Notes on Feynman Quantum Circuits for Seminar, Fifth Draft

April 6, 2019

Chapter 6: Quantum Mechanical Computers from "Feynman Lectures on Computation"

Aim: The aim of this talk is to discuss whether it is feasible to perform computation using a quantum system.

These sorts of discussions initially arose because Feynman and others (Bennett, Fredkin, Toffoli) were analyzing whether there were physical limitations to computation, such as whether there was a minimum free energy dissipation associated with performing a "unit" of computation. Bennett found that, provided a reversbile computation is run slowly enough, there is essentially no energy dissipation (if the reading or clearing of output is not counted as part of the computation). It suggested that the only theoretical limitations for completing computation were those of the system performing the computation, i.e. those due to quantum mechanics.

What we will be considering in this talk then, is if we can perform, and how we could go about performing, a classical computation on a quantum system. I want to emphasize "classical" here because we will not really be going into any of the specific attributes of quantum computing. We will also essentially disregard any aspects related to how one would input an initial state or measure an output state and any aspects surrounding efficiency or practicality of implementation. We also do not discuss any errors that may be occur; we are assuming a simplified and idealized system.

Let's make a few comments about what system we will be considering. Throughout this talk our bits are going to be atoms, where an atom is in the $|0\rangle$ state if there is no associated electron, and is in the $|1\rangle$ state if there is. I say "bits" to further emphasise that we are considering classical computation. Of course, there are many other systems that can be used to implement such computation, but this system obeys Fermi-Dirac statistics which simplifies the discussion somewhat.

We know from last week that a quantum system evolves according to the Schrödinger equation

$$i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle = H|\psi(t)\rangle$$

which has solution

$$|\psi(t)\rangle = U(t, t_0)|\psi(t_0)\rangle$$
$$U(t, t_0) = e^{-\frac{i}{\hbar}(t-t_0)H}$$

where H is the Hamiltonian (i.e. Hermitian/self-adjoint operator) and U is unitary.

Our aim then is write down a Hamiltonian H that is polynomial in some fundamental operators (to be defined shortly) that describes the evolution from an input state to our desired output state that defines our desired computation. In particular, we want our Hamiltonian to include operators that describe "nearest neighbour" interactions, so as to be most appropriate to our physical system.

The fundamental operations that will provide the building blocks for our Hamiltonian are:

$$a = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \text{ and}$$
$$a^* = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

It is important to note the behaviour of these operators acting on the states $|0\rangle$ and $|1\rangle$

$$a \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$a \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$
$$a^* \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$a^* \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

We will also take a second to note a couple of other properties to be referred back to

$$a^*a = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$
$$aa^* = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \text{ and so}$$
$$a^*a + aa^* = I$$

where I denotes the 2×2 identity matrix. The last line defines part of the anticommutation relations that are satisfied by our fermionic system.

Now we recall the content of Will's talk two weeks ago, specifically regarding reversible computation and complete (or universal) gate sets. Will introduced the NOT, CNOT and CCNOT gates, the last of which is

a universal single gate set. We now show that each of these gates can be written in terms of the operators defined above.

The case for NOT is easy

$$\text{NOT} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = a + a^*$$

The $\text{CNOT}_{A;B}$ gate where A is the control and B is the target, has truth table

Input A:	Input B:	Output A:	Output B:
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$

To show that CNOT can be written in terms of a and a^*, we note that from the truth table (or from the matrix representation) that when the input on A is in the $|0\rangle$ state, the gate acts like the identity on the input of B. When the input of A is in the $|1\rangle$ state, then the gate acts like NOT on the input of B. We can write down products of a and a^* that "select" for each of these two cases. For example $a_A^* a_A$, where the subscript denotes the operator acting on the atom A, selects the cases when A is in the $|1\rangle$ state since a takes $|1\rangle$ to $|0\rangle$ and then a^* takes to $|0\rangle$ back to $|1\rangle$. As we have already seen a takes $|0\rangle$ to 0, so $a_A^* a_A$ nullifies the $|0\rangle$ state. Similarly, $a_A a_A^*$ selects the $|0\rangle$ state. Thus we can write the CNOT gate as

$$\begin{aligned} \text{CNOT}_{A;B} &= a_A^* a_A \otimes (\text{NOT}_B) + a_A a_A^* \otimes I_B \\ &= a_A^* a_A \otimes (a_B + a_B^*) + a_A a_A^* \otimes (a_B^* a_B + a_B a_B^*) \end{aligned}$$

where '$\otimes$' denotes the tensor product.

Finally, the truth table for the $\text{CCNOT}_{A,B;C}$ where A, B are the controls and C the target is

Input A:	Input B:	Input C:	Output A:	Output B:	Output C:
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$

We will use a similar "selection" method to the treatment of CNOT above. It is only the last two lines of the truth table where the output of C is different to the input. Thus, for all the cases where at most one of

A or B is in state $|1\rangle$, then the gate acts as the identity on C, and when A and B are both in the $|1\rangle$ state, the gate acts as NOT on C. Thus we can write

$$\begin{aligned}\text{CCNOT}_{A,B;C} &= a_A a_A^* a_B a_B^* + a_A a_A^* a_B^* a_B + a_A^* a_A a_B a_B^* + a_A^* a_A a_B^* a_B(\text{NOT}_C) \\ &= a_A a_A^* a_B a_B^* + a_A a_A^* a_B^* a_B + a_A^* a_A a_B a_B^* + a_A^* a_A a_B^* a_B(a_C + a_C^*)\end{aligned}$$

At this point things are looking good; we've been able to write a complete gate set for reversible computation in terms of these fundamental operators. But that is not the same as being able to write down a Hamiltonian that describes an evolution of our system corresponding to each of these gates. Let us try and do that for a system of 3 atoms and a computation that is simply a CCNOT.

Let us take $H = \text{CCNOT} = a_A a_A^* a_B a_B^* + a_A a_A^* a_B^* a_B + a_A^* a_A a_B a_B^* + a_A^* a_A a_B^* a_B(a_C + a_C^*)$ (will write it as CCNOT for short) and see what happens. Let $|\psi_{IN}\rangle$ be our input state. Then

$$\begin{aligned}|\psi(t)\rangle &= e^{-\frac{i}{\hbar}(t-t_0)H}|\psi_{IN}\rangle \\ &= \sum_{n-0}^{\infty} \frac{1}{n!}(\frac{-it}{\hbar})^n H^n |\psi_{IN}\rangle \\ &= \sum_{m=0} \frac{1}{(2m)!}(\frac{-it}{\hbar})^{2m}(\text{CCNOT})^{2m}|\psi_{IN}\rangle + \sum_{m=0} \frac{1}{(2m+1)!}(\frac{-it}{\hbar})^{2m+1}(\text{CCNOT})^{2m+1}|\psi_{IN}\rangle \\ &= \sum_{m=0} \frac{1}{(2m)!}(\frac{-it}{\hbar})^{2m} I|\psi_{IN}\rangle + \sum_{m=0} \frac{1}{(2m+1)!}(\frac{-it}{\hbar})^{2m+1}\,\text{CCNOT}\,|\psi_{IN}\rangle \\ &= \cos(\frac{t}{\hbar})|\psi_{IN} + i\sin(\frac{t}{\hbar})\,\text{CCNOT}\,|\psi_{IN}\rangle\end{aligned}$$

Again, things are looking pretty good. The second term contains the output we desire: $\text{CCNOT}\,|\psi_{IN}\rangle$. But it occurs in superposition with $|\psi_{IN}\rangle$, and only occurs with probability $|i\sin(\frac{t}{\hbar})|^2$. We only know which state we are in by measuring our computational register, which then breaks the superposition, possibly by projecting into the wrong output. Since the output depends on t, we could try optimising the time at which we measure the system to improve our chance of measuring the correct ouput, but perhaps that is a tough problem to do for a more complicated Hamiltonian. For a more complicated computation, we also encounter the issue of not being sure if the output we measure is the correct one or not. For the simple case above, we can already compute what the output should be, so when we measure the system, we will know if we got the correct output or not. Generally this will not be the case, otherwise we don't need to do the computation in the first place.

We thus introduce, as Feynman does, the program counter and cursor. The program counter will be another set of atoms that we append to our computational register, all but one of which will be in the $|0\rangle$ state. The atom that is in the $|1\rangle$ state represents the cursor, which will move along the atoms of the program counter to keep track of where in the computation the system is.

How do we use this new set of atoms? Suppose we have a computation $U = U_k U_{k-1} ... U_1$ where the U_i represent gates from the universal gate set being used. The program counter in this case consists of $k+1$

atoms, with k of them in the $|1\rangle$ state and one atom acting as the cursor at any given time (we see why we want only one cursor shortly).

To see how the program counter and cursor are used, let us return to our example above. We denote our new input as

$$|\tilde{\psi}_{IN}\rangle = |\psi_{IN}\rangle \otimes |0\rangle \otimes |1\rangle$$

Since $U = \text{CCNOT}$ we have 2 additional atoms in the program counter, and the cursor starts in the 0^{th} spot (again we will see why later). We now write our Hamiltonian as

$$H = a_1^* a_0 \,\text{CCNOT} + a_0^* a_1 \,\text{CCNOT}^*$$

A couple of things to note. Firstly, I have dropped the tensor product notation, as this will get exponentially annoying to write out. Secondly, the subscripts on the a denote the which atom of the program counter they act upon. Thirdly, our new Hamiltonian now has two terms in it rather than just the one, since H needs to be self-adjoint (CCNOT is already self-adjoint so $\text{CCNOT}^* = \text{CCNOT}$).

We will consider $H^n|\hat{\psi}_{IN}\rangle$ for a few values of n to get an idea of what is happening. Start with $n = 1$:

$$\begin{aligned} H|\hat{\psi}_{IN}\rangle &= (a_1^* a_0 \,\text{CCNOT} + a_0^* a_1 \,\text{CCNOT})|\psi_{IN}\rangle \otimes |0\rangle \otimes |1\rangle \\ &= \text{CCNOT}\,|\psi_{IN}\rangle \otimes a_1^*|0\rangle \otimes a_0|1\rangle \\ &= \text{CCNOT}\,|\psi_{IN}\rangle \otimes |1\rangle \otimes |0\rangle \end{aligned}$$

What has happened here? Well, the CCNOT gate is applied to our original input $|\psi_{IN}\rangle$ and the cursor has moved from the 0th spot of the program counter to the 1st spot. Let's look at H^2:

$$\begin{aligned} H^2|\hat{\psi}_{IN}\rangle &= H(\text{CCNOT}\,|\psi_{IN}\rangle \otimes |1\rangle \otimes |0\rangle) \\ &= (a_1^* a_0 \,\text{CCNOT} + a_0^* a_1 \,\text{CCNOT})(\text{CCNOT}\,|\psi_{IN}\rangle \otimes |1\rangle \otimes |0\rangle) \\ &= (\text{CCNOT})(\text{CCNOT})|\psi_{IN}\rangle \otimes a_1|1\rangle \otimes a_0^*|0\rangle \\ &= |\psi_{IN}\rangle \otimes |0\rangle \otimes |1\rangle \end{aligned}$$

This case describes the reversibility of the computation/system. The initial input state has effectively had the identity operating on it, and the cursor has moved from the 0th spot to the first spot and back to the 0th spot.

Following the same process as earlier, we get

$$|\psi_{OUT}\rangle = \cos(\frac{t}{\hbar})|\psi_{IN}\rangle \otimes |0\rangle \otimes |1\rangle + i\sin(\frac{t}{\hbar})\,\text{CCNOT}\,|\psi_{IN}\rangle \otimes |1\rangle \otimes |0\rangle$$

This looks essentially the same as before, but now we have some extra utility in being able to measure the atoms on the program counter to determine whether the computational register, once measured, will give the desired output.

So the Hamiltonian for an arbitrary computation $U = U_k U_{k-1} ... U_1$ will be

$$H = \sum_{i=0}^{k-1} a_{i+1}^* a_i U_{i+1} + \sum_{i=0}^{k-1} a_i^* a_{i+1} U_{i+1}^*$$

Another example may be useful here to illustrate some of what will occur when we look at the general U above. Let us consider $U' = U_3 U_2 U_1$, input state $|\psi_{IN}\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle$ and Hamiltonian

$$H' = a_1^* a_0 U_1 + a_2^* a_1 U_2 + a_3^* a_2 U_3 + a_0^* a_1 U_1^* + a_1^* a_2 U_2^* + a_2^* a_3 U_3^*.$$

Consider the following calculations:

$$\begin{aligned}
H'|\hat{\psi}_{IN}\rangle &= (a_1^* a_0 U_1 + a_2^* a_1 U_2 + a_3^* a_2 U_3 + a_0^* a_1 U_1^* + a_1^* a_2 U_2^* + a_2^* a_3 U_3^*)|\psi_{IN}\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle \\
&= U_1|\psi_{IN}\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \\
H'^2|\hat{\psi}_{IN}\rangle &= H'(U_1|\psi_{IN}\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle) \\
&= U_2 U_1|\psi_{IN}\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \otimes |0\rangle + U_1^* U_1|\psi_{IN}\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle \\
&= U_2 U_1|\psi_{IN}\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \otimes |0\rangle + I|\psi_{IN}\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle \\
&= U_2 U_1|\psi_{IN}\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \otimes |0\rangle + |\hat{\psi}_{IN}\rangle \\
H'^3|\hat{\psi}_{IN}\rangle &= H'(U_2 U_1|\psi_{IN}\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \otimes |0\rangle + U_1^* U_1|\psi_{IN}\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle) \\
&= U_3 U_2 U_1|\psi_{IN}\rangle \otimes |1\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle + U_2^* U_2 U_1|\psi_{IN}\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \\
&+ U_1 U_1^* U_1|\psi_{IN}\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \\
&= U'|\psi_{IN}\rangle \otimes |1\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle + 2U_1|\psi_{IN}\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle
\end{aligned}$$

We see that in this example, our desired output state $U'|\psi_{IN}\rangle$ only occurs when the cursor is in the leftmost spot of the program counter. This will be discussed further shortly.

We take a moment here to discuss why we need exactly one cursor. If there was no cursor, then none of the $a_{i+1}^* a_i$ or $a_i^* a_{i+1}$ would be non-zero, and the program counter adds no extra utility. If there was more than one cursor, then it could occur that the cursor is in the leftmost spot of the program counter, but not all the U_i have operated on our computational system. For example, if our Hamiltonian above acted on $|\psi_{IN}\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \otimes |1\rangle$, the result would be a superposition:

$$\begin{aligned}
H'|\psi_{IN}\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle \otimes |1\rangle = &U_1|\psi_{IN}\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle + U_3|\psi_{IN}\rangle \otimes |1\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle \\
&+ U_2^*|\psi_{IN}\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle \otimes |1\rangle
\end{aligned}$$

where one of the terms has a cursor in the "terminating spot", but the correct computation hasn't occurred. Thus, we ensure our program counter contains only one cursor, and we define our input to include the cursor in the rightmost (0th) spot of the program counter, i.e. $|\psi_{IN}\rangle \otimes |0\rangle \otimes ... \otimes |0\rangle \otimes |1\rangle$.

Back to $U = U_k...U_1$, input state $|\hat{\psi}_{IN}\rangle = |\psi_{IN}\rangle \otimes |0\rangle^{\otimes k} \otimes |1\rangle$ and Hamiltonian

$$H = \sum_{i=0}^{k-1} a_{i+1}^* a_i U_{i+1} + \sum_{i=0}^{k-1} a_i^* a_{i+1} U_{i+1}^*$$

Passing to the exponential, we get

$$\begin{aligned}
e^{-\frac{it}{\hbar}H}|\hat{\psi}_{IN}\rangle &= \sum_{n=0}^{\infty} \frac{1}{n!}(\frac{-it}{\hbar})^n H^n |\hat{\psi}_{IN}\rangle \\
&= \sum_{n=0}^{\infty} \frac{1}{n!}(\frac{-it}{\hbar})^n [C_k^n (U_k U_{k-1}...U_1|\psi_{IN}\rangle \otimes |1\rangle \otimes |0\rangle^{\otimes k}) + C_{k-1}^n (U_{k-1}...U_1|\psi_{IN}\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle^{\otimes k-1}) + \\
&... + C_0^n (|\psi_{IN}\rangle \otimes |0\rangle^{\otimes k} \otimes |1\rangle)] \\
&= [\sum_{n=0}^{\infty} \frac{1}{n!} i^n t^n C_k^n] U|\psi_{IN}\rangle \otimes |1\rangle \otimes |0\rangle^{\otimes k} + [\sum_{n=0}^{\infty} \frac{1}{n!} i^n t^n C_{k-1}^n] U_{k-1}...U_1|\psi_{IN}\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle^{\otimes k-1} + \\
&... + [\sum_{n=0}^{\infty} \frac{1}{n!} i^n t^n C_0^n] |\psi_{IN}\rangle \otimes |0\rangle^{\otimes k} \otimes |1\rangle \\
&= D_k(t) U|\psi_{IN}\rangle \otimes |1\rangle \otimes |0\rangle^{\otimes k} + D_{k-1}(t) U_{k-1}...U_1|\psi_{IN}\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle^{\otimes k-1} + ... + D_0(t)|\psi_{IN}\rangle \otimes |0\rangle^{\otimes k} \otimes |1\rangle
\end{aligned}$$

where the C_i^n count the number of states that end up as $U_i...U_1|\psi_{IN}\rangle \otimes |0\rangle \otimes ... \otimes |1\rangle \otimes ...|0\rangle$ for a given n, with the cursor in the ith spot. The sums in the second last line all converge, and we represent their limits as the $D_i(t)$. For a given t, these determine the probability of measuring each of the constituent states of the superposition. By measuring the kth spot of the program counter, we project onto either the state that represents the solution on our computational atoms when (when we measure the kth spot in the $|1\rangle$ state), or some other state that is not the full computation on our computational atoms (when we measure the $|0\rangle$ state). This is useful since, we either collapse the wave function into the state we want, or do not collapse the superposition when we don't measure a $|1\rangle$, meaning we can keep measuring until we do get a $|1\rangle$.

Note that if the kth spot has been measured to be in the $|1\rangle$ state, then removing the cursor will ensure that the computational register atoms remain in our desired output state, and can be measured at any stage.

Also note that we could change our program counter and computation slightly, so that we are considering $U = U_{k+m}U_{k+m-1}...U_k...U_1$ where all the $U_{k+1}, ..., U_{k+m}$ are the identity. This means we have a program counter with $k+m+1$ additional atoms, and if we measure the cursor in any of the m positions $k+1$ to $k+m+1$, then the computation the output is what we want. It may also be possible to place our quantum system, or more specifically the program counter, into some field such that the movement of the cursor is heavily bisaed to the forward direction, again increasing the likelihood that we measure the cursor in a terminating position.

Now some final, more broad comments to finish off. Firstly, the universal gate set for reversible classical computation is not a universal gate set for quantum computation. Because we are considering a quantum system, the state of a particle can be any superposition $\alpha|0\rangle + \beta|1\rangle$, i.e. can be any vector lying on the 2-sphere.

That means there is greater variety with the operations we can perform, such as rotations. Furthermore, we saw earlier that we obtained the desired output by measuring the cursor in a certain state, which projected the quantum system into that output state, in essence throwing away a lot of information. This again suggests that quantum computation is more powerful than classical computation since it is possible to perform computation using the superposition of all the states produced in the evolution of the system.