A

Seminar report

On

# Quantum Computing

Submitted in partial fulfillment of the requirement for the award of degree
of Bachelor of Technology in Computer Science

**SUBMITTED TO:**

www.studymafia.org

**SUBMITTED BY:**

www.studymafia.org

# Acknowledgement

I would like to thank respected Mr…….. and Mr. ……..for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present a seminar report. It helped me a lot to realize of what we study for.

Secondly, I would like to thank my parents who patiently helped me as i went through my work and helped to modify and eliminate some of the irrelevant or un-necessary stuffs.

Thirdly, I would like to thank my friends who helped me to make my work more organized and well-stacked till the end.

Next, I would thank Microsoft for developing such a wonderful tool like MS Word. It helped my work a lot to remain error-free.

Last but clearly not the least, I would thank The Almighty for giving me strength to complete my report on time.

# Preface

I have made this report file on the topic **Quantum Computing**; I have tried my best to elucidate all the relevant detail to the topic to be included in the report. While in the beginning I have tried to give a general view about this topic.

My efforts and wholehearted co-corporation of each and everyone has ended on a successful note. I express my sincere gratitude to …………..who assisting me throughout the preparation of this topic. I thank him for providing me the reinforcement, confidence and most importantly the track for the topic whenever I needed it.

# *WHAT'S QUANTUM COMPUTING?*
## *An introduction*

What is quantum computing? It's something that could have been thought up a long time ago - an idea whose time has come. For any physical theory one can ask: what sort of machines will do useful computation? or, what sort of processes will count as useful computational acts? Alan Turing thought about this in 1936 with regard (implicitly) to classical mechanics, and gave the world the paradigm classical computer: the Turing machine.

But even in 1936 classical mechanics was known to be false. Work is now under way - mostly theoretical, but tentatively, hesitantly groping towards the practical - in seeing what quantum mechanics means for computers and computing.

In a trivial sense, everything is a quantum computer. (A pebble is a quantum computer for calculating the constant-position function - you get the idea.) And of course, today's computers exploit quantum effects (like electrons tunneling through barriers) to help do the right thing and do it fast. For that matter, both the computer and the pebble exploit a quantum effect - the "Pauli exclusion principle", which holds up ordinary matter against collapse by bringing about the kind of degeneracy we call chemistry - just to remain stable solid objects. But quantum computing is much more than that.

The most exciting really new feature of quantum computing is **quantum parallelism**. A quantum system is in general not in one "classical state", but in a "quantum state" consisting (crudely speaking) of a superposition of many classical or classical-like states. This superposition is not just a figure of speech, covering up our ignorance of which classical-like state it's "really" in. If that was all the superposition meant, you could drop all but one of the classical-like states (maybe only later, after you deduced retrospectively which one was "the right one") and still get the time evolution right. But actually you need the whole superposition to get the time evolution right. The system really is in some sense in all the classical-like states at once! If the superposition can be protected from unwanted entanglement with its environment (known as decoherence), a quantum computer can output results dependent on details of all its classical-like states. This is quantum parallelism - parallelism on a serial machine. And if that wasn't enough, machines that would already, in architectural terms, qualify as parallel can benefit from quantum parallelism too - at which point the mind begins to seriously boggle!

# *PRINCIPLE*

The advantage of quantum computers arises from the way they encode a bit, the fundamental unit of information. One number, 0 or 1 specifies the state of a bit in a classical digital computer. An n bit binary word in a typical computer is accordingly described by a string of n zeros and ones. An atom might represent a quantum bit, called a qubit in one or two different states, which can also be denoted as 0 or 1. Two qubits like two classical bits can attain four different well-defined states (00, 01, 10 and 11).

But unlike classical bits, qubits can exist simultaneously as o and 1, with the probability for each state given by a numerical coefficient; describing a two-qubit quantum computer thus requires four coefficients. In general, n qubits demand 2 exp n numbers, which rapidly becomes a sizable set for larger values of n. for example, if n equals 50, about 10 exp 15 numbers are required to describe all the probabilities for all the possible states of a quantum machine – a number that exceeds the capacity of the largest conventional computer, a quantum computer promises to be immensely powerful because it can be in multiple states at once-a phenomenon called superposition—and because it can act on all its possible states simultaneously. Thus a quantum computer could naturally perform myriad operations in parallel, using a single processing unit.

## *QUANTUM TECHNOLOGY*

A quantum computer - a new kind of computer far more powerful than any that currently exist - could be made today, say Thaddeus Ladd of Stanford University, Kohei Itoh of Keio University in Japan, and their co-workers. They have sketched a blueprint for a silicon quantum computer that could be built using current fabrication and measurement techniques[1].

The microelectronics industry has decades of experience of controlling and fine-tuning the structure and properties of silicon. These skills would give a silicon-based quantum computer a head start over other schemes for putting one together.

**Silicon technology has a head start over other quantum**

Quantum and conventional computers encode, store and manipulate information as sequences of binary digits, or bits, denoted as 1s and 0s. In a normal computer, each bit is a switch, which can be either 'on' or 'off'.

In a quantum computer, switches can be on, off or in a superposition of states - on and off at the same time. These extra configurations mean that quantum bits, or qubits, can encode more information than classical switches.

That increase in capacity would, in theory, make quantum computers faster and more powerful. In practice it is extremely difficult to maintain a superposition of more than a few quantum states for any length of time. So far, quantum computing has been demonstrated with only four qubits, compared with the billions of bits that conventional silicon microprocessors handle.

Several quantum-computing demonstrations have used nuclear magnetic resonance (NMR) to control and detect the quantum states of atoms floating in solution. But this beaker-of-liquid approach is unlikely to remain viable beyond ten or so qubits.

Many researchers suspect that making a quantum computer with as many qubits as a Pentium chip has transistors will take the same kind of technology, recording the information in solid-state devices.

# *QUANTUM COMPUTER BASICS*

In the classical model of a computer, the most fundamental building block, the bit, can only exist in one of two distinct states, a 0 or a 1. In a quantum computer the rules are changed .Not only can a 'quantum bit', usually referred to as a 'qubit', exist in the classical 0 and 1 states, it can also be in a coherent superposition of both. When a qubit is in this state it can be thought of as existing in two universes, as a 0 in one universe and as a 1 in the other. An operation on such a qubit effectively acts on both values at the same time. The significant point being that by performing the single operation on the qubit, we have performed the operation on two different values. Likewise, a two-qubit system would perform the operation on 4 values, and a three-qubit system on eight. Increasing the number of qubits therefore exponentially increases the 'quantum parallelism' we can obtain with the system. With the correct type of algorithm it is possible to use this parallelism to solve certain problems in a fraction of the time taken by a classical computer.

## THE PITFALL OF QUANTUM COMPUTING-DECOHERENCE

The very thing that makes quantum computing so powerful, its reliance on the bizarre subatomic goings-on governed by the rules of quantum mechanics, also makes it very fragile and difficult to control. For example, consider a qubit that is in the coherent state. As soon as it measurable interacts with the environment it will decohere and fall into one of the two classical states. This is the problem of decoherence and is a stumbling block for quantum computers as the potential power of quantum computers depends on the quantum parallelism brought about by the coherent state This problem is compounded by the fact that even looking at a qubit can cause it to decohere, making the process of obtaining a solution from a quantum computer just as difficult as performing the calculation itself.

## GETTING A RESULT

Once a calculation that makes use of quantum parallelism has been performed, there will be any number of different results in different universes. The fact that the results are not in this universe means that we can only obtain a solution to a computation by looking at the interference of the various results. It is important to note that looking at the result (or any intermediate state) of a quantum computer prevents any further interference between the different versions from taking place, i.e. prevents any useful quantum computations from continuing. Such interference is best illustrated with a simple example; In Young's two slit experiment, light is shone through two parallel slits onto a screen. The resulting pattern of light and dark fringes displayed on the screen is a result of constructive and destructive interference. In a similar way, the results from each universe's calculation will constructively and destructively interfere to give a measurable result. This result has a different significance for different algorithms, and can be used to deduce the solution to the problem in hand (For an example see Shor's algorithm - An example).

## *Computing at atomic scale*

Quantum computers will perform computations at the atomic scale. Fig. 1 shows a survey made by Keyes in 1988 .The number of dopant impurities required for logic in the bases of bipolar transistors is plotted against the year. This plot may be thought of as showing the number of electrons required to store a single bit of information. An extrapolation of the plot suggests that we might be within the reach of atomic-scale computations within the next two decades.

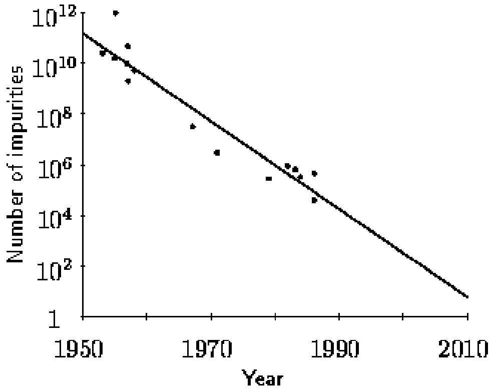


**Fig. 1** Plot showing the number of dopant impurities involved in logic in bipolar transistors with year.

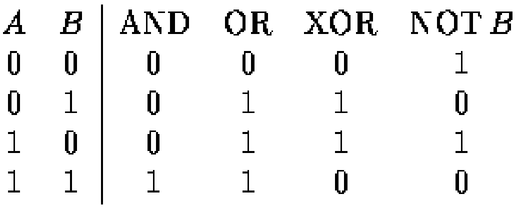
## *Reversible computation*

What are the difficulties in trying to build a classical computing machine on such a small scale? One of the biggest problems with the program of miniaturizing conventional computers is the difficulty of dissipated heat. As early as 1961 Landauer studied the physical limitations placed on computation from dissipation .Surprisingly, he was able to show that almost all operations required in computation could be performed in a reversible manner, thus dissipating no heat! The first condition for any deterministic device to be reversible is that its input and output be uniquely retrievable from each other. This is called logical reversibility. If, in addition to being logically reversible, a device can actually run backwards then it is called physically reversible and the second law of thermodynamics guarantees that it dissipates no heat.

The work on classical, reversible computation has laid the foundation for the development of quantum mechanical computers. On a quantum computer, programs are executed by unitary evolution of an input that is given by the state of the system. Since all unitary operators **U** are invertible with$U^{-1} = U^{\dagger}$, we can always ``uncompute" (reverse the computation) on a quantum computer.

## *Classical universal machines and logic gates:*

We now review the basic logic elements used in computation and explain how conventional computers may be used for any ``reasonable" computation. A reasonable computation is one that may be written in terms of some (possibly large) Boolean expression, and any Boolean expression may be constructed out of a fixed set of logic gates. Such a set (e.g., AND, OR and NOT) is called universal. In fact we can get by with only two gates, such as AND and NOT or OR and NOT. Alternatively, we may replace some of these primitive gates by others, such as the exclusive-OR (called XOR); then AND and XOR form a universal set. The truth tables for these gates are displayed in Table 1. Any machine which can build up arbitrary combinations of logic gates from a universal set is then a universal computer.



| $A$ | $B$ | AND | OR | XOR | NOT $B$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |

**Table 1** Truth table defining the operation of some elementary logic gates. Each row shows two input values **A** and **B** and the corresponding output values for gates AND, OR and XOR. The output for the NOT gate is shown only for input **B**.

Which of the above gates is reversible? Since AND, OR, and XOR are many-to-one operations they are not, as they stand, logically reversible. Before we discuss how these logic gates may be made reversible we consider some non-standard gates that we shall require.

## *Elementary quantum notation:*

A simple quantum system is the two-level spin-1/2 particle. Its basis states, spin-down $|\downarrow\rangle$and spin-up $|\uparrow\rangle$, may be relabelled to represent binary zero and one, i.e., |0>and |1>, respectively. The state of a single such particle is described by the wave function$\psi = \alpha|0\rangle + \beta|1\rangle$. The squares of the complex coefficients $|\alpha|^2$and $|\beta|^2$represent the probabilities for finding the particle in the corresponding states. Generalizing this to a set of **k** spin-1/2 particles we find that there are now $2^k$ basis states (quantum mechanical vectors that span a Hilbert space) corresponding say to the $2^k$ possible bit-strings of length **k**. For example, $|25\rangle = |11001\rangle = |\uparrow\uparrow\downarrow\downarrow\uparrow\rangle$is one such state for **k=5**.

The dimensionality of the Hilbert space grows exponentially with **k**. In some very real sense quantum computations make use of this enormous size latent in even the smallest systems.

## *Logic gates for quantum bits:*

In this section we describe how arbitrary logic gates may be constructed for quantum bits. We start by considering various one-bit unitary operations and a single two-bit one---the XOR operation. Combinations of these are sufficient to construct a Toffoli gate for quantum bits or indeed any unitary operation on a finite number of bits.

Start with a single quantum bit. If we represent the states $|\downarrow\rangle$and $|\uparrow\rangle$(i.e., |0> and |1>) as the vectors $\binom{1}{0}$ and $\binom{0}{1}$, respectively, then the most general unitary transformation corresponds to a 2x2 matrix of the form

$$U_\theta \equiv \begin{pmatrix} e^{i(\delta+\sigma+\tau)}\cos(\theta/2) & e^{-i(\delta+\sigma-\tau)}\sin(\theta/2) \\ -e^{i(\delta-\sigma+\tau)}\sin(\theta/2) & e^{i(\delta-\sigma-\tau)}\cos(\theta/2) \end{pmatrix} ,$$

Where we typically take $\delta = \sigma = \tau = 0$. Using this operator we can flip bits via:

$$U_\pi|0\rangle = -|1\rangle \text{ , and } U_\pi|1\rangle = |0\rangle \text{ .}$$

The extraneous sign represents a phase factor that does not affect the logical operation of the gates and may be removed if we wish, now or at a later stage. Such one-bit computations are illustrated schematically as a quantum circuit in Fig. 5 .

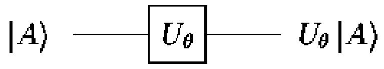


**Fig. 5** Schematic of the quantum circuit diagram for a one-bit gate. The line represents a single quantum bit (such as a spin-1/2 particle). Initially, this bit has a state described by |A>; after it has ``passed" through this circuit it comes out in the state $U_\theta\,|A\rangle$.

Another important one-bit gate is $U_{-\pi/2}$ which maps a spin-down particle

$$U_{-\pi/2}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \text{ ,}$$

to an equal superposition of down and up. Consider a string of **k** spin-1/2 particles initially spin-down. If we apply this gate independently to each particle we obtain a superposition of every possible bit-string of length **k**:

$$|0\rangle \longrightarrow \frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a\rangle \text{ ,}$$

where $q = 2^k$. Our computer is now in a superposition of an exponentially large number of integers **a** from **0** to $2^k - 1$. Suppose we could now construct a unitary operation which maps a pair of bit-strings |a;0> into the pair |a;f(a)> for some function f(a). Then such a unitary operator acting on the superposition of states

$$\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a;0\rangle \longrightarrow \frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a;f(a)\rangle \ ,$$

would compute f(a) in parallel an exponentially large number of times for the various inputs **a**.

To see how such unitary operators may be constructed from a few elementary ones we must also consider the XOR gate . Writing the two-particle basis states as the vectors

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix},$$
$$|10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix},$$

we may represent the XOR gate as a unitary operator

$$U_{\text{XOR}} \equiv \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix} .$$

Here the first particle acts as a conditional gate to flip the state of the second particle. It is easy to check that the state of the second particle corresponds to the action of the XOR gate given in Table. The quantum circuit for an XOR gate is illustrated in Fig. 6. This circuit is equivalent to the elementary instruction

if ( |A>  = 1 ) |B> = NOT |B>

which may be thought of as example of quantum computer code. The ket-brackets | >are reminders that we are dealing with quantum rather than classical bits. The XOR gate allows us to move information around as is illustrated in Fig. 7.

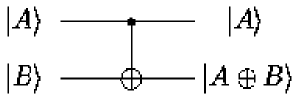


**Fig. 6** Quantum circuit diagram for an XOR gate. The lower bit |B> is flipped whenever the upper bit |A> is set.

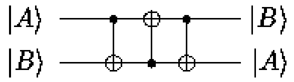


**Fig. 7** Circuit for swapping a pair of bits.

How do we construct the Toffoli gate? One major problem with this gate is that it requires three bits in and three out. Quantum mechanically, this seems to correspond to a scattering process involving three-particle collisions calling for a (possibly) unreasonable control of the particles . Fortunately, the Toffoli gate may be constructed by two-particle scattering processes alone . In particular, we show a construction here involving the XOR gate and some one-bit gates $U_\theta$ (Fig. 8) . Not only is the XOR sufficient for all logic operations on a quantum computer, but it can be used to construct arbitrary unitary transformations on any finite set of bits. Numerous proposals for producing such gates have been considered .

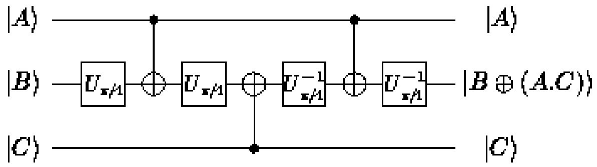


**Fig. 8** Toffoli gate built from two-bit XOR gates plus some one-bit gates . This circuit introduces some extra signs in the unitary matrix $U_{\mathrm{XOR}}$ which may be removed at a later stage.

## *Quantum parallelism: Period of a sequence:*

We now have sufficient ingredients to understand how a quantum computer can perform logical operations and compute just like an ordinary computer. In this section we describe an algorithm which makes use of the quantum parallelism that we have hinted at already: finding the period of a long sequence.

Consider the sequence

$$f(0),\ f(1),\ \ldots,\ f(q-1)\ ,$$

where $q = 2^k$; we shall use quantum parallelism to find its period. We start with a set of initially spin-down particles which we group into two sets (two quantum registers, or quantum variables):

$$|0;0\rangle = |\downarrow,\downarrow,\ldots;\downarrow,\downarrow,\ldots\rangle\ ,$$

the first set having **k** bits; the next having sufficient for our needs. (In fact other registers are required, but by applying Bennett's solution to space management they may be suppressed in our discussion here.) On each bit of the first register we perform the $U_{-\pi/2}$ one-bit operation, yielding a superposition of every possible bit-string of length **k** in this register:

$$\longrightarrow \frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a;\, 0\rangle \,.$$

The next stage is to break down the computation, corresponding to the function f(a), into a set of one-bit and two-bit unitary operations. The sequence of operations is designed to map the state |a;0> to the state |a;f(a)> for any input **a**. Now we see that the number of bits required for this second register must be at least sufficient to store the longest result f(a) for any of these computations. When, however, this sequence of operations is applied to our exponentially large superposition, instead of the single input, we obtain

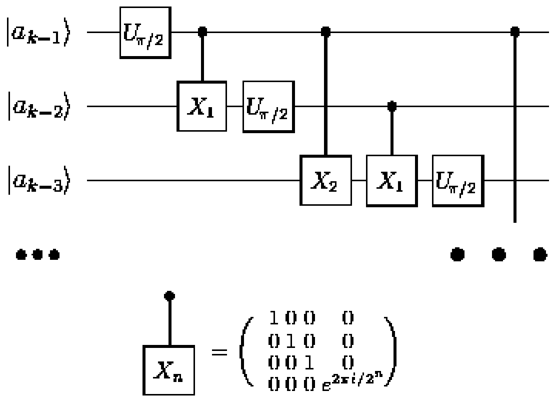
$$\longrightarrow \frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a;\, f(a)\rangle \,.$$

An exponentially large amount of computation has been performed essentially for free.

The final computational step, like the first, is again a purely quantum mechanical one. Consider a discrete ``quantum'' Fourier transform on the first register

$$|a\rangle \longrightarrow \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} e^{2\pi i a c/q} |c\rangle \,.$$

It is easy to see that this is reversible via the inverse transform and indeed it is readily verified to be unitary. Further, an efficient way to compute this transform with one-bit and two-bit gates has been described by Coppersmith (Fig. 10)



$|a_{k-1}\rangle$
$|a_{k-2}\rangle$
$|a_{k-3}\rangle$
$U_{\pi/2}$
$X_1$
$U_{\pi/2}$
$X_2$
$X_1$
$U_{\pi/2}$
$X_n$
$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{2\pi i/2^n} \end{pmatrix}$

**Fig. 10** Circuit for the quantum Fourier transform of the variable $|a_{k-1}\ldots a_1 a_0\rangle$ using Coppersmith's fast Fourier transformation approach The two-bit ``$X_n$'' gate may itself be decomposed into various one-bit and XOR gates

When this quantum Fourier transform is applied to our superposition, we obtain

$$\longrightarrow \frac{1}{q}\sum_{a=0}^{q-1}\sum_{c=0}^{q-1} e^{2\pi i a c/q}|c;\, f(a)\rangle \,.$$

The computation is now complete and we retrieve the output from the quantum computer by measuring the state of all spins in the first register (the first **k** bits). Indeed, once the Fourier transform has been performed the second register may even be discarded .

What will the output look like? Suppose f(a) has period **r** so f(a+r)=f(a). The sum over **a** will yield constructive interference from the coefficients $e^{2\pi i a c/q}$ only when c/q is a multiple of the reciprocal period 1/r All other values of c/q will produce destructive interference to a greater or lesser extent. Thus, the probability distribution for finding the first register with various values is shown schematically by Fig. 11.

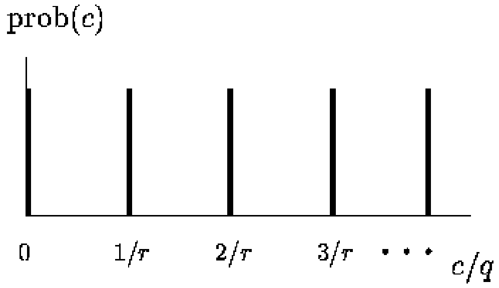


**Fig. 11** Plot of the probability of each result prob c versus c/q. Constructive interference produces narrow peaks at multiples of the inverse period of the sequence 1/r .

One complete run of the quantum computer yields a random value of c/q underneath one of the peaks in the probability of each result prob( c ). That is, we obtain a random multiple of the inverse period. To extract the period itself we need only repeat this quantum computation roughly $\log\log r/k$ times in order to have a high probability for at least one of the multiples to be relatively prime to the period **r**---uniquely determining it .Thus, this algorithm yields only a probabilistic result. Fortunately, we can make this probability as high as we like.

All the above work may appear a little anti-climactic. We have gone to a lot of trouble to design a quantum computer to find the period of a sequence. The point is, however, that the sequence is calculated in parallel and is exponentially long---even for a small value of say **k=140** bits in the first register the quantum computer has calculated and stored more results than there are particles in the universe. We now describe the simple structure that exists in the mathematical problem of factoring which allows us to apply the above quantum computer algorithm.

## *BUILDING A QUANTUM COMPUTER*

A quantum computer is nothing like a classical computer in design; you can't for instance build one from transistors and diodes. In order to build one, a new type of technology is needed, a technology that enables 'qubits' to exist as coherent superpositions of 0 and 1 states. The best method of achieving this goal is still unknown, but many methods are being experimented with and are proving to have varying degrees of success.

### *Quantum dots*

An example of an implementation of the qubit is the 'quantum dot' which is basically a single electron trapped inside a cage of atoms. When the dot is exposed to a pulse of laser light of precisely the right wavelength and duration, the electron is raised to an excited state: a second burst of laser light causes the electron to fall back to its ground state. The ground and excited states of the electron can be thought of as the 0 and 1 states of the qubit and the application of the laser light can be regarded as a controlled NOT function as it knocks the qubit from 0 to 1 or from ' to 0.

If the pulse of laser light is only half the duration of that required for the NOT function, the electron is placed in a superposition of both ground and excited states simultaneously, this being the equivalent of the coherent state of the qubit. More complex logic functions can be modeled using quantum dots arranged in pairs. It would therefore seem that quantum dots are a suitable candidate for building a quantum computer. Unfortunately there are a number of practical problems that are preventing this from happening:

- The electron only remains in its excited state for about a microsecond before it falls to the ground state. Bearing in mind that the required duration of each laser pulse is around 1 nanosecond, there is a limit to the number of computational steps that can be made before information is lost.
- Constructing quantum dots is a very difficult process because they are so small. A typical quantum dot measures just 10 atoms (1 nanometer) across. The technology needed to build a computer from these dots doesn't yet exist.
- To avoid cramming thousands of lasers into a tiny space, quantum dots could be manufactured so that they respond to different frequencies of light. A laser that could reliably retune itself would thus selectively target different groups of quantum dots with different frequencies of light. This again, is another technology that doesn't yet exist.

### *Computing liquids*

Quantum dots are not the only implementation of qubits that have been experimented with. Other techniques have attempted to use individual atoms or the polarization of laser light as the information medium. The common problem with these techniques is decoherence. Attempts at shielding the experiments from their surroundings, by for instance cooling them to within a thousandth of a degree of absolute zero, have proven to have had limited success at reducing the effects of this problem.

The latest development in quantum computing takes a radical new approach. It drops the assumption that the quantum medium has to be tiny and isolated from its surroundings and instead uses a sea of molecules to store the information. When held in a magnetic field, each nucleus within a molecule spins in a certain direction, which can be used to describe its state; spinning upwards can signify a 1 and spinning down, a 0. Nuclear Magnetic Resonance (NMR) techniques can be used to detect these spin states and bursts of specific radio waves can flip the nuclei from spinning up (1) to spinning down (0) and vice-versa.

The quantum computer in this technique is the molecule itself and its qubits are the nuclei within the molecule. This technique does not however use a single molecule to perform the computations; it instead uses a whole 'mug' of liquid molecules. The advantage of this is that even though the molecules of the liquid bump into one another, the spin states of the nuclei within each molecule remain unchanged. Decoherence is still a problem, but the time before the decoherence sets in is much longer than in any other technique so far. Researchers believe a few thousand primitive logic operations should be possible within time it takes the qubits to decohere.

Advancing beyond a 10-qubit system may prove to be more difficult. In a given sample of 'computing liquid' there will be a roughly even number of up and down spin states but a small excess of spin in one direction will exist. It is the signal from this small amount of extra spin, behaving as if it were a single molecule that can be detected and manipulated to perform calculations while the rest of the spins will effectively cancel each other out. This signal is extremely weak and grows weaker by a factor of roughly 2 for every qubit that is added. This imposes a limit on the number of qubits a system may have as the readable output will be harder to detect.

# *APPLICATIONS OF QUANTUM COMPUTING*

It is important to note that a quantum computer will not necessarily outperform a classical computer at all computational tasks. Multiplication for example, will not be performed any quicker on a quantum computer than it could be done on a similar classical computer. In order for a quantum computer to show its superiority it needs to use algorithms that exploit its power of quantum parallelism. Such algorithms are difficult to formulate, to date the most significant theorized being Shor's algorithm and Grover's algorithm. By using good these algorithms a quantum computer will be able to outperform classical computers by a significant margin. For example, Shor's algorithm allows extremely quick factoring of large numbers, a classical computer can be estimated at taking 10 million billion billion years to factor a 1000 digit number, where as a quantum computer would take around 20 minutes.

### *Shor's algorithm*

This is an algorithm invented by Peter Shor in 1995 that can be used to quickly factorise large numbers. If it is ever implemented it will have a profound effect on cryptography, as it would compromise the security provided by public key encryption (such as RSA).

**At Risk - Public Key Encryption**

This is currently the most commonly used method for sending encrypted data . It works by using two keys, one public and one private. The public key is used to encrypt the data, while the private key is used to decrypt the data. The public key can be easily derived from the private key but not visa versa. However, an eavesdropper who has acquired your public key can in principle calculate your private key as they are mathematically related. In order to do so it is necessary to factorise the public key, a task that is considered to be intractable.

For example, multiplying 1234 by 3433 is easy to work out, but calculating the factors of 4236322 is not so easy. The difficulty of factorizing a number grows rapidly with additional digits. It took 8 months and 1600 Internet users to crack RSA 129 (a number with 129 digits). Cryptographers thought that more digits could be added to the key to combat increasing performance in computers (it would take longer than the age of the universe to calculate RSA 140). However, using a quantum computer, which is running Shor's algorithm, the number of digits in the key has little effect on the difficulty of the problem. Cracking RSA 140 would take a matter of seconds.

**Shor's algorithm - An example**

The purpose of this section is to illustrate the basic steps involved in Shor's Algorithm. In order to keep the example relatively easy to follow we will consider the problem of finding the prime factors of the number 15. Since the Algorithm consists of three key steps, this explanation will be presented in 3 stages...

## Stage 1

The first stage of the algorithm is to place a memory register into a coherent superposition of all its possible states. The letter 'Q' will be used denote a qubit that is in the coherent state.

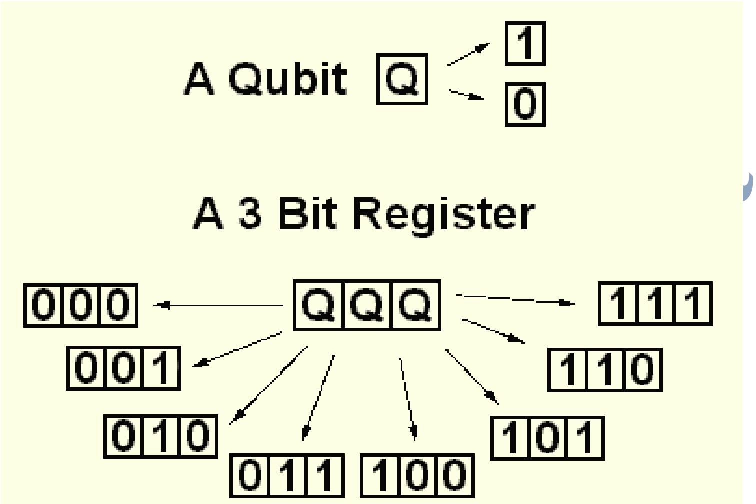


Figure 3 - A three-qubit register can represent 8 classical states simultaneously.

When a qubit is in the coherent state, it can be thought of as existing in two different universes. In one universe it exists as a '1' and in the other it exists as a '0' (See Figure 1). Extending this idea to the 3 bit register we can imagine that the register exists in 8 different universes, one for each of the classical states it could represent (i.e. 000, 001, 010, 011, 100, 101, 110, 111). In order to hold the number 15, a four bit register is required (capable of representing the numbers 0 to 15 simultaneously in the coherent state).

A calculation performed on the register can be thought of as a whole group of calculations performed in parallel, one in each universe. In effect, a calculation performed on the register is a calculation performed on every possible value that register can represent.

## Stage 2

The second stage of the algorithm performs a calculation using the register. The details of which are as follows:

- The number N is the number we wish to factorise, N = 15
- A random number X is chosen, where 1 < X < N-1
- X is raised to the power contained in the register (register A) and then divided by N
- The remainder from this operation is placed in a second 4 bit register (register B).

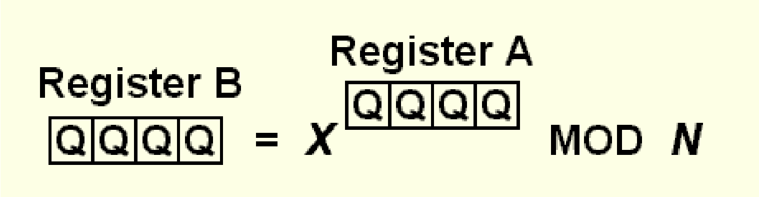


Figure 4 - Operation performed in stage 2.

After this operation has been performed, register B contains the superposition of each universes results. This is best illustrated with an example, if we choose X to be 2, then the contents of register B, for every possible value in register A are as follows.

| Register A | Register B |
|---|---|
| 0 | 1 |
| 1 | 2 |
| 2 | 4 |
| 3 | 8 |
| 4 | 1 |
| 5 | 2 |
| 6 | 4 |
| 7 | 8 |
| 8 | 1 |
| 9 | 2 |
| 10 | 4 |
| 11 | 8 |
| 12 | 1 |
| 13 | 2 |

Notice that the contents of register B follows a repeating sequence (1,2,4,8,1,2,4,8...), the frequency at which this repeats can be named f. In this case the repeating sequence (1, 2, 4, 8) has four values so f = 4.

| | |
|---|---|
| 14 | 4 |
| 15 | 8 |

Table 1 - Contents of Register B, when N = 15 and X = 2.

### Stage 3

The final stage is perhaps the most difficult to follow. The frequency of repetition, f, can be found using a quantum computer. This is done by performing a complex operation on register B and then looking at its contents which causes the results from every universe to interfere with each other. The resulting value for f is then used in the following equation to calculate a (possible) factor.

$$\text{Factor P} = X^{\frac{f}{2}} - 1$$

Figure 5 - Equation used to calculate factor.

The resulting number cannot be guaranteed to be a prime factor, but there is a good chance that it is one. The interference that produces the value for f tends to favor the correct answer as incorrect answers cancel each other out.

In our example the value f = 4 does give a correct answer of 3.

The fact that the answer cannot be guaranteed to be correct is of little consequence as it can be easily checked with multiplication. If the answer is incorrect, there is a very strong chance that repeating the calculation a few times with different values of X will produce the right answer.

### *Grover's algorithm*

Lov Grover has written an algorithm that uses quantum computers to search an unsorted database faster than a conventional computer .Normally it would take N/2 number of searches to find a specific entry in a database with N entries. Grover's algorithm makes it possible to perform the same search in root N searches. With the increasing size and integration of databases, this saving in time becomes significant. The speed up that this algorithm provides is a result of quantum parallelism. The database is effectively distributed over a multitude of universes, allowing a single search to locate the required entry. A further number of operations (proportional to root N) are required in order to produce a readable result.

Grover's algorithm has a useful application in the field of cryptography. It is theoretically possibly to use this algorithm to crack the Data Encryption Standard (DES), a standard which is used to protect, amongst other things, financial transactions between banks. The standard relies on a 56-bit number that both participants must know in advance, the number is used as a key to encrypt/decrypt data.

If an encrypted document and its source can be obtained, it is possible to attempt to find the 56-bit key. An exhaustive search by conventional means would make it necessary to search 2 to the power 55 keys before hitting the correct one. This would take more than a year even if one billion keys were tried every second, by comparison Grover's algorithm could find the key after only 185 searches. For conventional DES, a method to stop modern computers from cracking the code (i.e. if they got faster) would be simply to add extra digits to the key, which would increase the number of searches needed exponentially. However, the effect that this would have on the speed of the quantum algorithm is negligible.

### *Simulation of quantum mechanical systems*

In 1982, Feynman conjectured that quantum computers would be able to simulate quantum mechanical systems with a much greater degree of accuracy than is possible with classical computers. It is speculated that a quantum computer with a few tens of quantum bits could perform simulations that would take an unfeasible amount of time on a classical computer. This is due to the use of computer time and memory growing as an exponential function of the size of the quantum system in question.

On classical computers, the dynamics of a quantum system can be simulated using approximations. A quantum computer however, can be "programmed" to simulate the behaviour of a system by inducing interactions between its variables. These imitate the characteristics of the system in question. A quantum computer would, for example, allow the "Hubbard Model" (which describes the movement of electrons within a crystal) to be simulated, a task that is beyond the scope of current conventional computers.

### *Quantum Cryptography*

This is not really quantum computing but rather the use of quantum mechanics to transmit a key which is only known to the encoder (Alice) and the decoder (Bob). A better name than quantum cryptography would be quantum key distribution since quantum mechanics is used to create a method of cryptographic key distribution which can detect the presence of an eavesdropper (Eve) listening in.

A message N, which can be stored in an L-bit register is encoded with the use of a key K which is also a number between 0 and 2L-1. The encoded message M is simply

$$M \quad = \quad N \oplus K \qquad (\oplus \text{ means exclusive or - XOR })$$

The decoding is effected by again performing the XOR operation with K

$$M \oplus K \quad = \quad N \oplus K \oplus K \quad = \quad N \oplus 0 \quad = \quad N$$

The key is transmitted from encoder to decoder (or vice versa) by transmitting a large number of qubits (usually one will need at least 2L of these). The qubits are either in one of the two eigenstates of Sz or in one of the two eigenstates of Sx. These are chosen at random, but with equal probability by the encoder. For each qubit the encoder, Alice, records the eigenvalue of the qubit as well as the direction of spin (z or x) in which the qubit was an eigenstate. The decoder, Bob, measures either the z-component or the x-component of the spin of each qubit (at random, but with equal probability) and records the result as well as which direction of spin was measured.

In about half the cases Bob will have measured the spin in the same direction as Alice prepared it ("good" qubits). For such qubits Bob will obtain a result for the eigenvalue which is always equal to the eigenvalue corresponding to the eigenstate in which it was transmitted. In the remaining half, in which Bob measured the spin in a different direction from the direction in which they were prepared ("bad" qubits) the result will have equal probability of being equal or opposite to the eigenvalue of the prepared state. These "bad" bits must be discarded, but it is safe to build a key, K, from the remaining "good" qubits.

It is therefore sufficient for Bob to tell Alice (on an open line if necessary) in which direction the spin o each qubit was measured but not the result. Alice can then tell Bob (again on an open line) which are th "good" qubits and which are the "bad" ones. Although this is public information, no third party ca reconstruct the key, since the third party still does not know the eigenvalues of the "good" qubits.

One important feature of this technique is that the presence of an eavesdropper, Eve, can be detected. If Eve intercepts the signal from Alice, she does not know which setting, z or x, that Alice used. She must therefore choose a setting at random and then retransmit this result, using her setting, to Bob. Since Eve will not guess correctly every time, when Alice and Bob first make contact over the phone, they compare not only the settings but the results. If there is an eavesdropper then Alice and Bob will find that there are some "good qubits" on which they disagree. They then know that the security of the quantum channel is compromised. If they find perfect agreement, and can conclude there is no eavesdropper, they can then go ahead and exchange only setting information as described above.

Quantum key distribution, both over optical fibres and in free space, has been successfully demonstrated by a number of different groups.

# *QUANTUM COMMUNICATION*

The research carried out on quantum computing has created the spin-off field of quantum communication. This area of research aims to provide secure communication mechanisms by using the properties of quantum mechanical effects

### *How quantum communication works?*

Quantum communications makes use of the fact that information can be encoded as the polarisation of photons (i.e. the orientation of a photon's oscillation) . An oscillation in one direction can be thought of as 0 and in another as a 1. Two sets of polarisations are commonly used, rectilinear and diagonal (see figure 6).

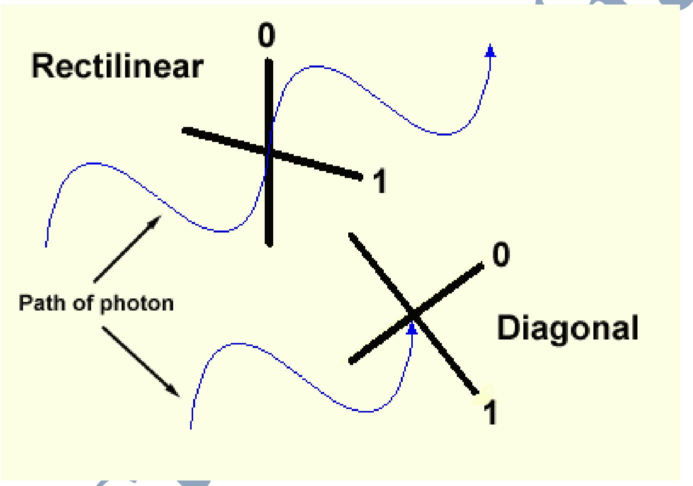


Figure 6 - The polarisation of photons can be used to encode data. In order to receive the data, the polarisation of the filter must match that of the photons.

The property that quantum communication exploits is that in order to receive the correct information, photons have to be measured using the correct filter polarisation e.g. the same polarisation that the information was transmitted with. If a receiver is in rectilinear polarisation, and a diagonally polarised photon is sent, then a completely random result will appear at the receiver. Using this, property information can be sent in such a way as to make it impossible for an eavesdropper to listen undetected. The mechanism by which this works is as follows:

1. The sender transmits information to the receiver using random polarisations.

2. The receiver detects this information (also at random polarisations) and records it.
3. The sender then informs the receiver of the polarisations that he used over a public channel.
4. The receiver and sender compare a random selection of the information that was received at the correct polarisation.
5. If an eavesdropper has intercepted and forwarded the information, the receiver and sender will be alerted as a higher percentage of errors will be present than expected.
6. If an eavesdropper has been detected, then the whole process has to be repeated.

For example, say there is a sender named Alice who wishes to transmit information to Bob without an eavesdropper Eve listening. They would follow the steps as described above. If Eve tries to eavesdrop, she will have to measure the bits coming from Alice and then forward them to Bob (she can't simply look at the information as doing so will alter it's content). She must use random polarisations to do this, as she does not know which ones Alice used. The chances are that Eve will correctly receive 50% of the information; the other 50% will consist of random values. The fact that approximately half of the random values will be correct means that Eve can at best forward 75% of the correct information to Bob.

Assuming that there is negligible noise on the communication line, Bob will be able to detect that Eve has eavesdropped, as the information he received at the correct polarisation will contain more than 25% errors. He checks for errors by comparing a random selection with Alice on a public channel.

Another way that Eve could attempt to subvert communication between Bob and Alice is to intercept the information and send her own instead. Eve will be thwarted by the fact that Alice and Bob discuss a randomly selected group of values, giving away the fact that Eve changed the information. It does not matter how subtly Eve intercepts the signal, Alice and Bob will always be able to discover that she has been listening to the line. This system can only work if the noise on the communication line is negligible, if the line had 25% noise for example it would be impossible to distinguish an eavesdropper from the noise itself. British Telecom has managed to implement a line with only 9% error over a distance of 10km, giving quantum communications a promising future.

### *Quantum bit commitment*

A different method of quantum communication is quantum bit commitment. Using this method, people can compare or combine information while keeping each individual contribution secret. A possible use for this would be in contract bidding (making firms bid their best possible offer instead of simply higher than the highest opposition).

The basic operation of this method is as follows:

1. Alice sends a string of photons to Bob, all of which are at the same polarisation.
2. Bob receives the photons, randomly changing his polarisation and recording the results.
3. Alice can prove to Bob that she sent the information by telling him the pattern of 1's and 0's he saw when his polarisation was the same as hers.

The weakness in this system is that Alice can cheat by creating pairs of photons and sending only one to Bob . These matched photons have the strange quantum property, which no matter how far apart, an observation of one will effect how the other appears at the receiver. Alice can now change Bob's photons by manipulating the copy she kept. Researchers knew of this problem for some time, and Mayor has recently proven that this is a general weakness of all quantum bit commitment systems.

## *ADVANTAGES of QUANTUM COMPUTING*

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 super positions of encoded numbers into different super positions. 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 super positions 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 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.

# *FUTURE SCOPE*

What are the prospects for quantum computation? In this section we discuss the search for other algorithms and describe the largest difficulty in building a quantum computer.

In this paper we discussed a single algorithm yielding an exponential speed-up over conventional methods: Effectively the calculation of the period of a long sequence. To date this is the only algorithm displaying such a speed-up. This algorithm was applied to a traditional computer-science problem, factoring, only by recognizing a deeper structure within that problem. This requirement appears to be a general one: Quantum parallelism will only yield an exponential speedup in problems whose structure avoids the need to try exponentially many solutions .Thus, a brute force approach to some of the hardest computational questions, known as NP-complete problems, will not succeed with the aid of quantum parallelism. Any progress for such problems will require finding a deeper structure within them. Instead, quantum computers are likely to be most useful for simulating or manipulating small quantum systems .

How difficult will it be to build a quantum computer? Even within apparently small atomic-scale systems, quantum computation runs on the enormous size of Hilbert space. Quantum computation involves building a trajectory from a standard initial state to a complex final state. The main difficulty is keeping to this trajectory. To fail is to be lost in Hilbert space. The largest problem is hypersensitivity to perturbations, shifting the computational trajectory randomly from its path. Such perturbations come from an unintentional coupling to external noise . It is too soon to predict the gravity of this problem. It appears, though, that there is no fundamental limit to how well we can isolate a quantum system. Currently, several implementations are being considered by theoreticians and experimentalists worldwide . One promising scheme involves ion-traps the next generation of atomic-clock standards. Over the next two decades conventional computers will approach the atomic scale, perhaps quantum computers will get there first.

## *CONCLUSION*

With classical computers gradually approaching their limit, the quantum computer promises to deliver a new level of computational power. With them comes a whole new theory of computation that incorporates the strange effects of quantum mechanics and considers every physical object to be some kind of quantum computer. A quantum computer thus has the theoretical capability of simulating any finite physical system and may even hold the key to creating an artificially intelligent computer.

The quantum computers power to perform calculations across a multitude of parallel universes gives it the ability to quickly perform tasks that classical computers will never be able to practically achieve. This power can only be unleashed with the correct type of algorithm, a type of algorithm that is extremely difficult to formulate. Some algorithms have already been invented; they are proving to have huge implications on the world of cryptography. This is because they enable the most commonly used cryptography techniques to be broken in a matter of seconds. Ironically, a spin off of quantum computing, quantum communication allows information to be sent without eavesdroppers listening undetected.

For now at least, the world of cryptography is safe because the quantum computer is proving to be vary difficult to implement. The very thing that makes them powerful, their reliance on quantum mechanics, also makes them extremely fragile. The most successful experiments only being able to add one and one together. Nobody can tell if the problems being experienced by researchers can be overcome, some like Dr. Gershenfield are hopeful that they can whilst others believe that the quantum computer will always be to fragile to be practical.

## **Reference**

□ www.google.com

□ www.wikipedia.org

□ www.studymafia.org