Quantum Computing

Physics 521

Armaan Sethi

November 26th, 2018

UNC Honor Pledge: I certify that no anauthorized assistance has been received or given in the completion of this work.

Signature _____

1 Introduction

Richard P. Feynman once said "[Quantum mechanics] describes nature as absurd from the point of view of common sense. And yet it fully agrees with experiment. So I hope you can accept nature as She is - absurd." [9] Quantum mechanics is one of the most precisely tested theories in the history of science. Even though the world around us is described by quantum mechanics, it is incredibly difficult to model any large quantum mechanical system. However, we know that quantum mechanics can easily model classical mechanics because when we look around, that's how we perceive the world. This begs the question about the possibilities of computing using the properties of quantum mechanics. Quantum mechanics has many special properties that can be mathematically manipulated in specific ways to compute things that would be nearly impossible to compute otherwise.

2 Quantum Bits

2.1 Singular Quantum Bits

In classical computation or classical information theory, the smallest unit of data and building block for all computation is a bit, or binary digit. However, in quantum computing or quantum information theory, a quantum bit is the fundamental unit and building block for all computation. A quantum bit or qubit is any mathematical object that follows the rules of quantum mechanics, usually described by a wave function, allowing for superposition, entanglement and is fundamentally stochastic in nature. Qubits are represented very similarly to classical bits as they collapse to either a 0 or a 1 when observed. However, the most convenient way to mathematically represent the state of a qubit is using Dirac notation with orthonormal basis vectors $|0\rangle$ and $|1\rangle$. It is possible to have any linear combination, or superposition, of these basis vectors:

$\psi = \alpha\,|0\rangle + \beta\,|1\rangle$

Where the coefficients, or amplitudes, alpha and beta are complex numbers that are normalized such that $|\alpha|^2 + |\beta|^2 = 1$ and the probability of measuring a qubit in a given state is the squared value of the coefficient of that state. Since the system is a quantum system, complex numbers are required to fully describe the superposition of states, interference, or entanglement that may be present. In this example we would measure the result 0 with probability $|\alpha|^2$ and the result 1 with the probability $|\beta|^2$. The computational basis is the most common basis used in quantum computing and represented by:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{1}$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2}$$

However, it is possible to use any orthonormal basis.

2.2 Multiple Quantum Bits

Qubits become increasingly useful as we begin discussing multiple qubits. In a classical system, it would only take 2 numbers to describe a 2 classical bit system. It would be either 00,01,10, or 11. However, a two qubit system can be any superposition of the four computational basis states $|00\rangle, |01\rangle, |10\rangle, |11\rangle$.

Thus, to completely describe the system you would need four normalized complex coefficients, or amplitudes:

$\psi = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$

Where the probability of measuring each state is the square of its coefficient. For example, the probability of the system being in the state $|00\rangle$ is $|\alpha_{00}|^2$ while the probability of being in the state $|01\rangle$ is $|\alpha_{01}|^2$. For an N qubit system, the computational basis states of the system are of the form $|x_1, x_2, ...x_n\rangle$. As the number of qubits increases, the number of probabilities scale as 2^N, where N is the number of qubits. This means for each additional qubit, the computational power doubles. For example, in a 49 qubit system, there would be 2^{49} coefficients that can be manipulated. Although no one has achieved this goal yet, it has been calculated that quantum supremacy, or when a quantum computer can outperform a classical supercomputer, can be comfortably demonstrated with 49 qubits, a circuit depth exceeding 40, and a two-qubit error below 0.5%. With 500 qubits, we would have 2^{500} amplitudes, which is a number larger than the estimated number of atoms in the Universe. This is the power of exponential scaling.

2.3 Physical Realization of Quantum Bits

Although we have discussed qubits as theoretical mathematical objects, they would not be very useful if they did not have potential physical realizations. There are many potential physical realizations, all of which have their own unique strengths and weaknesses.

The polarization state of a photon is one potential realization of a quantum bit. This is because photons exhibit all the necessary quantum mechanical effects. For example, a photon can be described as having horizontal or vertical linear polarization, or a superposition of the two. Photon loss is very similar to decoherence, and can similarly be mitigated by quantum error correction techniques to some extent, but having minimal photon loss would greatly increase how effective photons are as qubits. Another potential qubit is the spin states of nuclei within molecules. Using nuclear magnetic resonance the quantum states, which exhibit all the necessary quantum mechanical properties for quantum computation, can be examined. The initial approach involved using spin properties of atoms of particular molecule in a liquid as qubits. Now it is possible to use solid state nuclear magnetic resonance to achieve the same effects. Another promising potential physical realization of a qubit is a quantum dot. Quantum dots are very small semiconductor particles that exhibit

all of the quantum mechanical properties necessary for quantum computation. Finally, superconductors can be used to create physical qubits. There are three different ways of using of superconductors to implement qubits: charge, flex, and phase. They are all closely related since they use superconductors at extremely low temperatures, but are implemented slightly differently. Although there are many potential ways of realizing quantum bits, none of them are perfect and they all suffer from significant information loss and decoherence today.

3 Quantum Gates and Quantum Circuits

Similar to how classical computer circuits consist of wires and logic gates, quantum computers consist of quantum gates and quantum circuits. However quantum gates must follow the rules of quantum mechanics. This makes it such that all quantum gates act linearly, meaning they can be mathematically represented as transformation matrices, and that all quantum gates must be unitary, which is that for a matrix U, $U^\dagger U = I$ where $U^\dagger$ is the adjoint of U (the transpose of the conjugate of the U matrix). This also means that all quantum gates must be reversible since the inverse would be $U^\dagger$. However, since this is the only constraint, any unitary matrix is a valid quantum gate.

3.1 Single Quantum Bit Gates

One example of a quantum gate would be the NOT gate, (which is represented as X for historical reasons):

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{3}$$

Then if the quantum state is represented as $\alpha\,|0\rangle + \beta\,|1\rangle$, it can also be written as the column vector:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{4}$$

Then the output from the quantum gate would be:

$$X \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

Essentially swapping the coefficients or amplitudes associated with each gate. Another extremely important example of a quantum gate is the Hadamard operator, an operator that is often referred to as a "fair coin flip." The Hadamard operator applied to a qubit that is in the state $|0\rangle$ or $|1\rangle$, meaning the state is in no superposition (one of the coefficients is 0), will create an equal superposition of the states $|0\rangle$ and $|1\rangle$, having an equal probability of the qubit being observed in the state $|0\rangle$ or $|1\rangle$. Mathematically the Hadamard operator, H, can be written as:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \langle 0| + \frac{|0\rangle - |1\rangle}{\sqrt{2}} \langle 1|$$

Through simple algebra we can prove that $H^2 = I$, so applying the H operator twice to a state results in nothing.

3.2 Multiple Quantum Bit Gates

Quantum computing also makes use of multiple qubit gates, or controlled operations. These operations change the state of a qubit based on the values of other qubits. The controlled-NOT or CNOT gate performs the NOT operator to a qubit only if the controlling bit has the value $|1\rangle$. It is possible to have controlled application of any single qubit gate. This can be represented mathematically as the matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & x_{00} & x_{10} \\ 0 & 0 & x_{01} & x_{11} \end{bmatrix}$$

Where $x_{00}, x_{01}, x_{10}, x_{11}$ are elements of any unitary matrix or single qubit gate.

It is also possible to have controlled operations with any n control qubits and any unitary operator on k qubits. The most famous example of this is the Toffoli gate, or the controlled-controlled-NOT gate. The Toffoli gate can implement the logical AND, NOT, XOR, and FANOUT operations depending upon the input. Since all boolean functions can be created from the logical AND and NOT, the Toffoli gate shows that quantum circuits can perform all operations performed by classical circuits.

4 Quantum Algorithms

Quantum algorithms utilize important properties of quantum computation such as quantum parallelism and interference in quantum mechanics in order to solve various problems. Quantum parallelism is the idea that quantum computers are able to simultaneously evaluate a function $f(x)$ for many different values of x. However, this simple explanation overstates the usefulness of this property, since when we observe the system, we will only be able to observe one output for the system and not the individual outputs of $f(x)$ for many values of x. Since we know that the probability of observing a state is the square of that states coefficient or amplitude, we can use interference to manipulate the amplitudes of the states in a way that algorithmically gives us the output we would like. We will discuss three different quantum algorithms with varying complexity and potential impact on the world.

4.1 Deutsch's algorithm

Deutsch's algorithm solves a problem that is not very important in Computer Science, but instead is a contrived problem that demonstrates an instance in how quantum circuits can outperform classical circuits. The problem that Deutsch discovered was "Suppose there is a function f that maps $\{0,1\}$ into $\{0,1\}$. We wish to know whether the function is one to one or not. Note that there are only four possible functions mapping $\{0,1\}$ into $\{0,1\}$. Both $f(0)$ and $f(1)$ can map to either 0 or 1." [4] Classically one would solve this by evaluating the XOR of $f(0)$ and $f(1)$. If the function is one to one, then $f(0)$ XOR $f(1)$ would be 1, otherwise it would be equal to 0. This means that classically we must evaluate the function twice, once for $f(0)$ and once for $f(1)$.

Deutsche's Algorithm consists of five steps. First prepare the system. Since the function $f(x)$ has a domain and range of two values, we must prepare two qubits. One in the state $|0\rangle$ and one in the state $|1\rangle$. The combined state can be written as $\psi_0 = |01\rangle$ Second, apply the Hadamard gate on the two qubit system such that both qubits are in an equal superposition.

$$|\psi_1\rangle = \left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$$

Third we apply the transformation defined by the map $|x,y\rangle\,|x,y\oplus f(x)\rangle$, which is known as U_f [7] to $|\psi_1\rangle$. This leaves us with two possibilities:

$$|\psi_2\rangle = \begin{cases} \pm\left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] & f(0)=f(1) \\ \\ \pm\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] & f(0)\neq f(1) \end{cases}$$

Fourth we apply the Hadamard gate to the first qubit resulting in:

$$|\psi_3\rangle = \begin{cases} \pm|0\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] & f(0)=f(1) \\ \\ \pm|1\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] & f(0)\neq f(1) \end{cases}$$

This shows us that if $f(0) = f(1)$ the first qubit will be observed as $|0\rangle$ and if $f(0) \neq f(1)$, the first qubit will be observed as $|1\rangle$. This shows that in only one evaluation of $f(x)$ we are able to determine a global property of $f(x)$, whether or not it is one to one. The quantum algorithm only evaluates $f(x)$ once while the classical algorithm would need to compute the function twice. This shows that if $f(x)$ is sufficiently complicated to compute, a quantum computer can provide an improvement to classical computers. Although this problem may not seem very significant and the speed improvement is not massive, it is a simple problem that proves that there are problems that quantum computers are better at than classical computers.

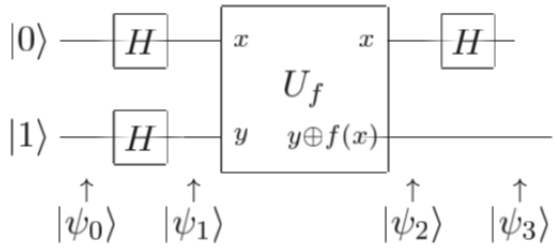

Figure 1: Quantum Circuit implementation of Deutsch's algorithm. [7, p. 33]

4.2 Grover's Algorithm

Grover's algorithm is a search algorithm that searches an unordered set of items to find the unique element that satisfies some condition. Classically searching an unordered set would require $O(N)$ time where N is the number of elements in the unordered set, but Grover's algorithm is able to accomplish this same task in $O(\sqrt{N})$ operations. Grover's algorithm uses both quantum parallelism and amplitude amplification, or exploiting the properties of quantum amplitudes that differentiate it from simple probabilities. This usually involves iteratively shifting the phase of specific states of a quantum system that satisfy some condition.

The first step to Grover's algorithm is to prepare the system such that all n qubits, where $N = 2^n$, are initialized to $|0\rangle$.

$$|0\rangle^{\otimes n} = |0\rangle$$

Then, similarly to Deutsch's algorithm, we apply a Hadamard transform to all n qubits so that the system is put into an equal superposition of states:

$$|\psi\rangle = H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$$

where x is the base-10 integer representation of a length-n number in base-2.

We then apply what is called the Grover iteration $\frac{\pi}{4}\sqrt{2^n}$ times. The first part of the Grover iteration is calling the quantum oracle, a quantum black-box that can observe and modify the system without collapsing into a classical state. The quantum oracle checks if the system is in the configuration we are searching for. If the system is in the correct state, the oracle will rotate the phase by π radians marking the correct state for further modification by later operations. If the system is not in the correct state, the oracle will do nothing. This can be written mathematically as:

$$|x\rangle \xrightarrow{\text{Oracle}} (-1)^{f(x)} |x\rangle$$

where $f(x) = 1$ if the oracle finds that the system is in the correct state (rotating by a phase of π is the same as negating the amplitude), and $f(x) = 0$ if the oracle finds that the system is in the incorrect state (nothing happens).

The next part of the Grover iteration is called the diffusion transform. The diffusion transform performs what is called an inversion about the average, meaning that it transforms the amplitude of each state such that it is now as far above the average as it was below the average before the transformation, and vice versa. This can be written mathematically as a Hadamard transform on all qubits, followed by a phase shift by -1 of all states except $|0\rangle$, followed by another Hadamard transform. Mathematically the diffusion transform simplifies to:

$$2\left|\psi\right\rangle\left\langle\psi\right| - I$$

and the entire Grover iteration simplifies to:

$$\left[2\left|\psi\right\rangle\left\langle\psi\right| - I\right] Oracle$$

Since we repeat the Grover iteration $\frac{\pi}{4}\sqrt{2^n}$ times, the entire process runs in $O(\sqrt{N})$ time. This is significantly faster than the classical implementation, as N gets large, that would run in $O(N)$. A worked three qubit example of Grover's algorithm can be found in "An Introduction to Quantum Algorithms"[5, p. 24]

4.3 Shor's algorithm

Shor's algorithm is arguably the most dramatic example of how quantum computing can drastically change what problems are difficult to compute. Finding the prime factors of an integer has always been a hard problem for classical computers. This is why there are many encryption schemes responsible for the security of our online transactions that depend upon the fact that factoring integers that have thousands of digits is practically impossible to compute. In 1995, this assumption was challenged by Peter Shor when he proposed a polynomial-time quantum algorithm for factoring large integers.

The problem that Shor's algorithm solves is to factor an integer N with d decimal digits. The general number field sieve is the most efficient classical factoring algorithm and achieves an asymptotic runtime that is exponential in $d^{\frac{1}{3}}$. Since it is exponential in time, it gets enormous very quickly as d increases. The largest number factored by this algorithm had 232 digits and took roughly 2,000 CPU years to compute. In contrast, the quantum algorithm Shor proposed is polynomial in d with a runtime of d^3, requiring $10d$ qubits, being a huge improvement over the exponential time of classical algorithms. This is accomplished by a clever use of quantum parallelism and constructive interference, similarly to Deutsch's Algorithm, but much more complicated.

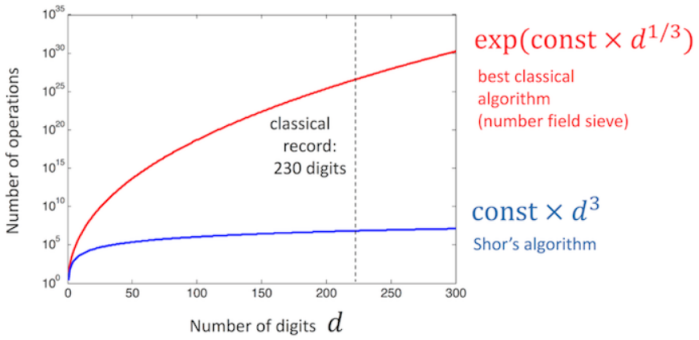

Figure 2: Runtime of Shor's Algorithm vs Number Field Sieve [8]

5 Quantum Noise and Quantum Error Correction

In the real world there are no perfectly closed systems. This makes it nearly impossible to create an environment in which quantum systems would not suffer any unwanted interactions with the outside world, thereby causing quantum noise. In order to describe the dynamics of open quantum systems and mitigate their effects, the mathematical formalism of quantum operations can be used. Quantum operations can describe systems that are weakly coupled to their environment, systems that are strongly coupled to their environment, and even closed systems that are opened suddenly and subject to measurement. Other tools that are used by physicists to describe quantum noise are the master equations and quantum process tomography. The master equations are derived from quantum optics and they describe the time evolution of an open system with a differential equation that properly describes non-unitary behavior. Quantum process tomography is a procedure used to determine an unknown quantum state. It is important to keep in mind that noise is not a new problem in computing and was once a problem that plagued classical computing, but has now been solved by countless error-correcting codes. However, this problem is still an extremely active area of research for quantum computing and is one of the many reasons we have not created large-scale physical quantum computers.

There are many potential ways to reduce the effects of quantum noise such as quantum error-correcting codes and fault-tolerant quantum computation. It was also proven that there exists a certain threshold for noise in individual quantum gates for which it is possible to reliably perform arbitrarily long quantum computation. This means that as long as we can keep physical error below a certain threshold, it is possible to use quantum error-correction schemes to minimize logical error rates to negligible levels. This is called the threshold theorem in quantum computing. There are countless quantum error-correcting schemes that can be used to potentially satisfy the threshold theorem, the first of which

was discovered by Peter Shor. He discovered that by storing the information of one qubit into a highly entangled state of nine qubits, the Shor code can protect a single qubit against arbitrary error effects. The quantum circuit for the Shor code is relatively simple and can be constructed from nine $|0\rangle$ qubits and three Hadamard gates, seen in Figure 3. Since it is impossible to create a perfectly closed system, a lot of research must be done in order to minimize quantum noise and satisfy the threshold theorem.

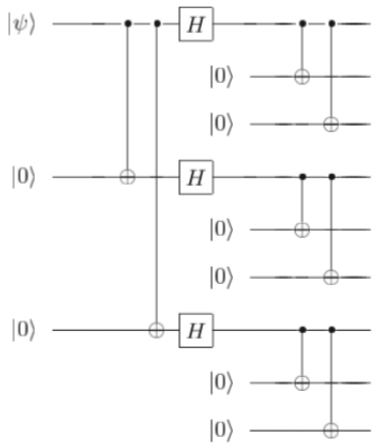

Figure 3: Shor code encoding circuit for the Shor nine qubit code. [7, p. 432]

6 Conclusion

The mathematical formalism behind quantum information theory and quantum computation shows amazing potential. However, there is still a lot of work to be done until quantum computers can be used to solve significant problems. Properties in qubits such as decoherence cause significant information loss in quantum computers today. Additionally, it is impossible to create a system that does not suffer from unwanted interactions with the outside world. There is still significant research to be done in order to create optimal qubits, reduce quantum noise, and improve quantum error-correction techniques in order to satisfy the threshold theorem in quantum computation. However, if we are able to overcome these obstacles, as we have done in the past with classical computing, the potential impact of quantum computers is immense.

It has been shown that quantum supremacy can happen with as few as 49 ideal qubits and we seem to be getting extremely close. In the last few years, Google and NASA have partnered to create many proof of concept quantum computers for research. Additionally, IBM has created quantum computers that anybody is able to program using the cloud. Microsoft and IBM have created

new programming languages that can be used in order to program quantum computers in addition to the open source libraries for existing languages that have been improving tremendously. There are also countless quantum computer and quantum circuit simulators that can be used by anybody so they can become comfortable with quantum computation.

Even though quantum computing has extraordinary potential to solve previously impossible-to-solve problems, the effects of quantum computation can be a little overstated. For example, quantum computation may cause some encryption techniques to become useless, but there are countless encryption techniques that are in place today that are safe from quantum computing. Additionally, it will be incredibly expensive to use and maintain quantum computers due to the measures that must be taken in isolate quantum computers from the world in order to minimize quantum noise. Although quantum computing may not be the perfect solution to every problem, they have the potential to have a serious impact on the world.

References

[1] Patrick J Coles, Stephan Eidenbenz, Scott Pakin, Adetokunbo Adedoyin, John Ambrosiano, Petr Anisimov, William Casper, Gopinath Chennupati, Carleton Coffrin, Hristo Djidjev, et al. Quantum algorithm implementations for beginners. *arXiv preprint arXiv:1804.03719*, 2018.

[2] R Courtland. Google plans to demonstrate the supremacy of quantum computing. *IEEE Spectrum [online], available at http://spectrum.ieee.org/computing/hardware/google-plans-todemonstrate-the-supremacy-of-quantum-computing (http://spectrum.ieee.org/computing/hardware/google-plans-to-demonstrate-the-supremacyof-quantum-computing)*, 2017.

[3] Richard P Feynman. Simulating physics with computers. *International journal of theoretical physics*, 21(6-7):467–488, 1982.

[4] Jeff Kinne. Deutsche's algorithm.

[5] PG Kwiat, JR Mitchell, PDD Schwindt, and AG White. Grover's search algorithm: an optical approach. *Journal of Modern Optics*, 47(2-3):257–266, 2000.

[6] Thaddeus D Ladd, Fedor Jelezko, Raymond Laflamme, Yasunobu Nakamura, Christopher Monroe, and Jeremy Lloyd O'Brien. Quantum computers. *Nature*, 464(7285):45, 2010.

[7] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[8] IBM Research and the IBM QX team. Shor's algorithm.

[9] QEQ Richard Feynman. The strange theory of light and matter, 1985.

[10] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[11] Emma Strubell. An introduction to quantum algorithms. *COS498 Chawathe Spring*, 13:19, 2011.