Assignment 4.: Deutsch algorithm

Dr. Suyong Eum

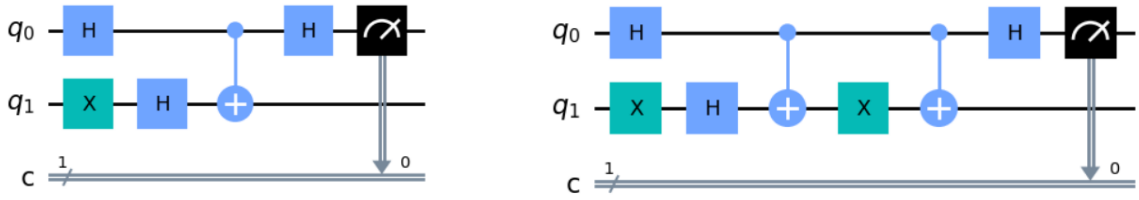
1^{st} Semester, 2023

1 Description

The aim of this assignment is to understand the operation of Deutsch algorithm in detail through hands-on experience on its implementation using qiskit[1], which is an open-source quantum computing framework that allows developers to build and execute quantum circuits.

The Deutsch algorithm is a quantum algorithm proposed by David Deutsch in 1985. It utilizes quantum superposition and interference to determine if a given function is balanced or constant. The algorithm prepares a quantum state, puts qbits in a superposition representing all possible inputs, and evaluates the function using quantum gates. By analyzing the interference pattern, it can determine the function's nature in a single evaluation. The Deutsch algorithm showcases the power of quantum computing, demonstrating its potential to outperform classical algorithms for certain problems.

2 Required Task(s)



(a) case: 1 (b) case: 2

Figure 1: Implementation of Deutsch algorithm

In this assignment, you are expected to create two quantum circuits as shown in Fig.1a and Fig.1b, and discuss the differences between them. The discussion should answer the questions below;

1. What is the measured value at the end of each circuit?
2. What are the states of both qbits, q_0 and q_1, just before and after the last Hadamard gate at q_0 for both cases? Visualize the states of both cases using Bloch Sphere.
3. Answer the above two questions again when the first X gate is added to q_0 not q_1.

The code, the implementation of the case 1 in Fig.1a, is given in Appendix A to help you complete this assignment. You should be able to understand the code reading the comments provided. Please, extend the code to create the quantum circuit shown in Fig.1b, and complete this assignment[2].

3 Administrative

- Due: 24:00, July 31, 2023
- Submission to (suyong@ist.osaka-u.ac.jp)
  - Please, zip the report. Then, name with your student number and assignment number, e.g., 32A18041_4.zip
- Late submission will be penalized at the rate of 10% reduction per day

[1] https://qiskit.org/

[2] You need to install qiskit. Please, refer the link; https://qiskit.org/documentation/getting_started.html.

Appendices

A Implementation of Deutsch algorithm shown in Fig.1a

```
##################################################
# Quantum circuit creation for Deutsch algorithm
##################################################

from qiskit import *
from qiskit.tools.visualization import plot_histogram

# Number of quantum circuits
num_qc   = 2

# Number of classical circuits
num_cc   = 1

# Create a circuit object
qcircuit = QuantumCircuit(num_qc, num_cc)

# Add x gate to the quantum circuit (q_1)
qcircuit.x(1)

# Add Hadamard gate to the quantum circuits, (q_0) and (q_1), respectively
qcircuit.h(0)
qcircuit.h(1)

# Add CNOT gate which connects the quantum circuit from (q_0) to (q_1)
qcircuit.cx(0,1)

# Add Hadamard gate to the quantum circuit (q_0)
qcircuit.h(0)

# Measure the quantum circuit (q_0) and show the result in (C)
qcircuit.measure(0,0)

# Draw the quantum circuit which includes all the gates added
qcircuit.draw(output='mpl')

##################################################
# Run the circuit on a quantum simulator
##################################################

backend=Aer.get_backend('qasm_simulator')
result=execute(qcircuit,backend=backend,shots=1000).result()
counts=result.get_counts(qcircuit)
plot_histogram(counts)
```