CIS 492/593 Hands-On Lab 3 Quantum Gates and Circuits Fall 2023

The aim of this hands-on lab is to familiarize yourself with:

- entanglement with more than 2 qubits in Qiskit
- constructing circuits with quantum gates such as X, H, CNOT, and Toffoli
- use of quantum gates to emulate classical gates (AND, NAND, and OR)

Login to a workstation and open a terminal (either middle click the terminal icon on the left side bar or press CTRL-ALT-T). In the terminal window, type

```
cd QC
jupyter notebook
```

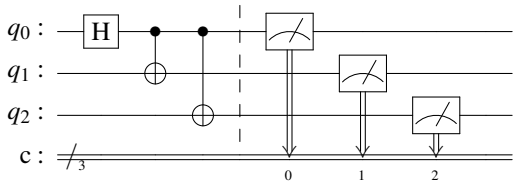
Inside the browser, click "New" and choose "Python 3 (ipykernel)". For each experiment, you need to put the experiment number as a comment (e.g. `# Experiment 1`) in the top of the code.

Experiment 1: 3-qubit Entanglement

- In the cell, type

  ```
  %load entangle2.py
  ```

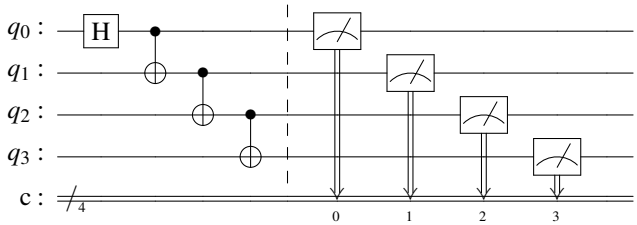
  and click "Run" to load the Qiskit program which you did in Lab 2 into the cell.
- Modify the code in the cell to build the circuit below which entangles 3 qubits:



  Note that you may call `qc.barrier()` before the measurements. This will generate a synchronization point (i.e. a dashed line in the circuit diagram), though it will not affect the result.
- Click "Run" to get, record, and explain the result in your report.

Experiment 2: 4-qubit Entanglement

- Click the cell which contains the code done in Experiment 1. Click the "Edit" button and select "Copy Cells". Click "Edit" again and choose "Paste Cells Below".
- In this experiment, we use an alternative way to entangle 4 qubits. Modify the code in the cell to build the circuit below:

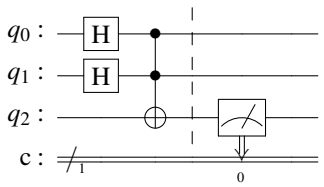


- Click "Run" to get, record, and explain the result in your report.

Experiment 3: Toffoli gate

In the class, we study the the Toffoli gate (also called CCNOT gate or "controlled-controlled-not" gate) which is a universal reversible logic gate. That is, if the first two control bits are both set to 1, it inverts the third bit, otherwise all bits remain the same.

- Copy and paste the code cell done in Experiment 1. Modify the code in the cell to build the circuit below:



  Note that we only need one classical bit and it has the value measured from q_2. In Qiskit, the interface of the Toffoli gate is

  ```
  ccx(c0,c1,t)
  ```

  where `c0` and `c1` are the control qubits and t is the target qubit.
- Click "Run" to get and record the result. Explain the result.

Experiment 4: Implementing AND and NAND gates

- Type the following in a new empty cell:

  ```
  %load ~cis492s/pub/ANDgate.py
  ```

  Click "Run" to load the code template into the cell.
- The AND function is incomplete. Add one line of code to complete it. Describe your circuit design in the report.
- Click "Run" to execute the code and verify your result. Is your AND quantum circuit reversible?
- Copy and paste the code cell into a new cell. To implement the NAND gate function, simply add a NOT gate (i.e. the quantum X gate) to flip the result of qubit 2 at the end. Also, replace every occurrence of "AND" with "NAND" in the code, including comments.
- Click "Run" to execute the code and verify your result

Experiment 5: Implementation of the OR gate

- Type the following in a new empty cell:

  ```
  %load ~cis492s/pub/ORgate.py
  ```

  Click "Run" to load the code template into the cell.
- The OR function is incomplete. Add three lines of code to complete it. Describe your circuit design in the report.
  Hint: Use two CNOT gates and one Toffoli gate to flip the target qubit in succession. An alternative way is to invert both inputs, use a NAND gate to get the output, and then invert inputs back.
- Click "Run" to execute the code and verify your result.

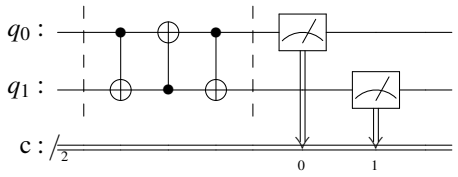
Experiment 6: Application of CNOT gates

- Type the following in a new empty cell:

  ```
  %load ~cis492s/pub/UNKNOWNfunc.py
  ```

  Click "Run" to load the code template into the cell.

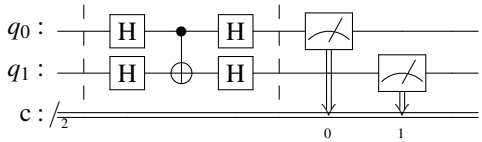
- The function UNKNOWN is incomplete. Add the code between two barrier calls based on the circuit below:



- Click "Run" to execute the code. Figure out what the the unknown function is.
  Hint: The CNOT function is like an XOR gate. Hence, the circuit is like executing the following instructions in sequence:
  $q_1 = q_0 \oplus q_1$
  $q_0 = q_0 \oplus q_1$
  $q_1 = q_0 \oplus q_1$

Experiment 7: Application of one CNOT and four H gates

- Type the following in a new empty cell:
  `%load ~cis492s/pub/UNKNOWNfunc.py`
  Click "Run" to load the code template into the cell.
- Add the code between two barrier calls in the UNKNOWN function based on the circuit below:



- Click "Run" to execute the code. Figure out what the the unknown function is.
  Hint: How does the qubit 1 affect the qubit 0?

Click "File" and choose "Save as". Type "lab3" in the entry box and click "Save" to save your work today into the file `lab3.ipynb`. To turn in your file, use CTRL-ALT-T to open a terminal and type

```
ssh grail
```

and type your password to login to the server grail. Then, type

```
cd QC
turnin -c cis492s -p lab3 lab3.ipynb
```

to electronically submit your file `lab3.ipynb`.

Shutdown the jupyter notebook and logout the workstation.

Hand in your lab report before your leave.