CMPT 409/981: Quantum Circuits and Compilation
Assignment 2

Due October 28th at the start of class
on paper or by email to the instructor

In this assignment we will investigate efficient implementation of the *quantum Fourier transform* over Clifford+T via an alternative to *gate approximation* called *catalytic embedding*.

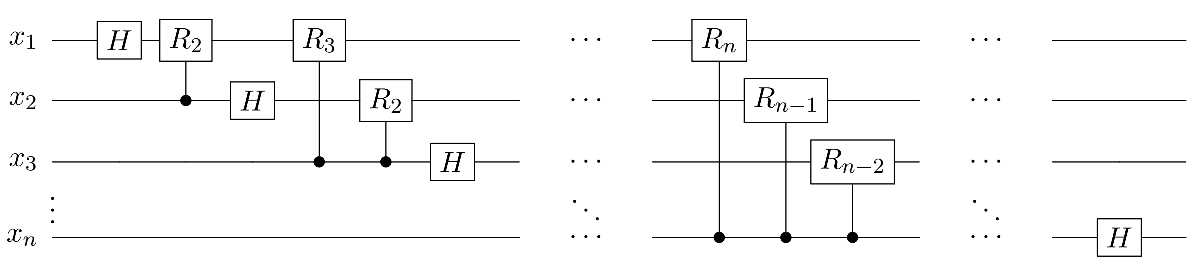
The quantum Fourier transform is a crucial building block of many quantum algorithms. It can be defined as the unitary transformation on n qubits

$$QFT_n : |\vec{x}\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{\vec{y}\in\{0,1\}^n} e^{\frac{2\pi i}{2^n}\vec{x}\vec{y}}|\vec{y}\rangle$$

where $\vec{x}\vec{y}$ is interpreted as integer multiplication, which can be expanded explicitly as

$$\vec{x}\vec{y} = (2^{n-1}x_1 + \cdots + 2x_{n-1} + x_n)(2^{n-1}y_1 + \cdots + 2y_{n-1} + y_n).$$

The QFT can be implement via a circuit over H gates and controlled $R_k := \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{bmatrix}$ gates with quadratic gate complexity. In particular, the circuit can be written as



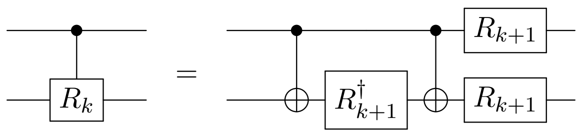
or in pseudo-code,

1. for i from 1 to n:
   (a) For j from i to n
      i. Apply a controlled-R_{j-i+1} to qubits j and i
   (b) Apply H to qubit i

Sinse the QFT is important in many algorithms suited to Fault-tolerant quantum computation, including Shor's algorithm, we wish to gain an understanding of how expensive it is to compute. We will assume in this assignment that our fault-tolerant quantum computer can implement gates from the Clifford+T gate set, that is

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad T = \begin{bmatrix} 1 & 0 \\ 0 & \omega := e^{i\pi/4} \end{bmatrix}, \qquad CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Note that the controlled R_k gate is symmetric in the target and control, and that



Question 1 [3 points]: Concrete resource estimates

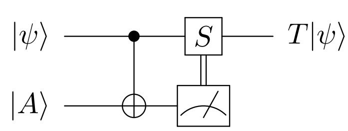
While R_k is not in Clifford+T for any $k \geq 4$, we can implement the QFT over Clifford+T by *approximating* R_k gates. The Ross-Selinger algorithm produces ϵ-approximations of diagonal gates (e.g. R_k) over Clifford+T with (approximately) $3\log_2(1/\epsilon)$ T gates. How many T gates would be used to implement the QFT on 32 qubits to **overall precision** 10^{-7} if the Ross-Selinger algorithm is used for single-qubit gate approximations? This is called a **resource estimate**, and is important in quantifying how much quantum advantage there is for real-world problems, and at what point we may start to see real advantage from quantum computers.

Do NOT simply give the big-O complexity — the (leading) constants are the important factor here!

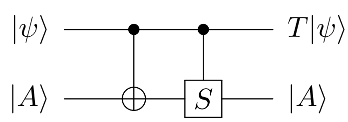
Question 2 [10 points]: Catalytic QFT

The previous resource estimate seems a little high. We'll now develop a technique which can get our resource counts down to something more manageable.

Recall the T gate teleportation circuit from class using the resource state $|A\rangle = TH|0\rangle$:



1. Verify that if measurement is deferred and the classically-controlled S gate is replaced with a quantum controlled S, then the final state is $(T|\psi\rangle) \otimes |A\rangle$. That is,



   This is an example of a more general technique called *catalytic embedding*, whereby a unitary over a ring extension $R[\alpha]$ is embedded into a unitary over the base ring R together with a resource state. Likewise, catalytic embedding generalizes the classic representation of $\mathbb{C}$ using $\mathbb{R}$-valued matrices.

2. Give a circuit using a single $|A\rangle$ state to perform 2 T gates using only $CNOT$ and CS gates. Note that with gate teleportation, a single $|A\rangle$ state can only be used to perform one T gate, as it is destroyed at the end.

   Note: this doesn't help us a whole lot at this point, because the controlled-S gate is non-Clifford, and in fact requires 3 T gates to implement.

3. Show that this construction generalizes to all R_k gates with the resource state $|A_k\rangle = R_k H|0\rangle = \begin{bmatrix} 1 \\ e^{2\pi i/2^k} \end{bmatrix}$.

   Explicitly, show that the R_k gate can be constructed from $|A_k\rangle$ states, $CNOT$ and controlled-R_{k-1} gates.

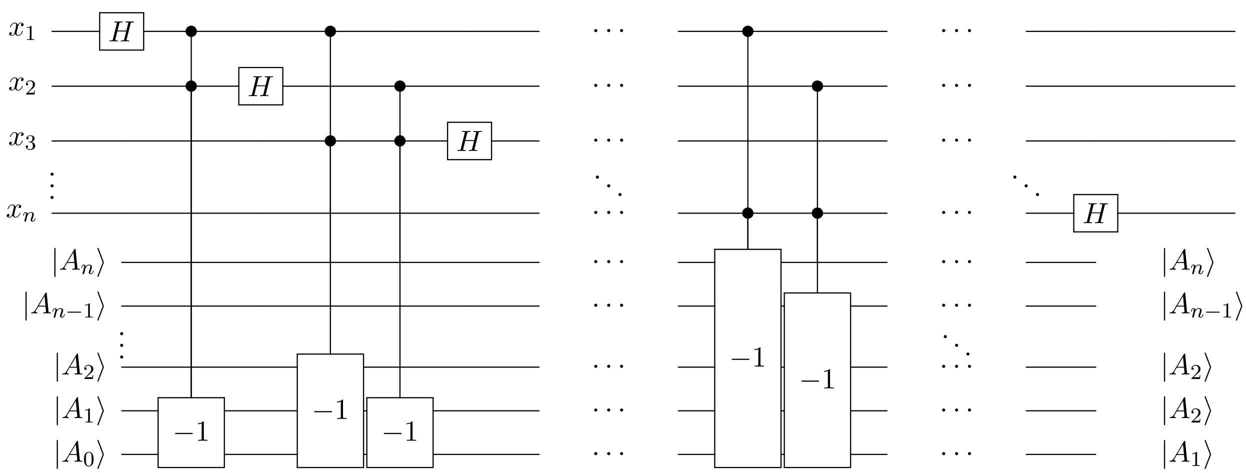
4. Does the T-gate teleportation circuit also generalize to allow the teleportation of R_k gates given an $|A_k\rangle$ state and fault-tolerant $CNOT, CR_{k-1}$ gates?

5. Try to implement a $R_4 := \sqrt{T}$ gate over Clifford+T using the construction from the previous question and the fact that $CT = (\sqrt{T} \otimes \sqrt{T})CNOT(I \otimes \sqrt{T}^\dagger)CNOT$. What is the problem?

6. Extend your R_k construction to a construction of the multiply-controlled R_k gate using $|A_k\rangle$ **states, multiply-controlled Toffolis and multiply-controlled R_{k-1} gates**.

7. Now use your construction recursively to give an explicit circuit implementing a controlled-R_4 (controlled-$\sqrt{T}$) gate using **only multiply-controlled Toffoli gates and ancillary resource states $|A_k\rangle$ for any k.** It may help to note that $R_1 := Z, R_0 := I$.

At this point we have shown that the QFT can be implemented as follows, where -1 denotes the *decrement* function on a binary register, which we will explore in the next question...



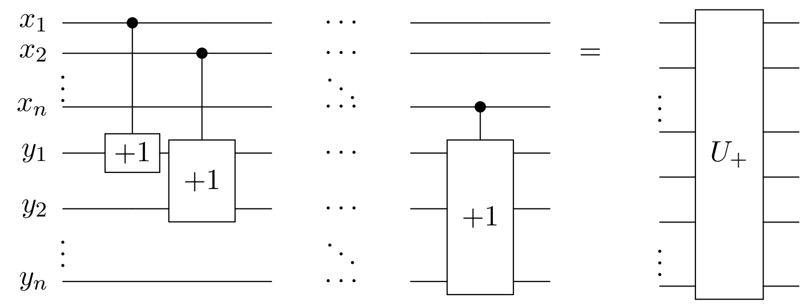
Question 3 [8 points]: Reversible arithmetic

We define the *modular increment function* as

$$(+1) : |\vec{x}\rangle \mapsto |\vec{x} + 1 \mod 2^n\rangle$$

where $\vec{x}$ is taken as a big-endian integer, i.e. x_1 is the high-order bit. That is, $(+1)$ adds 1 to a positive integer represented in big endian binary as an n-bit string $\vec{x} \in \{0,1\}^n$ ignoring overflow (i.e. $\vec{x}+1 \mod 2^n$). Note that the inverse of a modular increment is a modular decrement.

1. Verify that a cascade of controlled-increment circuits corresponds to binary addition. That is, let $U_+|\vec{x}\rangle|\vec{y}\rangle = |\vec{x}\rangle|\vec{y}+\vec{x} \mod 2^n\rangle$. Then



2. Design an **efficient** reversible circuit performing modular addition

$$(U_+) : |\vec{x}\rangle|\vec{y}\rangle \mapsto |\vec{x}\rangle|\vec{y}+\vec{x} \mod 2^n\rangle$$

   To be efficient, your implementation should use a **linear** number (i.e. $O(n)$) of Toffoli gates and as many clean ancillas and X and $CNOT$ gates as needed. The ancillas **must be returned to their initial state**. Note that since this is an **in-place** addition (i.e. an input register is modified directly), this means you cannot uncompute your ancillary states with the Bennett trick. Give the number of Toffoli gates your circuit uses as a function of n.

   You may describe your circuit via pseudo-code rather than a circuit diagram if you prefer.

   Hint: try to add two binary numbers by hand using long addition and see if you can replicate this process in a reversible circuit.

3. How would you go about adding a control to this entire circuit to make a *controlled* modular adder? How many extra Toffoli gates does this cost? Can you do it using only roughly $2n$ extra Toffoli gates? (Hint: compute all the carry bits first)

Question 4 [4 points]: Resource estimate redux

Questions 2 and 3 together give an implementation of the QFT on n qubits using n resource states and n controlled modular adders, or $O(n^2)$ Toffoli gates. Recalling that the resource states may be implemented as $|A_k\rangle := R_k H|0\rangle$, we can obtain a fault-tolerant circuit over Clifford+T gates by approximating an R_k gate for each resource state, and expanding the Toffoli gates over Clifford+T.

Calculate the number of T gates this implementation of the QFT on 32 qubits uses for $\epsilon = 10^{-7}$ and compare to the estimate you calculated in question 1. Can you see any further benefit from this implementation if *multiple* QFTs are needed within a single algorithm?