An Introduction to Quantum Computation

Ben Johnson

Graduate Group in Applied Math
bjohnson@math.ucdavis.edu

Introduction

There seem to be three primary driving forces behind research in quantum computation. We label these as *curiousity, possibility* and *necessity*. As with many scientific endeavors, there is the question of *can we do it?*, and quantum computation is no different. The ability to construct and use a quantum computer would be an achievement on its own, regardless of whether or not it was a useful technology. There are many possible realizations of a quantum computer, and research seems to be exploring many different avenues in this regard.

Now, assuming it were possible to create one, there is the question of *what could it do?* There has been a large volume of research over the last thirty years that shows that there are a diverse collection of extremely useful applications for a quantum computer. It can be shown that certain problems are faster to compute on a quantum computer than on a classical computer. This is as much the focus of current research as the actual construction of a quantum computer.

This leaves necessity. As Moore's Law continues to describe the increase is computing power, researchers have noted that there is a fundamental limit to the number of transistors that may be placed on a chip, at least with our current production methods. For better or worse, computers have become a necessary component of our lives, and the notion of an upper limit on computing power does not rest well with most people. This has led researchers to look for other forms of computation, ones that don't rely on classical circuit board and transistor technology. Of these, perhaps the best publicized is quantum computation.

So what is quantum computation? Many people have heard the term, almost as a catch-phrase, but there are many misconceptions about what it is, and what it can and cannot do. Fundamentally, quantum computation is computation that takes advantage of specific aspects of quantum mechanics, in particular, superposition and entanglement. Our objective then, is to clarify what these things mean, how quantum computation works, highlight differences between classical and quantum computation, and discuss theoretical potential for this new method of computation.

Despite its relative youth as compared to other fields of computational theory, quantum computation has a rich and diverse theoretical base, which extends far beyond what we might hope to cover in a brief report on the subject. As such, we focus on a few fundamental concepts, in the hope of providing a foundational understanding, and to highlight the number of different directions of study within quantum research.

History

The concept of quantum computation was first mentioned in a 1982 article by Paul Benioff, where he devised the notion of a quantum Turing Machine. Later that same year, Richard Feynman suggested that one of the greatest potential uses for a quantum computer would be to run quantum simulations; using the exponential resources available from a quantum computer to model the exponential parameters of a quantum system. Finally, in a 1985 paper, David Deutsch proposed the first formal definition of a quantum computer. In that same paper, he highlighted the potential power of a quantum computer by demonstrating a quantum algorithm that could solve a particular problem faster than any classical algorithm.

The papers by Feynman and Deutsch are often hailed as the advent of quantum computing. Over the next ten years, there were several other milestones in the field as well. Perhaps most notable is Peter Shor's 1994 factoring algorithm. He demonstrated that a quantum computer could find the prime factors of a number in $\mathcal{O}(\log n^3)$, which is an exponential speed-up from the best known classical algorithm. Two years later, Lov Grover devised a search algorithm that could

search an unsorted list in $\mathcal{O}(\sqrt{n})$-time, a quadratic speed-up versus the classical $\mathcal{O}(n)$ method. Both of these results have practical applications to real-world situations. Their discovery provided a large part of the fuel that has carried quantum research through to today.

Background

The fundamental unit of quantum computation is the *qubit*, short for *qu*antum *bit*. It is the quantum analog of the bit used in classical computation. We represent a single qubit $|\phi\rangle$ as a linear combination of the *computational basis states*, $|0\rangle$ and $|1\rangle$. We then write

$$|\phi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle \qquad \alpha_0, \alpha_1 \in \mathbb{C}$$

subject to the normalization restriction $|\alpha_0|^2 + |\alpha_1|^2 = 1$. Thus we can interpret $|\phi\rangle$ as a unit vector in the two dimensional complex vector space $\mathbb{C}^2$.

The square amplitudes of the basis states represent the probabilities of observing those states on measurement. For example, if we are given the state

$$|\phi\rangle = \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle,$$

we will observe state $|0\rangle$ with probability $\frac{1}{4}$ or $|1\rangle$ with probability $\frac{3}{4}$.

A larger dimensional case is generally useful to fully appreciate qubit representation. Consider the two qubit system

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

which lives in $(\mathbb{C}^2)^{\otimes 2} \simeq \mathbb{C}^4$. We again have a normalization condition:

$$|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1.$$

The interpretation of the amplitudes is the same as before. For example, we will observe $|10\rangle$ with probablity $|\alpha_{10}|^2$. From this, it is not difficult to extend this notion to the case of n qubits, which live in a Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n} \simeq \mathbb{C}^{2^n}$.

We now consider operations that we may perform on qubits. Landauer's Principle, simply stated, says that the loss of information into a system requires the dissipation of energy. As our quantum systems are isolated, we cannot allow for the dissipation of energy, and thus, we cannot lose information. This means that any operation performed on a qubit must be reversible. We also require that operations take qubits to qubits. This means that we only allow unitary operators. That is, operators such that $\mathbf{U}\mathbf{U}^* = \mathbf{U}^*\mathbf{U} = \mathbf{I}$. Note that we will use the terms operator and gate interchangably. This is due to an analogous circuit representation of these same concepts.

For illustrative purposes, and for later reference, we consider a few operators here. We present single qubit gates here. These can be extended to arbitrary size in an obvious way by appropriate use of tensor products. The Hadamard, Z, and NOT gates, respectively, are:

$$\mathbf{H} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \mathbf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The Hadamard gate is arguably the most important operator in quantum computation. When applied to the basis states, the result is:

$$\mathbf{H}|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \qquad \text{and} \qquad \mathbf{H}|1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right).$$

These two states are important in their own right. We denote them as $|+\rangle$ and $|-\rangle$, respectively. For later use, we note that if we apply a Hadamard gate again, we will return to our original qubit. That is, $\mathbf{H}|+\rangle = 0$, and $\mathbf{H}|-\rangle = 1$. Notice that these two states have identical probabilities but different amplitudes. This turns out to be significant when performing computations.

We make one final comment before moving on. We mentioned previously that all qubit operations are reversible. The exception to this rule is measurement. When we measure a qubit, we observe a particular value, so its probability of being in any other state becomes zero. Once the state is fixed, we cannot recover the probabilistic representation, and that information is lost. Note that this agrees with Landauer's Principle above.

Algorithms

A quantum algorithm is the same, conceptually, as a classical algorithm. It is simply the application of specifically chosen operations on the basis states. The result is, in general, something of interest. The difference is that a quantum algorithm can take advantage of the superposition of states, while a classical algorithm cannot. If the qubits are prepared in a specific way, the output is characterized by all 2^n possible states of n qubits having been evaluated by our algorithm. This feature is referred to as *quantum parallelism*.

There is an important caveat that must not be overlooked. We do not have access to this information, at least not directly. When we measure the output of an algorithm under these circumstances, we observe a single evalution of the function for some randomly chosen input. The trick then, is to manipulate the qubits before making an observation so that we get more than just the function evaluation as a result. This concept is made clear below when we proceed step-wise through a quantum algorithm.

We must note here another important result in quantum computation: the *no-cloning theorem*. This says that it is impossible to copy qubits, and thus we cannot gain extra information on the 2^n evaluations from the above algorithm.

We now show one example of the theoretical power of a quantum computer by means of the Deutsch problem. As mentioned above, Deutsch's 1985 paper provided the first example of a problem that could be solved explicity faster using quantum computation. We note that Deutsch's original solution to this problem was not deterministic, and we present a variant on his solution.

Consider a function $f : \{0, 1\} \to \{0, 1\}$. There are four such functions:

	$f(0)$	$f(1)$
f_0	0	0
f_1	0	1
f_2	1	0
f_3	1	1

The problem is to determine whether f is *constant* (f_0 or f_3) or *balanced* (f_1 or f_2).

On a classical computer, this will require 2 calls to the function f. It is clear that we cannot characterize f with a single invocation. Deutsch's algorithm shows that we can answer this question with a single function call on a quantum computer. We begin with single qubits on the input and output registers of our quantum computer. We start both qubits in the $|0\rangle$ state. Let $\mathbf{U}_f$ denote the unitary operator that implements the function f. Then on input $|x\rangle|y\rangle$, we have $\mathbf{U}_f(|x\rangle|y\rangle) = |x\rangle|y \oplus f(x)\rangle$. It is not difficult to verify that the unitary operators representing our function are given by:

$$\mathbf{U}_{f_0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{U}_{f_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{U}_{f_2} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{U}_{f_3} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

We prepare the input and output registers by first applying NOT gates and then Hadamard gates to both. Then we have:

$$(\mathbf{H} \otimes \mathbf{H})(\mathbf{X} \otimes \mathbf{X})(|0\rangle|0\rangle) = \frac{1}{\sqrt{2}}[|0\rangle - |1\rangle]\frac{1}{\sqrt{2}}[|0\rangle - |1\rangle].$$

We use these as the input to the our function $\mathbf{U}_f$, giving us:

$$\frac{1}{2}[|0\rangle|f(0)\rangle - |1\rangle|f(1)\rangle - |0\rangle|1 + f(0)\rangle + |1\rangle|1 + f(1)\rangle]$$

Depending on whether f is constant or not, we will get two different factorizations of the above expression:

$$\frac{1}{\sqrt{2}}\left[(|0\rangle - |1\rangle)\frac{1}{\sqrt{2}}(|f(0)\rangle - |1 + f(0)\rangle)\right], \qquad \text{if } f(0) = f(1)$$

or

$$\frac{1}{\sqrt{2}}\left[(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|f(0)\rangle - |1 + f(0)\rangle)\right], \qquad \text{if } f(0) \neq f(1).$$

Now, we recognize that when f is constant, the input register is in state $|-\rangle$ and when f is balanced, the register is in state $|+\rangle$. We know that the Hadamard gate distinguishes these two states, so we apply that gate to the input registers. (And apply the identity operator to the output register.) The result is:

$$|1\rangle\frac{1}{\sqrt{2}}(|f(0)\rangle - |1 + f(0)\rangle), \qquad \text{if } f(0) = f(1)$$

or

$$|0\rangle\frac{1}{\sqrt{2}}(|f(0)\rangle - |1 + f(0)\rangle), \qquad \text{if } f(0) \neq f(1).$$

We read the input register (*not* the output, as we would typically expect), and we have our result with probability 1. We may express this result concisely as

$$(\mathbf{H} \otimes \mathbf{I})\mathbf{U}_f(\mathbf{H} \otimes \mathbf{H})(\mathbf{X} \otimes \mathbf{X})(|0\rangle|0\rangle) = \begin{cases} 1 & \text{if } f(0) = f(1) \\ 0 & \text{if } f(0) \neq f(1) \end{cases}$$

There are two important observations that may be made at this point. First, we have shown Deutsch's problem for a specific case, but this algorithm extends in a very natural way (think tensor products). It is not difficult to show that we may take input from $\{0,1\}^n$ instead of $\{0,1\}$, and it will still only require a single function call to determine if f is constant or balanced, where a classical method could require up to $2^n + 1$ calls. Even with exponential speed up, this problem has no (known) practical application, other than highlighting the potential of quantum computation.

Second, we did not determine *which* function was represented by $\mathbf{U}_f$. In fact, we did not even learn the value $f(0)$ or $f(1)$. We sacrificed this information to learn if f was constant or not.

Limits

Now that we have the framework for how quantum computation works, and we have seen some of its potential, it is only appropriate to discuss its limitations. A common misconception, largely due to naïveté, is that quantum computers are arbitrarily more powerful than classical computers. While we have demonstrated that quantum computers can offer an exponential speed-up to certain problems, there are still many problems that they cannot compute efficiently. (By efficiently, we mean in polynomial time). In fact, we saw this result already with Grover's search algorithm. Quantum computation provides a quadratic improvement, but it still requires superpolynomial time, $\sim 2^{n/2}$. Grover's algorithm can be applied to any NP-complete problem (just search all possible solutions), so we have a quadratic speed up to an entire class of very important problems. But it has been shown that Grover's algorithm is optimal, that is, there is no faster algorithm, and thus a quantum computer cannot solve any NP-complete problem in polynomial time, at least with a brute-force search. It is still unknown if there are efficient quantum algorithms for NP-complete problems, but it is conjectured to be false.

Further research

To illustrate the diversity of the subject, we briefly mention a number of subfields within quantum computation, and try to characterize their focus.

- *Quantum error correction* - Quantum computers are extremely sensitive to environmental influence, so the potential for error is high. Research has shown that arbitrarily long computations are possible, so long as the error is below a particular threshold value. The goal is to construct error-correcting codes and fault-tolerant computation that will make the use of a quantum computer feasible.

- *Physical realization/construction* - As mentioned in the introduction, quantum computation can be realized in a number of different ways. A quantum computer will require a balance between theoretical needs (isolated system with qubits) and physical constraints (cost to construct, energy required to run it, ease of performing measurement). Research in this field focuses on further development of existing models, and creation of altogether new methods for performing quantum computation.

- *Quantum information theory* - The quantum analog of Shannon's classical information theory, this deals with different measurements of information and entropy in a quantum system. What type of information is generated by a quantum system, and what can be learned by measuring it. This also deals with data compression, and how to handle noise on a quantum channel.

- *Quantum cryptography* - If quantum computers of even moderate size are ever constructed, they will pose a significant threat to current encryption schemes. Quantum cryptography deals with encryption schemes based on quantum properties, such as entanglement, rather than the intractability of decryption. As measurement changes qubits, and copying is not allowed, any information that is intercepted by a third party will be known to the intended recipients. This principle is used to guarantee secure communication.

Conclusions

Quantum computation is an incredibly rich field with vast potential for research. We have barely scratched the surface here, but hope that we have elucidated certain aspects of the subject.

We expect that research will continue unabated for a long time, and would not be surprised if the next thirty or forty years saw quantum computers have the same dispersal as classical computers in the 1940s (few and far between, but gaining momentum).

References

N. D. Mermin. *Quantum Computer Science, An Introduction.* Cambridge University Press, 2007.

M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

J. Preskill. *Quantum Computating: Pro and Con. Proc. Roy. Soc. Lond.* A454 (1998) 469-486.

J. Preskill. Quantum Computation lecture notes. http://www.theory.caltech.edu/people/preskill/ph229/