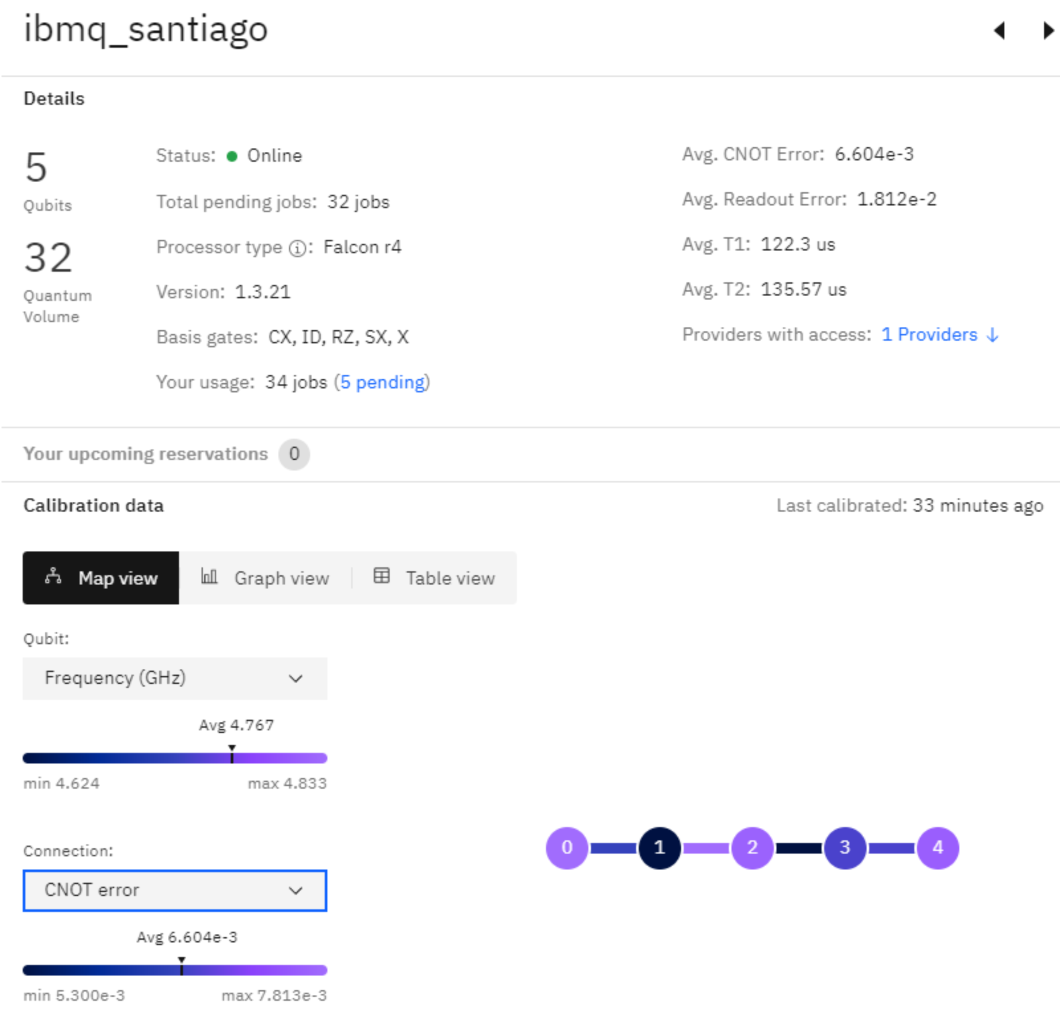


Figure 6: ibmp-santiago Information on May 21st

```python
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
from numpy import pi

qreg_q = QuantumRegister(3, 'q')
creg_c = ClassicalRegister(3, 'c')
circuit = QuantumCircuit(qreg_q, creg_c)

circuit.h(qreg_q[1])
circuit.cx(qreg_q[0], qreg_q[1])
circuit.tdg(qreg_q[1])
circuit.cx(qreg_q[2], qreg_q[1])
circuit.t(qreg_q[1])
circuit.cx(qreg_q[0], qreg_q[1])
circuit.t(qreg_q[0])
circuit.tdg(qreg_q[1])
circuit.cx(qreg_q[2], qreg_q[1])
circuit.t(qreg_q[1])
circuit.h(qreg_q[1])
circuit.cx(qreg_q[2], qreg_q[0])
circuit.tdg(qreg_q[0])
circuit.t(qreg_q[2])
circuit.cx(qreg_q[2], qreg_q[0])
circuit.measure(qreg_q[0], creg_c[0])
circuit.measure(qreg_q[1], creg_c[1])
circuit.measure(qreg_q[2], creg_c[2])
```

Figure 7: Original circuit code

```python
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
from numpy import pi

qreg_q = QuantumRegister(5, 'q')
creg_c = ClassicalRegister(3, 'c')
circuit = QuantumCircuit(qreg_q, creg_c)

circuit.rz(1.5707963267948966, qreg_q[1])
circuit.sx(qreg_q[1])
circuit.rz(1.5707963267948966, qreg_q[1])
circuit.cx(qreg_q[2], qreg_q[1])
circuit.rz(5.497787143782137, qreg_q[1])
circuit.cx(qreg_q[0], qreg_q[1])
circuit.rz(0.7853981633974487, qreg_q[1])
circuit.cx(qreg_q[2], qreg_q[1])
circuit.rz(5.497787143782137, qreg_q[1])
circuit.cx(qreg_q[1], qreg_q[0])
circuit.cx(qreg_q[0], qreg_q[1])
circuit.rz(2.356194490192345, qreg_q[0])
circuit.sx(qreg_q[0])
circuit.rz(1.5707963267948966, qreg_q[0])
circuit.rz(0.7853981633974487, qreg_q[2])
circuit.cx(qreg_q[1], qreg_q[2])
circuit.rz(0.7853981633974487, qreg_q[1])
circuit.rz(5.497787143782137, qreg_q[2])
circuit.cx(qreg_q[1], qreg_q[2])
circuit.barrier(qreg_q[1], qreg_q[0], qreg_q[2], qreg_q[3], qreg_q[4])
circuit.measure(qreg_q[2], creg_c[0])
circuit.measure(qreg_q[0], creg_c[1])
circuit.measure(qreg_q[1], creg_c[2])
```

Figure 8: Transpiled circuit code

Then based on many experiments, we decided to implement our circuit on **ibmp-santiago**, the result indicates that this device has lowest error rate running this circuit since it has lowest CNOT and SX gate error rate on that day. The detailed information of this device on May 21st is shown in figure 6. The **circuit image** is shown in figure 5. The **Qiskit code** of both original circuit and transpiled circuit are shown in figure 7 and figure 8

The **summary of the results** for each input is show in table 1.

Input	Fidelity Rate(%)
000	90.3
001	89.7
010	90.5
011	86.6
100	88.8
101	87.0
110	85.3
111	85.6

Table 1: Result of Different Input on ibmq-santiago

2.2 (b)

QUI link:**https://qui.science.unimelb.edu.au/circuits/60b06cbdf7500200548dfe3f**

The distribution of the measurement outcome for state $|110\rangle$ in IBM device is shown in figure 9 and table 2

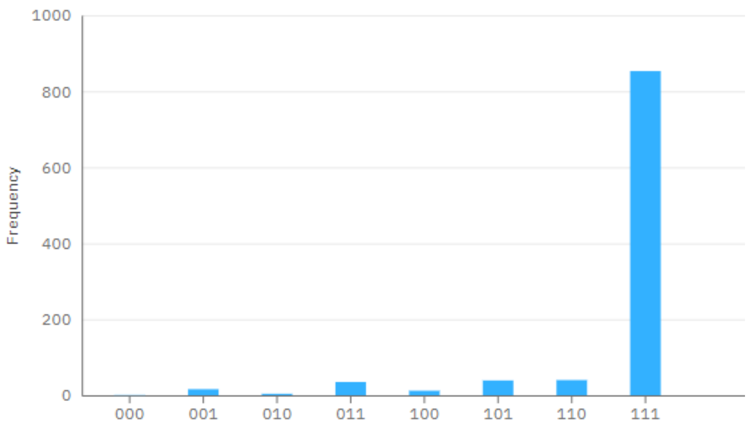


Figure 9: Distribution of Input **110**

State	Frequency	Probability(%)
000	1	0.1
001	16	1.6
010	4	0.4
011	35	3.5
100	12	1.2
101	39	3.9
110	40	4.0
111	853	85.3

Table 2: Distribution of Measurement outcome of Input 110 on IBM device

Then we implemented our toffoli gate circuit with input state $|110\rangle$ into QUI. The circuit is shown in figure 10, and the measurement result of the error model is shown in figure 11 and table 3.

After many experiments, we chose the error model with Rotation standard Deviation as 0.05π, and the Rotation Mean as 0.08. We can see the error model closely matched the distribution of probabilities we get for the $|110\rangle$ state in a).

(**Note:** Since we used middle qubit as target qubit, we need to switch the position of second qubit and third qubit of the outputs in order to get the same read order as normal. For example, the state 001 shown in the QUI measurement output is actually the state $|010\rangle$)

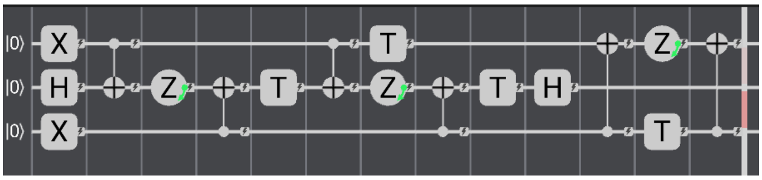


Figure 10: Distribution of Input **110**

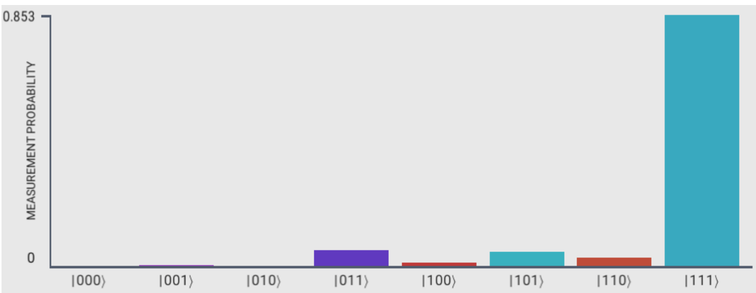


Figure 11: Distribution of Input **110**

State	Probability(%)
000	0.1
001	0
010	0.3
011	5.5
100	1.2
101	2.8
110	4.8
111	85.3

Table 3: Probability of different state in error model

2.3 (c)

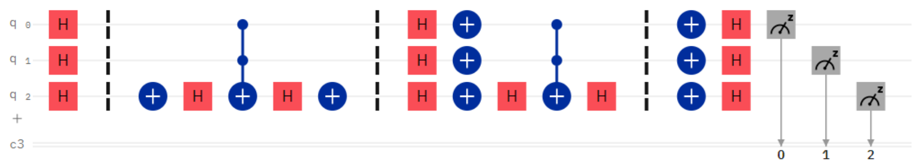


Figure 12: Grover search circuit

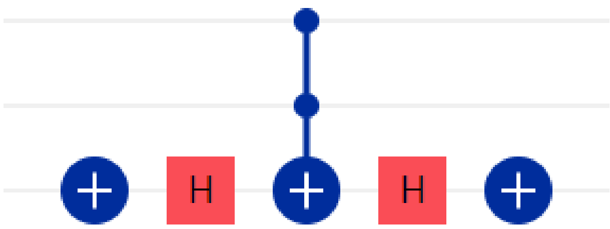


Figure 13: Oracle circuit

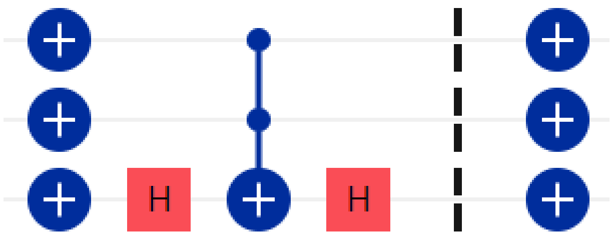


Figure 14: Inversion circuit

```
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
from numpy import pi

qreg_q = QuantumRegister(3, 'q')
creg_c = ClassicalRegister(3, 'c')
circuit = QuantumCircuit(qreg_q, creg_c)

circuit.h(qreg_q[0])
circuit.h(qreg_q[1])
circuit.h(qreg_q[2])
circuit.barrier(qreg_q[0], qreg_q[1], qreg_q[2])
circuit.x(qreg_q[2])
circuit.h(qreg_q[2])
circuit.ccx(qreg_q[0], qreg_q[1], qreg_q[2])
circuit.h(qreg_q[2])
circuit.x(qreg_q[2])
circuit.barrier(qreg_q[0], qreg_q[1], qreg_q[2])
circuit.h(qreg_q[0])
circuit.h(qreg_q[1])
circuit.h(qreg_q[2])
circuit.x(qreg_q[0])
circuit.x(qreg_q[1])
circuit.x(qreg_q[2])
circuit.h(qreg_q[2])
circuit.ccx(qreg_q[0], qreg_q[1], qreg_q[2])
circuit.h(qreg_q[2])
circuit.barrier(qreg_q[2], qreg_q[1], qreg_q[0])
circuit.x(qreg_q[0])
circuit.x(qreg_q[1])
circuit.x(qreg_q[2])
circuit.h(qreg_q[0])
circuit.h(qreg_q[1])
circuit.h(qreg_q[2])
circuit.measure(qreg_q[0], creg_c[0])
circuit.measure(qreg_q[1], creg_c[1])
circuit.measure(qreg_q[2], creg_c[2])
```

Figure 15: Circuit code

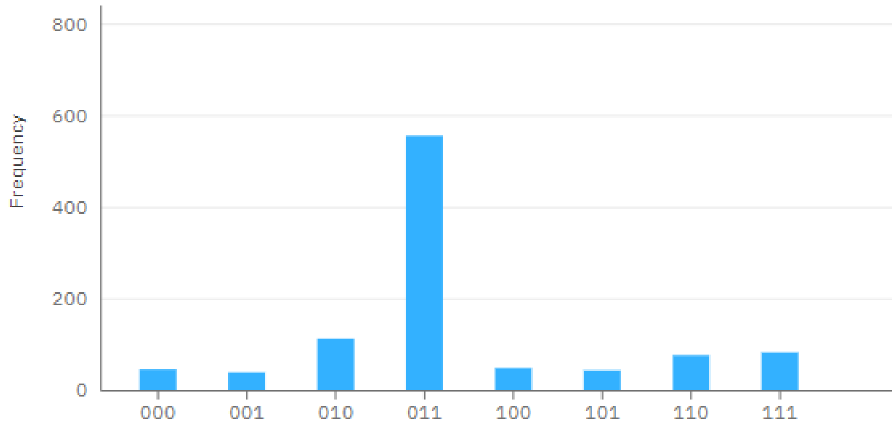


Figure 16: Result on ibmq_santiago

The circuit is shown in figure 12, and the oracle part is shown in figure 13, the inversion part is shown in 14.

(**Note:** Since the transpiled circuit of our Toffoli gate in (a) is totally same as the transpiled circuit of 3-qubit Toffoli gate, we will use the 3-qubit toffoli gate to show our works here)

Then the screshot of code is shown in figure 15, We also put a code in text at last of this question. The final result is shown in figure 16, and the detail is shown in table 4

State	**Frequency**
000	45
001	48
010	112
011	76
100	39
101	43
110	555
111	82

Table 4: Distribution of Measurement outcome

From the QUI, we know that ideal probability of this circuit to find $|110\rangle$ is 78.125%, and the rest states are all 3.125%. After 1000 shots experiments, we found the actual probability of finding $|110\rangle$ is 55.5%. Even though the actual probability is smaller than the ideal probability, we can still claim that the circuit amplify the state correctly. Because without this search algorithm, the superposition input should get a equally probability for all states which is 12.5%. After one iteration of grovers search, the state $|110\rangle$ we want to find has a probability over 50%. And the other states has decreased its probability of being searched, all of them has a probability lower than 12.5%.

```
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
from numpy import pi

qreg_q = QuantumRegister(3, 'q')
creg_c = ClassicalRegister(3, 'c')
circuit = QuantumCircuit(qreg_q, creg_c)

circuit.h(qreg_q[0])
```

```
circuit.h(qreg_q[1])
circuit.h(qreg_q[2])
circuit.barrier(qreg_q[0], qreg_q[1], qreg_q[2])
circuit.x(qreg_q[2])
circuit.h(qreg_q[2])
circuit.ccx(qreg_q[0], qreg_q[1], qreg_q[2])
circuit.h(qreg_q[2])
circuit.x(qreg_q[2])
circuit.barrier(qreg_q[0], qreg_q[1], qreg_q[2])
circuit.h(qreg_q[0])
circuit.h(qreg_q[1])
circuit.h(qreg_q[2])
circuit.x(qreg_q[0])
circuit.x(qreg_q[1])
circuit.x(qreg_q[2])
circuit.h(qreg_q[2])
circuit.ccx(qreg_q[0], qreg_q[1], qreg_q[2])
circuit.h(qreg_q[2])
circuit.barrier(qreg_q[2], qreg_q[1], qreg_q[0])
circuit.x(qreg_q[0])
circuit.x(qreg_q[1])
circuit.x(qreg_q[2])
circuit.h(qreg_q[0])
circuit.h(qreg_q[1])
circuit.h(qreg_q[2])
circuit.measure(qreg_q[0], creg_c[0])
circuit.measure(qreg_q[1], creg_c[1])
circuit.measure(qreg_q[2], creg_c[2])
```

3 Q3 Three qubit K-state

3.1 (a)

QUI link: **https://qui.science.unimelb.edu.au/circuits/60a94163f0bd640096537bd3**

The circuit is shown in figure 17 and the output is shown in 18. The key idea to create this circuit is to use Y gate to generate states with different probability, and then use CNOT and RZ gate to modify the phase and state.

The probability of the state $|001\rangle$, $|010\rangle$, $|100\rangle$ is 0.5, $\frac{1}{6}$, $\frac{1}{3}$. We can see that the K can be written as

$$|K\rangle = \frac{\sqrt{2}}{\sqrt{6}}|100\rangle - \frac{e^{i3\pi/4}}{\sqrt{6}}|010\rangle + \frac{\sqrt{3}}{\sqrt{6}}|001\rangle \tag{1}$$

To generate this, we firstly need to apply a RY gate to the third qubit and we can get

$$\frac{1}{\sqrt{2}}|001\rangle + \frac{1}{\sqrt{2}}|000\rangle$$

Then, we need to preserve a state with probability $\frac{1}{2}$ and want to generate states with probability of $\frac{1}{3}$ and $\frac{1}{6}$. To achieve this, we need to use RY gate to generate states and use control on 1 to keep another half unchanged. The angle of RY gate is 1.919. Then we can get

$$\frac{\sqrt{2}}{\sqrt{6}}|011\rangle + \frac{1}{\sqrt{6}}|001\rangle + \frac{\sqrt{3}}{\sqrt{6}}|000\rangle$$

The form we get is quite similar to the K-state we want. Then we used two CNOT gate to flip the first and second qubit into the form we want.

$$\frac{\sqrt{2}}{\sqrt{6}}|101\rangle + \frac{1}{\sqrt{6}}|011\rangle + \frac{\sqrt{3}}{\sqrt{6}}|000\rangle$$

Then, we can apply X gate to flip the third qubit.

$$\frac{\sqrt{2}}{\sqrt{6}}|100\rangle + \frac{1}{\sqrt{6}}|010\rangle + \frac{\sqrt{3}}{\sqrt{6}}|001\rangle$$

At last, we can get the K-state by changing the phase of second qubit using RZ gate. The RZ gate here is actually the $T^{\dagger}$ gate.

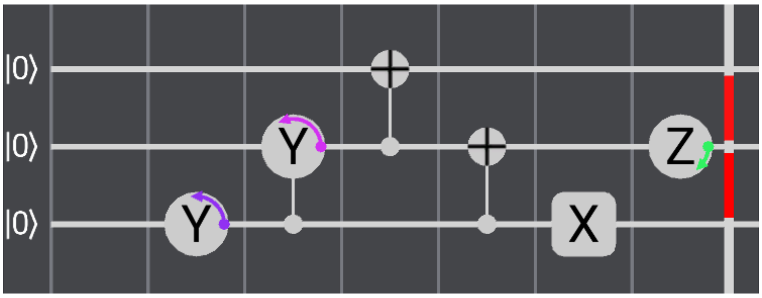


Figure 17: Circuit on QUI

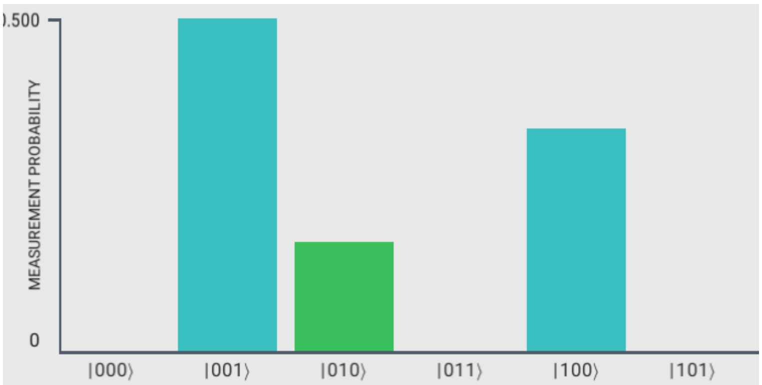


Figure 18: Output of circuit

3.2 (b)

As mentioned before, different devices is also an important factors that influence the fidelity. We experimenttd the circuit in different devices and the results are shown below:

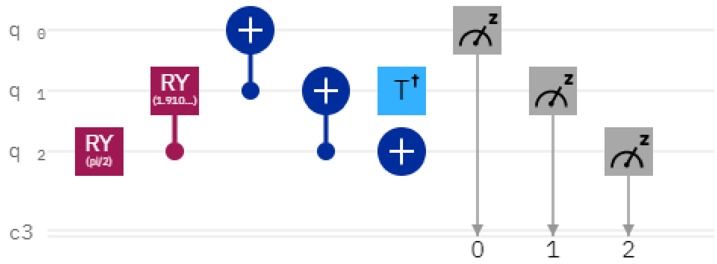


Figure 19: Circuit on IBM device

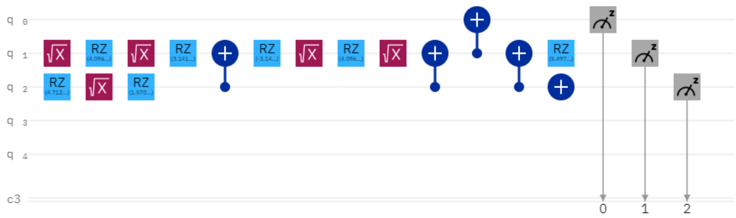
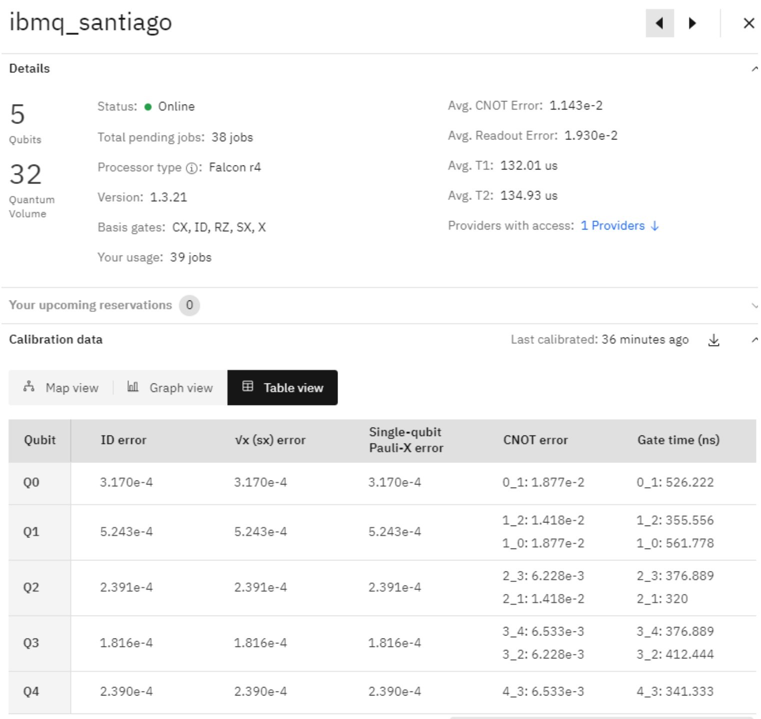


Figure 20: Transpiled circuit

Device	$\|100\rangle$	$\|010\rangle$	$\|001\rangle$	**Error**
manila	316	160	440	74
santiago	331	142	457	**70**
athens	293	151	515	71
belem	286	183	429	134
quito	319	208	363	202
melbourne	241	144	361	254
lima	361	166	394	135

From the results, we decided to further experiment on the **santiago**, which has lowest error rate.


ibmq_santiago

Details

5 Qubits

32 Quantum Volume

Status: Online

Total pending jobs: 38 jobs

Processor type ⓘ: Falcon r4

Version: 1.3.21

Basis gates: CX, ID, RZ, SX, X

Your usage: 39 jobs

Avg. CNOT Error: 1.143e-2

Avg. Readout Error: 1.930e-2

Avg. T1: 132.01 us

Avg. T2: 134.93 us

Providers with access: 1 Providers

Your upcoming reservations 0

Calibration data

Last calibrated: 36 minutes ago

Map view Graph view Table view

Qubit	ID error	√x (sx) error	Single-qubit Pauli-X error	CNOT error	Gate time (ns)
Q0	3.170e-4	3.170e-4	3.170e-4	0_1: 1.877e-2	0_1: 526.222
Q1	5.243e-4	5.243e-4	5.243e-4	1_2: 1.418e-2 1_0: 1.877e-2	1_2: 355.556 1_0: 561.778
Q2	2.391e-4	2.391e-4	2.391e-4	2_3: 6.228e-3 2_1: 1.418e-2	2_3: 376.889 2_1: 320
Q3	1.816e-4	1.816e-4	1.816e-4	3_4: 6.533e-3 3_2: 6.228e-3	3_4: 376.889 3_2: 412.444
Q4	2.390e-4	2.390e-4	2.390e-4	4_3: 6.533e-3	4_3: 341.333

Figure 21: Santiago detail on May 24th

We campared the CNOT error rate, SX error rate and X error rate of santiago. We found that the error rate on Q3 is much lower than it on Q1, Q0. Since our transpiled circuit has many CNOT and SX gates, we decided to implement our circuit on Q2, Q3, Q4. The results indicates that it has significant improvement on fidelity. The result is shown below:

Device	$\|100\rangle$	$\|010\rangle$	$\|001\rangle$	**Error**
santiago	332	167	451	50

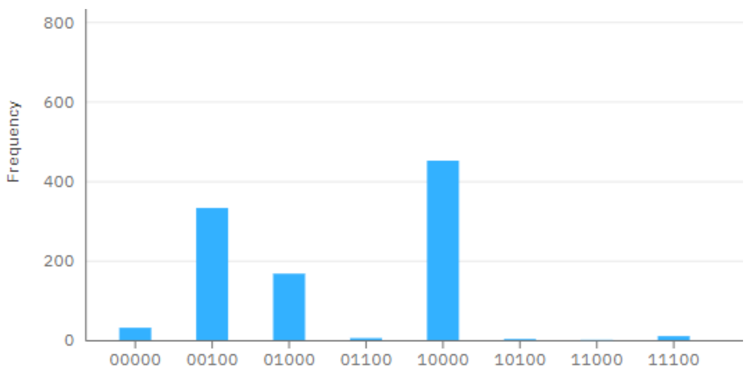


Figure 22: Circuit on Q2,Q3,Q4

To summarize, as shown in above, our initial circuit is already well optimised and very simple, so there is no need to further reduce the number of gates. And considering the different qubits and devices, we implement our circuit on Q2,Q3,Q4 on ibmq-santiago device and get the highest fidelity.

3.3 (c)

The measured results are shown in Figure 22 and Tabel 5. The error rate is

$$|33.2\% - 33.3\%| + |16.7\% - 16.7\%| + |45.1\% - 50.0\%| = 5\%$$

The error rate is pretty low since the circuit has been optimized well and was implemented on the qubit and devices with lowest error rate. And we noticed that the frequency of 000 is relatively higher than the other error outcome. One reason for this may be that all the distance between 000 and the three expected outcomes (001,010,100) are 1. Therefore, once an error occured, the final outcome will be more likely to be measured as 000.

Measurement Outcome	Frequency	Probability(%)
000	31	3.1
001	451	45.1
010	167	16.7
011	1	0.1
100	332	33.2
101	3	0.3
110	5	0.5
111	10	1.0

Table 5: Distribution of Measurement outcome

3.4 (d)

The implementation circuit is shown in figuire 23 and the measurement output is shown in figure 24. The probability of measuring the state $|000\rangle$ is 0.302. And the ideally probability is 0.320. The proof is shown in figuire 26. We also implemented the circuit on QUI and the result also proved that the probability is 0.320 which is shown in figure 25.

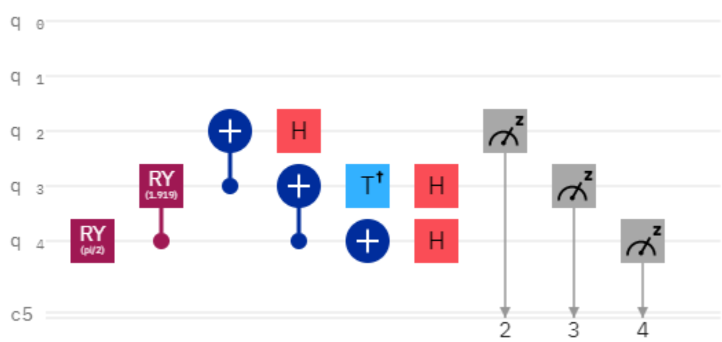


Figure 23: Circuit

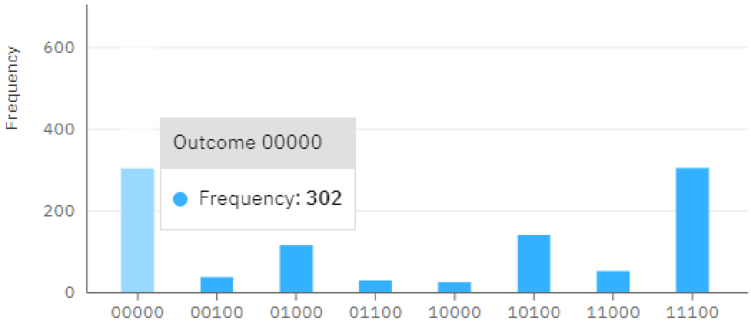


Figure 24: Measurement output

Figure 25: QUI output

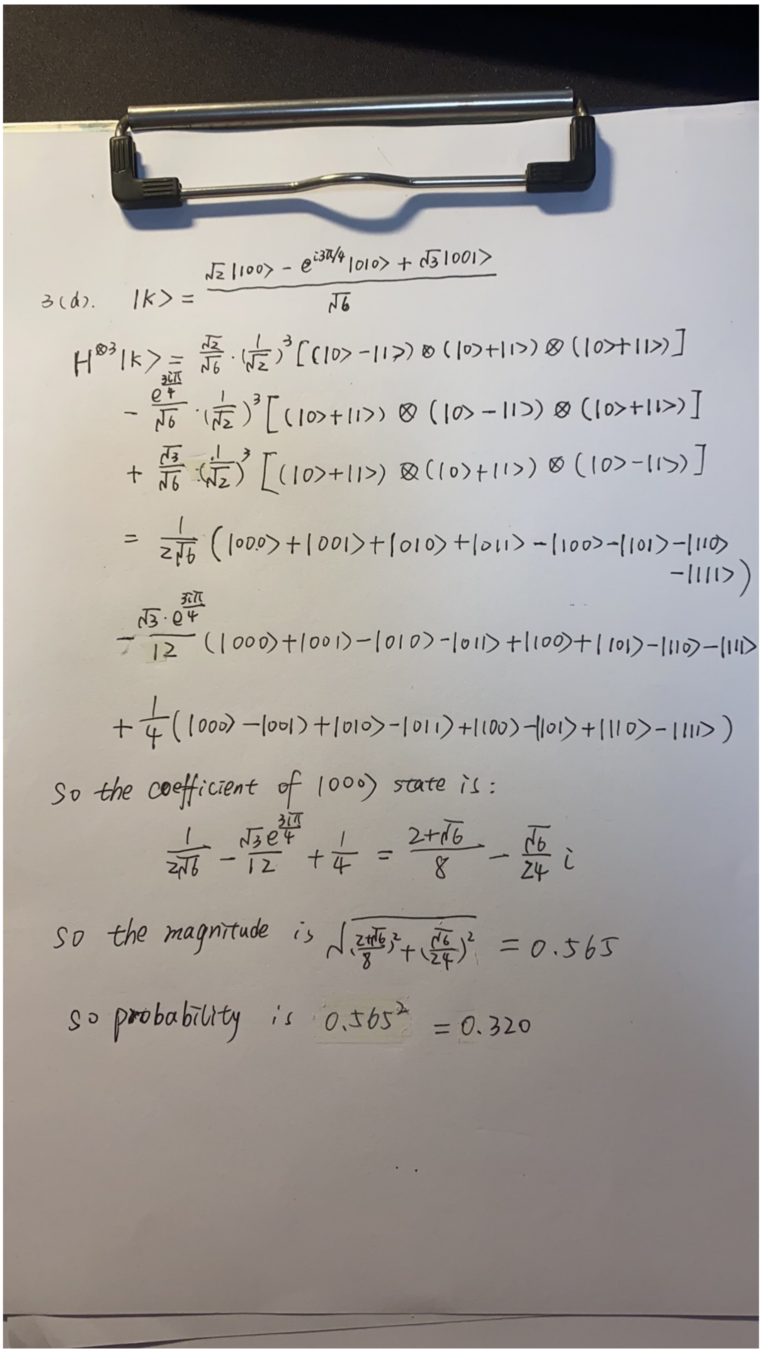


Figure 26: Idealy probability calculation

4 Q4 Variarional Quantum Eigensolver

4.1 (a)

The QUI link is here : **https://qui.science.unimelb.edu.au/circuits/60acbedbeb73ad00284250a1**
The circuit is shown in figure 27, the rotation angle of RZ gate in circuit is 2θ.

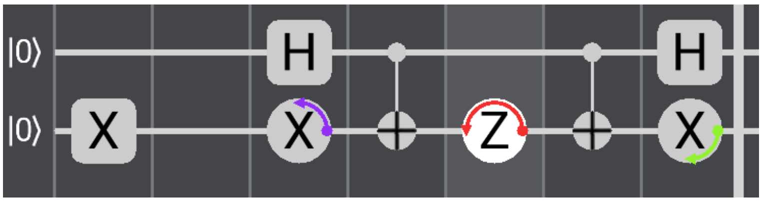


Figure 27: circuit of $|\phi(\theta)\rangle$

Proof:
We know that

$$|\phi(\theta)\rangle = (\cos\theta I - i\sin\theta X_1 Y_2)|01\rangle$$

So we let

$$U = \cos\theta I - i\sin\theta X_1 Y_2$$

Then we will convert U into zz coupling. According to the fact that $X = HZH$ and $Y = (\sqrt{X})^\dagger Z\sqrt{X}$. We can get:

$$\begin{aligned} U &= \cos\theta I - i\sin\theta X_1 Y_2 \\ &= \exp[-i\theta \cdot (X_1 \otimes Y_2)] \\ &= \exp[-i\theta \cdot (H_1 Z_1 H_1) \otimes ((\sqrt{X_2})^\dagger Z_2 \sqrt{X_2})] \\ &= \exp[-i\theta \cdot (H_1 \otimes (\sqrt{X_2})^\dagger)(Z_1 \otimes Z_2)(H_1 \otimes \sqrt{X_2})] \end{aligned} \tag{2}$$

Since H_1 is real matrix, we can get that

$$H_1 \otimes (\sqrt{X_2})^\dagger = H_1^\dagger \otimes (\sqrt{X_2})^\dagger = (H_1 \otimes \sqrt{X_2})^\dagger = (H_1 \otimes \sqrt{X_2})^{-1}$$

Suppose $P = H_1 \otimes (\sqrt{X_2})^\dagger$, we can get that

$$U = \exp[-i\theta P(Z_1 Z_2)P^{-1}] = \exp[P(-i\theta Z_1 Z_2)P^{-1}]$$

Because $Z_1 Z_2$ is diagonal matrix, so we can get

$$U = P\exp[-i\theta Z_1 Z_2]P^{-1}$$

4.2 (b)

We can divide the $\langle H\rangle$ into four parts in order to measure them simply. We assume

$$\begin{aligned} \langle H_1\rangle &= c_0 + c_1\langle Z_1\rangle + c_2\langle Z_2\rangle + c_3\langle Z_1 Z_2\rangle \\ \langle H_2\rangle &= c_4\langle Y_1 Y_2\rangle \\ \langle H_3\rangle &= c_5\langle X_1 X_2\rangle \\ \langle H_4\rangle &= c_6\langle X_1 Y_2\rangle \end{aligned} \tag{3}$$

So

$$\langle H\rangle = \langle H_1\rangle + \langle H_2\rangle + \langle H_3\rangle + \langle H_4\rangle$$

For a given value of θ, to measure each part of the $\langle H\rangle$, we need to change the base of the qubit to Z base. With the help of QUI, we do not need to experiment many times. We can simply use the probabilities of each state and calculate the expectations.

For H_1, since all of the components are in Z bases. We can simply measure them through QUI using circuit in 4(a). Then according to the page 38 in lecture 20, we knwon that we need to add a H gate before measure X bases and need to add $S^\dagger$, H gate before measure Y. This is shown in figure 28

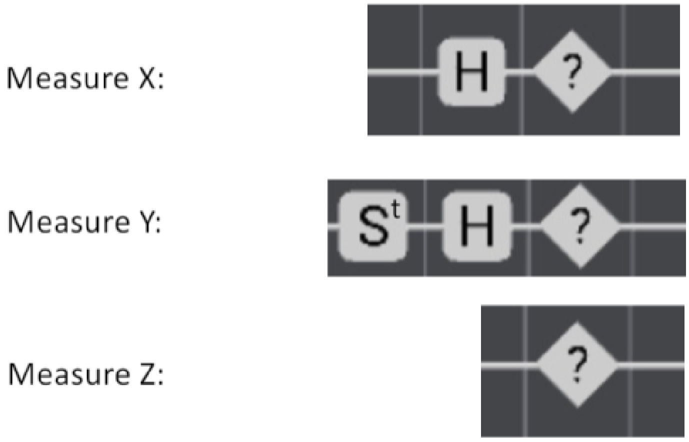


Figure 28: Measure Paulis

Then to get $\langle H_2 \rangle$, we add $S^\dagger$ and H gate at the end. The circuit is shown below:

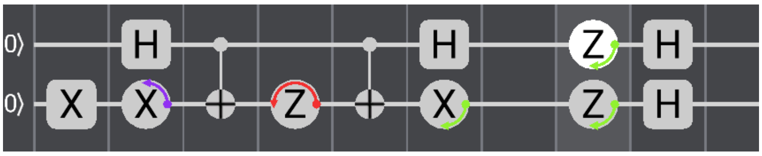


Figure 29: H_2 measurement circuit

Then to get $\langle H_3 \rangle$, we need to add H gate at the end. The circuit is shown below:

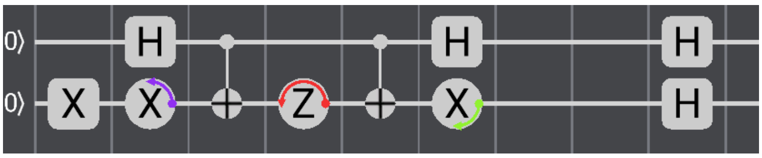


Figure 30: H_3 measurement circuit

Then to get $\langle H_4 \rangle$, we need to add $S^\dagger$ and H gate at the end of qubit 2 and add H at the qubit 1. The circuit is shown below:

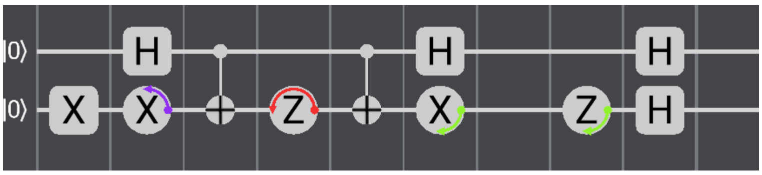


Figure 31: H_4 measurement circuit

At last we need to sum up these four parts to get the final expectation.

4.3 (c)

State	Z_1	Z_2	Z_1Z_2	Y_1Y_2	X_1X_2	X_1Y_2
00	+1	+1	+1	+1	+1	+1
01	+1	-1	-1	-1	-1	-1
10	-1	+1	-1	-1	-1	-1
11	-1	-1	+1	+1	+1	+1

According to the table, we get

$$\begin{aligned}
\langle Z_1\rangle &= P_{00}\cdot(+1) + P_{01}\cdot(+1) + P_{10}\cdot(-1) + P_{11}\cdot(-1)\\
\langle Z_2\rangle &= P_{00}\cdot(+1) + P_{01}\cdot(-1) + P_{10}\cdot(+1) + P_{11}\cdot(-1)\\
\langle Z_1Z_2\rangle &= P_{00}\cdot(+1) + P_{01}\cdot(-1) + P_{10}\cdot(-1) + P_{11}\cdot(+1)\\
\langle H_1\rangle &= c_0 + c_1\langle Z_1\rangle + c_2\langle Z_2\rangle + c_3\langle Z_1Z_2\rangle\\
&= -1.470 + 0.442\langle Z_1\rangle - 0.246\langle Z_2\rangle + 0.373\langle Z_1Z_2\rangle\\
\langle Y_1Y_2\rangle &= P_{00}\cdot(+1) + P_{01}\cdot(-1) + P_{10}\cdot(-1) + P_{11}\cdot(+1)\\
\langle H_2\rangle &= c_4\langle Y_1Y_2\rangle\\
&= 0.191\langle Y_1Y_2\rangle\\
\langle Z_1Z_2\rangle &= P_{00}\cdot(+1) + P_{01}\cdot(-1) + P_{10}\cdot(-1) + P_{11}\cdot(+1)\\
\langle H_3\rangle &= c_5\langle X_1X_2\rangle\\
&= 0.191\langle X_1X_2\rangle\\
\langle Z_1Z_2\rangle &= P_{00}\cdot(+1) + P_{01}\cdot(-1) + P_{10}\cdot(-1) + P_{11}\cdot(+1)\\
\langle H_4\rangle &= c_6\langle X_1Y_2\rangle\\
&= 0.010\langle X_1Y_2\rangle\\
\langle H\rangle &= \langle H_1\rangle + \langle H_2\rangle + \langle H_3\rangle + \langle H_4\rangle
\end{aligned} \tag{4}$$

We experimented different θ, and the result is shown below.

θ	**Energy**
0.10π	-1.510336
0.20π	-1.993384
0.25π	-2.224999
0.30π	-2.419944
0.35π	-2.556200
$\mathbf{0.40\pi}$	**-2.624896**
0.45π	-2.615632
0.50π	-2.531000
0.55π	-2.380702
0.70π	-1.692616
0.80π	-1.266056
0.90π	-1.061104

From the table, we noticed when θ is around 0.40π, the energy is quite low, so we further experiment some θ value around 0.40π.

θ	**Energy**
0.36π	-2.576848
0.37π	-2.591688
0.38π	-2.605152
0.39π	-2.615712
0.40π	-2.624896
0.41π	-2.628424
0.42π	**-2.629048**
0.43π	-2.62692
0.44π	-2.623416

So the minumum energy for the Hamiltonian is **-2.629048**, when θ is 0.42π.

A Appendix A

From figure we can see that when we set the middle qubit as target, the transpiled circuit has least gates.

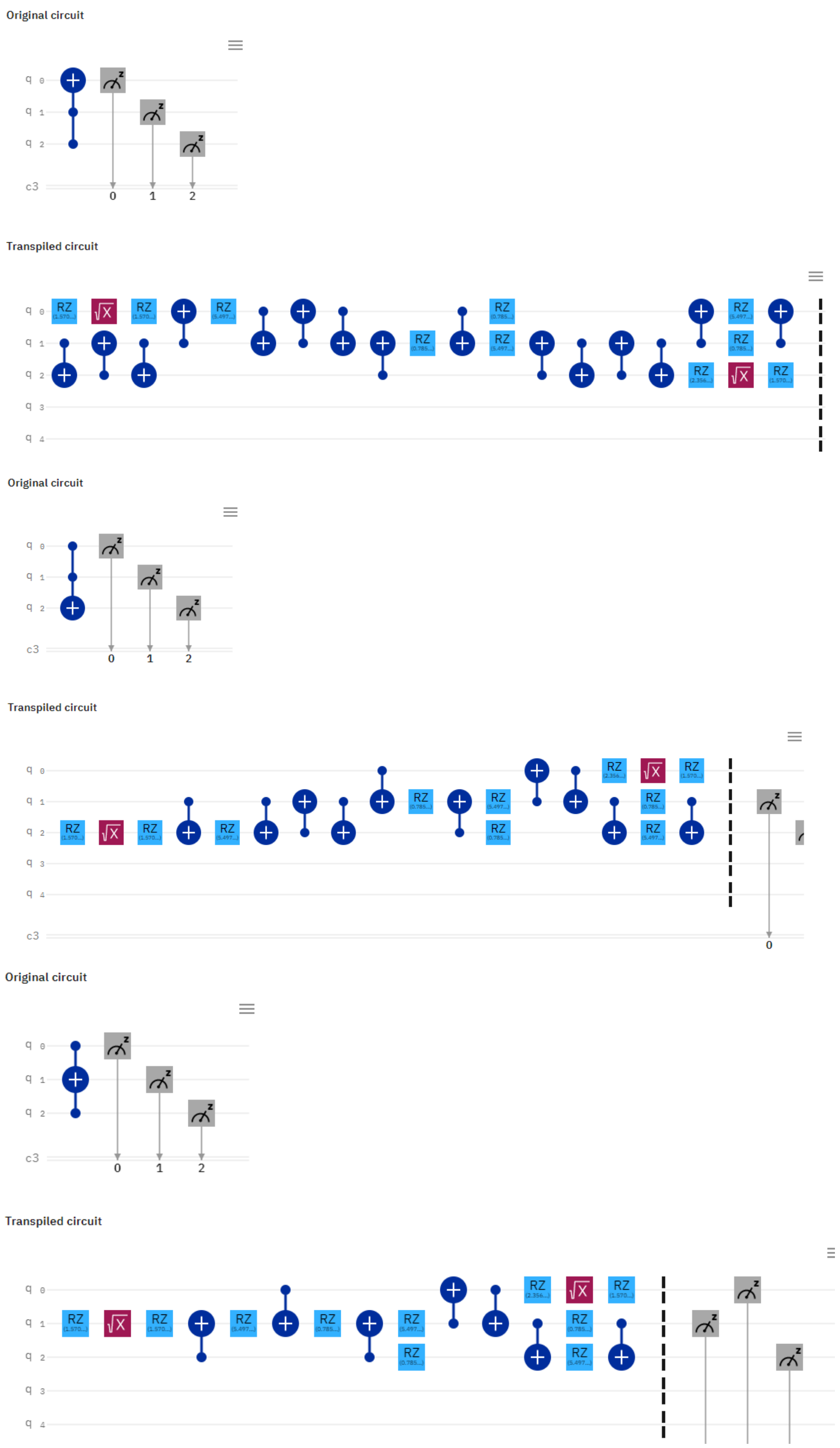


Figure 32: Comparison of different target qubit