Middle East Technical University Department of Computer Engineering

CENG 487

Introduction to Quantum Computing

Fall 2021-2022

Assignment 3 & 4

Due date: 25.01.2022, 23:59

1. The Environment

a. You will implement quantum computing programs using real quantum computers and simulators on IBM quantum computing environment(quantum-computing.ibm.com).

b. The easiest way to create your quantum circuits visually or using OpenQASM 2.0 (an assembly-like language for quantum computing) is IBM Quantum Composer(quantum-computing.ibm.com/composer). You can also use python library Qiskit (v0.3.2.0) on IBM Quantum Lab(https://lab.quantum-computing.ibm.com)

c. You can get help from the tutorials to start with quantum computing: quantum-computing.ibm.com/docs/

2. Quantum Error Correction (50 pts)

For classical error correction, we can just duplicate the bits (i.e. encodig with 3 bit repetition code) and do correction by methods such as “majority voting”. However in quantum computing we can not simply apply same process because we can not clone the phase of a qubit due to the non-cloning theorem. Moreover, in addition to bit flips, phase flips are also possible in quantum computing and quantum errors are more continuous in nature.

One of the methods for error correction in quantum computing is using 9 qubits Shor code which is a combination of 3 qubit bit flip correction code and 3 qubit phase flip code.

In this part, you will construct your own qubit using Haddamard and Phase-Shift gates and measure it on both real quantum computers and simulators.

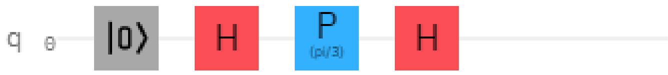
Then, you are going to implement quantum error correction using Shor code and test for it’s correctness using only bit-flip, only shape-flip and both bit-flip and shape-flips.(The tests will be done on simulators because of the limitation of real computers on IBM environment)

For Shor code implementation you can use the one on the lecture notes. (Another example of a Shor code error correction implementation is here. You can also implement taking that one as reference)

Note that you will only check for the error correction on the initial qubit. (It is not in the scope of this asssignment but you may also want to try and see what happens when there are errors on other qubits as an excercise)

a. Create $|\psi> = H \cdot P\left(\frac{2a-20}{a}\pi\right) \cdot H \cdot |0>$, where a is the sum of the digits of your METU student ID and test it both on a real quantum computer and on a simulator with shots=1024. Compare the results with the expected ones. (The results of the simulator will be used as a base for checking the error correction on the following parts.)

For example, if your student ID is 2530101, a will be 12 and $|\psi>$ can be created using the following circuit:



b. Implement error correction using 9 qubit Shor code for $|\psi>$ and check its correctness by adding a NOT gate to the qubit after the encoding part. Test on a simulator with shots=1024 to see if your implementation corrects the error. (i.e. bit flip check)

c. Implement error correction using 9 qubit Shor code for $|\psi>$ and check its correctness by adding a Z gate to the qubit after the encoding part. Test on a simulator with shots=1024 to see if your implementation corrects the error. (i.e. phase flip check)

d. Implement error correction using 9 qubit Shor code for $|\psi>$ and check its correctness by adding both NOT and Z gates to the qubit after the encoding part. Test on a simulator with shots=1024 to see if your implementation corrects the error.

3. Quantum Fourier Transform (50 pts)

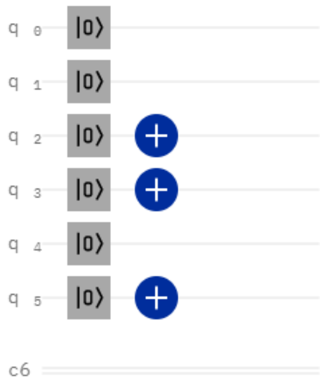
In this part you are expected to construct a 6 qubit Quantum Fourier Transform(QFT) Circuit and apply it to a quantum state determined by your Student ID then test it both on a simulator and a real quantum computer.

For 6 qubit QFT, you can take the 3 qubit one on the lecture notes as reference. (Another examples of QFT implementations are here and here. You can also implement taking these as reference)

Hint: On IBM Quantum Composer, you can construct controlled R_S gates by combining phase shift gates ($P_{\frac{\pi}{2^S}}$) and control gate modifier. (cp() function in OpenQASM and QuantumCircuit.cp() in Qiskit).

a. Create $|\psi> = |q_5q_4q_3q_2q_1q_0>$, where $q_5q_4q_3q_2q_1q_0$ is the binary representation of a and a is the sum of the digits of your METU student ID.

For example, if your student ID is 2687948, a will be 44 and $|\psi> = |101100>$ can be created using the circuit on the right hand side.



b. Implement 6 bit QFT and apply Quantum Fourier Transform on $|\psi>$. Execute your circuit on both on a simulator and a real Quantum Computer. Comment on the results.
c. What is the advantage of QFT over Fourier Transformation on classical computing? Can this advantage be seen on your application? (Explain). How does this advantage change as the input size grows?

4. Submission

a. For this assignment, we **expect** you to upload three files: The first one is your report as a **pdf** file consisting of your answers, comments and comparisons and the **images** of your circuits. The second and third are the files containing OpenQASM or Qiskit **codes** of your circuits for only **part 2.d** and **3.b**.

b. Try to finish as early as possible. The sytems to run your work are shared between many different users across the earth. Therefore some systems may have longer queues that can cause for you to wait. (look for systems that have smaller number of jobs on queue).

c. You can still submit your work if the **deadline** is passed, however with an increasing **penalty** of **5*days*days**. (i.e. first day -5 points, second day -5*2*2=-20 points and so on). Note that even a minute late means that it is the other day.

d. The assignments are for individual work. Your work should be done by only you and it should be genuine. We have zero tolerance policy for cheating. People involved in cheating will be punished according to the university regulations and will get zero.