Quantum Computing: A Theoretical Outlook on Building a Quantum Computer

Paul Liu - 50168103

Science One - Term 1 Project

Abstract

In this paper the theoretical basis behind a quantum computer as well as physical methods to constructing one is examined. Using previous spectroscopic techniques in new applications one can extend the usage of logic gates - the foundation to all classical computers - to a quantum analogue. Moreover, the differences between classical and quantum computing will be explored by analyzing the decoherence problem, the main obstacle to the construction of a quantum computer. Finally, a glimpse is given at possible applications of quantum computing by examining Shor's Prime Factorization Algorithm, thus demonstrating the effectiveness of quantum constructions over the classical computer.

1 Introduction

1.1 Why quantum computers?

As the need for faster and better computers rise, the ability to create such computers using traditional methods become impossible [5]. Due to the limiting size of the transistor and the relatively large amount of heat that silicon transistors generate, the current method of computer construction is projected to only last until 2020 to 2025 [5]. Thus, many alternatives are being explored to continue the growth of computer processing speeds. One alternative, quantum computation, involves computing on individual atoms as opposed to the transistor-based circuits in traditional computers. Thus, quantum computing allows one to scale components of computers down to the tiniest size possible: the scale of individual atoms [1]. With increasing research efforts in this field, it is hoped that quantum computers will soon overtake and replace classical computers, offering a previously unseen level of computing at the disposal of the average citizen.

1.2 How do quantum computers work?

In a computer, operations are carried by tiny two-level switches called bits. At any given time, the bits of an ordinary computer are in a definite state. At the lowest level, we describe these states mathematically by a binary representation of all the bits in the processor, such as 1010110... When bits are scaled to the size of atoms however, quantum mechanical effects drastically change how these bits behave [1]. Atoms are no longer in a single place, as quantum mechanics state that their locations will always have some immeasurable uncertainty to them [5]. Thus, instead of the classical bit, we have the analogous quantum *qubit.* The qubit is a quantum system that has two states, such as the spin-up (denoted by 1) and spin-down (denoted by 0) states of a half-spin elementary

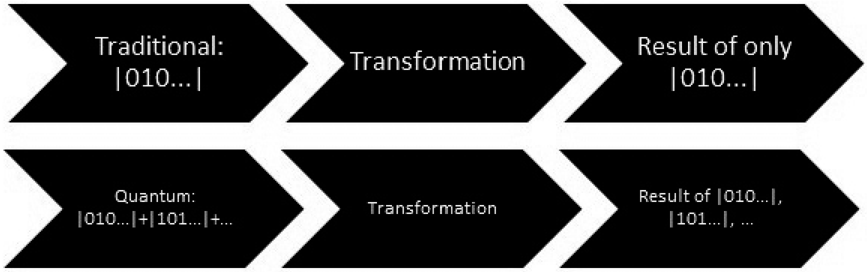

Figure 1.2.1*: The difference of transforming a single qubit versus a single bit. Note that in a quantum computer, we cannot extract all the results from each superposition. We can only obtain one useful result when measured.*

particle [1]. For qubits, the state of the computer is no longer definite, but rather a superposition of all the states it could be in [1]. These possible states can be mathematically represented by a *wave function*, for example

$$\psi = a|101010...\rangle + b|101011...\rangle + ... \tag{1.1}$$

where the coefficients *a,b,...* are complex numbers [1]. As the wave function describes the probability of the qubit in a certain wave state, the sum probability of all the states (calculated by $|a|^2+|b|^2+...$) must equal 1 [1]. This wave function describes a system similar to the waves one sees in everyday life, and so quantum systems have wave properties such as phases and amplitudes. However, it is critically important to note that the qubits are not in any single one of these states at a given time, but are rather in all of them at the same time according to the laws governing quantum phenomena [1]. It is only when we measure these states that the wave function collapses (by the Copenhagen interpretation of quantum mechanics) and a definite result is obtained [1]. Thus, an unmeasured qubit is both a $|1\rangle$, $|0\rangle$, and any superposition of $|1\rangle$and $|0\rangle$. While it is known that an n-bit computer can only be in one of 2^n states at any time, the superposition of quantum states means that an n-qubit computer is in all of 2^n states at any time [1]. This superposition property implies that when we apply an effect to one qubit, the effect is carried to all of its states at the same time, saving one the trouble of doing many repeated calculations (see figure 1.2.1) [5]. This critical difference is what makes quantum computers vastly different from the ones we use in everyday life.

1.3 Constructing a computer - Logic gates

In ordinary computers, calculations are carried out by a set of transformations on the original state of the computer's bits. These transformations are called logic gates, and hold logic functions such as *AND, NOT,* or *OR.* Adopting the convention of representing 1 as true and 0 as false, the *NOT* gate may have a function that changes a bit from 0 to 1 (false to true), while the *AND* gate may take two bits 1 and 0 to produce 0 (true and false to false) [5]. It can be shown that the logic gate *NOT* along with any of the gates *AND or OR* produces a logically complete system [5]. In other words, all other logic gates stated can be expressed as a combination of *NOT* and the chosen logic gate, providing the foundations to ordinary computation. It will be seen in this paper that there are quantum parallels to the classical logic gate, making quantum computations theoretically feasible.

2 Discussion

2.1 Creating Quantum Logic Gates

With traditional computing, logic gates are created by an electric circuit that accepts two input currents, compares them, and outputs a current based on the two inputs [5]. In quantum computing, a suitable analogue for a logic gate can be implemented by well known spectroscopic techniques [1]. For example, the *NOT* gate can be created by applying a *tipping pulse* to invert the state of the qubit (i.e. from 0 to 1). This tipping pulse is created by a precisely-applied time dependent magnetic field, which acts to change only the spin of the qubit and not the amplitude associated with its wave function [1].

When trying to create multi-qubit systems, the physics become much harder. To ensure the survival of a multi-qubit system, all of the qubits must be in phase with one another as to remove unwanted destructive interference of wave functions [1]. Thus, the number of techniques one could use to change the qubit system decreases [1]. Still, for a system with two qubits (say a and b), the XOR (exclusive or) logic gate can be constructed by using the polarization transfer technique developed by Feher in Electron-Nucleus Double Resonance (ENDOR) [1]. In this technique, two tipping pulses are applied to the qubit system, leaving a in the state of *a XOR b* and b in the state of a. Another *XOR* technique described by DiVincenzo involves only one tipping pulse and leaves a in the state of *a XOR b* with no changes to b itself [1]. For a system with three qubits (a, b, c), one can make an *AND* gate by applying the *XOR* transformation three times to store the value of *a AND b* in c [1].

From the operations of *NOT* and *AND* developed above, a logically complete system for a quantum computer can be created. With the development of these “quantum gates”, the theoretical parts to making a rudimentary quantum computer are complete.

2.2 Applications of Quantum Computation - Shor’s Factorization

With the theoretical quantum computer constructed, we can now apply the merits of quantum computation to something beyond the traditional computer’s reach: Peter Shor’s Prime Factorization Algorithm.

Shor’s algorithm has two essential steps that are unique to quantum computation:

1. Create a superposition of all possible states of the number to be calculated.

2. Apply each state to Shor’s function f. Since the states are a superposition of a group of qubits, this function only needs to be run once to be applied to all states. Transform the superposition to another state that will return the correct answer upon measurement with high probability.

As the algorthim determines the correct answer by running the algorithm repeatedly, one needs to run this algorithm many times [1]. Still, this algorithm is exponentially faster than the current classical algorithm for factorization due to the quantum computer’s ability to have Shor’s function applied to the superposition of all states simultaneously. As a result, Shor’s factorization can be run

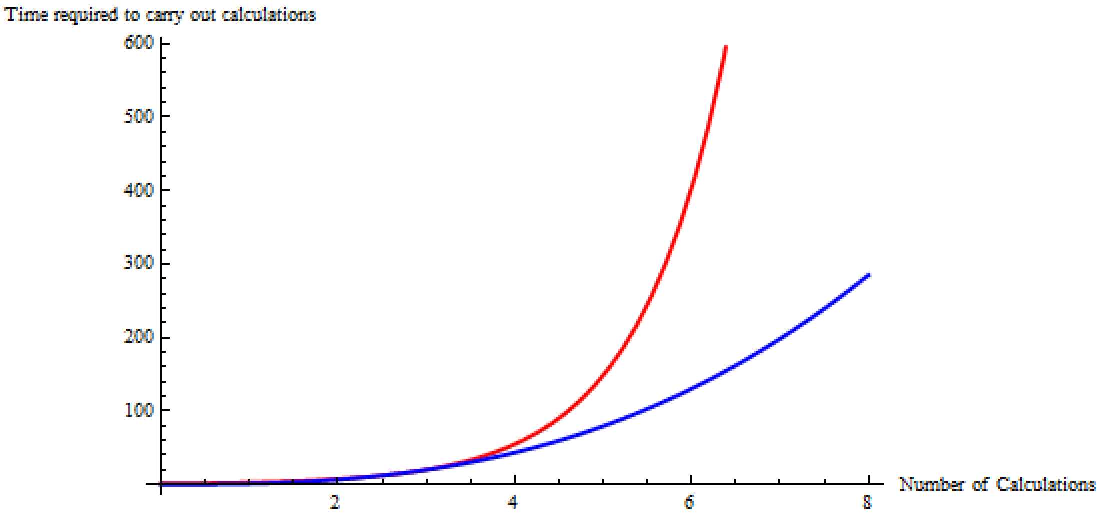

***Figure 2.2.1**: A graph of an exponential function (red) compared with a polynomial function (blue) on scales with arbitrary units. It can be seen above that as the number of calculations gets large, the time required for the exponential function grows far beyond the polynomial function.*

in polynomial time (in this case k^2) on a quantum computer, meaning time increases quadratically as the size of the number grows. For classical computers however, the fastest factorization algorithm runs in exponential time, meaning that computation time is far lengthier than quadratic time at higher number of calculations (see figure 2.3.1).

Thus, realization of Shor's function on a quantum computer would allow for much greater speeds of computation, doing tasks in seconds for which the classical computer would take years.

2.3 Issues with Quantum Computer Construction

Although it is easy to state that three *XOR* gates can be connected to make one *AND* gate, it is notoriously difficult to physically carry this process out [1]. By suggesting that one knows how to "connect" these quantum gates, it is implied that one knows how to control qubit systems to the utmost precision - something currently unfeasible. To date, one cannot manipulate atoms finely enough for such arbitrary connection of quantum gates to happen [1].

Even if one was able to somehow arbitrarily connect quantum gates, another significant issue arising in the construction of quantum computers is the decoherence problem [1]. With a quantum system being as fragile as it is, the decoherence problem arises when an attempt is made to isolate and stabilize qubit systems from changes in its surroundings [1]. After a certain decoherence time, small disturbances in the surrounding environment will cause the quantum system to go out of phase, destroying the intricately built quantum computer [1]. At the current level of technological sophistication, even the slightest disturbance of any type to the qubit would render it unusable [1]. For now, the only solution to this problem is greater accuracy and control in experimental procedures and technology, something which will undoubtedly become better with time [1].

3 Significance

With the steadfast shrinking of silicon-based computers, we see that the current classical model of computer construction is close to collapsing [5]. As stated, it is projected that in ten to fifteen years at the time of writing this paper, classical computers will reach a plateau in computing speeds, effectively limiting transistor-based computing power [5]. With the development of quantum computers however, an alternative to the classical construction of computers is provided that allows growth of computational power for a much longer time [5]. Using quantum computers, realization of computationally intensive prospects such as Artificial Intelligence could be feasible [5]. Additionally, even a quantum computer consisting of a few qubits could be of great scientific interest, allowing physicists to further study the validity of quantum theory [1].

Following the publication of techniques and theories outlined by physicists such as DiVincenzo, there has been a significant number of attempts at building quantum devices [4]. Recently, attempts by IBM researchers following DiVincenzo's theoretical outline have resulted in a 7-qubit quantum computer [2]. This computer, using Shor's Algorithm, has managed to obtain the world record thus far for quantum computation: factoring 15 into 3 × 5 [2]. Despite the success using methods outlined by DiVincenzo, it should be noted that there are several other methods of creating a quantum computer [5]. Although DiVincenzo suggests using a hydrogen qubit with a tipping pulse produced by a magnetic field, there has been notable success with methods such as light-controlled quantum computers [3] as well aluminum ion qubit systems [4]. While DiVicenzo's approach is the most direct, other approaches have their own advantages in areas such as energy conservation and size reduction [5]. Thus, there is no standard in quantum computer construction at this time.

Largely due to its youth, quantum computing remains a much elusive and relatively unexplored subject [5]. With crucial theoretical literature such as that of DiVincenzo's, light is being shed on the dim field of quantum computing. As many experimentalists are following DiVincenzio's model of construction, it is likely that DiVincenzio's work and others like it will become pivotal in establishing a standard for quantum computer construction. With little doubt, one can see that these theoretical works lay the basic foundation of quantum computation and guide experimentalists in creating the first generation of quantum computers.

4 References

1. DiVincenzio, D. 1995. Quantum Computation. Science, **270**: 255-261.

2. Vandersypen, L.M.K., Steffen M., Breyta G., Yannoni C.S., Sherwood M.H., Chuang I.L. Experimental realisation of Shor's quantum factoring algorithm using nuclear magnetic resonance. 2001. Nature, **414**: 883-887.

3. Politi, A., Mathews, J., O'brien, Jeremy. 2009. Shor's Quantum Factoring Algorithm on a Photonic Chip. Science, **325**: 1221.

4. Dicarlo, L., Chow, J.M., Gambetta, J.M., *et al.* 2009. Demonstration of two-qubit algorithms with a superconducting quantum processor. Nature, **460**: 240-244.

5. Kaku, M. 1999. Visions: How Science Will Revolutionize the 21st Century and Beyond. New York: Oxford University Press.