EECE 435 - Introduction to Quantum Computing
"Quantum Programming Project"

Due date: 11:59 pm (before midnight), Friday, December 6, 2024

Instructions: The project must be done individually. Ensure that your code is properly tested and well-commented. Submit both your code and a PDF report discussing your solutions, results, and answers to the provided questions.

Submission Guidelines: Make sure all your files are zipped in one folder. Use the following format for naming your files: `YourLastName_Project.zip`.

Oral Exam: Each student will participate in a 5-minute oral examination, during which I will ask questions related to the project. The questions may pertain to the code, results, discussion, or underlying concepts. Please ensure you have a thorough understanding of your code and findings. Oral exams will take place during the week of December 9, 2024.

Problem 1: Order of Quantum Gates

In this exercise, you will investigate how the order of applying quantum gates affects the final output of a quantum circuit. Specifically, you will compare the outcomes of two circuits that apply the following rotation gates in different orders:

- Circuit 1: Apply an $RX(\theta_1)$ gate followed by an $RY(\theta_2)$ gate on a single qubit.
- Circuit 2: Apply an $RY(\theta_2)$ gate followed by an $RX(\theta_1)$ gate on a single qubit.

Task:

- Construct both circuits in Pennylane and measure the expectation values of the Pauli-X observable for both circuits.
- Choose two angles, θ_1 and θ_2, and compute the probability distributions of the measurement outcomes for each circuit.
- Calculate the absolute difference between the Pauli-X expectation values of the two circuits:
$$\text{Absolute Difference} = |\langle\sigma_x\rangle_1 - \langle\sigma_x\rangle_2|$$
- Next, measure the expectation values of the Pauli-Z observable for both circuits.
- Provide your code, results, and a comparison of the difference between the Pauli-Z measurements and the Pauli-X results.

Also, answer the following questions:

1. What differences do you observe between the outcomes of the two circuits when measuring in the Pauli-X basis?
2. How does the result change when measuring in the Pauli-Z basis?
3. Does the order of operations (RX and RY) affect the results in the same way for both Pauli-X and Pauli-Z measurements?
4. What insights can you derive about the nature of these rotation gates based on your results?

Problem 2: GHZ State Creation and Optimization

The GHZ state (Greenberger–Horne–Zeilinger state) is a special type of entangled quantum state involving multiple qubits. For three qubits, the GHZ state is written as:

$$|\mathrm{GHZ}\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

For five qubits, the GHZ state is written as:

$$|\mathrm{GHZ}_5\rangle = \frac{1}{\sqrt{2}}(|00000\rangle + |11111\rangle)$$

Task:

- Design a quantum circuit that generates a 3-qubit GHZ state. Then, measure the state and verify that it is the GHZ state. You are free to use any gates available in Pennylane.
- After running the simulation, verify the final state by performing measurements and calculating the probability distribution.
- Extend your quantum circuit to create a 5-qubit GHZ state. Make sure your circuit has an optimized depth and number of gates. The smaller the depth of your circuit, the higher your grade will be.
- Provide your code, results, and a comparison of the number of gates and the overall circuit depth between the 3-qubit and 5-qubit cases.

Problem 3: Counting SWAP Gates Needed for a CNOT Gate

In some quantum hardware architectures, qubits may not be fully connected. If two qubits are not directly connected and a CNOT gate needs to be applied between them, SWAP gates must be used to move the qubits to adjacent positions.

You are given a quantum hardware architecture represented by the following graph:

$$\texttt{graph} = \begin{cases} 0 : [1], \\ 1 : [0, 2, 3, 4], \\ 2 : [1], \\ 3 : [1], \\ 4 : [1, 5, 7, 8], \\ 5 : [4, 6], \\ 6 : [5, 7], \\ 7 : [4, 6], \\ 8 : [4] \end{cases}$$

Each key represents a qubit, and the values represent the qubits that are directly connected to it.

Task:

Implement a function that computes the minimum number of SWAP gates required to implement a CNOT gate between two qubits on this hardware. The function should:

- Take a CNOT operation (with specific control and target qubits) as input.
- Return the minimum number of SWAP gates needed to move the qubits into adjacent positions for the CNOT operation.

Example: For a CNOT gate between qubits 0 and 4, determine the number of SWAP gates necessary to perform the operation on the given hardware. Provide your code along with the results for this example.

Problem 4: Deutsch's Algorithm Implementation

The Deutsch algorithm is a quantum algorithm that determines whether a given single-bit function $f : \{0, 1\} \to \{0, 1\}$ is constant (gives the same output for all inputs) or balanced (gives different outputs for different inputs). In this problem, the function f is implemented as an oracle, which is a quantum black-box function that modifies the state of the qubits based on the encoded function f.

Task:

1. Implement Deutsch's algorithm using a quantum circuit that determines whether a function is constant or balanced in PennyLane.
2. The quantum circuit should take as input a single oracle (function) f and output a measurement result:
   - Output '0' if the function is constant.
   - Output '1' if the function is balanced.

Problem 5: Deutsch-Jozsa Algorithm Implementation

The Deutsch-Jozsa algorithm is an extension of the Deutsch algorithm that works for n-bit functions $f : \{0,1\}^n \rightarrow \{0,1\}$. This algorithm determines whether the given function is constant (gives the same output for all inputs) or balanced (gives an equal number of 0 and 1 outputs). Like in the Deutsch algorithm, the function f is encoded as an oracle, which acts on multiple qubits. Your oracle will be a quantum operation, or a series of quantum gates, that is applied to a set of qubits.

Task:

1. Implement the Deutsch-Jozsa algorithm using a quantum circuit that determines whether a function is constant or balanced based on the measurement outcome.
2. You must choose two different oracles (functions):
   - One oracle that encodes a constant function.
   - One oracle that encodes a balanced function.
3. Test your implementation with your two oracles and provide the results, including:
   - A brief description of each function you selected and how it is encoded in the oracle.
   - The output of your Deutsch-Jozsa algorithm for each function.
   - A short explanation of why the algorithm produced the given results.
4. **What is the role of the Hadamard gates in the Deutsch and Deutsch-Jozsa algorithms?** Explain how the Hadamard gates are used to prepare the input qubits and how they affect the output qubits.
5. **What makes an oracle balanced or constant?** Give two examples of constant oracles, and two examples of balanced oracles.

Problem 6* (Bonus): Quantum Superdense Coding Using Bell Pairs

Superdense coding is a quantum communication protocol that allows Alice to send two classical bits of information to Bob by sending only one qubit, using a shared entangled state.

Task:

1. Construct a quantum circuit that implements the superdense coding protocol using Bell states.
   - Alice and Bob share a Bell pair of entangled qubits. Alice wants to communicate two classical bits b_1 and b_2 to Bob.

- Depending on the value of these bits, Alice applies certain quantum operations to her qubit.
- Alice sends her qubit to Bob. Bob then performs quantum operations and measurements on both qubits to retrieve the two classical bits b_1 and b_2.

2. Your quantum circuit should take an integer as input representing Alice's classical bits (0, 1, 2, or 3), which correspond to binary values b_1b_2, and it should output the two classical bits b_1b_2 that Bob retrieves after his operations.

3. Test your circuit using two different Bell states:

   - $|\Phi^+\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$
   - $|\Psi^-\rangle = \frac{|01\rangle-|10\rangle}{\sqrt{2}}$

   Show the results for each Bell state and explain the outcomes.