2.2 Fourier Transform

Fourier transform turns out to be the key element of quantum algorithms for a class of problems including factoring. It may be seen as a generalization of the Hadamard transform, which has proved essential in the quantum algorithms that we have seen.

2.2.1 Discrete Fourier Transform (DFT)

The DFT of a vector of N complex numbers $x = (x_1, \ldots, x_N)^T$ is another vector $y = (y_1, \ldots, y_N)^T$ of complex numbers such that

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi ijk/N}.$$

The inverse DFT (IDFT) is the transformation

$$x_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k e^{-2\pi ijk/N}.$$

Exercise: Show that DFT is a unitary transformation.
Example: The matrix for DFT for N=8 is given by

$$W_8 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\ 1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^1 & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\ 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\ 1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega & \omega^6 & \omega^3 \\ 1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\ 1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega \end{bmatrix}$$

where $\omega = e^{2\pi i/8} = \sqrt{i}$.

2.2.2 Quantum Fourier Transform (QFT)

The QFT of a set of basis vectors $|0\rangle, \ldots, |N-1\rangle$ is the unitary operator that transforms the basis vectors as follows

$$|j\rangle \to \frac{1}{N} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle.$$

By superposition, the QFT of an arbitrary state is

$$\sum_{j=0}^{N-1} x_j |j\rangle \to \sum_{k=0}^{N-1} y_k |k\rangle,$$

where $\{y_k\}$ is the DFT of $\{x_j\}$.

We may define an inverse QFT analogously to the IDFT.

The QFT is a unitary operation and hence can be implemented using quantum gates. There is a very efficient implementation for the case $N = 2^n$ and the quantum system consists of n qubits. This construction is based on the fact that the QFT of the basis vector $|j\rangle = |j_1, \ldots, j_n\rangle$ has the following alternative form

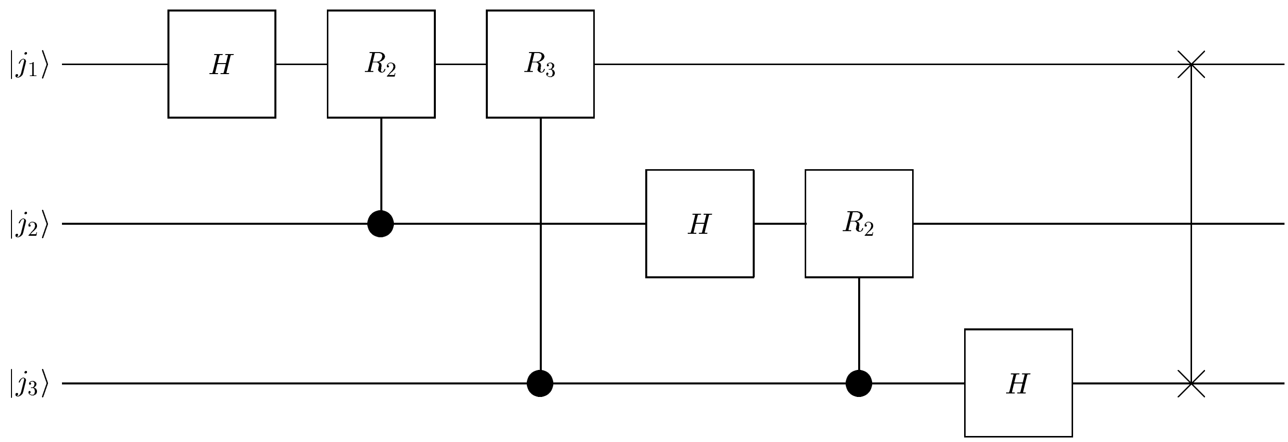
$$\frac{(|0\rangle + e^{2\pi i 0.j_n}|1\rangle)(|0\rangle + e^{2\pi i 0.j_n j_{n-1}}|1\rangle)\cdots(|0\rangle + e^{2\pi i 0.j_1 \cdots j_n}|1\rangle)}{2^{n/2}}. \tag{2.1}$$

Proof. See Nielsen and Chuang p. 218.

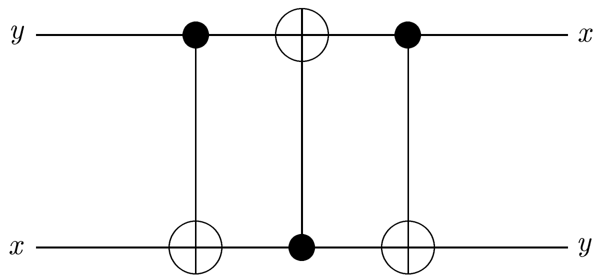
This form is unentangled and it can be implemented very easily using the Hadamard, CNOT and the controlled $C(R_k)$ gates where

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix}.$$

As an example, the QFT for a 3-qubit basis state is implemented by the following circuit.



The final stage in the above diagram swaps the first and the third qubits. Without this the qubits at the output of the circuit would be in reverse order compared to (2.1). Qubit swap operation can be implemented using three CNOT gates as shown below.



2.2.3 Complexity of Implementing QFT

The QFT circuit in general requires

- n Hadamard gates

- $\binom{n}{2}$ controlled R_k gates
- At most $n/2$ swaps each requiring 3 CNOT gates

for a total of $O(n^2)$ gates.

This is better than classical complexity of DFT which has circuit complexity $N\log(N)$ where $N = 2^n$ is the dimension. This does not directly mean however that quantum circuits can be used to compute DFT more efficiently. For a discussion of this point, see Nielsen and Chuang p. 220.

Since the R_k gates require exponentially increasing phase precision, they cannot be implemented using polynomial number of discrete gates. This does not cause a major problem because it can be shown that if each R_k gate is approximated to precision $1/p(n)$ for some polynomial $p(n)$, the error in the overall QFT operation scales as $n^2/p(n)$.

In conclusion, QFT can be implemented using a number of elementary gates (H, T, and CNOT) bounded polynomially in n, the number of qubits.