ECE 592-100 – Signal Processing Tour of Quantum Computing

Final Exam – Spring 2023

May 1, 2023

Please remember to justify your answers carefully.

Last name: ____________________ First name: ____________________

Question 1 (Two-qubit quantum circuit.)
Consider a simple quantum circuit with two inputs,

$$|\phi_1\rangle = a_1 |0\rangle + b_1 |1\rangle ,$$

$$|\phi_2\rangle = a_2 |0\rangle + b_2 |1\rangle .$$

In this question, you will pass these qubits through two controlled NOT (CNOT) gates. You will analyze how the state vector of the two-qubit system evolves, and specify your state vectors as length-4 column vectors.

(a) The initial state is $|\psi_1\rangle = |\phi_1\rangle \otimes |\phi_2\rangle$. Compute $|\psi_1\rangle$. (Recall that you must provide a length-4 vector.)

Solution.

$$\begin{aligned} |\psi_1\rangle &= (a_1 |0\rangle + b_1 |1\rangle) \oplus (a_2 |0\rangle + b_2 |1\rangle) \\ &= a_1 a_2 |00\rangle + a_1 b_2 |01\rangle + b_1 a_2 |10\rangle + b_1 b_2 |11\rangle \\ &= \begin{pmatrix} a_1 a_2 \\ a_1 b_2 \\ b_1 a_2 \\ b_1 b_2 \end{pmatrix} . \end{aligned}$$

(b) The next state, $|\psi_2\rangle$, is obtained by applying a CNOT, where the first qubit ($|\phi_1\rangle$ in our case) is the control and the second qubit ($|\phi_2\rangle$) is the target. Compute $|\psi_2\rangle$.

Solution. We swap the 10 and 11 states,

$$|\psi_2\rangle = \begin{pmatrix} a_1 a_2 \\ a_1 b_2 \\ b_1 b_2 \\ b_1 a_2 \end{pmatrix} .$$

(c) The last state, $|\psi_3\rangle$ is obtained by applying a CNOT in an opposite way, meaning that the first qubit is the target and the second qubit is the control. Compute $|\psi_3\rangle$.

Solution. We swap the 01 and 11 states,

$$|\psi_2\rangle = \begin{pmatrix} a_1a_2 \\ b_1a_2 \\ b_1b_2 \\ a_1b_2 \end{pmatrix}.$$

Question 2 (Implementing Toffoli gates.)
A Toffoli gate can be thought of as a CNOT with two control qubits. Denoting the control qubits by c_1 and c_2 and the target qubit by t, a Toffolio gate takes the classical triple (c_1, c_2, t) to $(c_1, c_2, t \oplus c_1c_2)$.
More generally, a quantum gate with two control qubits that applies the single qubit operation U to t when c_1 and c_2 are both 1 is denoted by $C^2(U)$. That is,

$$C^2(U) |c_1c_2t\rangle = |c_1c_2\rangle U^{c_1c_2} |t\rangle .$$

It is well known in the quantum computing literature (details in Exercise 4.21 and Figure 4.8 in Nielsen & Chuang) that a controlled-U gate can be implemented using several CNOT, V, and $V^\dagger$ gates, where $V^2 = U$. Nielsen & Chaung describe the V matrix needed to implement the Toffoli gate, for which $U = X$. Compute V such that $V^2 = Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$.

Solution. The question itself is a somewhat long motivation for the actual problem that needs to be addressed, which is to find V such that $V^2 = Z$. This solution can be identified in multiple ways, for example through an eigen decomposition, or just realizing that we can take the square roots of entries along the diagonal. In any event, the answer is

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}.$$

It can be verified that $V^2 = Z$.

Question 3 (Fast Hadamard transform.)
An important result in the history of signal processing was the fast Fourier transform (FFT), which computes the discrete Fourier transform (DFT) of a length-N vector, $|\psi\rangle$, in runtime proportional to $N \log_2(N) = Nn$, instead of N^2 for the DFT. In this question, you will derive an analogous fast Hadamard transform (FHT) that computes the discrete Hadamard transform (DHT) of $|\psi\rangle$, and show a similar computational improvement.
(a) Consider H_n, a Hadamard matrix of size $N \times N$. (Recall that one of our homework problems considered H_3 of size 8×8.) Suppose that H_n is stored in memory. Explain why computing $H_n |\psi\rangle$ using conventional matrix vector multiplication will require computation

proportional to N^2. (In words, the DHT requires N^2 runtime, similar to the DFT.)

Solution. A matrix vector product involves taking each entry of the matrix, multiplying it by the corresponding entry of the vector, and summing over pairs of products. With N^2 entries in the matrix, there will be N^2 memory accesses, N^2 multiplications, and $N(N-1)$ additions. Overall, the computation is proportional to N^2.

(b) The Hadamard matrix can be defined recursively,

$$H_{n+1} = \frac{1}{\sqrt{2}} \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix},$$

where

$$H_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Given a vector $|\psi\rangle$ of length $N = 2^n$, describe a divide and conquer style algorithm that applies H_{n-1} recursively to two halves of $|\psi\rangle$, and then combines the result. Describe the basis case and recursive part of this algorithm. (It may be simplest to describe your algorithm using some combination of pseudocode and mathematical notation.)

Solution. Consider a function or routine $\text{FHT}(n, |\psi\rangle)$. Its first input is n, the problem size. Its second input is $|\psi\rangle$, a vector that we will multiply by H_n. If $n = 1$, then we apply the basis case and output $H_1 |\psi\rangle$. Else $n > 1$, and we partition the matrix vector product into parts. Owing to the block structure of H_n, we partition $H_n |\psi\rangle$ into four subproblems where H_{n-1} multiplies either the first or second half of $|\psi\rangle$, which will be denoted by $|\psi_1\rangle$ and $|\psi_2\rangle$, respectively. Writing in a somewhat informal manner, the following steps are performed,

$$\begin{aligned} term_1 &= \text{FHT}(n-1, |\psi_1\rangle) \\ term_2 &= \text{FHT}(n-1, |\psi_2\rangle) \\ \text{Output} & \quad \frac{1}{\sqrt{2}} \begin{bmatrix} term_1 + term_2 \\ term_1 - term_2 \end{bmatrix}. \end{aligned}$$

(c) Explain why applying this algorithm will require computation proportional to $N \log_2(N)$. (In words, the FHT requires $N \log_2(N)$ runtime, similar to the FFT.)

Solution. This is a divide and conquer algorithm. The step that combines the solutions of the recursive runs of the function is the last one where we form $term_1 + term_2$ and $term_1 - term_2$ into a matrix. Because these terms are each vectors of length $N/2$, performing basic arithmetic requires computation proportional to N. We have seen that similar divide and conquer algorithms are $N \log_2(N)$.

Question 4 (Quantum phase estimation.)
In quantum phase estimation (QPE), we apply a controlled-U^n matrix to one of its eigenvalues, $|u\rangle$. Suppose that

$$U = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}.$$

(a) Show that $|u\rangle = |1\rangle$ is an eigen-vector of U. What is the corresponding eigen-value?

Solution. $U|1\rangle = \begin{pmatrix}0 & i\end{pmatrix} = i|1\rangle$. Therefore, the eigen-value is i.

(b) Suppose that you run QPE. What do you expect to sample at the QPE algorithm's output? (Hint: $i = e^{i\pi/2}$.)

Solution. The eigen-value is $e^{i2\pi 0.25}$, meaning that the phase ϕ in QPE is 0.25. If QPE has 2 qubits in its first register, it will output 01, which is the binary fraction that corresponds to the decimal 0.25. With 4 qubits, we will get 0100, and so on.

Question 5 (Projects.)

Briefly describe one of the other projects that were presented. What did you learn from either the presentation in class or the report?

Solution. Any answer that describes any of the other projects is fine.