CMPT409 Assignment3 Report

Zhuo Ning 301421723
Shengyao Wang 301356790
Yue Zhao 301385561
Shifan He 301386458
Luoyijie Zhang 301355551

July 2022

1 Problem Description

1.1 What is Deutsch–Jozsa algorithm

1.2 Motivation

1.3 Classical way vs Quantum way

2 Oracle Formulation

2.1 Constant Oracle

2.2 Balanced Oracle

3 Deutsch Jozsa Algorithm Principle

3.1 Algorithm working principle

3.2 Phase Kickback

4 Our Implementation

5 Demo and Visualization

6 Reference

1 Problem Description

1.1 What is Deutsch–Jozsa algorithm

Deutsch's algorithm can be thought as a solution to combine quantum parallelism with a property of quantum interference. It is a deterministic quantum algorithm proposed by David Deutsch and Richard Jozsa in 1992 with improvements by Richard Cleve, Artur Ekert, Chiara Macchiavello, and Michele Mosca in 1998. Although of little current practical use, it is one of the first examples of a quantum algorithm that is exponentially faster than any possible deterministic classical algorithm.[1]

1.2 Motivation

The algorithm is specifically designed that allowed a quantum algorithm easy to solved where the problem would be very hard for a classical deterministic algorithm. The original idea for this algorithm is to prove that a quantum computer can solve the black box problem without error, and for the classical computer, it will need a large amount of time and queries to the black box in order to solve the problem.[2]

1.3 Classical way vs Quantum way

For a traditional deterministic algorithm and n represent number of bits, in the worst case, it will use at most 2^{n-1} +1 of evaluations. To prove that f is constant, just over half the set of inputs must be evaluated and their outputs found to be identical (remembering that the function is guaranteed to be either balanced or constant, not somewhere in between). The best case occurs where the function is balanced and the first two output values that happen to be selected are different.
The Deutsch-Jozsa quantum algorithm produces an answer that is always correct with a single evaluation of f.[3]

2 Oracle Formulation

The problem that Deutsch-Jozsa algorithm attempts to solve is expressed using binary functions. Hence, we need to create a "black box" U_f implementing some oracle functions f:$\{0, 1\}^n \rightarrow \{0, 1\}$ quantum circuit model and determine whether f is "constant" or "balanced". An oracle is a function that maps n bits (0 or 1) to f(0) or f(1), and it can be either constant or balanced.

2.1 Constant Oracle

In the constant function, since the input has no effect on the output so we can applying Pauli X gate to randomly set the output qubit to be 0 or 1,which means f(x) = 0 for all x or f(x) = 1 for all x.[4]

Algorithm 1 Constant Oracle

```
$n \leftarrow 5$
$flip \leftarrow np.random.randint(2)$
**if** $flip == 1$ **then**
  **for** n in range (n-1) **do**
    qc.x(i) // apply Pauli X gate in each input
  **end for**
**end if**
**return** qc
```

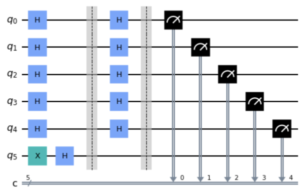


Figure 1: Const QuantumCircuit

After running the circuit on a quantum simulator, we get the following result:

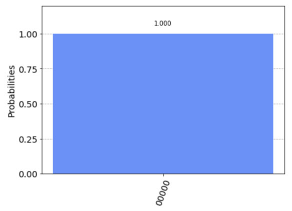


Figure 2: Constant oracle plot

As We can see from the results, the measurement is 00000 per each time. Hence, the oracle function is constant.

2.2 Balanced Oracle

By applying CNOT gates with each input qubit as a control and the output bit as the target, a balanced oracle can be created.[5] When the oracle is a balanced function, f(x) = 0 for half of the input values x and f(x) = 1 for the other half.

Algorithm 2 Balanced Oracle

```
$n \leftarrow 5$
for n in range (n-1) do
    qc.cx(i) // apply CNOT gate in each input
end for
return qc
```

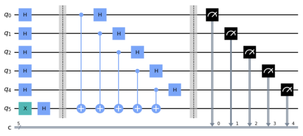


Figure 3: Balanced QuantumCircuit

After running the circuit on a quantum simulator, we get the following result:

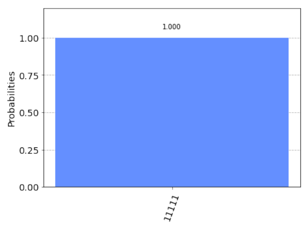


Figure 4: Balanced oracle plot

As We can see from the results, there is no chance of measuring 00000. Hence, the oracle function is balanced.

3 Deutsch Jozsa Algorithm Principle

3.1 Algorithm working principle

3.1.1 Constant Oracle

The quantum states does not change before or after querying when it is constant. In Deutsch Jozsa Algorithm the H-gate is own inverse and apply H-gate to implement obtain the initial quantum state of $|00...0>$in the first register.

3.1.2 Balanced Oracle

After apply H-gate to each qubit, input register is an equal superposition of all the states in the computational basis. If the oracle is balanced, phase kickback will add a negative phase to exactly half these states: There will never get all-

$$U_f \frac{1}{\sqrt{2^n}} \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2^n}} \begin{bmatrix} -1 \\ 1 \\ -1 \\ \vdots \\ 1 \end{bmatrix}$$

Figure 5:

zero state, because after applying the H-gate, a quantum state will be orthogonal to $|00...0>$ and then the algorithm end up with it.[5]

3.2 Phase Kickback

Phase kickback is where the eigenvalue added by a gate to a qubit is 'kicked back' into a different qubit via a controlled operation and it will help to simplify the calculation of the Deutsch Jozsa algorithm.[6]

Conclusion: Quantum mechanics are counter-intuitive, with help of phase kickback, even though the phase is not directly measurable, there are ways to exploit differences in the phase between states.[7]

4 Our Implementation

Algorithm 3 Pseudo code

```
1: Apply H to every input qubits which will lead to a uniform superposition of $2^n$ possible states.
2: Apply X gate to output qubit, so that the output is in the $|1\rangle$ state.
3: Run the oracle on the input register and the output qubit.
4: Apply H to all of the input qubits.
5: Measure all the input qubits by applying the H gate. If all of them are $|0\rangle$, then the function is constant. Otherwise the function is balanced. [8]
```

5 Demo and visualization

By using four different oracles to exhibit demo and visualization.
Oracle with constant 1, case comes out thought Deutsch-Jozsa Algorithm:

```
The given Oracle is constant.
```

Figure 6: Algorithm output

After running the circuit on a quantum simulator, we get the following result:

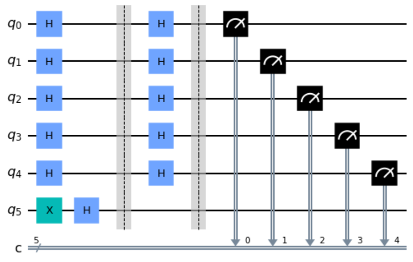


Figure 7: Constant QuantumCircuit

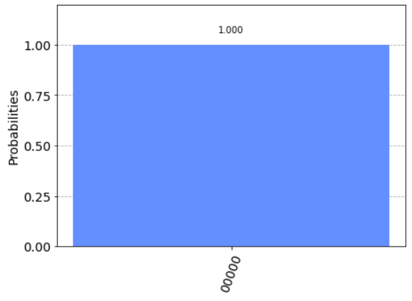


Figure 8: Constant oracle plot

Second test case, oracle with constant 0, case comes out thought Deutsch-Jozsa Algorithm:

```
The given Oracle is constant.
```

Figure 9: Algorithm output

After running the circuit on a quantum simulator, we get the following result:

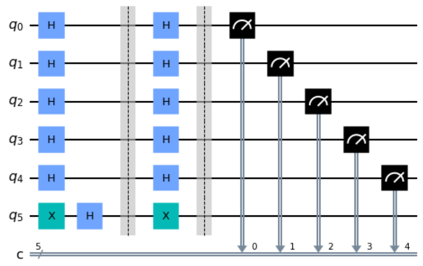


Figure 10: Constant QuantumCircuit

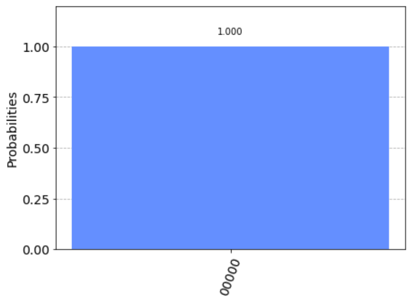
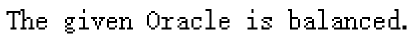


Figure 11: Constant oracle plot

Third test case, oracle has even number of input bit flip into 1, case comes out thought Deutsch-Jozsa Algorithm:



```
The given Oracle is balanced.
```

Figure 12: Algorithm output

After running the circuit on a quantum simulator, we get the following result:

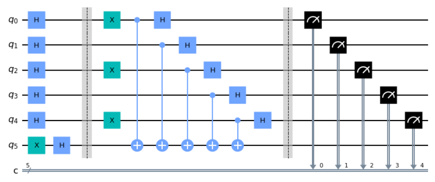


Figure 13: Constant QuantumCircuit

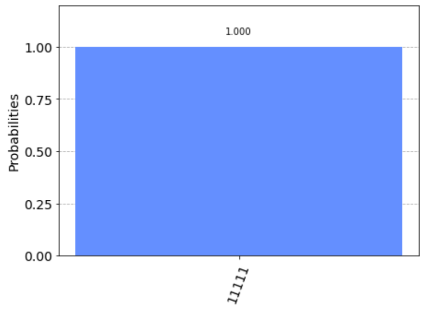
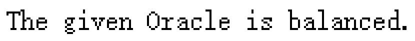


Figure 14: Constant oracle plot

Fourth test case, oracle has even number of input bit flip into 1, case comes out thought Deutsch-Jozsa Algorithm:



```
The given Oracle is balanced.
```

Figure 15: Algorithm output

After running the circuit on a quantum simulator, we get the following result:

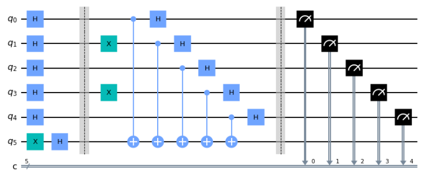


Figure 16: Constant QuantumCircuit

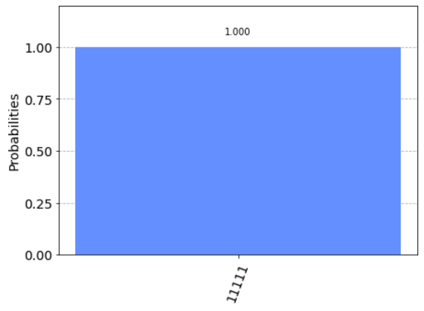


Figure 17: Constant oracle plot

6 Reference

[1][2][3]“Deutsch–Jozsa Algorithm.” Wikipedia, Wikimedia Foundation, 4 July 2022, https://en.wikipedia.org/wiki/Deutsch
[4][5]”Deutsch-Jozsa Algorithm” Qiskit, Qiskit.org, 12 July 2022, https://qiskit.org/textbook/ch-algorithms/deutsch-jozsa.html
[6]”Phase Kickback” Qiskit, Qiskit.org, 12 July 2022, https://qiskit.org/textbook/ch-gates/phase-kickback.html
[7]”Quantum Phase Kickback” Frank Zickert, towardsdatascience, 14 July, 2022, https://towardsdatascience.com/quantum-phase-kickback-bb83d976a448.
[8]Preston, Joe Clapis amp; Richard. “Deutsch-Jozsa Algorithm¶.” Deutsch-Jozsa Algorithm - Intro to Quantum Software Development, 26 June 2021, https://stem.mitre.org/quantum/quantum-algorithms/deutsch-jozsa-algorithm.html.