Lab 3 Report

Quantum Computing Algorithms

Author: B08901049 Yuan-Chia Chang

Instructor: Professor Hao-Chung Cheng

TA: Chia-Yi Chou

Created: 11/3/2021(Wed)

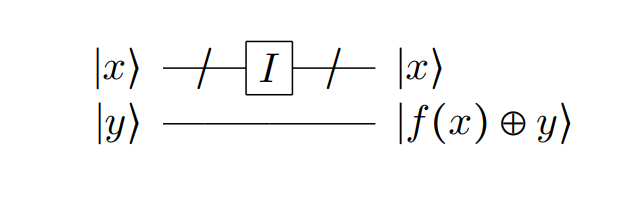
Last edited: 11/3/2021(Wed)

Contact: b08901049@ntu.edu.tw

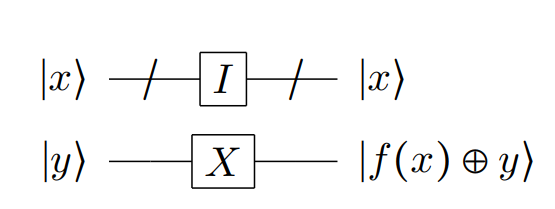
Collaborator: B08901002 Chen-Han Lin, B08901209 Yu-Hsiang Lin

Q1.

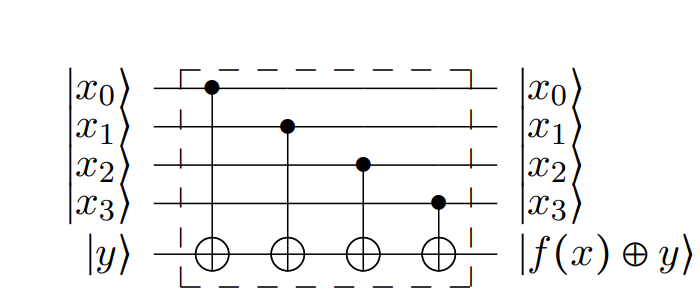
(a)



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x0 | x1 | x2 | x3 | y | result |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 |



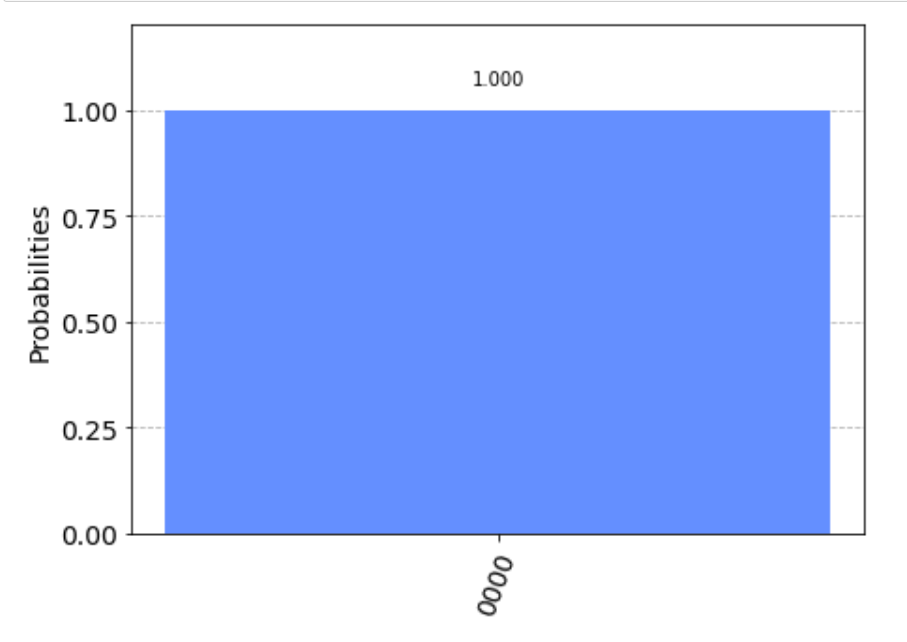
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x0 | x1 | x2 | x3 | y | result |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |



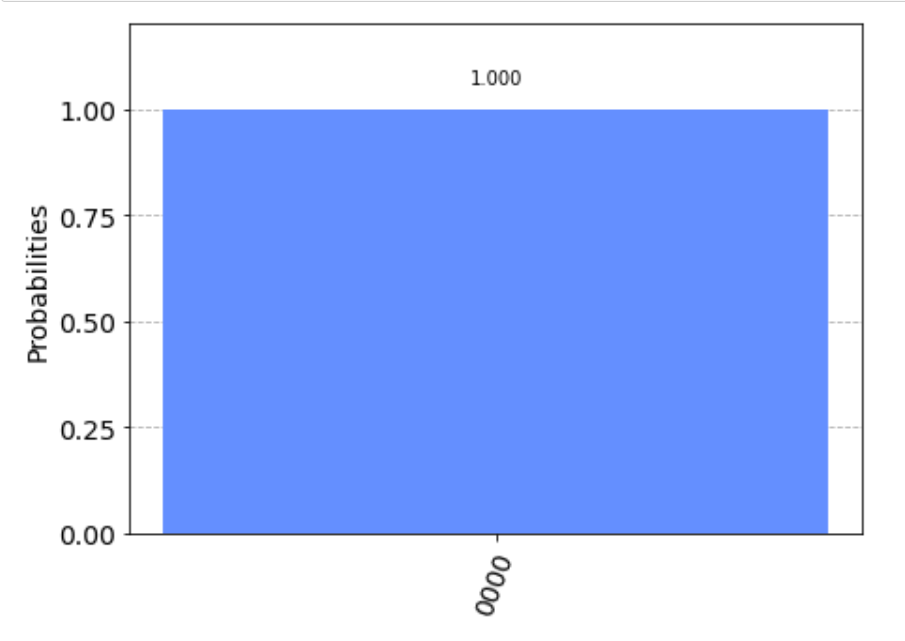
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x0 | x1 | x2 | x3 | y | result |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 |

(b)

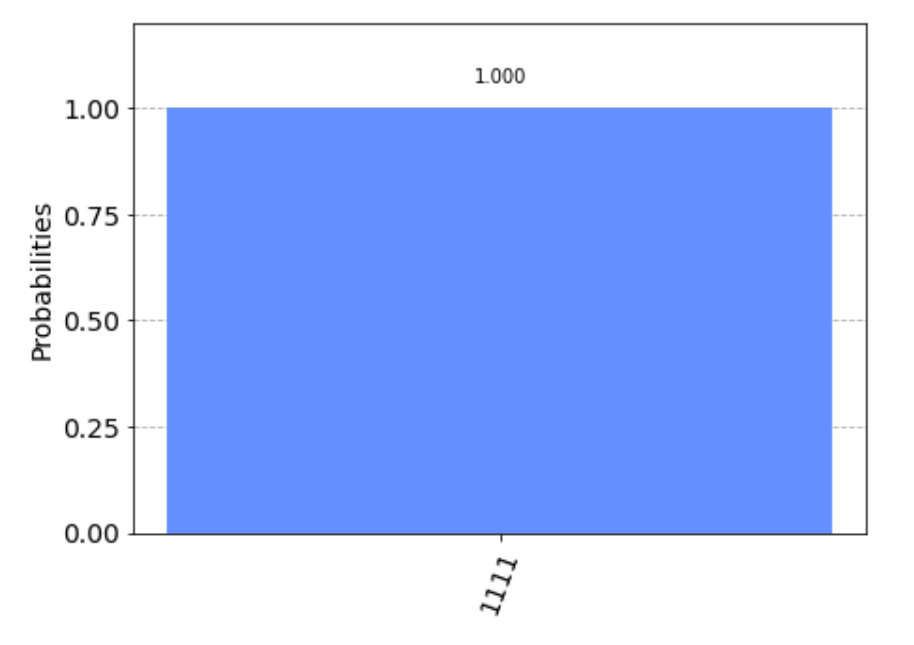
All 0



All 1



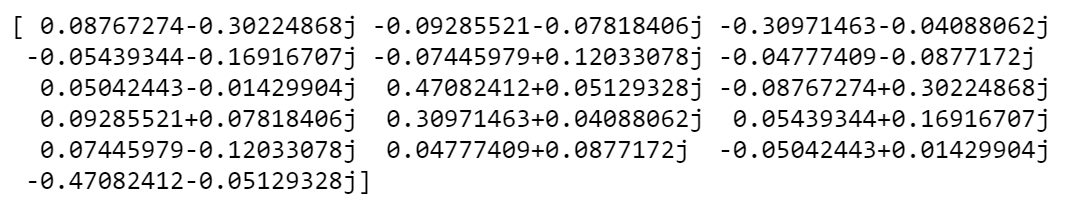
Balanced



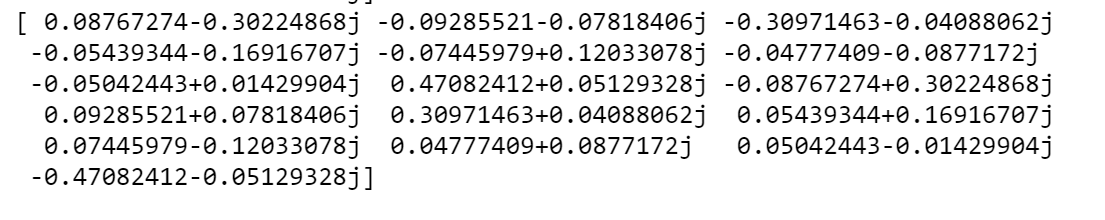
Q2

(a)

Before the gate [y,x2,x1,x0]

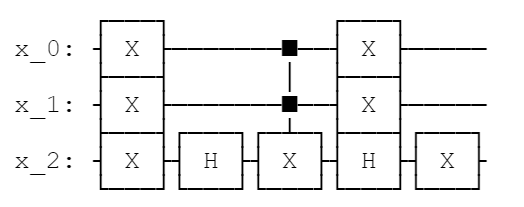


After the gate [y,x2,x1,x0]



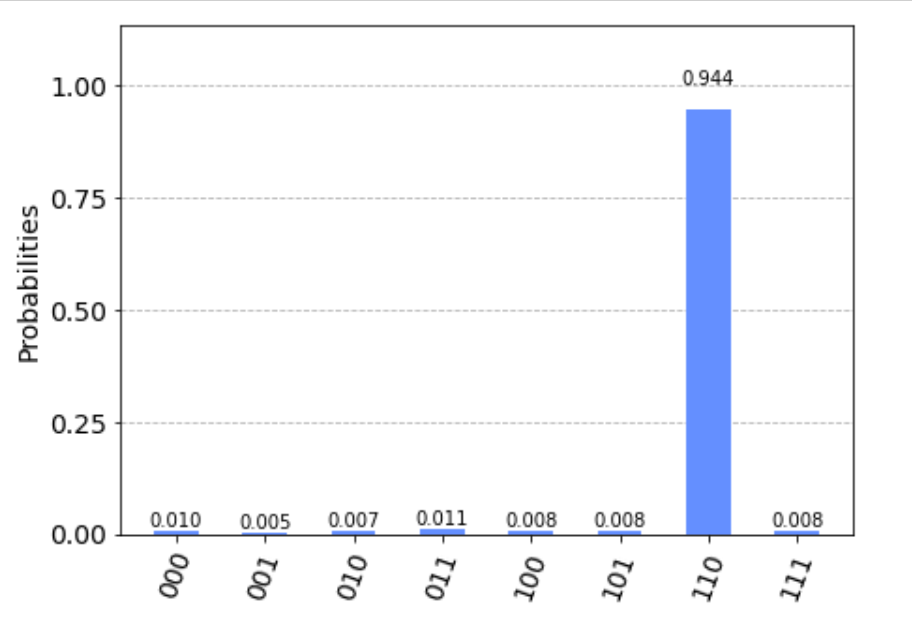
We find that number 6 and 14 number are the minus sign, which corresponds to [x2,x1,x0] = [1,1,0], which is [x0,x1,x2] = [0,1,1].

(b)



(c)

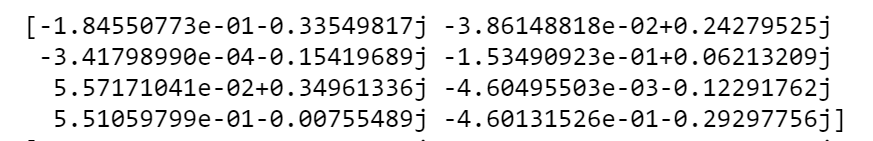
After iterations, the result is shown below



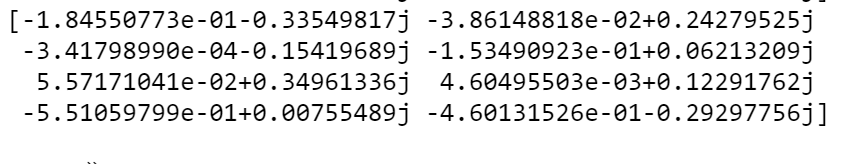
Which we get 011 with high probability, but not always.

(d)

Before the gate [x2,x1,x0]



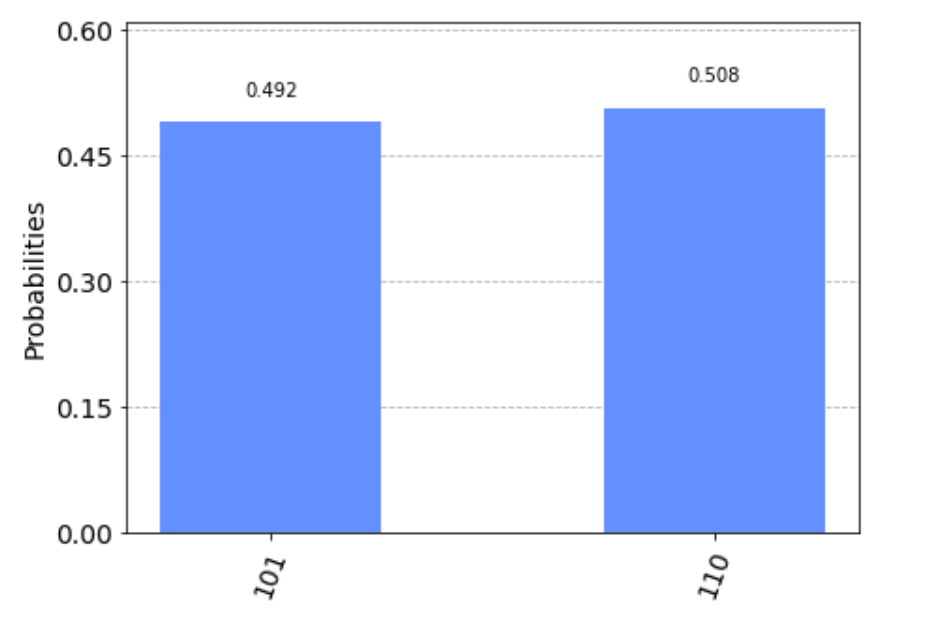
After the gate [x2,x1,x0]



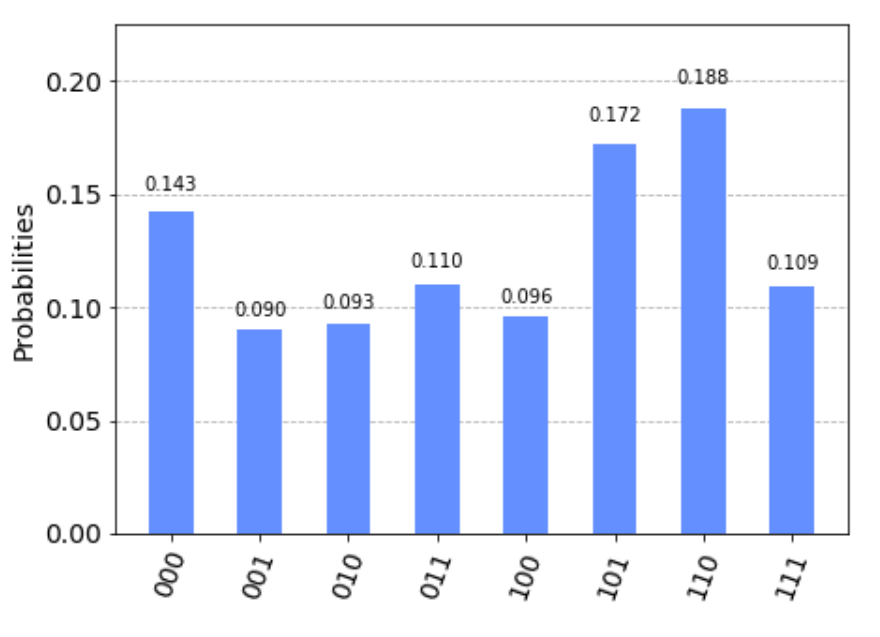
We find that number 5 and 6 number are the minus sign, which corresponds to [x2,x1,x0] = [1,0,1] and [1,1,0], which is [x0,x1,x2] = [1,0,1] and [0,1,1].

(e)

On qasm\_simulator, we need 4 iterations of queries.

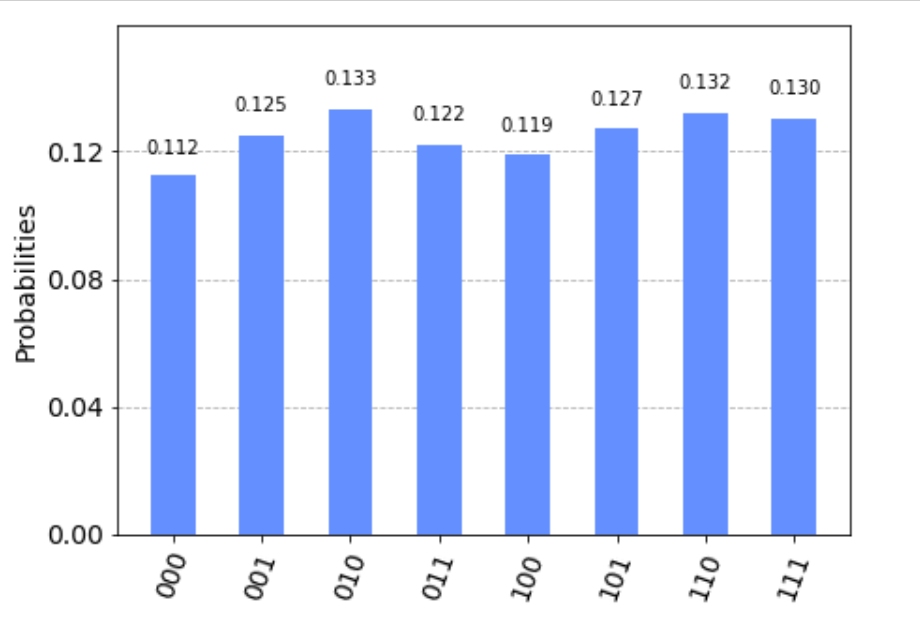


On real device,



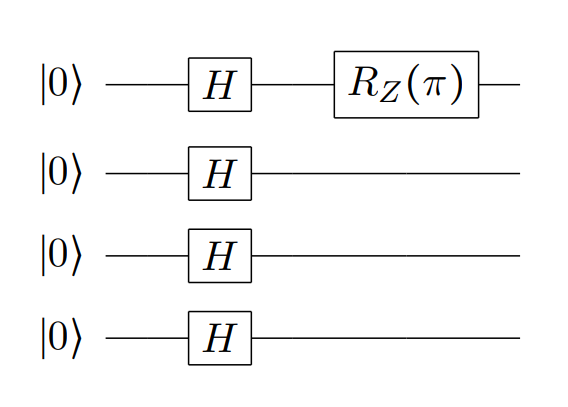
We get the highest probability on 101 and 011, but the error is really huge.

(f)

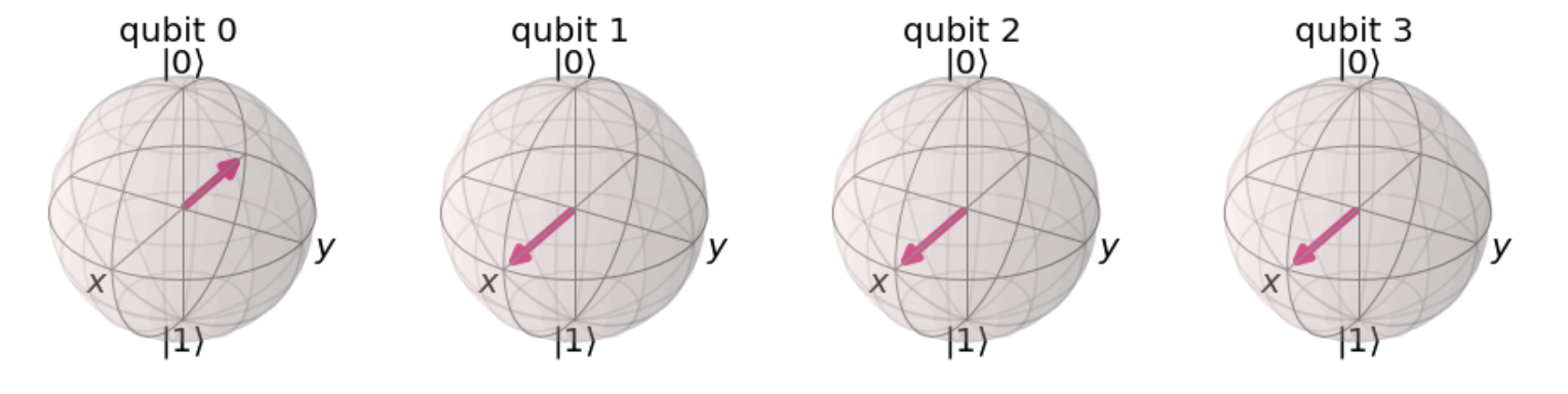


We will always get the same probability of all qubit states. The reason is we flip the half amount of qubits rather than one, which cause nothing difference. In other words, Grover search cannot search too much targets.

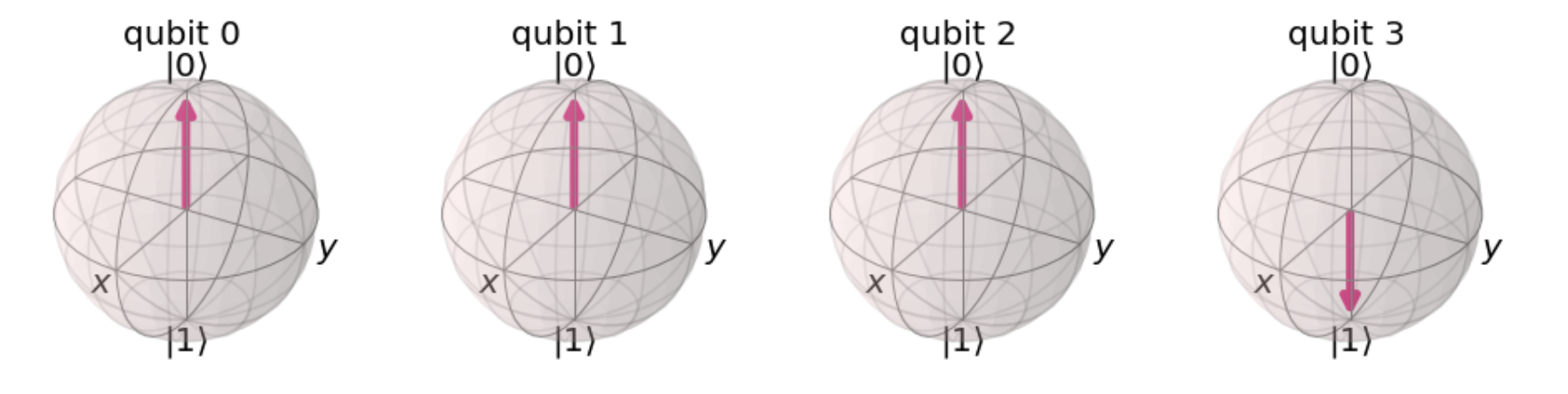
Q3.



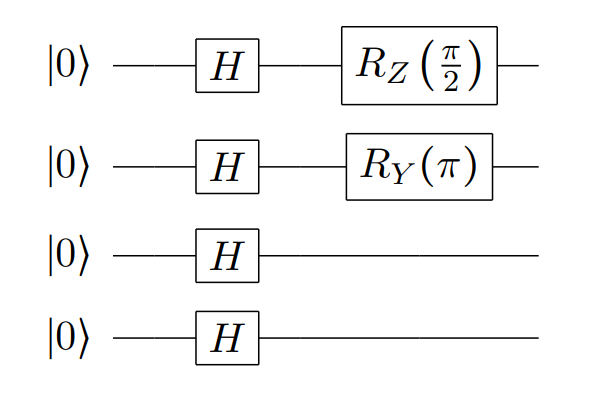
State in Fourier Basis



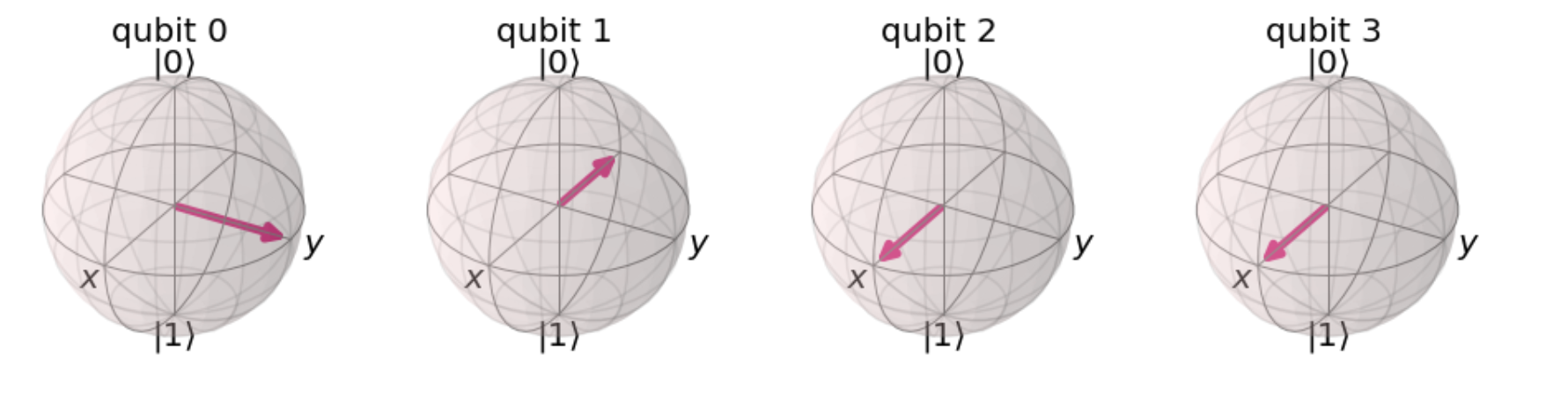
State in Computational Basis



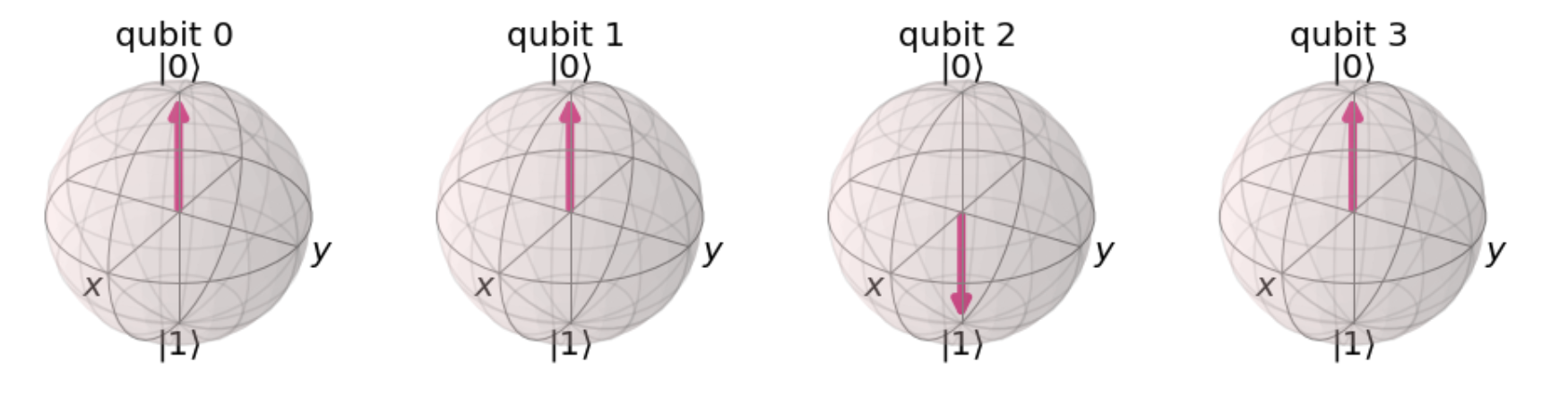
For the Fourier basis, we find the qubit 0 rotate to -x, also the qubit 0 rotate each flip in computational basis. Hence, there are 8 flips in computational basis. The representation <q3,q2,q1,q0> will be 1000. Hence <q0,q1,q2,q3> will be 0001.



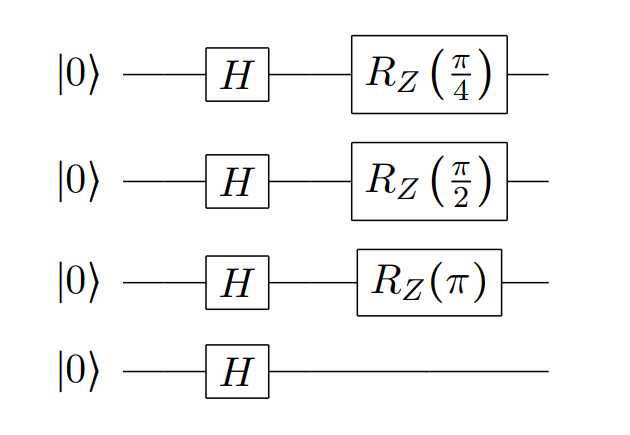
State in Fourier Basis



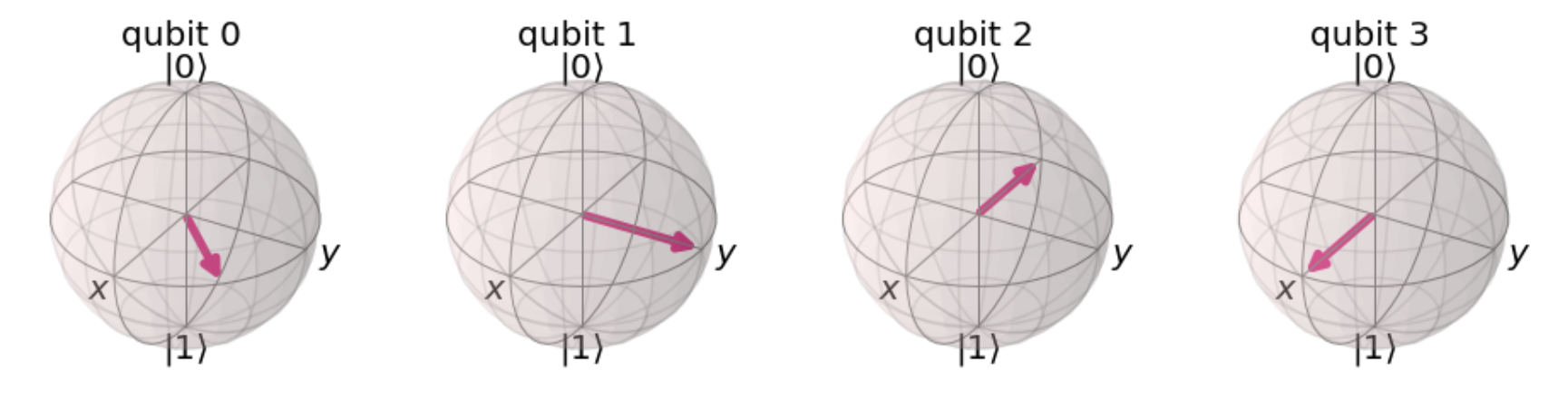
State in Computational Basis



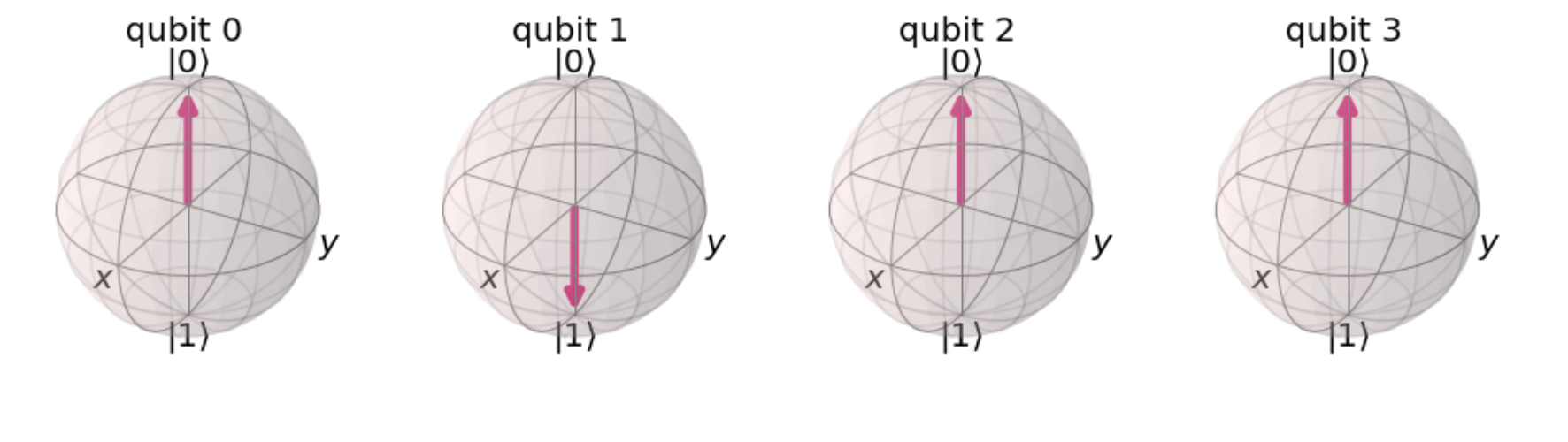
For the Fourier basis, we find the qubit 0 rotate to y, also the qubit 0 rotate each flip in computational basis. Hence, there are 4 flips in computational basis. The representation <q3,q2,q1,q0> will be 0100. Hence <q0,q1,q2,q3> will be 0010.



State in Fourier Basis

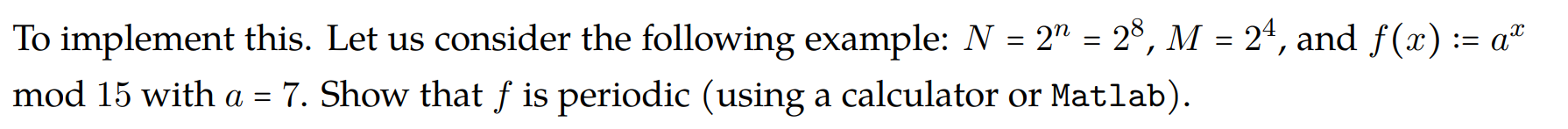


State in Computational Basis

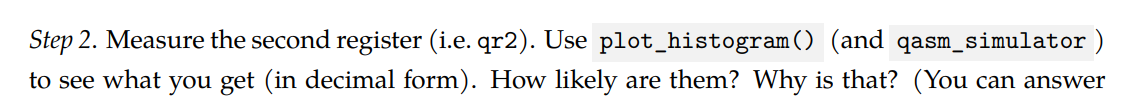


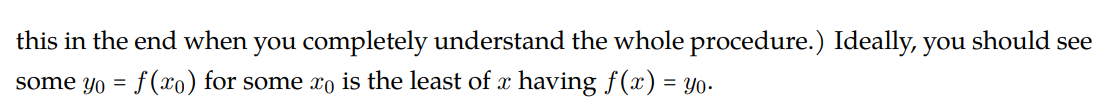
For the Fourier basis, we find the qubit 0 rotate exactly between x and y, also the qubit 0 rotate each flip in computational basis. Hence, there are 2 flips in computational basis. The representation <q3,q2,q1,q0> will be 0010. Hence <q0,q1,q2,q3> will be 0100.

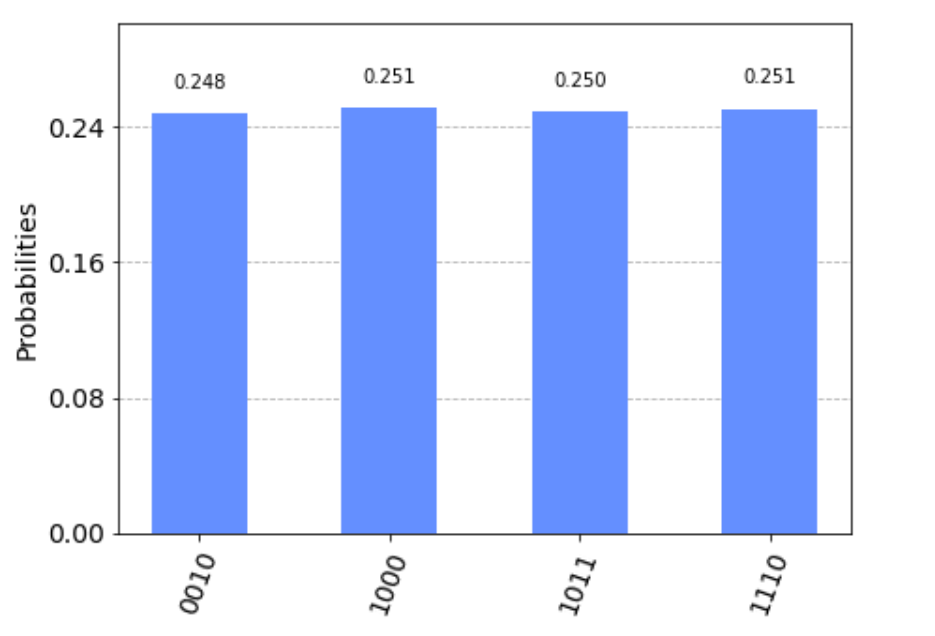
Q4.



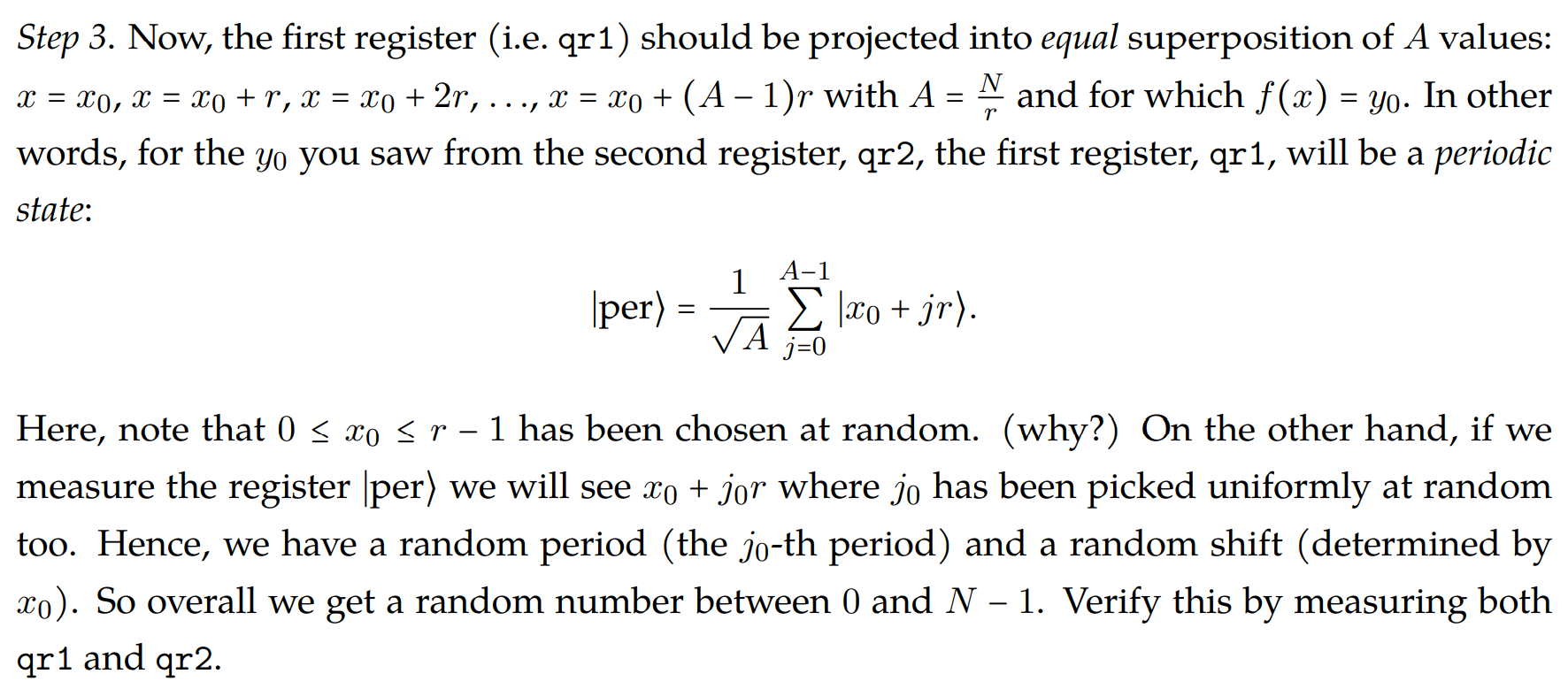
f(1) = 7, f(2) = 4, f(3) = 13, f(4) = 1, f(5) = 7... Hence f is periodic with period 4.





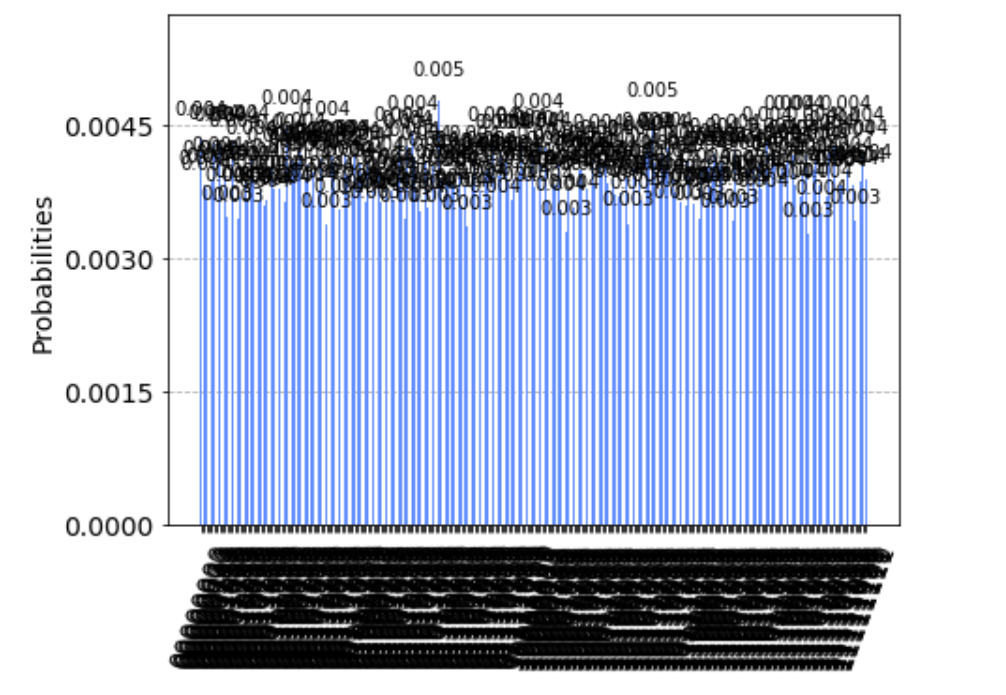


We get 2,8,11,14 as equal probability 0.25. It is easily find that these numbers are twice modulus of the power of 7 mod 15(7,4,13,1).(for 11, the half modulus 11 mod 15 is (11+15)/2=13). The reason why these numbers are twice modulus of the power of 7 mod 15 is still a mystery for me.



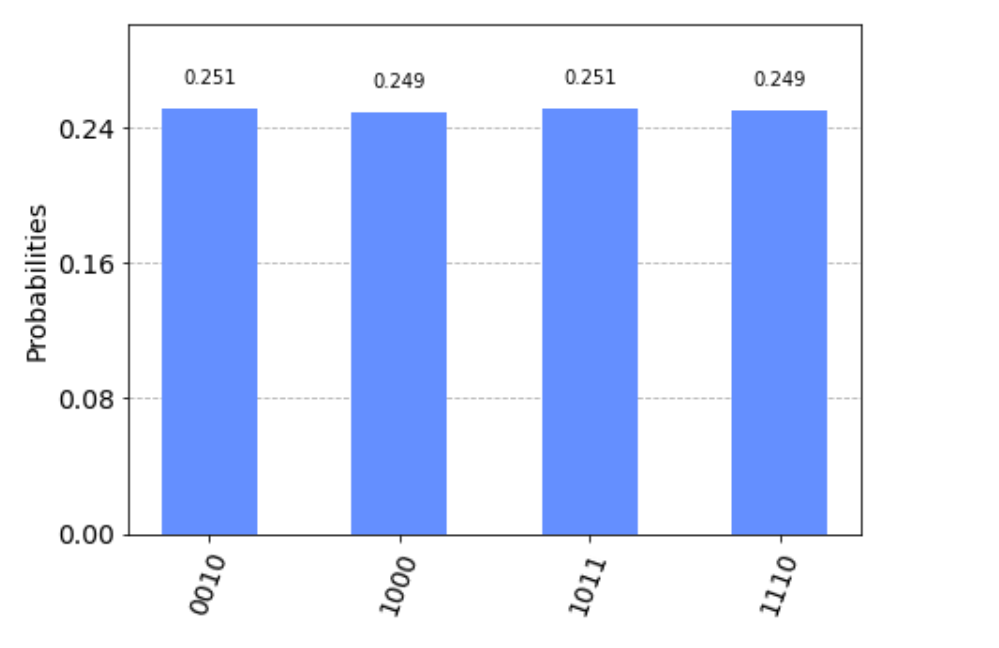
If we didn’t randomize chosen x0, x0+jr may lead all the state be the same. And the per state may be A times the same state, which is not preferred.

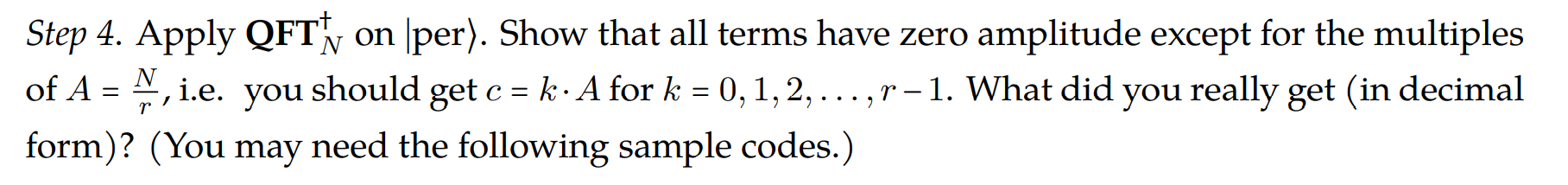
qr1

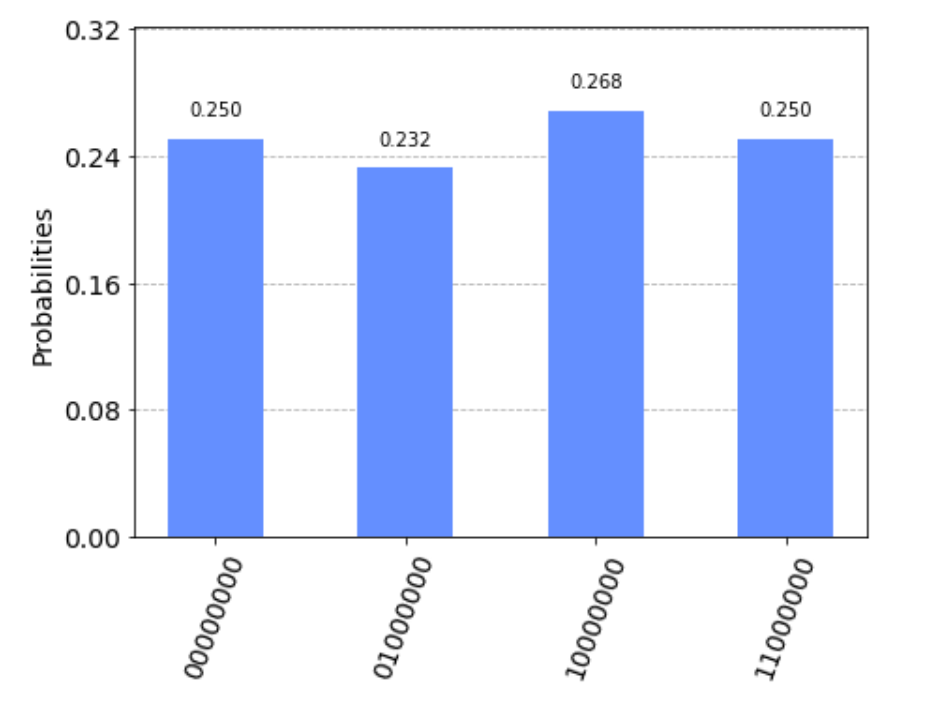


Uniform distribution between 0 to N-1

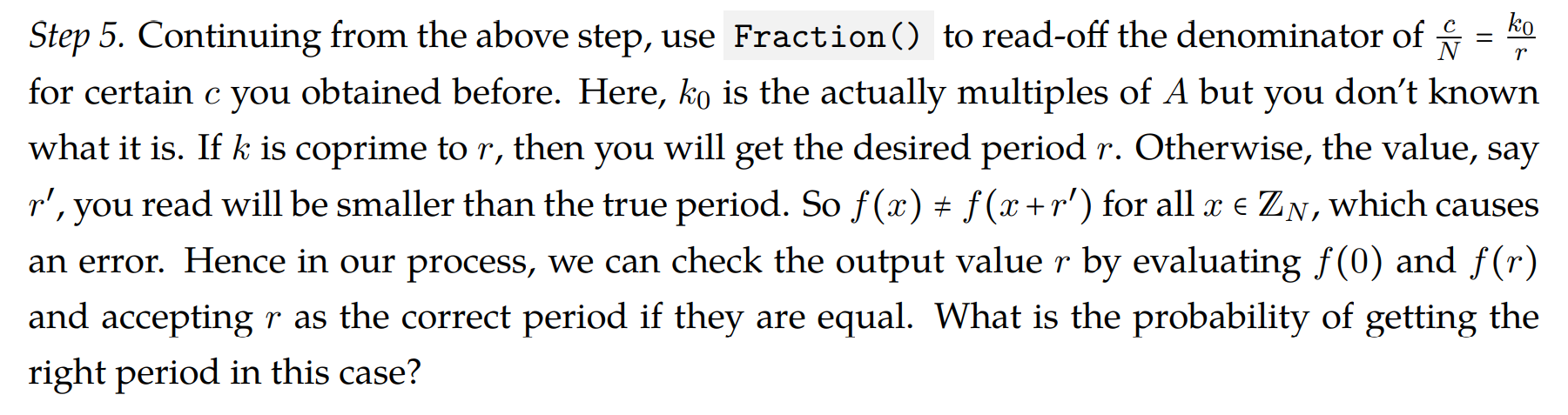
qr2





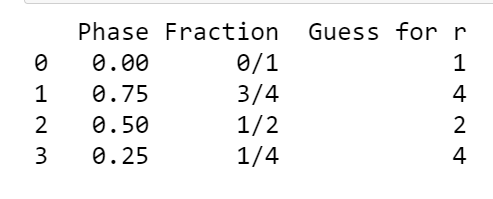


N = 256, r = 4, A = 256/4 = 64. We get c = 0,64,128,192.



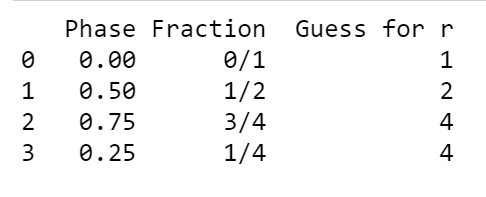
There are 4 cases in a = 7, where two cases represent right period 4. Hence the probability of getting the right period in this case is 50%.

a=7



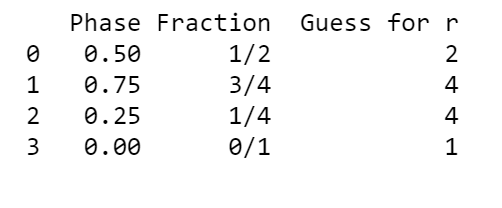
After validating, the period is 4.

a= 2



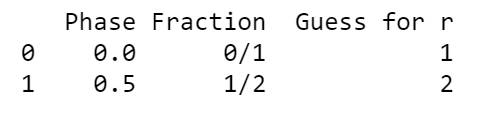
After validating, the period is 4.

a=8



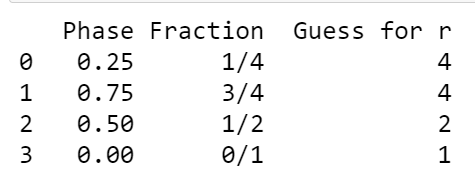
After validating, the period is 4.

a=11



After validating, the period is 2.

a=13



After validating, the period is 4.

Appendix

Code: <https://github.com/yuanchiachang/CommLab/blob/main/Lab3/src>

Reference :

[1] https://qiskit.org/textbook/ch-algorithms/deutsch-jozsa.html

[2] https://qiskit.org/textbook/ch-algorithms/grover.html

[3] https://qiskit.org/textbook/ch-algorithms/quantum-fourier-transform.html

[4] https://qiskit.org/textbook/ch-algorithms/shor.html