Week 6 Report

In the past week, I have worked on some important quantum algorithms and subroutines, as well as, understanding quantum gates and their implementation IBM QX.

Quantum Fourier Transform

$$QFT_N = \frac{1}{\sqrt{N}} * \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(N-1)} \\ \vdots & \ddots & & & \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)^2} \end{pmatrix} \tag{1}$$

where $\omega = e^{\frac{2\pi i}{N}} = \cos\frac{2\pi}{N} + i\sin\frac{2\pi}{N}$ and $N = 2^n$

ω is N^{th} root of unity. For example, N = 12

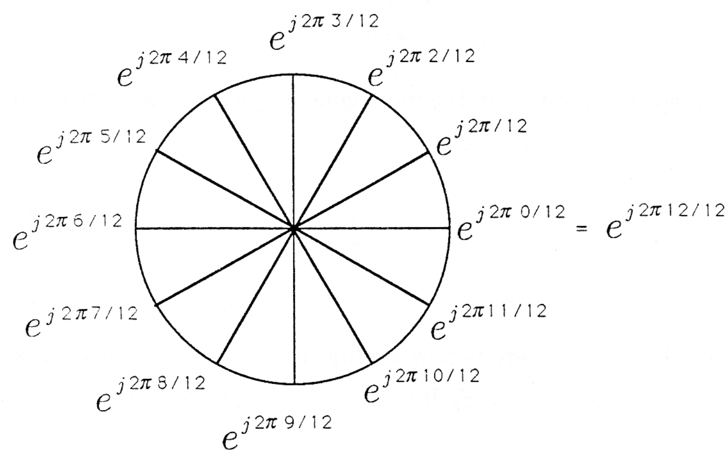

Figure 1: Roots of Unity

$$\frac{1}{\sqrt{N}} * \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(N-1)} \\ \vdots & \ddots & & & \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)^2} \end{pmatrix} * \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{N-1} \end{pmatrix} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \alpha_{N-1} \end{pmatrix} \tag{2}$$

Definition:

$$\sum_{j=0}^{N-1} \alpha_j \left|j\right\rangle \xrightarrow{\mathrm{QFT}_N} \sum_{k=0}^{N-1} \beta_k \left|k\right\rangle$$

where: $\beta_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \alpha_j \omega^{jk} = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \alpha_j e^{\frac{2\pi i j k}{N}}$

In case that x is a basis state, we have that:

$$|x\rangle \xrightarrow{\text{QFT}_N} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{xk} |k\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i j k}{N}} |k\rangle$$

More generally, we can use the binary representation of $j = j_1 j_2 \dots j_n$, or more formally $j = j_1 2^{n-1} + j_2 2^{n-2} + \dots + j_n 2^0$. Now, we have that:

$$\begin{aligned}
|j\rangle &\xrightarrow{\text{QFT}} \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{\frac{2\pi i j k}{2^n}} |k\rangle \\
&= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi i j (\sum_{l=1}^{n} k_l 2^{-l})} |k_1 \dots k_n\rangle \\
&= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \\
&= \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \Big[\sum_{k_l=0}^{1} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \Big] \\
&= \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} \Big[|0\rangle + e^{2\pi i j 2^{-l}} |1\rangle \Big] \\
&= \frac{(|0\rangle + e^{2\pi i 0.j_n} |1\rangle) \otimes (|0\rangle + e^{2\pi i 0.j_{n-1} j_n} |1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n} |1\rangle)}{2^{n/2}}
\end{aligned} \tag{3}$$

The product representation (3) makes it easy to derive an efficient circuit for the quantum Fourier transform.

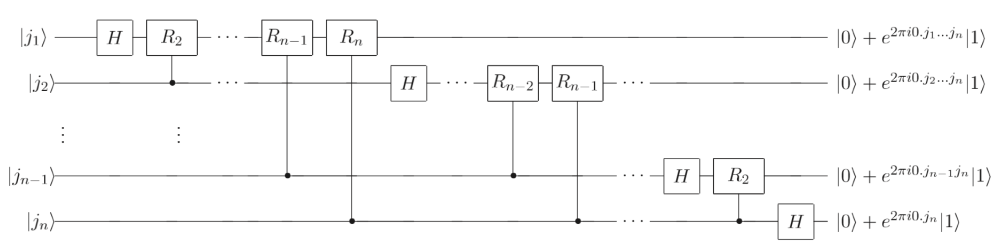

Figure 2: Quantum Fourier Transform circuit

where R_k denotes the unitary transformation:

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix}$$

R_k leaves the basis state $|0\rangle$ unchanged and map $|1\rangle$ to $e^{\frac{2\pi i}{2^k}}|1\rangle$
Input state $|j_1 j_2 \dots j_n\rangle$

- Apply the hadamard gate to the first bit produces the state:

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.j_1}|1\rangle)|j_2 \dots j_n\rangle$$

- Apply the controlled R_2 gate produces the state:

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.j_1 j_2}|1\rangle)|j_2 \dots j_n\rangle$$

- We continue applying controlled - R_3, R_4 through R_n gates, each of which adds an extra bit the the phase of the co-efficient of the first $|1\rangle$. At the end of the procedure, we have the state

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n}|1\rangle)|j_2 \dots j_n\rangle$$

- We perform a similar procedure on the second qubit. The Hadamard gate put us in the state:

$$\frac{1}{2}(|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n}|1\rangle)(|0\rangle + e^{2\pi 0.j_2}|1\rangle)|j_3 \dots j_n\rangle$$

- The controlled - R_2 through R_{n-1} gates yields the state:

$$\frac{1}{2}(|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n}|1\rangle)(|0\rangle + e^{2\pi 0.j_2 \dots j_n}|1\rangle)|j_3 \dots j_n\rangle$$

- We continue in this fashion for each qubit, giving a final state:

$$\frac{(|0\rangle + e^{2\pi i 0.j_n}|1\rangle)(|0\rangle + e^{2\pi i 0.j_{n-1} j_n}|1\rangle)\dots(|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n}|1\rangle)}{2^{n/2}}$$

Computational Cost for QFT

- For the first qubit, we use a Hadamard gate and n-1 controlled - R_k gates
- For the second qubit, we use a Hadamard gate and n-2 controlled - R_k gates
- Hence, in total, we use $n + (n-1) + (n-2) + \dots + 1 = \frac{n(n-1)}{2}$ gates
- Cost of QFT is $O(n^2)$
- Classical FFT costs $O(n2^n)$

Quantum Phase Estimation

The problem that Quantum Phase Estimation is trying to solve is that: Suppose a unitary operator U has an eigenvector $|u\rangle$ with eigenvalue $e^{2\pi i\varphi}$, where $0 < \varphi < 1$ and φ is unknown, we would like to know what is the eigenvalues corresponding to $|u\rangle$. Hence, the output of the algorithm would be $\widetilde{\varphi}$ approximation of φ. To perform the estimation we assume that we have available black boxes (oracles) capable of preparing the state $|u\rangle$ and performing the controlled - U^{2^j} operation (j non negative). The use of black boxes indicates that the phase estimation procedure is not a complete quantum algorithm in its own right. Rather, phase estimation is a kind of 'subroutine' that, when combined with other subroutines, can be used to perform interesting computational tasks.

Let's assume that φ can be expressed in exactly n - bit binary fraction (we will come to the case when φ is not later). So we have that:

$$\varphi = 0.x_1x_2\ldots x_n = \sum_{i=1}^{n} \frac{x_i}{2^i}$$

For k $\in \{0, 1, \ldots, n-1\}$ we have:

$$2^k\varphi = x_1x_2\ldots x_k.x_{k+1}\ldots x_n$$

$$\begin{aligned}
e^{2\pi i 2^k\varphi} &= e^{2\pi i(x_1x_2\ldots x_k.x_{k+1}\ldots x_n})\\
&= e^{2\pi i(x_1x_2\ldots x_k + 0.x_{k+1}\ldots x_n)}\\
&= e^{2\pi i(x_1x_2\ldots x_k)} * e^{2\pi i(0.x_{k+1}\ldots x_n)}\\
&= e^{2\pi i(0.x_{k+1}\ldots x_n)}
\end{aligned} \tag{4}$$

Know that: $U|u\rangle = e^{2\pi i\varphi}|u\rangle$ so $U^{2^j}|u\rangle = e^{2\pi 2^j\varphi}|u\rangle$. Thus:

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|u\rangle \xrightarrow{\mathrm{U}^{2^j}} \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi 2^j\varphi}|1\rangle)|u\rangle$$

Therefore, after applying the Hadamard gate and the controlled - $U^{2^{n-1}}$ on the first qubit, we have:

$$\frac{|0\rangle + e^{2\pi i 2^{n-1}\varphi}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + e^{2\pi i 0.x_n}|1\rangle}{\sqrt{2}}$$

We obtain this result from the previous derivation. Repeat the same procedure for the second qubit, we obtain:

$$\frac{|0\rangle + e^{2\pi i 2^{n-2}\varphi}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + e^{2\pi i 0.x_{n-1}x_n}|1\rangle}{\sqrt{2}}$$

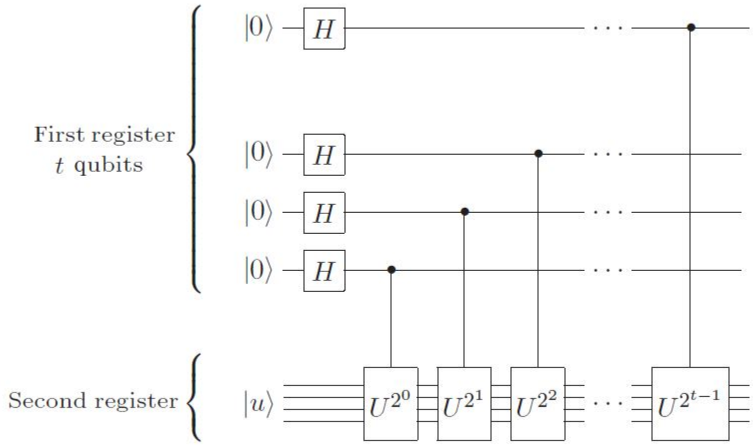

Figure 3: Quantum Phase Estimation circuit

Therefore, we evaluate the output of the previous circuit:

$$\begin{aligned}&\frac{|0\rangle + e^{2\pi i 2^{n-1}\varphi}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i 2^{n-2}\varphi}|1\rangle}{\sqrt{2}} \otimes \dots \\ &\otimes \frac{|0\rangle + e^{2\pi i 2^{1}\varphi}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i 2^{0}\varphi}|1\rangle}{\sqrt{2}} \\ &= \frac{|0\rangle + e^{2\pi i 0.x_n}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i 0.x_{n-1}x_n}|1\rangle}{\sqrt{2}} \otimes \dots \\ &\otimes \frac{|0\rangle + e^{2\pi i 0.x_1 x_2 \dots x_n}|1\rangle}{\sqrt{2}}\end{aligned} \tag{5}$$

This is exactly the Quantum Fourier Transform applied on $(|x_n\rangle \otimes |x_n n-1\rangle \otimes \dots \otimes |x_1\rangle)$. To get the values of $x_1, x_2, \dots, x_n$, apply the inverse of Quantum Fourier Transform.

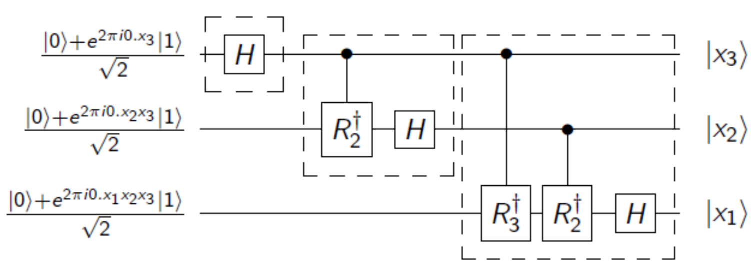

Figure 4: Inverse QFT for 3 qubits

where:

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix}$$

Summary

- We use the Hadamard gate and controlled - U^{2^k} to prepare the state:
$$\frac{|0\rangle + e^{2\pi i 0.x_{k+1}x_{k+2}\ldots x_n}|1\rangle}{\sqrt{2}}$$
- Using the previously determined bits $x_{k+2}, x_{k+3}, \ldots, x_n$ we change this state to:
$$\frac{|0\rangle + e^{2\pi i 0.x_{k+1}}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + (-1)^{x_{k+1}}|1\rangle}{\sqrt{2}}$$
- Apply the Hadamard gate to obtain $|x_{k+1}\rangle$
- Intuition: The controlled phase shifts enable us to reduce the problem of determining each bit to distinguishing between $|+\rangle$ and $|-\rangle$

When φ is abitrary and can be not an exact n-bit binary fraction.

Lemma:

Let $x = \sum_{i=1}^{n} x_i 2^{n-i}$ and $\varphi_x = 0.x_1x_2\ldots x_n = \frac{x}{2^n}$ be the corresponding bit fraction. For x be such that:

$$\frac{x}{2^n} \leq \varphi \leq \frac{x+1}{2^n}$$

Then, the probability of return one of the two closest bit fraction is at least $\frac{8}{\pi^2}$ with error less than $\frac{1}{2^n}$, i.e., $Pr(|\varphi_x - \varphi| < \frac{1}{2^n}) \geq \frac{8}{\pi^2}$

Others

Apart from what is represented on Quantum Fourier Transform and Quantum Phase Estimation, I have gone over the algorithm for finding period with Quantum Phase Estimation via Quantum Factoring Period Finding video, as well as, the Shor Algorithms via Quantum Shor's Factoring Algorithm video. But I don't have enough of time to sumarize everything for the presentation. Other than learning quantum algorithms, I also spend time working with IBM QX: basically, building a circuit for quantum SVM is a hard problem because qSVM consists of so many components and so many complex gates, which are not available on IBM QX. Hence, what I was doing is just trying to understand quantum gates and trying to implement more complex quantum gates and subroutines needed to implement QSVM on IBM QX.

References

[1] Michael Nielsen, Isaac L. Chuang. *Quantum Computing and Quantum Information*

[2] Umesh Vazirani. Quantum Fourier Transform