CSCI-B490: Quantum Programming
Homework 2
Due: Tues, Jan 28

This assignment will be submitted to Canvas under Homework 2. You'll upload a pdf document named `hw2.pdf` containing your written responses. You'll also upload your qasm files. For the sanity of the graders, name your qasm files using the following format: `hw2-exE-P.qasm` where `E` is the exercise number and `P` is the part number.

It is not currently possible to define a circuit in Open QASM which is parametrized by the number of input bits. For example, we know that for all $n > 0$, there exists an n-bit circuit which negates every input. We might call such a circuit $\mathtt{x}(n)$. For now, we will define such things recursively in mathematical/Python-style pseudocode:

```
def x(n):
  if n = 1:
    return single-bit circuit with an x gate
  if n > 1:
    c = x(n-1)
    add a new register q to c
    apply x to q
    return c
```

Exercise 1. (*6 points*) All circuits must be constructed from gates in $\{\mathtt{x}, \mathtt{cx}, \mathtt{ccx}\}$.

1. Design a reversible circuit which implements inclusive or (use two `ccx` gates).
2. Design a 2-bit reversible circuit which implements swap.
3. Design a reversible circuit which implements the Fredkin gate.
4. Explain how to implement fanout with `ccx`.
5. Design a reversible circuit which computes the 5-bit majority function.

Exercise 2. (*7 points*) All circuits must be constructed from gates in $\{\mathtt{x}, \mathtt{cx}, \mathtt{ccx}\}$.

1. Design `cccx` (i.e., `toffoli(4)`). Try to arrange the circuit so that in the top half of the truth table, only the last two rows appear swapped.
2. Design `cccccx` (i.e., `toffoli(6)`).
3. Show how to construct $\mathtt{toffoli}(n)$ for all $n > 0$.

Exercise 3. (*13 points*) All circuits must be constructed from gates in $\{\texttt{x}, \texttt{cx}, \texttt{ccx}\}$.

1. Design a 2-bit reversible circuit which swaps `00` and `01` and leaves all other inputs fixed.
2. Design a 2-bit reversible circuit which swaps `00` and `11` and leaves all other inputs fixed.
3. Design a 3-bit reversible circuit which swaps `000` and `001` and leaves all other inputs fixed.
4. Design a 3-bit reversible circuit which swaps `000` and `011` and leaves all other inputs fixed.
5. Design a 3-bit reversible circuit which swaps `000` and `111` and leaves all other inputs fixed.
6. Show how to construct, for all n, and for any two n-bit strings s_1 and s_2, a reversible circuit which swaps s_1 and s_2 and leaves all other inputs fixed.

Exercise 4. (*7 points*)

1. Design a 5-bit ripple adder (cf. Lecture on 1/23)
2. Show how to construct for all $n > 0$, $\texttt{adder}(n)$.