Homework 5

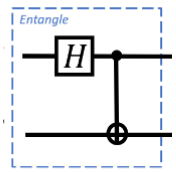
CSE 490q

Due: Fri, Nov 6th by 11pm PST

This homework requires to write quantum programs using Q#. First read the "Introduction to Q#" to learn how to install Q# on a computer and write programs using this quantum programming language.

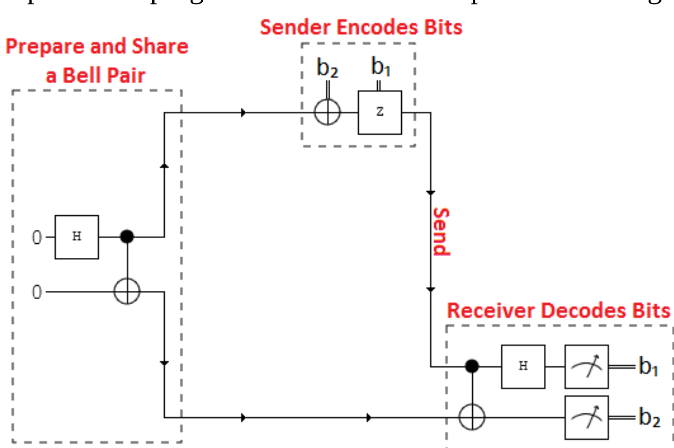
1. Implement a `RandomBit` operation that returns a 0 or 1 randomly. Create an `@EntryPoint` in your Q# file and call this `RandomBit` operation 10 times, printing the result to the console.

2. Implement an `Entangle` operation that given two qubits in the $|00\rangle$ state, prepares them into the $^1/_{\sqrt{2}}(|00\rangle + |11\rangle)$ state. As a reminder, this is accomplished by the following circuit:



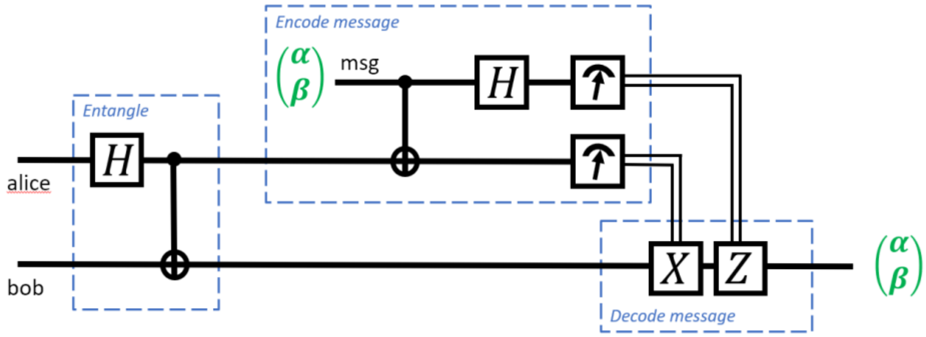
   Create an `@EntryPoint` in your Q# file, that allocates two qubits and calls this `Entangle` operation and measures the qubits 10 times. Print the resulting measurements to the console and verify that the values measured are always (0,0) or (1,1).

3. Implement a program that showcases Superdense coding:



   a. Create an operation `EncodeBits` that receives one qubit and two bool parameters. The operation then encodes the message using an X and a Z gate accordingly.
   b. Create an operation `DecodeBits` that receives two qubits and returns a tuple with two Results. The operation then decodes the original bits by applying a $CNOT_{1,2}$ and an H_1, it returns the measurement of both qubits.
   c. Create a main `@EntryPoint` that accepts two bool flags and puts together the other 3 operations (`Entangle`, `EncodeBits` and `DecodeBits`) to implement the circuit shown above, and print the values returned by the last one. These values should match the inputs.

4. Implement a program that showcases Teleport:



a. Create an operation `EncodeMessage` that receives one qubit and a bool parameter and returns two classical values. This operation allocates a new qubit and prepares it as $|+\rangle$ if the bool parameter is True, or $|-\rangle$ if False. The operation then encodes the message using a CNOT and an H, then return the measurement of the two qubits.
b. Create an operation `DecodeMessage`, that receives one qubit and two classical parameters, and decodes the message (sets the corresponding qubit to match the message state) by applying an X or a Z depending on the inputs.
c. Create a main `@EntryPoint` that accepts a bool flag and puts together the other 3 operations (`Entangle`, `EncodeMessage` and `DecodeMessage`) to implement the circuit shown above, and print if the final state of the "bob" qubit is $|+\rangle$ or $|-\rangle$.