**INTRODUCTION**

The history of computer technology has involved a sequence of changes from one type of physical realisation to another --- from gears to relays to valves to transistors to integrated circuits and so on. Today's advanced lithographic techniques can squeeze fraction of micron wide logic gates and wires onto the surface of silicon chips. Soon they will yield even smaller parts and inevitably reach a point where logic gates are so small that they are made out of only a handful of atoms; i.e. the size of the logic gates become comparable to the size of atoms.

On the atomic scale matter obeys the rules of quantum mechanics, which are quite different from the classical rules that determine the properties of conventional logic gates. So if computers are to become smaller in the future, new, *quantum technology* must replace or supplement what we have now. The point is, however, that quantum technology can offer much more than cramming more and more bits to silicon and multiplying the clock-speed of microprocessors. It can support entirely new kind of computation with qualitatively new algorithms based on quantum principles!

The story of quantum computation started as early as 1982, when the physicist Richard Feynman considered simulation of quantum-mechanical objects by other quantum systems. However, the unusual power of quantum computation was not really anticipated until the 1985 when David Deutsch of the University of
Quantum Computing

Oxford published a crucial theoretical paper in which he described a universal quantum computer. After the Deutsch paper, the hunt was on for something interesting for quantum computers to do. At the time all that could be found were a few rather contrived mathematical problems and the whole issue of quantum computation seemed little more than an academic curiosity. It all changed rather suddenly in 1994 when Peter Shor from AT&T's Bell Laboratories in New Jersey devised the first quantum algorithm that, in principle, can perform efficient factorisation. This became a 'killer application' --- something very useful that only a quantum computer could do.

**GENERAL CONCEPT OF INFORMATION**

To explain what makes quantum computers so different from their classical counterparts we begin by having a closer look at a basic chunk of information namely one bit. A bit is the basic unit of information in a digital computer. From a physical point of view, a bit is a physical system which can be prepared in one of the two different states representing two logical values --- no or yes, false or true, or simply 0 or 1. For example, in digital computers, the voltage between the plates in a capacitor represents a bit of information: a charged capacitor denotes bit value 1 and an uncharged capacitor bit value 0. One bit of information can be also encoded using two different polarizations of light or two different electronic states of an atom. In any of the systems listed above, a bit can store a value of logical 1
Quantum Computing

or logical 0 using some method which depends on the system used. The size of the register to be used would be determined by the maximum value that is to be used \((m)\) and the number of bits in each register is determined using the equation

\[
k = \log_2 n
\]

where \(n\) is the smallest power of 2 greater than or equal to \(m\).

**CONCEPT OF INFORMATION IN QUANTUM COMPUTERS – THE QUBIT**

In quantum computers also, the basic unit of information is a bit. The concept of quantum computing first arose when the use of an atom as a bit was suggested. If we choose an atom as a physical bit then quantum mechanics tells us that apart from the two distinct electronic states (the excited state and the ground state), the atom can be also prepared in what is known as a *coherent superposition* of the two states. This means that the atom can be both in state 0 and state 1 simultaneously. It is at this point that the concept of a quantum bit or a *qubit* arises. This concept is the backbone of the idea of quantum computing. For the same reason, let’s see in detail what actually coherent superposition is.
Coherent Superposition

In any quantum mechanical system, a particular state of the system is represented by a mathematical function called as the wavefunction of that state. A wavefunction is a complex exponential which includes all possible phases of existence of that particular state.

Considering any quantum mechanical system, let $\psi_1$ and $\psi_2$ be two wavefunctions that represent any two independent states of the system. Then quantum mechanics tells us that there exists a state of the same system that can be represented by the wavefunction $c_1\psi_1 + c_2\psi_2$. This state is called as a superposition of the two states represented by $\psi_1$ and $\psi_2$. This would mean that the system would be in both the states of $\psi_1$ and $\psi_2$ simultaneously. All superpositions of two quantum states of a system need not be stable. If the superposition is to be stable, then there should be some sort of coherence between the two states that are being superpositioned. Such a superposition is called as a coherent superposition.

There can be more than one coherent superposition for a particular pair of states of a quantum mechanical system. So in our talk the term ‘coherent superposition’ would refer to that superposition which is the most stable one.
AD VANTAGE OF USING COHERENT-SUPERPOSITIONED MEMORY

The importance of coherent-superpositioned storage can be understood from the following example. Consider a register composed of three physical bits. Any classical register of that type can store in a given moment of time only one out of eight different numbers i.e. the register can be in only one out of eight possible configurations such as 000, 001, 010, ... 111. Consider the case of a quantum register at that place. Since a qubit can store both the values of 0 & 1 simultaneously, a quantum register composed of three qubits can store in a given moment of time all the eight numbers in a quantum superposition. The catch is that the memory size grows exponentially when additional bits are added compared to the linear growth in classical computers. Also, once the register is prepared in such a superposition, operations can be performed on all the numbers simultaneously.
We have seen that once a register is prepared in a superposition of different numbers, we can perform operations on all of them. For example, if qubits are atoms then suitably tuned laser pulses affect atomic electronic states and evolve initial superpositions of encoded numbers into different superpositions. During such evolution each number in the superposition is affected and as the result we generate a massive parallel computation albeit in one piece of quantum hardware. This means that a quantum computer can in only one computational step perform the same mathematical operation on $2^L$ different input numbers encoded in coherent superpositions of $L$ qubits. In order to accomplish the same task, any classical computer has to repeat the same computation $2^L$ times or one has to use $2^L$ different processors working in parallel. In other words a quantum computer offers an enormous gain in the use of computational resources such as time and memory.

But this, after all, sounds as yet another purely technological progress. It looks like classical computers can do the same computations as quantum computers but simply need more time or more memory. The catch is that classical computers need exponentially more time or memory to match the power of quantum computers and this is really asking for too much because an exponential increase is really fast and we run out of available time or memory very quickly.
ALGORITHMS FOR QUANTUM COMPUTERS

In order to solve a particular problem computers follow a precise set of instructions that can be mechanically applied to yield the solution to any given instance of the problem. A specification of this set of instructions is called an algorithm. Examples of algorithms are the procedures taught in elementary schools for adding and multiplying whole numbers; when these procedures are mechanically applied, they always yield the correct result for any pair of whole numbers. Some algorithms are fast (e.g. multiplication) other are very slow (e.g. factorisation). Consider, for example, the following factorisation problem

\[ ? \times ? = 29083 \]

How long would it take you, using paper and pencil, to find the two whole numbers which should be written into the two boxes (the solution is unique)? Probably about one hour. At the same time, solving the reverse problem

\[ 127 \times 129 = ? , \]

again using paper and pencil technique, takes less than a minute. All because we know fast algorithms for multiplication but we do not know equally fast ones for factorisation.
What really counts for a "fast" or a "usable" algorithm, according to the standard definition, is not the actual time taken to multiply a particular pairs of number but the fact that the time does not increase too sharply when we apply the same method to ever larger numbers. The same standard text-book method of multiplication requires little extra work when we switch from two 3-digit numbers to two 30-digit numbers. By contrast, factoring a thirty digit number using the simplest trial division method is about $10^{13}$ times more time or memory consuming than factoring a three digit number. The use of computational resources is enormous when we keep increasing the number of digits. The largest number that has been factorised as a mathematical challenge, i.e. a number whose factors were secretly chosen by mathematicians in order to present a challenge to other mathematicians, had 129 digits. No one can even conceive of how one might factorise say thousand-digit numbers; the computation would take much more than the estimated age of the universe.

Apart from the standard definitions of a fast or a usable algorithm, computer scientists have a rigorous way of defining what makes an algorithm fast (and usable) or slow (and unusable). For an algorithm to be fast, the time it takes to execute the algorithm must increase no faster than a polynomial function of the size of the input. Informally, think about the input size as the total number of bits needed to specify the input to the problem, for example, the number of bits needed to encode the number we want to factorise. If the best algorithm we know for a particular problem has the execution time (viewed as a function of the size of the
Quantum Computing

input) bounded by a polynomial then we say that the problem belongs to class \( P \). Problems outside class \( P \) are known as hard problems. Thus we say, for example, that multiplication is in \( P \) whereas factorisation is not in \( P \) and that is why it is a hard problem. Hard does not mean ‘impossible to solve’ or ‘non-computable’ --- factorisation is perfectly computable using a classical computer, however, the physical resources needed to factor a large number are such that for all practical purposes, it can be regarded as intractable. Purely technological progress can only increase the computational speed by a fixed multiplicative factor which does not help to change the exponential dependence between the size of the input and the execution time. Such change requires inventing new, better algorithms. Although quantum computation requires new quantum technology, its real power lies in new quantum algorithms which allow exploiting quantum superposition that can contain an exponential number of different terms. Quantum computers can be programmed in a qualitatively new way. For example, a quantum program can incorporate instructions such as ‘... and now take a superposition of all numbers from the previous operations...’; this instruction is meaningless for any classical data processing device but makes lots of sense to a quantum computer. As the result we can construct new algorithms for solving problems, some of which can turn difficult mathematical problems, such as factorisation, into easy ones!
Demonstrating Quantum Computing

Due to technical obstacles, till date, a quantum computer has not yet been realized. But the concepts and ideas of quantum computing has been demonstrated using various methods. Here, we discuss four of the most important technologies that are used to demonstrate quantum computing. They are

1. Nuclear Magnetic Resonance
2. Ion Trap
3. Quantum Dot
4. Optical Methods

While reading the following "top four technologies", two things should be kept in mind. The first is that the list will change over time. Some of the approaches valuable for exploring quantum computing in the laboratory are fundamentally un-scalable, and so will drop out of contention over the next few years. The second thing to keep in mind is that although there are a bewildering number of proposed methods for demonstrating quantum computing (a careful search will yield many more options that what is listed here); all of them are variations on three central themes:

(a) manipulating the spin of a nucleus or subatomic particle
(b) manipulating electrical charge
(c) manipulating the polarization of a photon.
Quantum Computing

In variation "a" a qubit is derived from superposition of up and down spins. In variation "b" a qubit is derived from superposition of two or more discrete locations of the charge. In the last variation a qubit is derived from superposition of polarization angles. Of the three, the manipulation of spin is generally viewed as the most promising for practical large-scale application. Let’s now see each of these techniques in detail.

1. **Nuclear Magnetic Resonance**

Using nuclear magnetic resonance (NMR) techniques, invented in the 1940's and widely used in chemistry and medicine today, these spins can be manipulated, initialized and measured. Most NMR applications treat spins as little "bar magnets", whereas in reality, the naturally well-isolated nuclei are non-classical objects.

A Nuclear Magnetic Resonance (NMR) quantum computer is based on control of nuclear spin. In demonstrations to date this has been achieved by manipulating the nuclear spins of several atoms making up a molecule - the most recent effort using a molecule of five fluorine atoms and two carbon atoms. The spin manipulation is accomplished by application of magnetic pulses within a magnetic field produced by the NMR chamber.

The quantum behavior of the spins can be exploited to perform quantum computation. Magnetic field produced by the NMR chamber can be used to
Quantum Computing

manipulate the spin state of the nucleus. The manipulation of the spin is accomplished by application of magnetic pulses in the chamber. The entanglement of spins required to establish a qubit is created by the chemical bonds between neighboring atoms within a molecule - within $10^{18}$ molecules to be more precise, since a measurable signal (relative to the background noise from kinetic energy) is achieved by using a test tube of "processing liquid" rather than a single molecule.

For example, consider the carbon and hydrogen nuclei in a chloroform molecule to be representing two qubits. Applying a radio-frequency pulse to the hydrogen nucleus addresses that qubit, and causes it to rotate from a $|0\rangle$ state to a superposition state. Interactions through chemical bonds allow multiple-qubit logic to be performed. In this manner, applying newly developed techniques to allow bulk samples with many molecules to be used, small-scale quantum algorithms have been experimentally demonstrated with molecules such as Alanine, an amino acid. This includes the quantum search algorithm, and a predecessor to the quantum factoring algorithm.

The major drawback of this method is scalability; the signal strength of the answer decreases exponentially with the number of qubits.

2. **Ion Trap**

An Ion Trap quantum computer is also based on control of nuclear spin (although using vibration modes or "phonons" has also been considered). In this approach the individual ions are, as the name implies, trapped or isolated by
Quantum Computing

Quantum Computing

means of an electromagnetic field which is produced by means of an
electromagnetic chamber.

Ordinarily, the energy difference between different spin states in nuclei is
so small relative to the kinetic energy of the ions, that they are not measurable. In
the prior technique (NMR) this problem was overcome by operating
simultaneously on a large number of atoms. But in this case, the solution is a bit
different. The trapped ions are cooled to the point where motion is essentially
eliminated. They are then manipulated by laser pulses and a qubit arises from the
superposition of lower and higher energy spin states.

This technique is potentially scalable, but a great disadvantage is that it
requires a cryogenic environment - not to mention that to date no more than single
qubit systems have been demonstrated.

3. Quantum Dot

A quantum dot is a particle of matter so small that the addition or removal
of an electron changes its properties in some useful way. All atoms are, of course,
quantum dots, but multi-molecular combinations can have this characteristic. In
biochemistry, quantum dots are called redox groups. Quantum dots typically have
dimensions measured in nanometers, where one nanometer is $10^{-9}$ meter or a
millionth of a millimeter.
Quantum Computing

A Quantum Dot quantum computer can involve manipulation of electrical charge, spin, or energy state - the Australians have a patent on a spin based version. The idea is that a small number of electrons or possibly an individual electron is confined with a quantum dot, the quantum dot typically being a small "hill" of molecules grown on a silicon substrate. A computer would be made up of a regular array of such dots. As with the prior two methods, the most popular approach is to have spin up counted as zero, spin down counted as one, and use a superposition of spin states to create the qubit. Techniques for self assembly of large arrays of quantum dots have already been demonstrated and can be done using the industry standard silicon substrate. Thus, of all the approaches listed here, this one seems to have the highest potential for commercial scalability.

4. Optical

As the name indicates, an optical quantum computer uses the two different polarizations of a light beam to represent two logical states. As an example, we can consider the polarization of a light beam in the vertical plane to represent a logical 1 and the polarization of the beam in the horizontal plane to represent a logical 0. An Optical quantum computer would be based on manipulating the polarization of individual photons. Entanglement is achieved by coincident creation of identical photons. Identical photons in this context would mean photons having the same energy as well as same polarization. The superposition of
Quantum Computing

polarization or phase state is manipulated using polarizing lenses, phase shifters, and beam splitters.

It was originally believed that non-linear optics would be required in order to create a working quantum computer, and this was considered to be the major technical obstacle to a photon-based approach. However, recent theoretical advances indicate that linear optics is sufficient. Several laboratories are working on a practical demonstration. The new stumbling block has been creating beam splitters of sufficient accuracy. The entire process can be carried out at room temperature and there is no reason, in principle, that it is not scalable to large numbers of qubites.
Quantum computing principles use the principle of coherent superposition storage. As stated in the above example, it is quite remarkable that all eight numbers are physically present in the register but it should be no more surprising than a qubit being both in state 0 and 1 at the same time. If we keep adding qubits to the register we increase its storage capacity exponentially i.e. three qubits can store 8 different numbers at once, four qubits can store 16 different numbers at once, and so on; in general L qubits can store $2^L$ numbers at once. Once the register is prepared in a superposition of different numbers we can perform operations on all of them. For example, if qubits are atoms then suitably tuned laser pulses affect atomic electronic states and evolve initial superpositions of encoded numbers into different superpositions. During such evolution each number in the superposition is affected and as the result we generate a massive parallel computation albeit in one piece of quantum hardware. This means that a quantum computer can in only one computational step perform the same mathematical operation on $2^L$ different input numbers encoded in coherent superpositions of L qubits. In order to accomplish the same task any classical computer has to repeat the same computation $2^L$ times or one has to use $2^L$ different processors working in parallel. In other words a quantum computer offers an enormous gain in the use of computational resources such as time and memory. It looks like classical computers can do the same computations as quantum
Quantum Computing

computers but simply need more time or more memory. The catch is that classical computers need *exponentially* more time or memory to match the power of quantum computers and this is really asking for too much because an exponential increase is really fast and we run out of available time or memory very quickly.

**WHAT WILL QUANTUM COMPUTERS BE GOOD AT?**

These are the most important applications currently known:

- Cryptography: Perfectly secure communication.
- Searching, especially algorithmic searching (Grover's algorithm).
- Factorizing large numbers very rapidly (Shor's algorithm).
- Simulating quantum-mechanical systems efficiently.
Quantum Computing

**OBSTACLES AND RESEARCH**

The field of quantum information processing has made numerous promising advancements since its conception, including the building of two- and three-qubit quantum computers capable of some simple arithmetic and data sorting. However, a few potentially large obstacles still remain that prevent us from "just building one", or more precisely, building a quantum computer that can rival today's modern digital computer. Among these difficulties, error correction, decoherence, and hardware architecture are probably the most formidable.

**Decoherence**

We have seen that if a superposition of any two states of a quantum – mechanical system is to be stable over a period of time, there should be some sort of coherence between the states that are being superpositioned. But still, no superposition of any pair of given states are perfectly stable to any extent. This is because of the existence of a property by name *decoherence* which forces a quantum – mechanically superpositioned system to decay from a given quantum coherent – superpositioned state into an incoherent state as it entangles or interacts with the surroundings. The final effect is that the coherence between the two states that are superpositioned would be gradually “lost”. For the same reason, no quantum memory can, at present, be used to hold data that is to be used for
Quantum Computing

operations that take a long time. The time taken by the system to lose the coherence between the two states is known as decoh\r\nherence time.

Error Correction

Error correction is rather self explanatory, but what errors need correction? The answer is primarily those errors that arise as a direct result of decoherence, or the tendency of a quantum computer to decay from a given quantum state into an incoherent state as it interacts, or entangles, with the state of the environment. These interactions between the environment and qubits are unavoidable, and induce the breakdown of information stored in the quantum computer, and thus errors in computation. Before any quantum computer will be capable of solving hard problems, research must devise a way to maintain decoherence and other potential sources of error at an acceptable level.

Thanks to the theory (and now reality) of quantum error correction, first proposed in 1995 and continually developed since, small scale quantum computers have been built and the prospects of large quantum computers are looking up. Probably the most important idea in this field is the application of error correction in phase coherence as a means to extract information and reduce error in a quantum system without actually measuring that system. In 1998, researches at Los Alamos National Laboratory and MIT managed to spread a single bit of quantum information (qubit) across three nuclear spins in each molecule of a liquid solution of alanine or trichloroethylene molecules. They accomplished this using the techniques of nuclear magnetic resonance (NMR). This experiment is
Quantum Computing

significant because spreading out the information actually made it harder to corrupt. Quantum mechanics tells us that directly measuring the state of a qubit invariably destroys the superposition of states in which it exists, forcing it to become either a 0 or 1.

The technique of spreading out the information allows researchers to utilize the property of entanglement to study the interactions between states as an indirect method for analyzing the quantum information. Rather than a direct measurement, the group compared the spins to see if any new differences arose between them without learning the information itself. This technique gave them the ability to detect and fix errors in a qubit's phase coherence, and thus maintain a higher level of coherence in the quantum system. This milestone has provided argument against skeptics, and hope for believers. At this point, only a few of the benefits of quantum computation and quantum computers are readily obvious, but before more possibilities are uncovered theory must be put to the test. In order to do this, devices capable of quantum computation must be constructed. Quantum computing hardware is, however, still in its infancy.

As a result of several significant experiments, nuclear magnetic resonance (NMR) has become the most popular component in quantum hardware architecture. Only within the past year, a group from LOS Alamos National Laboratory and MIT constructed the first experimental demonstrations of a quantum computer using nuclear magnetic resonance (NMR) technology. Currently, research is underway to discover methods for battling the destructive
Quantum Computing

effects of decoherence, to develop optimal hardware architecture for designing and building a quantum computer, and to further uncover quantum algorithms to utilize the immense computing power available in these devices. Naturally this pursuit is intimately related to quantum error correction codes and quantum algorithms, so a number of groups are doing simultaneous research in a number of these fields.

To date, designs have involved ion traps, cavity quantum electrodynamics (QED), and NMR. Though these devices have had mild success in performing interesting experiments, the technologies each have serious limitations. Ion trap computers are limited in speed by the vibration frequency of the modes in the trap. NMR devices have an exponential attenuation of signal to noise as the number of qubits in a system increases. Cavity QED is slightly more promising; however, it still has only been demonstrated with a few qubits. The future of quantum computer hardware architecture is likely to be very different from what we know today; however, the current research has helped to provide insight as to what obstacles the future will hold for these devices.

Lack of Reliable Reading Mechanism

The techniques that exist till date have a big problem that trying to read from a superpositioned qubit would invariably make it to lose its superpositioned state and make it to behave just as a classical bit – i.e. it would store only one among the values of 0 and 1.
Quantum Computing

Also, if we are given a quantum register comprising of $n$ bits of which $m$ bits are superpositioned ones, none among the reading mechanisms available today is able to determine which value from the superposition is to be read out. i.e. if we are given a 3-bit register that contains, say, 4 values in a particular superposition (let them be 4,5,6,7), the reading mechanisms available today are unable to determine which how to access a specific value from the superposition.
Quantum Computing

**FUTURE OUTLOOK**

At present, quantum computers and quantum information technology remains in its pioneering stage. At this very moment obstacles are being surmounted that will provide the knowledge needed to thrust quantum computers up to their rightful position as the fastest computational machines in existence. Error correction has made promising progress to date, nearing a point now where we may have the tools required to build a computer robust enough to adequately withstand the effects of decoherence. Quantum hardware, on the other hand, remains an emerging field, but the work done thus far suggests that it will only be a matter time before we have devices large enough to test Shor's and other quantum algorithms. Thereby, quantum computers will emerge as the superior computational devices at the very least, and perhaps one day make today's modern computer obsolete. Quantum computation has its origins in highly specialized fields of theoretical physics, but its future undoubtedly lies in the profound effect it will have on the lives of all mankind.
**Desirable Features of an Ideal System**

Hoping for the best, if ever a quantum computer is being practically realized, then an ideal one should have some desirable features. They are as listed below.

1. **Be a Scalable Physical System**

   Scalability in this context refers to the change that occurs in the signal strength of the system that would take place when the number of qubits keeps on increasing. The clause ‘a scalable physical system’ refers to a system which, under ideal conditions, does not suffer from any kinds of loss of signal strength when the number of qubits increases. If a system suffers from such a loss, then the problem that arises would be that the system cannot afford to handle more than a fixed number of qubits totally, which would lead to the occurrence of a large obstacle in the construction of an efficient quantum computer.

2. **Be Initializable to a Simple State**

   The term *Initialization* would have the same meaning it has as far as classical computers are considered. i.e. If a particular address or a memory
Quantum Computing

location is given, then we should be able to initialize it to a particular state, may it be a single state or a superpositioned state by a single step, or at least, using a very few number of steps.

3. Have Much Long Decoherence Times

As we have seen, no quantum memory is till date completely free from the grip of decoherence. For the same reason, we cannot completely avoid this evil. So our aim should be as to reduce the troubles caused by it. One method to achieve this is to use memory having much longer decoherence time so that by the time the coherence between the states is lost, the required operation would have been performed on the stored data. Another approach is to improve the algorithms used, so that the time that a data is to be stored in a location would decrease. But the latter approach is left to the person who designs the algorithm, and therefore outside the control of the hardware designer or manufacturer.
CONCLUSION

Experimental and theoretical research in quantum computation is accelerating world-wide. New technologies for realizing quantum computers are being proposed, and new types of quantum computation with various advantages over classical computation are continually being discovered and analysed and we believe some of them will bear technological fruit. From a fundamental standpoint, however, it does not matter how useful quantum computation turns out to be, nor does it matter whether we build the first quantum computer tomorrow, next year or centuries from now. The quantum theory of computation must in any case be an integral part of the world view of anyone who seeks a fundamental understanding of the quantum theory and the processing of information.
Quantum Computing

REFERENCES

1. www.qubit.org

2. http://tph.tuwien.ac.at