Quantum Computing and Communication: a Technical Basis

Jean Bacon and Jon Crowcroft

Abstract
A highly mathematical treatment is usually the means to fully describe quantum computing. Here we give a minimal mathematical background with even the simplest mathematics spelled out in order to give an intuition of how quantum computing works. A companion paper presents the material with no mathematics. We first summarise aspects of classical computing then introduce a representation of a quantum object, the qubit. We then state the expert consensus on the properties of quantum systems.

Since what is often called quantum communication but is in practice quantum key distribution (QKD) builds on the quantum properties of photons, we describe this in some detail, although omitting mathematical formulation. QKD has been demonstrated experimentally in research contexts and wider-scale deployment is in progress.

We then return to quantum computing, in contrast to classical computing. By means of minimal mathematics and quantum circuits (gates), we cover how data might be initialised, represented and stored, how input and output might be done and how programs can be represented and executed. At present, quantum computers are not general "stored program computers" but their operational circuits are built to carry out specific algorithms.

Interest in quantum computing arises from its claimed potential for solving problems that are infeasible for classical computers, such as breaking public key encryption (PKI). We explain why breaking PKI seems a most likely early casualty of quantum computing. A companion paper discusses the implications, were it to happen, including the role of QKD in this scenario and post-quantum cryptography. We then outline other algorithms that have been suggested as appropriate for quantum computers.

A great deal of money is being directed towards research into quantum computing. We outline possible fabrication technologies for qubits and quantum computers and summarise briefly the progress that has been made in various research projects towards realising quantum computing.

Neils Bohr: "*Everything we call real is made up of things that cannot be regarded as real*"

Albert Einstein: "*In quantum mechanics, one can know everything about a system and nothing about its individual parts*"

"*Quantum mechanical systems are not deterministic. The results of experiments can be statistically random but if we repeat an experiment a large number of times, average quantities can follow the expectations of classical physics, at least up to a point.*" and "*Classical systems are built from probabilistic systems averaged over very large numbers.*" (From: L Susskind and A Friedman, *Quantum Mechanics, the Theoretical Minimum,* Penguin, 2014).

Contents

1. Classical and quantum computing
   - 1.1 Classical computing
   - 1.2 A quantum object or qubit
2. Fundamental properties of quantum systems
   - 2.1 Hiesenberg's uncertainty principle
   - 2.2 Superposition
   - 2.3 Entanglement
   - 2.4 Measurement and transformation of qubits
   - 2.5 Nonclonability
3. Quantum key distribution
   - 3.1 The BB84 protocol
   - 3.2 Using entanglement to transmit a secret key, the E91 protocol
   - 3.3 Wide area deployment of QKD
4. Quantum Computing
   - 4.1 Transformations of qubits
   - 4.2 Quantum gates and circuits
   - 4.3 Quantum computing programs
   - 4.4 Quantum error correction (QEC)
5. Quantum Algorithms and Protocols
   - 5.1 Quantum teleportation
   - 5.2 Superdense coding
   - 5.3 Deutsch and Deutsch-Jozsa algorithms
   - 5.4 Grover's algorithm for unstructured search
   - 5.5 Outline implementation of PKI and implications of breaking it
   - 5.6 Shor's quantum algorithm for prime factorisation
6 Technology for qubits and quantum computers
7 Quantum computers under development
   (cross reference to paper by Chris Norval and Jat Singh)
8 Challenges
   - 8.1 Noise
   - 8.2 Cost
   - 8.3 Algorithms
   - 8.4 Performance
9 Conclusions

1 Classical and quantum computing

The second world war provided an impetus for accelerating the development of computers. For example, Colossus was developed at Bletchley Park, UK, to decode the Enigma machine. After the war, a consensus was reached on what has become the classical design for computers – the von Neumann architecture. Binary was to be used as the basis for representing information of all kinds, both data and instructions (code). The greater breakthrough was "the stored program computer", that is, the code to manipulate the data was to be stored in the computer memory (in binary), as well as the data.

1.1 Classical computing

Figure 1 shows a program stored in memory as well as the data on which it is to operate. Figure 2 shows the classical "fetch-execute cycle" of a stored-program computer.

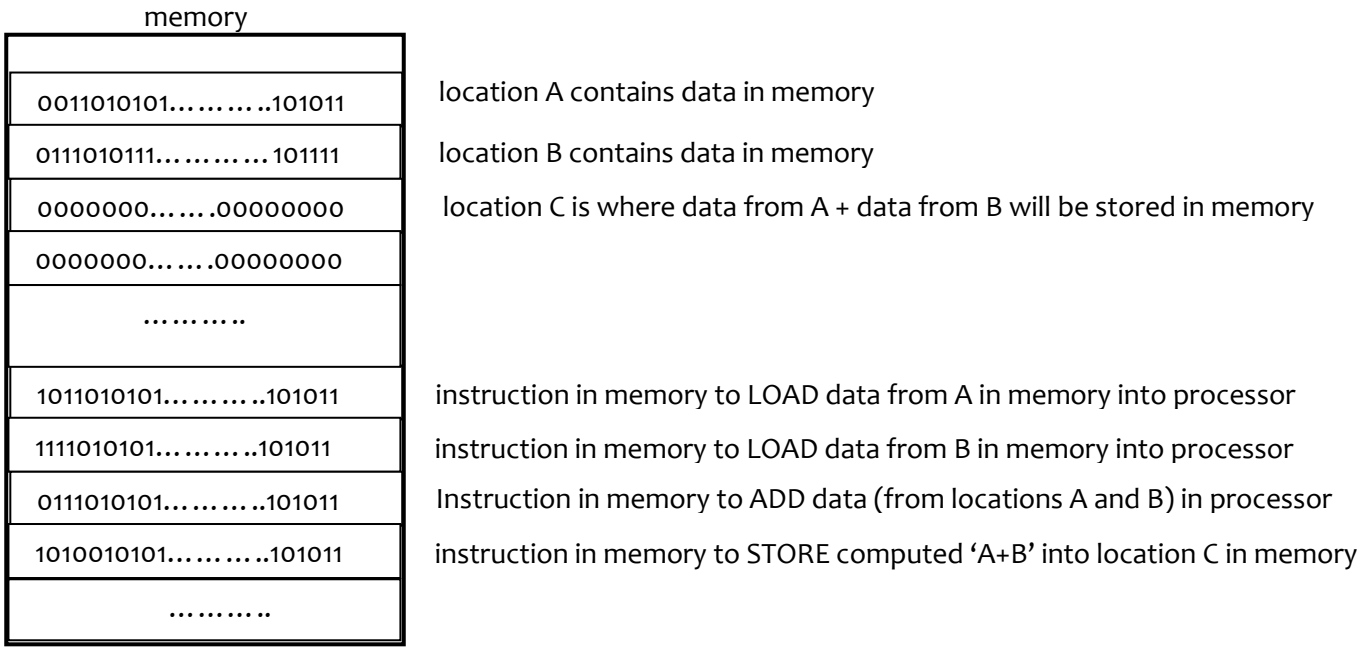


Fig. 1 program and data as bit patterns in classical architecture

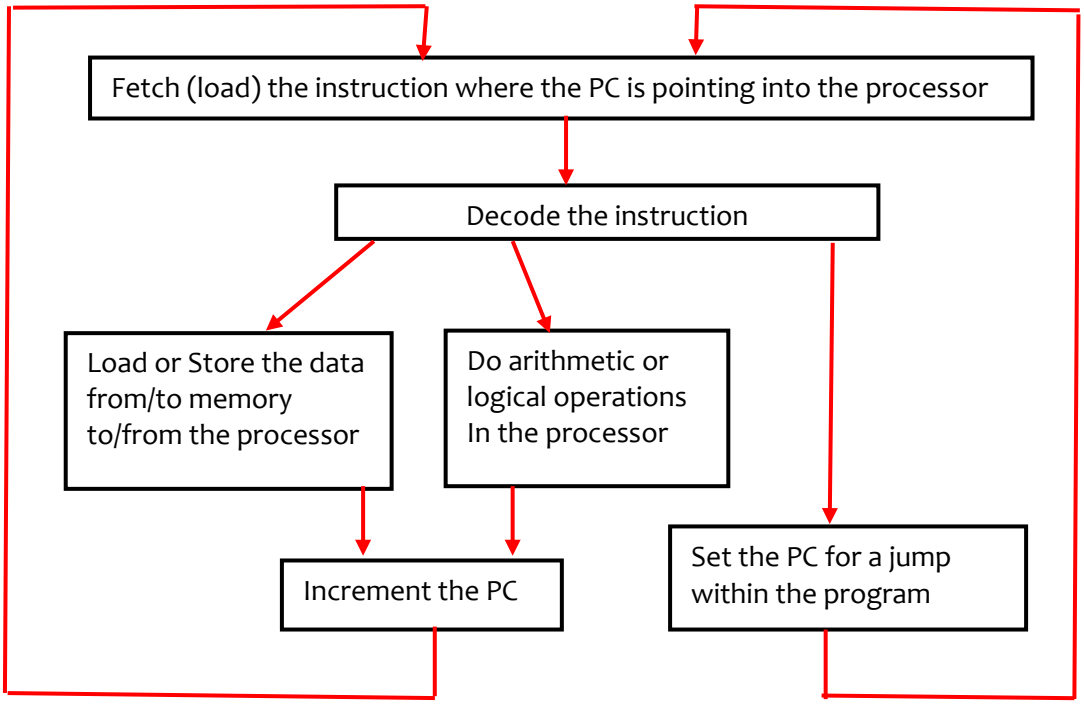


Fig. 2 The classical von Neumann architecture: The fetch-execute cycle
(PC = program counter which points to the next instruction)

1.2 A quantum object or qubit

As we have seen above, classical computing is based on binary, that is, a system of two stable states 0 and 1 with controlled transitions between them. In a classical system, the *physical representation of a bit in a computer* has the value 0 or 1. In classical programming, such as in probabilistic (Bayesian) programming, we can say that a given bit is either 0 or 1, each with some probability. The logic of the program keeps track of the probabilities and results are output with associated probabilities. In this respect, quantum computing seems similar. In a classical system, values can be copied and output throughout a process without affecting those values. Copying is ubiquitous in classical systems for securely logging data, to replicate bits for error correction, transmitted data is almost invariably a copy, etc. In quantum systems copying is impossible.

For a quantum object (qubit) there is a notion of motion (spin, momentum etc) as well as position. Historically, there was a period of intense research over whether a quantum object was to be modelled as a particle or a wave (or both), with many associated well-known experiments and thought-experiments, such as the two-slit experiment and Schrödinger's cat.

In a quantum system, the physical representation of a qubit is modelled as probabilistic. In a two-state quantum system, a qubit is often described as *simultaneously* having the value 0 with some probability p and the value 1 with probability (1-p). Measuring a qubit yields 0 or 1, according to these probabilities, after which the previous state is lost (perhaps the wave becomes a particle). The idea that the probabilities must sum to 1, representing probability 1 as the sum of all possible measurement outcomes, carries through to any number of qubits and permeates the modelling mathematics.

Although a given qubit can only be measured once, after which its state is lost, its representation in terms of probabilities implies that, if instead, we were able to carry out the measurement repeatedly we would get values 0 and 1 in proportion to the probabilities. For example, if the probabilities of 0 and 1 were each ½, half the measurements would yield 0 and half 1 over a large number of measurements. In quantum experiments, this idea is carried through to measuring a long stream of qubits in which the measurements are deemed to be probabilitstic.

In quantum physics, we visualise a qubit as a sphere with radius 1, but it has different properties from a classical sphere. We start from a classical sphere in real 3D space, as shown in the centre of Figure 3, then generalise to the ***Bloch sphere*** of quantum mechanics.

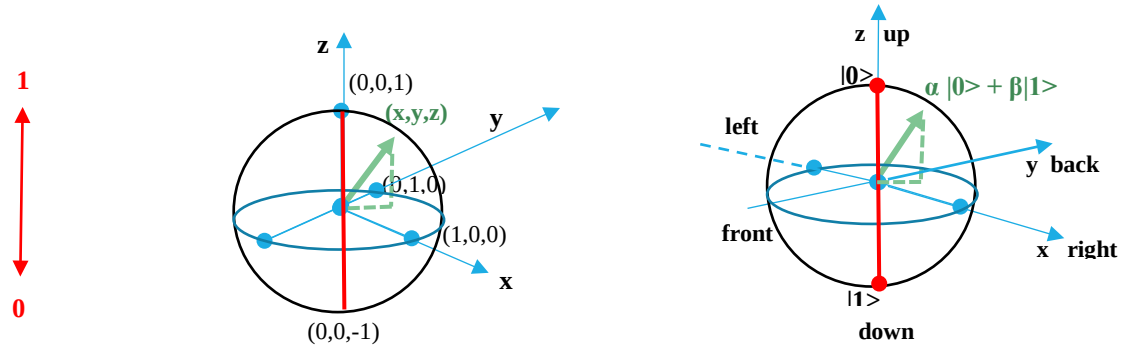


Fig. 3 classical bits, classical sphere and qubit sphere (Bloch sphere) for a 2-state quantum system

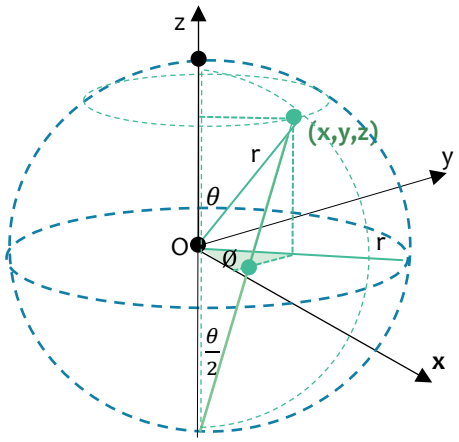
Consider first a 3D representation of the position of a stationary object on a sphere of radius 1. The coordinates of a point on the surface of the sphere are (x,y,z) where $x^2 + y^2 + z^2 = 1$. Note that the three

axes x, y, z are orthogonal (at right angles) i.e. any point on the x axis has coordinates $(x,0,0)$, since its projection on the y and z axes is zero, on the y axis $(0,y,0)$ and on the z axis $(0,0,z)$. The axes are said to be linearly independent, allowing the (x,y,z) representation of any point, i.e. a point can be represented as a linear combination of its projection (values) on the x, y and z axes.

- $x^2 + y^2 + z^2 > 1$ represents locations outside the unit sphere, so of no interest.
- $x^2 + y^2 + z^2 < 1$ represents locations inside the unit sphere so of no interest
- $x^2 + y^2 = 1, z = 0$ is true for points on the unit circle in the x,y plane (the equator).

When we come to interpret apparently similar values as qubits, points on the unit circle in the x,y plane could collapse to 1 or 0, each with probability 0.5. All other points on the surface of the sphere have some probability of collapsing to either 1 or 0. First thoughts are that points in the northern hemisphere are more likely to collapse to 1, those in the southern hemisphere more likely to collapse to 0. However, we shall see that qubits interact and their states can be composed and/or interfere with each other. Managing these interactions is the essence of quantum computing.

Figure 4 shows the classical sphere in polar coordinate form, useful for understanding the Bloch sphere.



For a general point on the sphere, $x^2 + y^2 + z^2 = r^2$,
$z = r\cos\theta$, $y = r\sin\theta\sin\emptyset$, $x = r\sin\theta\cos\emptyset$

In the horizontal/equatorial x,y plane:
$z=0$, $x^2 + y^2 = r^2$, $x = r\cos\emptyset$, $y = r\sin\emptyset$

Fig. 4 Polar coordinates for the classical sphere

A classical sphere can't be used to represent a qubit, although some properties carry forward. To understand the Bloch sphere representation we need to introduce both complex numbers and vectors. We require to represent a *two-state quantum system*, where a qubit can take any value on the surface of the Bloch sphere until it is measured, when it collapses to one of the two states I or 0 with some probability. We therefore still use a sphere with radius 1 (the radius is in fact a complex number with modulus 1). However, for a classical sphere the position of a static particle on a unit sphere is represented by three coordinates (x,y,z) or the corresponding polar coordinates. A qubit also has a further property which, depending on the type of particle can be such things as:

- the polarisation of a photon
- the spin direction of an electron
- the energy level of an atom
- the charge state of a quantum dot
- the mesoscopic current in a superconductor

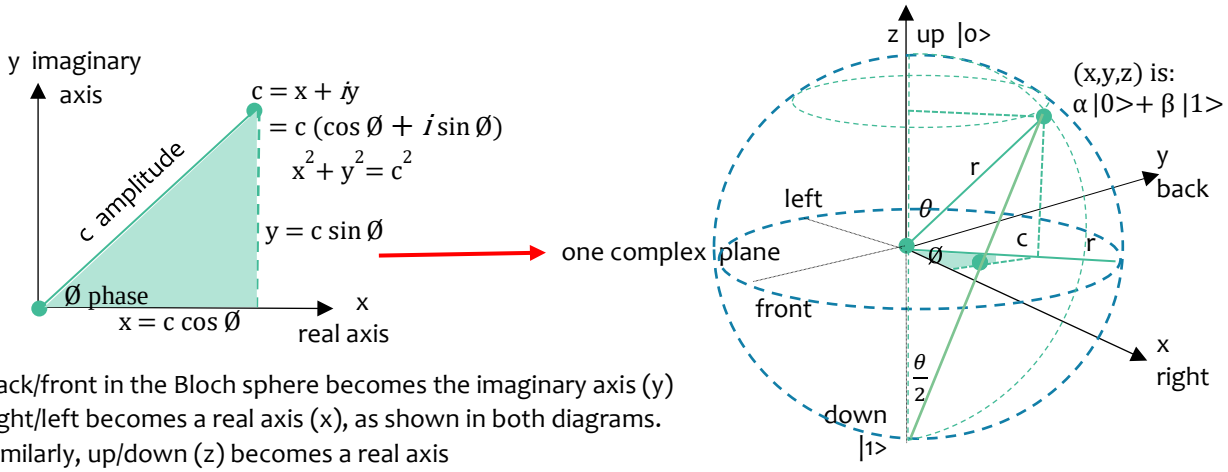
Heisenberg was concerned with the measurement of both the position and the momentum of a qubit. Heisenberg's ``uncertainty principle'' indicates that it is not possible to measure one property without disturbing another, thus collapsing the object. Further, if one component of a property is measured, e.g., the spin of an electron along one axis, it is not possible to measure its component along any other axis with

certainty. All that can be said is that if such a measurement could be carried out many times, the average value could be taken as the value of that component, i.e. it is probabilistic.

Complex numbers

To capture the particle/wave duality of a qubit in the mathematical modelling we need α and β in the expression for the state of a qubit α |0> + β |1> in Figure 3 to be complex numbers.
We define i as the square root of -1, i.e. $i = \sqrt{-1}$ $i^2 = -1$. Figure 5 illustrates on the left.
A complex number (say, c) takes the form $c = x + iy$ where x is the real part and y the imaginary part.
The magnitude (modulus) of a complex number is $|c| = \sqrt{x^2 + y^2}$.
The phase (argument) Ø is given by tan Ø= y/x.
The **complex conjugate** of c (c*) is defined as $c^* = x - iy$, so $c^*c = x^2 + y^2$.
In polar coordinates c can be expressed as $c = c\,(\cos Ø + i \sin Ø)$ where c is the radius of a circle in the complex plane and Ø is the angle between c and the real axis, as shown on the left in Figure 5.

If we make 'back' in the Bloch sphere on the right the imaginary axis, we have both the equatorial/horizontal (as indicated in Figure 5 on the right) and the polar/vertical planes as complex planes. The Bloch sphere has complex radius r with modulus (magnitude) = 1, as required to represent probability 1. Figure 4 gives the expressions for a general point (x,y,z) on the sphere as polar coordinates (but see below for vectors), which becomes α |0>+ β |1> on the Bloch sphere, where α and β are complex numbers. As the phase angle Ø changes, this "point" moves around the sphere at its line of latitude, as shown in the figure. Similarly, as the phase angle θ changes, the "point" moves around the sphere at its line of longitude, as shown.



back/front in the Bloch sphere becomes the imaginary axis (y)
right/left becomes a real axis (x), as shown in both diagrams.
Similarly, up/down (z) becomes a real axis

Fig. 5 Complex planes in the Bloch sphere

Vectors are also part of modelling a qubit:

- **Vector**: A directed line from the origin to a point is a vector. We are concerned with vectors of unit length, i.e. from the origin to points on the unit sphere. An example is shown in green in Figures 3, 4, 5. In quantum mechanics the state of a qubit is a vector which has complex components and which lies on the complex unit sphere, with the sum of the squares of the moduli (magnitudes) of its components adding up to probability 1.
- **Vector space**: Vectors to all points within the 3D sphere or, when the vector is constrained to be of length 1, to points on its surface, form a real vector space. In quantum mechanics, the state of a qubit inhabits a complex vector space, called a **Hilbert Space** or **inner product space**. In other words, the space of sets of qubits is not a set but a vector space. This means that *states can be added and subtracted,*

which makes no sense in classical mechanics. A Hilbert space can be n-dimensional but for modelling a qubit we show that we only need two dimensions in this space to represent any vector.

- **Dirac's bra-ket notation** Statistical averaging plays a fundamental part in determining the values of qubits. Paul Dirac introduced the bra-ket notation, where **<Q>** is the statistical average (expectation) of quantity **Q**.

For an introduction to Quantum Mechanics including a mathematical treatment, see (L Susskind and A Friedman, *Quantum Mechanics, the Theoretical Minimum,* Penguin, 2014). Here we outline the notations and properties behind the assertions made in the discussion.

First we show that in this complex vector space **up/down** are orthogonal (linearly independent), as are **back/front** and **right/left** (or any other line through the origin). Any point on the Bloch sphere can therefore be represented as a combination of one of these orthogonal pairs, say, α_1**up** + α_2**down**; this pair is used as the standard way of representing qubits, the so-called computational basis. (Note the difference from classical spherical coordinates where x, y, z are three orthogonal axes).

Vectors, inner products and orthogonal bases

In Dirac's bra-ket notation for quantum mechanics, a vector space is composed of elements |A> called ket vectors or simply kets. They are column vectors with complex components. The corresponding row vectors are called bra vectors, with notation <A|, where the elements in <A| are the complex conjugates of the elements in |A>.

Suppose $|A\rangle = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}$, $\langle A| = (\alpha_1^*, \alpha_2^*)$ and $|B\rangle = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}$, $\langle B| = (\beta_1^*, \beta_2^*)$

Bras and kets can be composed e.g. : <A|A> or<A|B> to give a complex number in general, since

$\langle A|A\rangle = (\alpha_1^*, \alpha_2^*) \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = |\alpha_1|^2 + |\alpha_2|^2$ and $\langle A|B\rangle = (\alpha_1^*, \alpha_2^*) \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = (\alpha_1^* \beta_1 + \alpha_2^* \beta_2)$.

<A|B> is called the **inner product** of |A> and |B>. |A><B| is called the outer product of |A> and |B>. The inner product is a way of associating a strictly positive length (the **norm**) with a non-zero vector in a vector space. The **norm** of a vector $|A\rangle = \sqrt{\langle A|A\rangle}$.

In the Bloch sphere in Figures 3 and 5, the upwards (z) direction (electron spin up) is conventionally labelled $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ with value 1 and the downwards direction $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ with value 0. Figure 6 gives a visualisation.

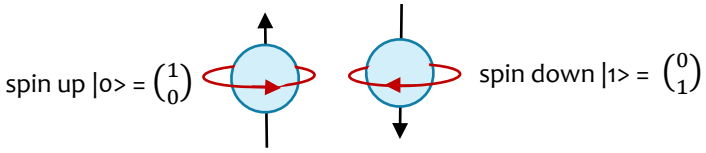


Fig. 6 Visualisation of spin up and spin down of a qubit

Definition of **orthogonal axes** in a vector space:
A and B are **orthogonal vectors** if <A|B> = 0 i.e. $\alpha_1^* \beta_1 + \alpha_2^* \beta_2 = 0$
We need orthogonal vectors to use as axes in a vector space (like x, y, z in classical space).

The qubits we are modelling lie on the Bloch sphere, represented by vectors of unit length. A **normalised vector** is such that <A|A> = 1, i.e. $|\alpha_1|^2 + |\alpha_2|^2 = 1$ i.e. |A> is a normalised vector (of length 1).
Orthonormal vectors have both properties, i.e. the vectors must be orthogonal and of unit length. Note that |0> and |1> are orthonormal, since $\langle 0|1\rangle = (1,0) \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 0$ and they are of unit length. We can therefore represent the state of any qubit on the Bloch sphere in terms of |0> and |1>, say: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$

Note that **up and down are orthogonal in the complex vector space** (unlike z and –z in real space), as are right and left in Figures 3 and 5 and front and back (or any line passing through the centre). |0> and |1> are taken as the standard axes, called the **standard or computational base or basis**, for representing qubits. But why are two axes sufficient when we need three, in classical space (x,y,z)? First, two complex numbers contain four components, but these are not all independent for qubit modelling because of the constraints imposed by normalisation and orthogonality. It is therefore the case that only two orthogonal axes (basis vectors or bases) are needed to span this space.

Probability amplitudes. Since up and down (|0> and |1>) are orthonormal, any vector can be represented as a linear combination of its projections onto these axes. That is, any vector on the Bloch sphere (shown in green in Figures 3 and 5) can be represented with respect to them as qubit state, $|\psi> = \alpha\,|0> + \beta|1>$ where $|\alpha|^2+|\beta|^2 = 1$ (because the vectors are on the sphere). α and β are called the **probability amplitudes** of the vector. Any attempt to measure the state results in |0> with probability $|\alpha|^2$ and |1> with probability $|\beta|^2$. The expression for $|\psi>= \alpha\,|0>+\beta\,|1>$ can be written in polar coordinates (where r=1):

$$|\psi> = \mathrm{r}\cos(\tfrac{\theta}{2})\,|0> + (\cos\emptyset + i\sin\emptyset)\,\mathrm{r}\sin(\tfrac{\theta}{2})\,|1> \quad \text{where } 0<\theta<\pi,\ 0<\emptyset<2\pi$$

Since the modulus of $(\cos\emptyset + i\sin\emptyset)^2 = \cos^2\emptyset + \sin^2\emptyset = 1$ and $\cos^2\frac{\theta}{2} + \sin^2\frac{\theta}{2} = 1$, we have the required $|\alpha|^2 + |\beta|^2 = 1$. The derivation of this expression for |ψ>is left for further reading. Whereas z is orthogonal with x and y at π/2 in the classical sphere, on the Bloch sphere, |0> and |1> are orthogonal at π.

On the complex (equatorial) plane in the Bloch sphere, $\theta = 90° = \pi/2$ so we have

$$|\psi> = \tfrac{1}{\sqrt{2}}\,|0> + \tfrac{1}{\sqrt{2}}(\cos\emptyset + i\sin\emptyset)\,|1>$$

which gives the expressions in Figure 7, as Ø moves around the equatorial circle from 0 to 2π.

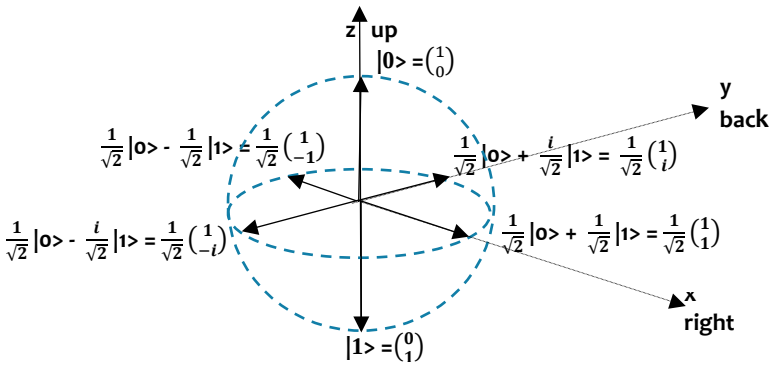


Fig. 7 Six states on the Bloch sphere, represented in the standard basis

As shown in Figure 7, the vectors on the sphere along the right, left, front and back axes are:

$\text{right} = \frac{1}{\sqrt{2}}|0> + \frac{1}{\sqrt{2}}|1> = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$, $\text{left} = \frac{1}{\sqrt{2}}|0> - \frac{1}{\sqrt{2}}|1> = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$ (orthogonal since $\alpha_1\,\beta_1 + \alpha_2\,\beta_2 = 0$)

$\text{front} = \frac{1}{\sqrt{2}}|0> + \frac{i}{\sqrt{2}}|1> = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\i\end{pmatrix}$, $\text{back} = \frac{1}{\sqrt{2}}|0> - \frac{i}{\sqrt{2}}|1> = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-i\end{pmatrix}$ (orthogonal since $\alpha_1\,\beta_1 + \alpha_2\,\beta_2 = 0$)

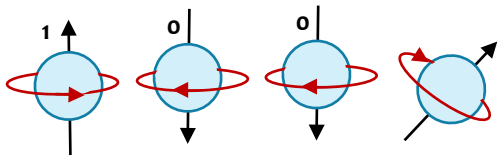
Note that all lengths, and therefore probabilities, are 1, since $\frac{1}{\sqrt{2}}^2 + \frac{1}{\sqrt{2}}^2 = 1$

Summarising this modelling of a qubit state:

1. The state of a qubit is represented by a unit (normalised) vector in a complex vector space of states.
2. In this complex space we need only two orthogonal basis vectors to specify a qubit state. The standard bases used are up |0> and down |1>

Representing binary numbers with qubits

As we saw in Figure 1, a classical binary value comprises (say) n bits. The range of values it can represent is therefore from 0 (all n bits are zero) to $2^n - 1$ (all bits are 1). For a two-state quantum system, when each bit is represented by a qubit, each bit has some probability of collapsing to 1 or to 0, according to its associated probability distribution. Figure 8 shows four qubits representing a binary number. On being measured, the number would collapse to 1000 or 1001 with some probability.



binary 1000 = decimal 8
or
binary 1001 = decimal 9
with some probability

Fig. 8 Example showing four qubits representing a binary number

The potential of quantum computing is that in some sense, the n bits can hold the whole range of possible values simultaneously. We shall see later that this is because the bits are not isolated, in which case there would be no gain above a classical representation, but because their fields interoperate due to entanglement. Each operation on the bits carried out by a quantum computing program can therefore be seen as operating on all the possible patterns at once. The challenge is to be able to manipulate, select, collapse and read the one bit pattern that represents the answer to the problem at hand. We return to quantum computing in Section 4.

In the next section we summarise the properties of quantum systems for reference throughout the paper.

2. Fundamental properties of quantum systems

Experts have defined the following fundamental properties of quantum systems. Our focus is two-state systems. We state the properties with some visualisations and minimal mathematical underpinning to provide intuition. In the next section we give a detailed example using the example of a photon as a two-state qubit and show how some of these properties have been used in this practical system.

2.1 Heisenberg's Uncertainty Principle

An unknown quantum state cannot be observed (measured) without being disturbed. Heisenberg initially had the properties of position and momentum of a particle in mind and described how any conceivable method of measuring a particle's position would disturb its conjugate property, its momentum, thus destroying its *coherence*. It is therefore impossible to simultaneously observe both properties with certainty.

A consequence of this is that once a qubit has been measured it stays in the measured state. You have collapsed it to measure it -- it is no longer a complex vector representing a wave, but a particle.

We discuss measurement in Section 2.4, how to model it is perhaps the most difficult and controversial problem in quantum mechanics and is fundamental to quantum computing.

2.2 Superposition

The term *superposition* comes from the wave-like (analogue) properties of qubits in that waves can be added or can cancel each other out. To pursue this classical analogy, Figure 9 gives an example of adding classical waveforms, showing a frequency and a harmonic.

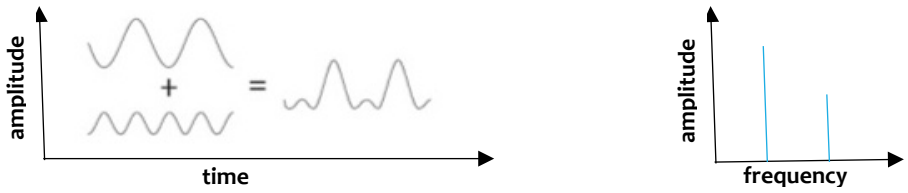


Fig. 9 Example of a frequency and a harmonic in classical waveforms

However, when combining waveforms, the *phases* of the constituent waveforms also affect the result. Figure 10 gives three examples from classical waveform composition, the waves to be added shown in blue and the result of composition shown in red : a) in-phase composition doubling the amplitude, b) cancellation of amplitude when "out of phase" and c) arbitrary phase difference.

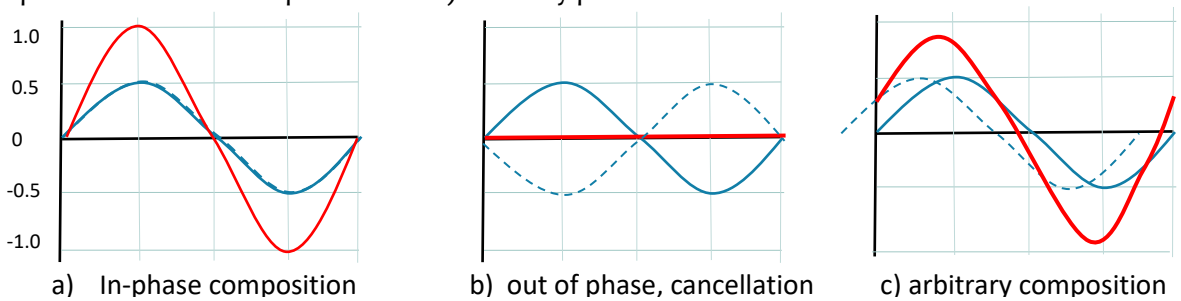


Fig. 10 Effect of phase on the combination of classical waveforms

We introduce phase because the phase of the spin of some fabrications of qubits is operated on by quantum gates (see Section 4.2). Also, cancellation effects are needed to explain certain quantum phenomena. Cases a) and b) in Figure 10 show waves constrained by boundaries, i.e. sharing endpoints. We have similar requirements for qubit modelling since the wavelengths must "fit around" the unit sphere as sketched in Figure 11.

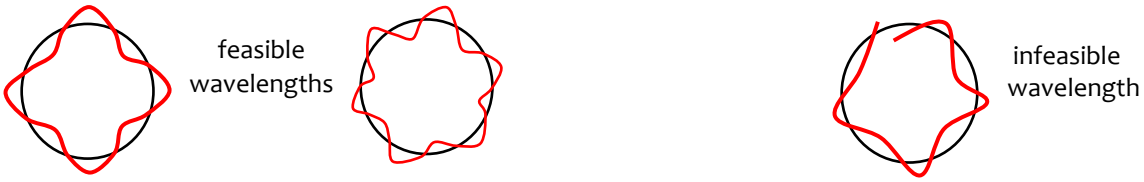


Fig. 11 A whole number of quantum wavelengths must fit into a quantum orbit.

Figure 7 shows six possible representations of a single qubit. First, up and down (the standard basis) shows up as $|0\rangle \begin{pmatrix}1\\0\end{pmatrix}$ which is measured as 1 with 100% probability and down as $|1\rangle \begin{pmatrix}0\\1\end{pmatrix}$ which is measured as 0 with 100% probability. The qubit might instead be in an **equal superposition** of states with 50% probability of being measured as 0 or 1 which means that if measured, a random value will be returned. One such pair of states is shown on the real axis x (right/left). Note that right$=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$ and left $=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$, demonstrate the equal superposition. Alternatively, the imaginary axes could be taken with back $=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\i\end{pmatrix}$and front $=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-i\end{pmatrix}$. Note that all the vectors lie on the unit circle with probabilities summing to 1 since $\frac{1}{\sqrt{2}}^2+\frac{1}{\sqrt{2}}^2=1$

The expression for qubit state $|\psi\rangle$ in polar coordinates is

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + (\cos\emptyset + i\sin\emptyset)\sin\left(\frac{\theta}{2}\right)|1\rangle, \quad \text{where } 0<\theta<\pi,\ 0<\emptyset<2\pi$$

The phase Ø represents a rotation of the vector around a line of latitude, so leaving the probability amplitudes unchanged. As the phase θ changes from 0 to π, the probability amplitudes of the qubit vary from |0> through $\frac{1}{\sqrt{2}}$|0>+$\frac{1}{\sqrt{2}}$|1> to |1>. Phase differences allow cancellation during quantum computations.

In summary, a qubit is said to exist in an arbitrary complex superposition (combination) of classical Boolean states. From Section 2.1, when a qubit's value is measured, a single value (0 or 1) emerges for the state.

2.2.1 Superposition of states of multiple qubits

After measurement, two qubits can represent 4 states, three qubits 8 states, and so on (n qubits represent 2^n states). But let's consider the states of 2 qubits before measurement. The following are used for the computational basis states for two qubits. Each 4x1 vector is called the tensor product ⊗ of the corresponding two 2x1 vectors, a multiplicative operation on the two vectors.

$$|00\rangle=|0\rangle\begin{pmatrix}1\\0\end{pmatrix}\otimes|0\rangle\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}1\,x1\\1x0\\0x1\\0x0\end{pmatrix}=\begin{pmatrix}1\\0\\0\\0\end{pmatrix} \qquad |01\rangle=|0\rangle\begin{pmatrix}1\\0\end{pmatrix}\otimes|1\rangle\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}1x0\\1x1\\0x0\\0x1\end{pmatrix}=\begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

$$|10\rangle=|1\rangle\begin{pmatrix}0\\1\end{pmatrix}\otimes|0\rangle\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}0x1\\0x0\\1x1\\1x0\end{pmatrix}=\begin{pmatrix}0\\0\\1\\0\end{pmatrix} \qquad |11\rangle=|1\rangle\begin{pmatrix}0\\1\end{pmatrix}\otimes|1\rangle\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}0x0\\0x1\\1x0\\1x1\end{pmatrix}=\begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

The superpositions of state that a single qubit can be in are $\alpha_0|0\rangle+\alpha_1|1\rangle$, where α_0 and α_1 are complex numbers, constrained by $|\alpha_0|^2+|\alpha_1|^2=1$, so the state space is infinite.

For a pair of qubits in superposition, using |00>, |01>, |10> and |11> as the computational basis, the expression for their state is:

$\alpha_0|00\rangle+\alpha_1|01\rangle+\alpha_2|10\rangle+\alpha_3|11\rangle$ where the α_i are complex, constrained by $\sum_{i=0}^{3}|\alpha_i|^2=1$

For a pair of qubits we have lost the intuitive model of a single unit sphere. However, the mathematics regarding norms and normalisation for the state space of 2 bits (indeed, n bits) yields the amplitude/probability = 1 condition.

For example, $\frac{1}{2}(|00\rangle+|01\rangle+|10\rangle+|11\rangle)=\frac{1}{2}\begin{pmatrix}1\\0\\0\\0\end{pmatrix}+\frac{1}{2}\begin{pmatrix}0\\1\\0\\0\end{pmatrix}+\frac{1}{2}\begin{pmatrix}0\\0\\1\\0\end{pmatrix}+\frac{1}{2}\begin{pmatrix}0\\0\\0\\1\end{pmatrix}=\frac{1}{2}\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$ is an equal superposition of states each with amplitude ½, where the squares of the amplitudes (4 x ¼) sum to 1.

Similarly, three bits have eight states with $\frac{1}{\sqrt{8}}$ as the corresponding amplitude, and so on.

Recall that **measurement destroys superposition** since each qubit is measured as 1 or 0.

2.3 Entanglement

Certain pairs (or more generally groups) of qubits can be inextricably interconnected. For such a pair, when a particular state is measured in one of the objects, the opposite state will instantaneously be observed on the entangled object. Note that there is no suggestion of communication between the objects; the effect is instantaneous. Einstein called this "spooky action at a distance" – it is called the Einstein, Podolsky, Rosen (EPR) paradox. The paradox is that if this related behavior were due to communication between the entangled objects it would be faster than the speed of light.

Entanglement was a controversial concept from the start. Physicists have investigated whether the phenomenon could be explained by "hidden local variables", i.e., that the entangled bits somehow have a hidden plan (metadata) on how they will behave under all measurement possibilities. In 1964 J. S. Bell produced a theorem to the effect that observed quantum results could not be explained by hidden local variables. Experiments have been carried out up to the present day, to attempt to validate the theorem as well as to close claimed loopholes in the various experimental setups. To date, experimental results have favoured the quantum (not hidden variable) explanation. The phenomenon is already in use in quantum communication.

An entangled pair could be created by splitting a single photon. The pair created have opposite polarisations, thus conserving the momentum of the original photon, see Figure 12.

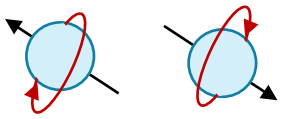


Fig. 12 Visualisation of a pair of entangled photons

Another example is a pair of electrons created together so that their spin sums to zero, as shown in Figure 6. Once a "random guess" of measurement axis has resulted in a measurement at the first object of an entangled pair, the value of the second is fixed on that axis. Although the values observed are random, after the first measurement the outcome is 100% predictable. This instantaneous effect is sometimes called quantum non-locality. Entanglement is fundamental to achieving speedup in quantum communications and computing. For this reason we'll spend some time spelling it out in detail.

An entangled state is a state consisting of multiple qubits which cannot be expressed in terms of individual qubits. Technically, the combined state cannot be tensor factorized.
The general expression for the state of two qubits can be written as their product:

$$(\alpha_0|0>+\alpha_1|1>)(\beta_0|0>+\beta_1|1>) = \alpha_0\beta_0|00>+\alpha_0\beta_1|01>+\alpha_1\beta_0|10>+\alpha_1\beta_1|11>$$

i.e. for tensor factorisation, given the combined expression on the right, we are looking for possible $\alpha_0, \alpha_1, \beta_0, \beta_1$. For example, the state |00> is not entangled because it can be described as |0> ⊗ |0> with α_1 and β_1 zero. Similarly, the states |01>, |10> and |11> are not entangled.

Consider the following 6 states, which are not so obvious as the above examples (in each case, we could have ± instead of + yielding 12 states):

(1) $\frac{1}{\sqrt{2}}|00>+\frac{1}{\sqrt{2}}|01> = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\0\end{pmatrix}+\frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\0\\0\end{pmatrix}=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\\0\\0\end{pmatrix}$ is not entangled since it can be factorised as

$$|0>(\frac{1}{\sqrt{2}}|0>+\frac{1}{\sqrt{2}}|1>)$$

(2) $\frac{1}{\sqrt{2}}|00>+\frac{1}{\sqrt{2}}|10> = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\0\end{pmatrix}+\frac{1}{\sqrt{2}}\begin{pmatrix}0\\0\\1\\0\end{pmatrix}=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\1\\0\end{pmatrix}$ is not entangled since it can be factorised as

$$(\frac{1}{\sqrt{2}}|0>+\frac{1}{\sqrt{2}}|1>)|0>$$

(3) $\frac{1}{\sqrt{2}}|00>+\frac{1}{\sqrt{2}}|11> = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\0\end{pmatrix}+\frac{1}{\sqrt{2}}\begin{pmatrix}0\\0\\0\\1\end{pmatrix}=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$ is an entangled state. It cannot be factorised since the terms |01> and |10> are missing and no non-zero values of $\alpha_0, \alpha_1, \beta_0, \beta_1$ can be found to achieve this.

(4) $\frac{1}{\sqrt{2}}|01> + \frac{1}{\sqrt{2}}|10> = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\0\\0\end{pmatrix} + \frac{1}{\sqrt{2}}\begin{pmatrix}0\\0\\1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\1\\0\end{pmatrix}$ is similarly an entangled state.

(5) $\frac{1}{\sqrt{2}}|01> + \frac{1}{\sqrt{2}}|11> = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\0\\0\end{pmatrix} + \frac{1}{\sqrt{2}}\begin{pmatrix}0\\0\\0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\0\\1\end{pmatrix}$ is not entangled since it can be factorised as

$$(\frac{1}{\sqrt{2}}|0> + \frac{1}{\sqrt{2}}|1>)|1>$$

(6) $\frac{1}{\sqrt{2}}|10> + \frac{1}{\sqrt{2}}|11> = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\0\\1\\0\end{pmatrix} + \frac{1}{\sqrt{2}}\begin{pmatrix}0\\0\\0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\0\\1\\1\end{pmatrix}$ is not entangled since it can be factorised as

$$|1>(\frac{1}{\sqrt{2}}|0> + \frac{1}{\sqrt{2}}|1>)$$

In states (1) and (6) the first qubit would be measured with 100% certainty as 0 in state (1) and 1 in state (6), which constrains the second bit to have the opposite value. In states (2) and (5) the second bit would be measured as 0 in state (2) and 1 in state (5) with 100% certainty which constrains the first bit to have the opposite value. In states (3) and (4) both qubits have 50% probability of being measured as 0 or 1. Measuring the first qubit immediately constrains the second to take the same value in state (3) and the opposite value in state (4). States (3) and (4) therefore show entangled pairs (EPR pairs). Before measuring, nothing is known. Measuring the first bit gives |0> or |1> with equal probability. After this, the second qubit is also determined. In neither case can their states be expressed as two single qubit states and their value predicted in advance of measurement.

Stated more generally, if one of these qubits is measured along any axis it behaves randomly but its random behaviour perfectly predicts how the other qubit would behave if measured along the same axis. No unentangled state exhibits this combination of individual randomness and perfect correlation.

Bell States

The four states $\frac{1}{\sqrt{2}}|00> \pm \frac{1}{\sqrt{2}}|11>$ and $\frac{1}{\sqrt{2}}|01> \pm \frac{1}{\sqrt{2}}|10>$ are called the *Bell states* (after John S. Bell's paper of 1964) and are said to be *maximally entangled*. As we saw when discussing superposition, a qubit pair can be modelled as being in infinitely many linear combinations of the 16 states $\begin{pmatrix}0\\0\\0\\0\end{pmatrix}$ to $\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$, most of which are not separable into individual qubits, i.e. the pairs are entangled. The four Bell states are an orthonormal basis for the entangled states of two qubits.

Entanglement applies to more than two qubits. It is the entanglement property that yields the large state space of quantum computing, arising from the complex probability amplitudes of qubits, and giving the potential for capturing multiple steps of complex algorithms simultaneously.

2.4 Measurement and transformation of qubits

The two-slit experiment. The particle/wave duality introduced above was explored in the early days of quantum physics. In the two-slit experiment, a single particle can behave as though it passes through *both* slits because interference patterns are observed on a light-sensitive screen beyond the slits. Interference patterns are caused by reinforcement and cancellation when waves are combined, as we see above for classical waveforms. However, if a detector is placed on the path to either or both slots, while allowing the particle to continue, then the interference effects disappear – the act of measurement has collapsed the qubit into a particle.

These observations of the quantum world are difficult to comprehend. To quote Niels Bohr ***"If quantum mechanics hasn't profoundly shocked you, you haven't understood it yet".***

We now consider measurement and transformation in more detail since they form the basis of quantum computing. At the end of this section we show that a qubit cannot be copied/cloned.

A little background is that Schrödinger's equation represents the evolution of a quantum system over time (the quantum equivalent of Newton's second law of motion), ignoring the system's environmental context. The solutions of Schrödinger's equation are independent stable states and are the possible bases for representing qubits. For example, |0> and |1> are independent in that |0> has no component in the |1> direction and vice versa. The general state of a qubit $\alpha|0> + \beta|1>$ means that α is the projection on the vector |0> and β the projection on the vector |1>. In Section 3 we see two alternative bases in use for a photon's polarisation, as shown in Figure 14. These bases (stable solutions of Schrödinger's equation) are called eigenvectors.

In German, *eigen* means self or own. An *eigenstate* of a quantised dynamic system (such as an atom, molecule or crystal) is one in which one of the variables defining the state (such as energy or angular momentum) has a determinable (measureable) fixed value. In the case of qubits we have only two dimensions and two eigenstates, i.e. two components in the representation of a qubit.

2.4.1 Measurement

In the mathematical modelling of physical processes, Hermitian matrices are associated with measurement. Hermitian matrices are complex, have real eigenvalues and their eigenvectors are orthogonal for different eigenvalues, so they can form a basis for the whole space. We now give some background from linear algebra for measurement and transformation. The Appendix gives definitions for reference.

It is a general principle of quantum mechanics that there is an operator (means of measuring) for every property that is physically observable. Every observable can be represented by a Hermitian operator (matrix), the eigenvalues of which are the possible values that can be obtained on measurement. Immediately after a measurement, the state of the system is the eigenstate associated with that eigenvalue. The real eigenvalues of a Hermitian matrix mean that the measured values are real and the orthogonal eigenvectors represent the bases of measurement. Projective measurement determines the projected values on the bases (eigenvectors) for each eigenvalue.

Introducing eigenvalues and eigenvectors in general, for our 2D vectors, an operator M is a 2x2 matrix and for an eigenvalue c (say), we require M|v> = c|v>. A general M changes both the direction of the vector it operates on and its magnitude.

e.g. $\frac{1}{\sqrt{2}}\begin{pmatrix}1 & -1\\ 1 & 1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$ and $\frac{1}{\sqrt{2}}\begin{pmatrix}1 & -1\\ 1 & 1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}-1\\1\end{pmatrix}$, see Figure 7 for the physical interpretation.

The projection of an eigenvector is onto itself, with a zero component in the orthogonal direction. Therefore, for eigenvectors, the output direction after applying M is the same as the input direction but in general, the magnitude changes,

e.g. $\begin{pmatrix}1 & 2\\ 2 & 1\end{pmatrix}\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}3\\3\end{pmatrix} = 3\begin{pmatrix}1\\1\end{pmatrix}$, 3 is an eigenvalue, $\begin{pmatrix}1\\1\end{pmatrix}$ is an eigenvector

$\begin{pmatrix}1 & 2\\ 2 & 1\end{pmatrix}\begin{pmatrix}1\\-1\end{pmatrix} = \begin{pmatrix}-1\\1\end{pmatrix} = -1\begin{pmatrix}1\\-1\end{pmatrix}$ -1 is an eigenvalue, $\begin{pmatrix}1\\-1\end{pmatrix}$ is an eigenvector.

The output of measurement of a qubit is a scalar, 0 or 1. For the general state $|\psi> = \alpha|0> + \beta|1>$, defined in Section 1, the outcome of measurement can be expressed as $<\psi|\psi>$ with the probability of measuring 0 (i.e. the probability amplitude of $|0>$) as $|\alpha|^2$ and of 1 as $|\beta|^2$.

The outcome of measurement after operation by M is $<\psi| M^\dagger M|\psi>$ where $M^\dagger$ is the complex conjugate of M. If we define operators M_α and M_β where the subscript refers to the measurement outcome, we have three postulates:

1. The probability of measurement outcome α is $<\psi| M_\alpha^\dagger M_\alpha|\psi>$, similarly for measurement outcome β.
2. Because the total probability over all measurement outcomes sums to 1, this implies that $M_\alpha^\dagger M_\alpha + M_\beta^\dagger M_\beta = 1$. This is known as the completeness relation.
3. Assuming outcome α has occurred, the state of the system immediately following the measurement is

$$|\psi> \rightarrow M_\alpha|\psi> / \sqrt{<\psi|M_\alpha{}^\dagger M_\alpha|\psi>}.$$ This is often called wave function collapse.

The operators M_α and M_β are known as projection operators, usually called P_0 (onto $|0>$) and P_1 (onto $|1>$). They are such that $P^2 = P$ and their eigenvalues must be 0 or 1. For $|\psi>$ they can be seen to be

$$P_0|\psi> = \begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \begin{pmatrix}\alpha\\ 0\end{pmatrix} = \alpha|0>$$

$$P_1|\psi> = \begin{pmatrix}0 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \begin{pmatrix}0\\ \beta\end{pmatrix} = \beta|1>$$

Checking postulate 1 above, if the probabilities of measuring 0 and 1 are $p(0)$ and $p(1)$, then

$$p(0) = <\psi|P_0^\dagger P_0|\psi> \text{ and } p(1) = <\psi|P_1^\dagger P_1|\psi>$$

since $<\psi|P_0^\dagger P_0|\psi> = <\psi|\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}|\psi> = <\psi|\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}|\psi> = <\psi|\alpha|0> = \alpha^*\alpha = |\alpha|^2$

then $p(0) = |\alpha|^2$ and similarly, $<\psi|P_1^\dagger P_1|\psi> = |\beta|^2$ so $p(1) = |\beta|^2$

Checking the completeness relation (item 2 above) for these matrices:

$$P_0^\dagger P_0 + P_1^\dagger P_1 = \begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix} + \begin{pmatrix}0 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}0 & 0\\ 0 & 1\end{pmatrix} = \begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix} + \begin{pmatrix}0 & 0\\ 0 & 1\end{pmatrix} = I.$$

Note that $P_0^2 = P_0$, $P_1^2 = P_1$, and $P_0 + P_1 = I$.

Checking item 3, $M_\alpha|\psi> = P_0|\psi> = \alpha|0>$, the denominator is the square root of $|\alpha|^2$ so the collapsed state is $|0>$. Similarly, for outcome β the collapsed state is $|1>$.

The analysis generalises to n qubits.

2.4.2 Partial measurement and probabilities

Under superposition in Section 2.2, we saw that the state of two bits can be written in the computational basis as:

$$\alpha_0|00> + \alpha_1|01> + \alpha_2|10> + \alpha_3|11> \text{ where } \sum_{i=0}^{3} |\alpha_i|^2 = 1$$

Now suppose that the first qubit is measured as $|0>$. We can remove the terms $\alpha_2|10> + \alpha_3|11>$ with first qubit $|1>$. The coefficients of the remaining terms must be recomputed to retain the condition on the probability amplitudes. This process is called **renormalisation** and results in:

$(\alpha_0|0> + \alpha_1|1>)/\sqrt{\alpha_0{}^2 + \alpha_1{}^2}$

If the first qubit is measured as $|1>$ the resulting second bit state is $\alpha_2|0> + \alpha_3|1>/\sqrt{\alpha_2{}^2 + \alpha_3{}^2}$

If the second qubit is measured as $|0>$ the resulting first bit state is $\alpha_0|0> + \alpha_2|1>/\sqrt{\alpha_0{}^2 + \alpha_2{}^2}$

If the second qubit is measured as $|1>$ the resulting first bit state is $\alpha_1|0> + \alpha_3|1>/\sqrt{\alpha_1{}^2 + \alpha_3{}^2}$

2.4.3 Changing the measurement basis

The standard computational basis is the orthogonal pair of vectors |0> |1>. A qubit can be measured with respect to any orthogonal basis, for example front/back: $\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle$ and $\frac{1}{\sqrt{2}}|0\rangle-\frac{1}{\sqrt{2}}|1\rangle$. These are sometime called |+> and |->. A vector expressed in one basis can be converted to another. For example

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle = |+\rangle, \qquad \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}0\\ 1\end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle-\frac{1}{\sqrt{2}}|1\rangle = |-\rangle.$$

Any vector in the basis |0>, |1>, $\alpha|0\rangle + \beta|1\rangle$, can be converted to the basis |+>, |-> by applying the same operator (matrix) $\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}\alpha\\ \beta\end{pmatrix}=\frac{1}{\sqrt{2}}\begin{pmatrix}\alpha+\beta\\ \alpha-\beta\end{pmatrix}$

i.e. $\alpha|0\rangle + \beta|1\rangle \rightarrow \frac{1}{\sqrt{2}}(\alpha+\beta)|+\rangle + \frac{1}{\sqrt{2}}(\alpha-\beta)|-\rangle$

2.4.4 General observations on measurement

Measurement is perhaps the most controversial and difficult topic in quantum mechanics and is essential for quantum computation. The analysis presented above omits much general discussion. It is worth highlighting some properties relevant to quantum computing and communication.

After a measurement in some orientation, if that observable is measured again in that orientation, the same result is obtained. Moreover, if one type of property becomes certain, through measurement, other types (or other aspects of the same type) become uncertain, as stated in Heisenberg's Uncertainty Principle. See Section 3 for a practical example of this principle in use.

Suppose we want to measure the state of a qubit A and the fabrication of A is such that we can (or by chance happen to) orientate our measuring apparatus in the correct direction of A's spin (say), as the up/down direction. If this is the case, then if the state of A is |0>, the value 0 will be measured with 100% probability; if the state of A is |1>, the value 1 will be measured with 100% probability. For example, if the value is SET to |1> it will be measured as 1 in the same orientation. If the value is CLEARed to |0> its value will be measured as 0; this is used in quantum computing. If the value of A is in some intermediate state then performing the measurement yields 0 or 1 with some probability.

If the value of A was actually in the up/down direction and measured in the right/left or back/front directions the returned values would be 0 or 1 with 50% probability, similar to a coin toss.
That is, the state $\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle$ yields 0 with 50% probability and 1 with 50% probability,
as do $\frac{1}{\sqrt{2}}|0\rangle-\frac{1}{\sqrt{2}}|1\rangle$, $\frac{1}{\sqrt{2}}|0\rangle+\frac{i}{\sqrt{2}}|1\rangle$ and $\frac{1}{\sqrt{2}}|0\rangle-\frac{i}{\sqrt{2}}|1\rangle$ with respect to the standard basis |0>, |1>.

If it was possible to measure the same qubit repeatedly (which is not the case), over many measurements the numbers of 0 and 1 would be in the proportions indicated by the probability amplitudes of the qubit. Similarly, if an algorithm is performed many times and the resulting value measured in the same orientation, over many repeats the result is a probability distribution. The same intuition on probabilities is applied to the results of measuring a long stream of qubits, for example, a photon stream. The results are regarded as probabilistic.

In all cases, after a qubit has been measured and yields 0 or 1, it remains in the state as measured, in that orientation, regardless of the state it was in before being measured, which is lost. Measurement has destroyed any superposition of states.

A qubit can in theory exist in an arbitrarily complex superposition (combination) of classical Boolean states, perhaps reflecting the chance orientation of our measuring device. In practice, a qubit will exist in a limited number of states, determined by such things as wavelength, phase, electron level, photon polarisation etc. This is manifested by the constraints on transformations that can be carried out.

2.4.5 Transformation

A unitary transformation preserves the lengths of vectors, that is, a qubit stays on the unit sphere under transformation, but its position on the sphere and its dynamic property such as spin or phase may be changed. Conversely, if a linear transformation of a unitary space (such as the surface of the unit sphere) preserves the lengths of all vectors, then it is unitary.

In quantum physics, **unitarity** is a restriction on the allowed evolution of quantum systems which ensures that the sum of probabilities of all possible outcomes of any event always equals 1. Unitary matrices have the property $U^{\dagger}U = UU^{\dagger} = I$, where $U^{\dagger}$ is the conjugate transpose of U, and preserve inner products and therefore length (see Appendix). A further point is that because a unitary operator is linear, it can be applied to the terms of an expression $\alpha\,|0\rangle + \beta\,|1\rangle$ in turn, i.e. to $\alpha\,|0\rangle$ then $\beta\,|1\rangle$. Together, unitarity and linearity can be used to prove that a qubit cannot be cloned, see Section 2.5.

Three matrices that represent quantum transformation operators are called the **Pauli matrices**, after their discoverer, Wolfgang Pauli.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ has eigenvectors $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ with eigenvalues $+1$ and -1,

For unitarity, check $U^{\dagger}U = UU^{\dagger} = I$: $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$

$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ has eigenvectors $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ i \end{pmatrix}$ and $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -i \end{pmatrix}$ with eigenvalues $+1$ and -1,

For unitarity check $\begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} = -I$

$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ has eigenvectors $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ with eigenvalues $+1$ and -1, check $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$

The Pauli matrices span the space of observables in 2D Hilbert Space. Each matrix represents the observable corresponding to spin along the corresponding axis in 3D classical space, see Figure 7. See Section 4.2 for these operators in use in quantum computing as Pauli gates. We'll see there again that they each flip a vector by 180° in relation to one of the axes of the Bloch sphere in Figures 3 and 5: up/down, right/left and front/back.

The effect of its environment on a system is captured by the notion of applying a Hamiltonian operator to the system. (After Sir William Rowan Hamilton, 1805 – 1865). The environmental effect is responsible for decoherence of the system which explains why a system must be isolated from its environment except through intended transformations. The operator which describes the progress of a physical system in time (its Hamiltonian) must be a unitary operator.

2.5 Nonclonability

Once a qubit is measured, the result cannot be used to generate a copy of the original qubit since the original state is lost. In general, it is impossible to generate a copy of a qubit's state. The nonclonability theorem was first published by Wootters, Zurek and Dieks in 1982 (*W Wootters and W Zurek 1982 "A single quantum cannot be cloned" Nature vol 299, 802-803. D Dieks 1982 "Communication by EPR devices" Physics Letters vol 92 no 6 271-272).*

This has Implications for audit since it is not possible to create a log as a process proceeds. Quantum error correction, that classically requires replication, is not straightforward, see Section 4.4.

The proof follows from the linearity and unitarity of quantum operators. Quantum operators are linear operators, so can be taken as operating on the components of a quantum state $\alpha |0\rangle + \beta |1\rangle$ in turn. Unitary operators are defined in Section 2.4.1 and the Appendix.
Quantum operators must have equal numbers of inputs and outputs for reversibility (see Section 4). To clone a qubit we need a Unitary operator U (let's call it a cloner) with first input $\alpha |0\rangle+\beta |1\rangle$ and (say) second input a cleared qubit $|0\rangle$. The required outputs are two copies of $\alpha |0\rangle+\beta |1\rangle$ with a combined state:
$(\alpha |0\rangle+\beta |1\rangle)(\alpha |0\rangle+\beta |1\rangle) = \alpha^2|00\rangle + \alpha\beta|01\rangle + \alpha\beta|10\rangle + \beta^2|11\rangle$.
Because of linearity, for the $|0\rangle$ component (omitting coefficients for simplicity), the first input is $|0\rangle$ taken with the second input $|0\rangle$. The cloner is required to produce two outputs $|0\rangle$ and $|0\rangle$ ($|00\rangle$). For the $|1\rangle$ component taken with the second input $|0\rangle$, the cloner is required to produce two outputs $|1\rangle$ and $|1\rangle$ ($|11\rangle$). It is not possible in general for the cloner to produce the full required state without violating linearity.

An alternative demonstration that such a cloner is not possible comes from unitarity: unitary matrices preserve inner products. Suppose we have a cloner U. For inputs $|\psi\rangle$ and $|0\rangle$ with inner product $\langle\psi|0\rangle$ the operation by U gives outputs $U|\psi\rangle$ and $U|0\rangle$ with inner product $\langle\psi| U^\dagger U|0\rangle=\langle\psi|0\rangle$. The required output from the cloner is $|\psi\rangle$ and $|\psi\rangle$ with inner product $\langle\psi|\psi\rangle$ which is not the same as $\langle\psi|0\rangle$. In terms of components of $|\psi\rangle= \alpha |0\rangle+\beta |1\rangle$, the inner product of the inputs $\langle\psi|0\rangle$ is α^* whereas the inner product of the outputs $\langle\psi|\psi\rangle$ is $\alpha\alpha^* +\beta\beta^*$, which are not equal unless $\beta=0$, $\alpha=1$.

In the next section we consider the example of a photon as a qubit. Photons are already being used for secure distribution of secret encryption keys in the research domain, exploiting some of the above properties. Wider deployment is underway in projects on secure networking. After seeing how photons are used in this application we will return to the above properties.

3 Quantum key distribution

Classical cryptography uses both secret key and public key methods. A Public Key Infrastructure (PKI) typically uses encryption based on a public and private key-pair to securely transmit a shared secret session key that is used in more efficient subsequent encryption. We shall return to this and give a more detailed description in Section 5.

The PKI approach is based on the complexity of breaking the public/private encryption scheme, which is based on a key being the product of two very large prime factors. The factorisation required to break PKI

would take many thousands of years on a classical computer. The possibility that Quantum Computing might allow classical encryption to be broken has led to research in Quantum key distribution (QKD). QKD has been shown to work in practice in the research domain and wider network deployments are underway.

QKD is an example of a system based on qubits, in this case photons. For photons, the qubit is not mapped onto two energy levels, but rather onto two *polarisation* degrees of freedom as the required two orthogonal basis states. We shall call them *rectilinear* and *diagonal*, modelled respectively on the Bloch sphere as up/down and right/left, as shown in Figure 14.

The requirement for QKD is that two parties, Alice and Bob, who wish to communicate securely must first have a copy each of a secret encryption key. The problem is how to transmit a secret key as a stream of photons while ensuring that any eavesdropping on the communication by Eve is detected. Figure 13 presents the scenario.

We assume a photon generator that generates a random stream of photons to be used as an encryption key. If the transmission and measurement basis is rectilinear on the classical sphere (see Figure 14), then V (vertical) can be encoded to represent 1 and H (horizontal) to represent 0. If the basis is diagonal (at 45° to the rectilinear basis), then L (left) 135° can be encoded to represent 1 and R (right) 45° to represent 0. *For each photon she transmits, Alice knows the base used and polarisation with respect to that base.*

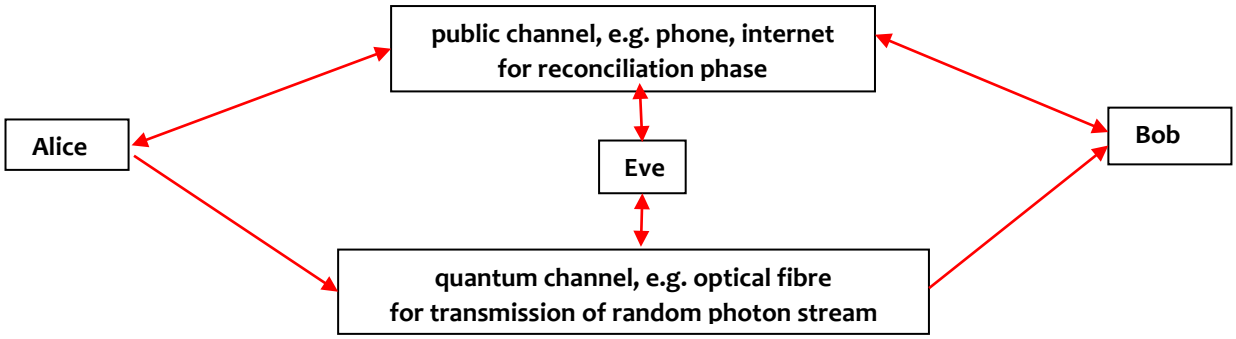


Fig.13 QKD scenario

3.1 The BB84 protocol

We now describe a protocol known as **BB84**, due to its inventors Charles Bennett and Gilles Brassard, to achieve the secure transfer of a secret key (*C H Bennett and G Brassard "Quantum cryptography: Public key distribution and coin tossing" Proc IEEE Int conf on Signal processing Vol 175 p8 New York 1984*).

Alice and Bob wish to communicate. Alice generates a random stream of photons which she sends to Bob. At the end of the protocol's execution Alice and Bob will agree on the secret key they will use. The protocol must take into account the possibility of an eavesdropper (Eve) both during the original transmission and during the reconciliation phase when the selected bit pattern is agreed by Alice and Bob.

In the original transmission, Bob receives the stream of photons generated by Alice but it is impossible to measure the base and polarisation of a single photon (Section 2.1). Instead, Bob has a polarisation beam splitter which allows him to make a random guess for each photon of the measurement basis that has been used; the polarisation can then be recorded as the corresponding value 0 or 1 depending on the observed polarisation and agreed encoding. If the random guess of base is correct, the correct polarization is recorded. If the random guess of base is wrong, a random polarization (and binary value) is recorded.

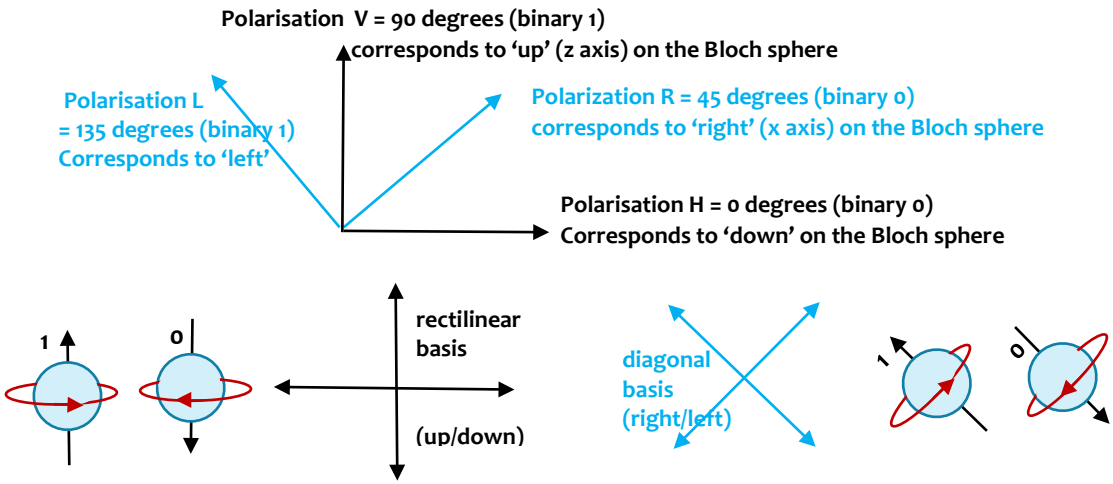


Fig. 14 Bases for photon polarization encoding

After the measurement, the photon is polarised in the state it was measured in, with all the information about its initial polarisation lost. (Section 2.1). The responses from the measurement of the photon are given in Table 1. The rows represent what Alice sent, the columns represent Bob's guesses and the contents show the response of the photon.

Table 1 Results of attempting to measure the value of a photon

	base +	**base x**
+ V	1	Random 0/1
+ H	0	Random 0/1
x L	Random 0/1	1
x R	Random 0/1	0

Rows are what Alice sends in the form base and polarisation: **+** (V or H) or **x** (L or R).
Columns are Bob's guesses of base (**+** or **x**).
If the base is correct, the polarisation is correct, if the base is wrong a random value is returned (due to uncertainty).

Table 2 Example showing transmitted and measured values and agreed bits of secret key

Alice's random bit	**1**	**0**	**1**	**1**	**0**	**1**	**0**	**0**
Alice's random sending basis	**+**	**+**	**x**	**+**	**x**	**x**	**x**	**+**
Alice's chosen Photon polarisation	**V**	**H**	**L**	**V**	**R**	**L**	**R**	**H**
Bob's random Measuring basis	**+**	**x**	**x**	**x**	**+**	**x**	**+**	**+**
Photon polarization Bob measures	**V**	**L**	**L**	**L**	**H**	**L**	**H**	**H**
Public Reconciliation								
Shared secret key	**1**		**1**			**1**		**0**

The table shows that if Bob guesses the right base he measures the correct polarisation V or H (which by agreement with Alice he can record as 1 or 0).

If Bob guesses the wrong base (at ±45° to the actual base used), he gets a random response for the polarisation (which again, he records as 1 or 0).

After Bob has received and measured the values of each of the photons in the stream, Bob and Alice now communicate via a public channel to agree their secret key.

We assume they are time-synchronised to the required accuracy for distinguishing and comparing the photons. This process is called ***reconciliation***. Alice knows for each bit the base she chose and the polarisation with respect to it. Bob transmits his guesses to her but cannot send the values 0 and 1 in case

Eve is eavesdropping. However, he can securely send his sequence of guesses for the base. For each guess Bob sends, Alice replies right or wrong. They use the values Alice says are correct (in the form of the corresponding stream of bits, 0 and 1) as the secret key. Recall they have an agreement of the correspondence between V, H, L, R and the values 0 and 1, e.g. V=1, H=0, L=1, R=0 but any adversary listening on the public channel does not see and cannot know the bit sequence. That is, if Eve eavesdrops during reconciliation, she does not learn the encryption key because each "correct" does not correspond to a unique base and polarisation.

Now suppose Eve intercepts the original transmission of the key. If Eve reads a photon, as described above for Bob she is not able to read and retransmit (copy) the full quantum state (Section 2.1) but can only measure its projection onto a plane according to one or other base. She has to guess the base and polarisation of the original photon and will be wrong on average in 50% of the cases.

When Eve is present and Alice and Bob go through the reconciliation procedure, they will detect a higher error rate than 50% which indicates that the bits have been read and incorrectly replaced during transmission. An alternative check is that they can agree to sacrifice some initial number of bits of the key which they now both have. Alice transmits the photons corresponding to these bits to Bob. If Bob receives them correctly, he knows there is no Eavesdropper on the channel, who could not avoid introducing errors on copying.

There are other variations on the BB84 protocol such as B92 but our next example uses a different property of quantum systems, that of entanglement (and teleportation).

3.2 Using entanglement to transmit a secret key, the E91 protocol

BB84 has one party generating and transmitting the photon stream to be used as the secret key. An alternative approach is for Alice and Bob to each have one photon from each pair of a stream of entangled photon pairs. These can be sent by a third party source (such as a satellite, but which might also be Eve) or it could be by Alice or Bob. Figure 15 shows the general scenario with a third party source.

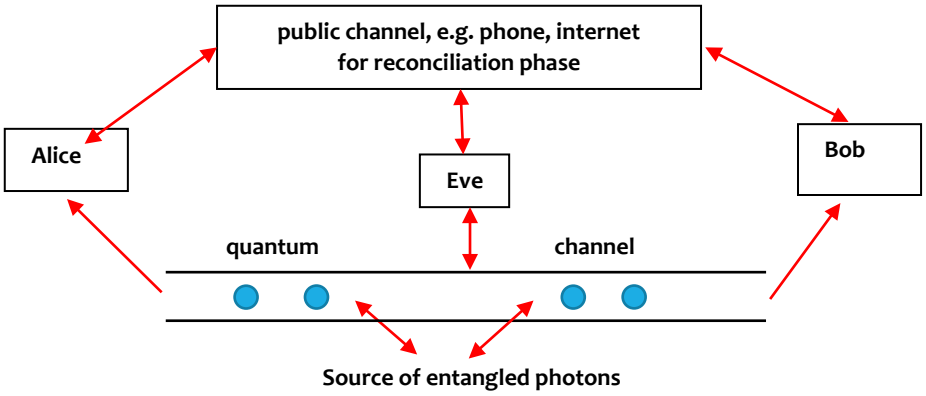


Fig. 15 QKD scenario using a stream of entangled photon pairs

We now describe the E91 protocol, due to Artur Ekert (*A K Ekert 1991 "Correlations in quantum optics" DPhil thesis, University of Oxford*)(*A Ekert, 1991, "Quantum cryptography based on Bell's theorem" Physical Review letters, American Physical Society, vol 67, pp661—663*) (*Ref Wikipedia Quantum Key Distribution and Quantum entanglement*)

The pairs of entangled photons emitted by the source are separated. Alice receives one photon from each pair and Bob receives the other. Alice and Bob can only choose their base on which to measure their particle at random (Section 2.1). As in BB84, they can discuss in clear on a public channel which bases they used for their measurements. For each measurement where Alice and Bob chose the same base, they expect opposite results due to the principle of quantum entanglement (Section 2.3). One party can invert their bits and they now have a shared secret key.

In other words, the scheme relies on two properties of entanglement. First, the entangled states are perfectly correlated, that is, if Alice and Bob both measure whether their particles have horizontal or vertical polarisations, they always get the same answer with 100% probability. This is also the case if both measure any other pair of complementary (orthogonal) polarisations.

The presence of an eavesdropper can be detected because as above in Section 3.1 , reading an entangled photon and reinserting a random attempt at copying it into the stream to be received by one or other party, will be detected.

3.3 Wide area deployment of QKD

A deployment phase of QKD is underway for use in larger scale networks, offering one-to-many communication, not only point-to-point. In traditional large-scale networks, classical bits tend to degrade and are periodically intercepted, cleaned by the removal of noise and retransmitted as 0 or 1. This is done by a **repeater** node. Attempting this for a QKD network would introduce the problem that qubits are not simply 0 or 1, so cannot be cleaned, and cannot be copied by measurement without losing their state (Section 2.5).

Work is in progress on designing repeaters and routers for quantum networks. Section 4 introduces quantum gates and there we see how two qubits can be entangled and unentangled, processes that are used in designing repeaters and routers. Quantum teleportation is described in Section 5.1 together with a design for a quantum repeater.

4 Quantum computing

Section 1 described the classical computer architecture: a stored program computer where both data and code are represented by bit sequences in memory, see Figure 1. The code is executed by fetching, decoding and executing the stored instructions one-at-a-time, see Figure 2. Instruction execution can involve loading data from memory into the processor, operating on data in the processor, storing results in memory and branching within the code to do repeated executions or to end the program when a terminating condition is detected.

We now look at the concepts from quantum physics that underlie how a qubit may be operated on. In Section 6 we look at the technologies and engineering involved in attempting to build a quantum computer. The aim of this section is to give some intuition on how qubits might be set up, operated on to solve problems and the solution read out.

4.1 Transformations of qubits

In Section 2.4 we introduced the mathematical basis of measurement and transformation of qubits. Whereas measurement collapses the state of a qubit, transformation (without measurement) operates on the state and allows subsequent transformations.

We saw that while Hermitian matrices represent measurement, Unitary matrices, which in addition preserve the unit lengths and therefore inner-products $\langle\psi|\psi\rangle$ of vectors, are needed to represent transformations. This is in order to preserve the probabilities of measured outcomes summing to 1. A necessary condition for a matrix to represent a unitary transformation is $U^\dagger U = UU^\dagger = I$, where $U^\dagger$ is the conjugate transpose of U. Unitary operators (U) can be shown to be norm-preserving because $\langle U\psi|U\psi\rangle=\langle\psi| U^\dagger U|\psi\rangle=\langle\psi|\psi\rangle$ (also see Appendix). Quite apart from engineered transformations, the operator which describes the progress of a physical system in time (its Hamiltonian) must be a unitary operator.

A further point is that because a unitary operator is linear, it can be applied to the terms of an expression in turn, e.g. to $\alpha|0\rangle + \beta|1\rangle$ in turn.

4.2 Quantum gates and circuits

A quantum logic gate (quantum gate) is a quantum circuit that operates on a small number of qubits, in the same way that classical logic gates (e.g. AND, OR, XOR, NOT) comprise the circuits in classical computers. It can be shown from the properties of Unitary operators that quantum gates must be reversible, unlike many classical gates. They therefore have the same number of inputs and outputs. Since it is possible to contrive to perform classical computing using only reversible gates, it follows that quantum circuits can perform all operations performed by classical circuits. In more detail, the following properties hold:

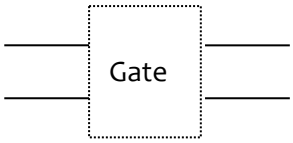
- Any unitary operation on n qubits can be implemented by a sequence of 2-qubit operations.
- Any unitary operation can be implemented by a combination of controlled NOT (C-NOT) gates and single qubit operations, for more detail see below.
- Exceptions to general Unitary operators are CLEAR (to value $|0\rangle$) and MEASURE, which we have already discussed as collapsing the state to $|0\rangle$ or $|1\rangle$ with some probability, losing the pre-measured state.

Mathematically, a quantum gate performs a reversible unitary transformation (Section 4.1) on the qubits. That is, quantum gates are represented by unitary operators. For a single qubit, unitary transformations correspond to rotations of the qubit (unit) vector on the Bloch sphere (Figures 3 and 5) to specific superpositions.

Quantum logic gates are the building blocks for quantum circuits in most quantum computers. (The D-Wave computer, see Section 7.6 instead uses a process called quantum annealing). Quantum gates can operate on one, two or three qubits. The number of qubits in the input and output of a gate must be equal. Figure 16 a, b, and c introduces some notation that is used for representing quantum circuits diagrammatically.

Fig. 16 a) single qubit gates

Fig. 16 b) two qubit gates take the form:



example of a gate including a control

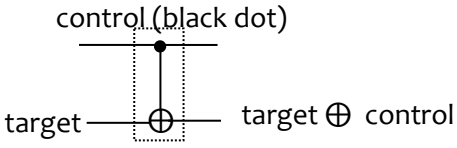


Fig. 16 c) notation for n qubit gates

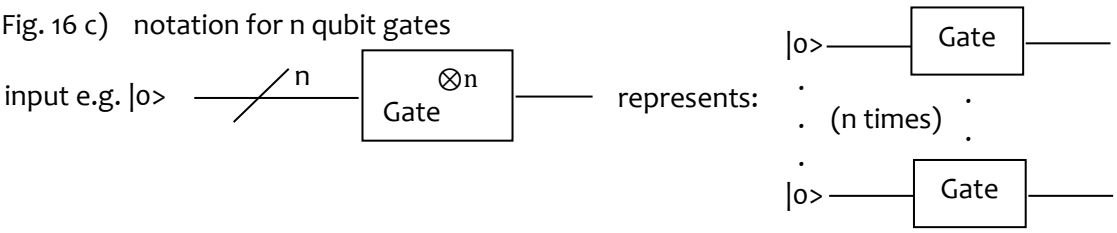


Fig. 16 a, b and c Notation for transformation of qubits via quantum gates

It is important to note that in the circuit diagrams, lines represent qubits under time evolution rather than the transmission along physical wires that classical circuit diagrams indicate. Note also that the circuit representation does not imply that the control qubit is copied (which is impossible due to the fundamental no-cloning property of qubits); we should think of a gate with n inputs and n outputs. Application of a gate represents a transformation achieved by electromagnetic fields applied to the qubits.

4.2.1 The quantum NOT gate (X) for a single qubit

The quantum NOT gate is defined as X |0> = |1> and N |1> = |0>

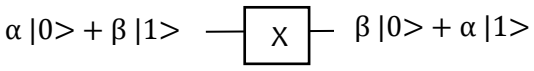
Its matrix form is $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ where $X|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

$X|1\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

More generally, a qubit input to a NOT gate is a superposition $\alpha|0\rangle + \beta|1\rangle$. The linearity of unitary operators means that a gate can be seen as being applied to each component in turn, so we have:

$X(\alpha|0\rangle + \beta|1\rangle) = \beta|0\rangle + \alpha|1\rangle$

The general circuit representation of X is



For unitarity, check $X^\dagger X = XX^\dagger = I$: $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$

4.2.2 Phase shift gates, for a single qubit

In the case of nuclear spin, phase can be manipulated by applying electric and/or magnetic fields. Recall from Section 1 that the state of a qubit is

$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + (\cos\emptyset + i\sin\emptyset)\sin\left(\frac{\theta}{2}\right)|1\rangle$, where $0<\theta<\pi$, $0<\emptyset<2\pi$

As the phase angle Ø changes the vector moves around the sphere at its line of latitude; as the phase angle θ changes it moves around the sphere at its longitude, see Figure 5. When measuring in the standard basis, a change in Ø does not affect the result.

Several gates manipulate the phase Ø, leaving the probability of measuring |0> or |1> unchanged, but engineering cancellations. This is part of quantum computer programming and occurs as part of the operation of some gates, e.g., the Hadamard gate (Section 4.2.3)

A phase shift operator has the form $\begin{pmatrix} 1 & 0 \\ 0 & \cos\emptyset + i\sin\emptyset \end{pmatrix}$

It has no effect on |0> and changes the phase of |1> by Ø (i.e. the phase of the |1> component in a superposition), that is:

$\begin{pmatrix} 1 & 0 \\ 0 & \cos\emptyset + i\sin\emptyset \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 1 & 0 \\ 0 & \cos\emptyset + i\sin\emptyset \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = (\cos\emptyset + i\sin\emptyset)\begin{pmatrix} 0 \\ 1 \end{pmatrix}$

When Ø = 180° = π, $\cos\emptyset + i\sin\emptyset$ = -1, inverting the |1> component of a state.

When Ø = 90° = $\pi/2$, $\cos\emptyset + i\sin\emptyset = i$, rotating the |1> component by $\pi/2$

When Ø = 45° = $\pi/4$, $\cos\emptyset + i\sin\emptyset = \frac{1}{\sqrt{2}}(1+i)$, rotating the |1> component by $\pi/4$.

4.2.3 The Hadamard operator for a single qubit

1. On first application to |0>, the Hadamard acts like a random NOT with 50% chance if measured of resulting in 0 or 1. In our previous notation this is $\frac{1}{\sqrt{2}}|0> + \frac{1}{\sqrt{2}}|1>$
2. On first application to |1>, as for |0>, the Hadamard acts like a random NOT with 50% chance if measured of resulting in 0 or 1. The difference is that *the phase of the resulting 1 is reversed*. This is $\frac{1}{\sqrt{2}}|0> - \frac{1}{\sqrt{2}}|1>$
3. Applying the Hadamard operator twice always returns the qubit to its original value

Figure 17 gives a visualisation of two applications of the Hadamard gate to a single qubit. (Ref: davidkemp.github.io/QuantumComputingArticle – which includes animations of operators).

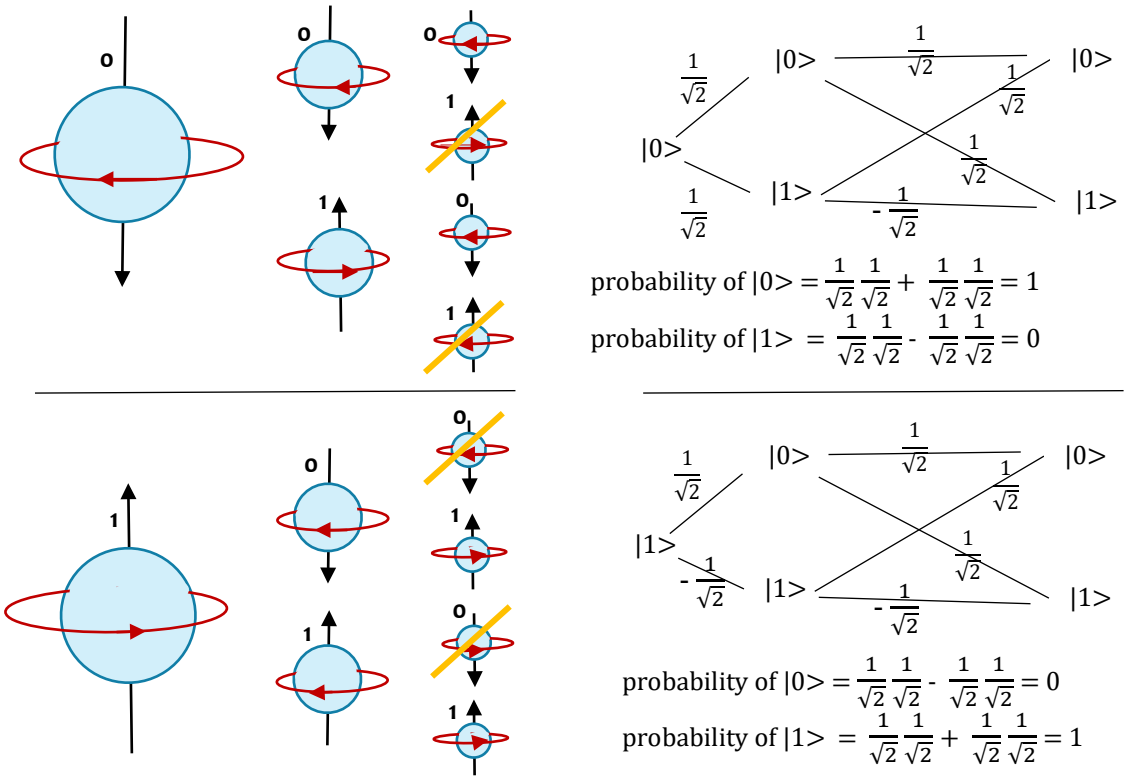


Fig. 17 Two applications of the Hadamard gate to a qubit

These results would be counterintuitive unless opposite phases can be seen as cancelling each other out; the notion of *interference*. Figure 10 gives the idea for classical waveforms. The calculation of probabilities uses the **Feynman path rule**. Note that all the paths stay active throughout the operations. It is results such as these that led Neils Bohr to comment "*If quantum physics hasn't profoundly shocked you, you haven't understood it yet*". It is also effects such as these that have to be exploited to achieve collapsing to the required solution in quantum computing algorithms.

A mathematical treatment makes the operation apparent and simple to understand in two different ways:
The Hadamard gate has matrix representation $H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$ which represents a reflection in the $\pi/8$ line.
As for all quantum operators, H is unitary since $H^\dagger = H$ and $H^\dagger H = HH^\dagger = I$:

$$H\,H\,|\psi> = \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}|\psi> = \tfrac{1}{2}\begin{pmatrix}2 & 0\\ 0 & 2\end{pmatrix}|\psi> = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}|\psi>$$ (the identity matrix)

Second, following the transformations of |0> and |1> visualised in Figure 17, we see

$H\,|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$ applying H again gives $\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix}$

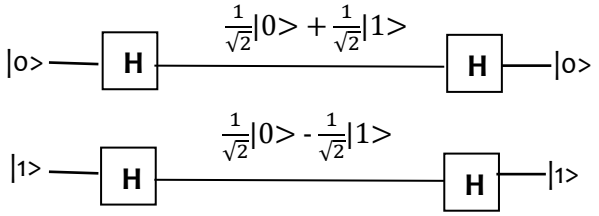
i.e. $H\,|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ $\quad H\,H\,|0\rangle = |0\rangle$

$H\,|1\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$ applying H again gives $\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix}$

i.e. $H\,|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$ $\quad H\,H\,|1\rangle = |1\rangle$

The algebra can similarly be worked through for the general superposed input $\alpha|0\rangle + \beta|1\rangle$. For any input state $|\psi\rangle$ applying H twice gives the same output state, i.e. $H\,H\,|\psi\rangle = |\psi\rangle$

The circuit representation is as follows:



The circuit also shows that H is reversible, as required for a quantum gate.

In Section 4.2.5 we show how a Hadamard gate can be used with a controlled C-NOT gate to entangle and unentangle two qubits.

Hadamard gates can be constructed to operate on any number of qubits. As introduced in Section 2.3.1, the combined quantum state of two qubits is equal to the tensor product ⊗ of the constituent qubits. An entangled state cannot be tensor factorised i.e. it cannot be written as a tensor product of its constituent qubit states. The tensor product of two n-qubit quantum gates generates the gate that is equal to the two gates in parallel, operating on 2n qubits. The Hadamard gate H operating on two qubits is H ⊗ H

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \otimes \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{pmatrix}$$

Note that $\frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{pmatrix}\begin{pmatrix}1\\0\\0\\0\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$ an equal superposition $\frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$

Section 4.3.1 discusses the use of H in initialising qubits in quantum computations.

4.2.4 The Pauli (single qubit) gates

We introduced the Pauli matrices in Section 2.4 and checked that they are unitary. There is a gate corresponding to each, which acts on a single qubit. Each causes a rotation of 180° around one of the axes.

The Pauli-X gate (X) is also the NOT gate $\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$. It is also called bit-flip since it maps |0> to |1> and vice versa i.e. it rotates a qubit by 180° around the up/down (z) axes as shown in Figure 18.

For Pauli-X = $\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$ $X\,|0\rangle = |1\rangle$ and $X\,|1\rangle = |0\rangle$ i.e. a rotation by 180° around the z (up/down) axis.

The Pauli-Y gate $\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}$ is a rotation by 180° around the front/back (y, imaginary) axes.

$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ $(\frac{1}{\sqrt{2}}|0> + \frac{i}{\sqrt{2}}|1>) = \frac{1}{\sqrt{2}}|0> - \frac{i}{\sqrt{2}}|1>$ and vice versa.

The Pauli-Z gate $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ is a rotation by 180° around the right/left (x, real) axes.

$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ $(\frac{1}{\sqrt{2}}|0> + \frac{1}{\sqrt{2}}|1>) = \frac{1}{\sqrt{2}}|0> - \frac{1}{\sqrt{2}}|1>$ and vice versa.

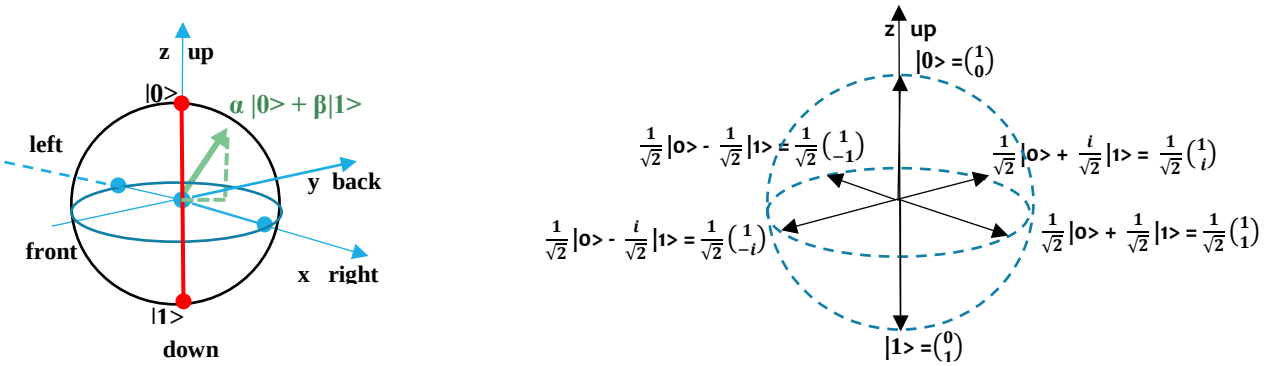


Fig. 18 The Pauli operators

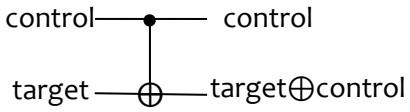
4.2.5 The controlled not gate C-NOT for two qubits (quantum XOR)

C-NOT is a two qubit gate.
If the first (most significant) qubit is |1> then C-NOT flips the second (least significant).
If the first (most significant) qubit is |0> then C-NOT leaves the second (least significant) unchanged.
We consider general superposed inputs, each of the form $\alpha\ |0> + \beta\ |1>$, after these cases.
C-NOT mirrors the reversible form of the classical XOR, $\oplus$ having two inputs and two outputs.

We introduced the notation for two qubits as ket vectors in Section 2.3.

$$|00> = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad |01> = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad |10> = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad |11> = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

The C-NOT operator is the 4x4 matrix: $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ with circuit:



The general structure is $\begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix}$.

For reference, controlled H $= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$, controlled Z $= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$

C-NOT maps |00> to |00>, |01> to |01>, |10> to |11> and |11> to |10>

Check: $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$ etc.

The specification of C-NOT for inputs of |0> and |1> and its circuit diagram give us the intuition that the control is unchanged by C-NOT. This is not the case for general superposed inputs of the form $\alpha\,|0> + \beta\,|1>$.

For example, for inputs: control $\frac{1}{\sqrt{2}}(|0> + |1>)$ target $\frac{1}{\sqrt{2}}(|0> - |1>)$

The combined input state is $\frac{1}{\sqrt{2}}$ (|00> - |01> + |10> - |11>)

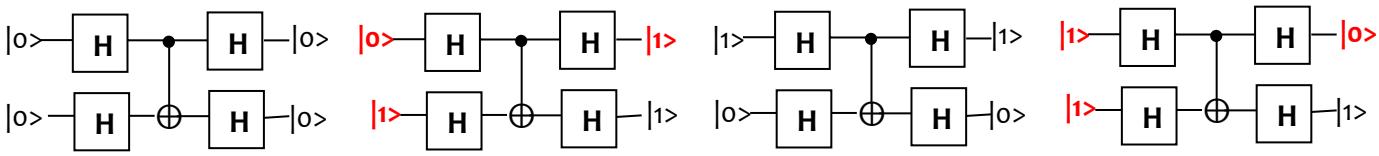
and the output $\frac{1}{\sqrt{2}}$ (|00> - |01> + |11> - |10>)

Separating the output gives: control $\frac{1}{\sqrt{2}}(|0> - |1>)$ target $\frac{1}{\sqrt{2}}(|0> - |1>)$

So in the general case, the control can change as well as the target: the control is affecting the target but the target is also affecting the control. Similar results can be obtained by working through with inputs as other combinations of

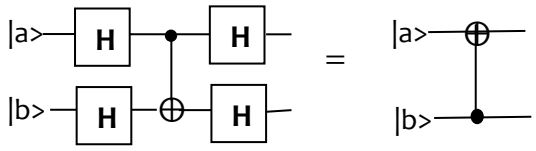
control $\frac{1}{\sqrt{2}}(|0> \pm |1>)$ target $\frac{1}{\sqrt{2}}(|0> \pm |1>)$

Putting together C-NOT with Hadamard gates gives a useful component for some quantum algorithms.



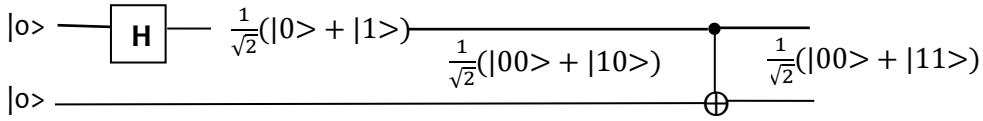
Note that the second output is always the same as the second input; the first output is unchanged when the second input is |0>; the first output is flipped when the second input is |1> (as in an inverted C-NOT). The Deutsch-Jozsa algorithms outlined in Section 5.3 build on these circuits.

Generalising |0>, |1> to |a>, |b>, it can be shown that for all superposed inputs:



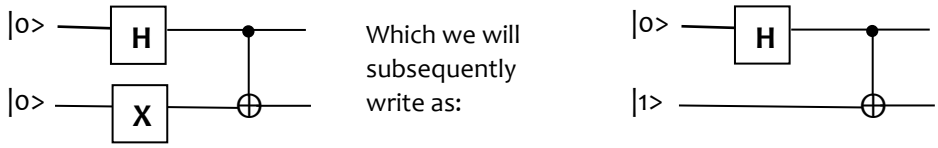
4.2.6 Entangling two qubits using C-NOT

The C-NOT gate has the important property of generating entangled states, introduced in Section 2.3. Consider the following circuit:



The Hadamard gate takes |0> to an equal superposition $\frac{1}{\sqrt{2}}(|0> + |1>)$. Since H is not applied to the second qubit, the input to the C-NOT gate is $\frac{1}{\sqrt{2}}(|0> + |1>)|0> = \frac{1}{\sqrt{2}}(|00> + |10>)$ which is clearly not entangled. Applying the C-NOT gate leaves |00> unchanged (since the control bit is 0) and takes |10> to |11> (since the control bit is 1). The output of this circuit is therefore $\frac{1}{\sqrt{2}}(|00> + |11>)$ which is entangled state (3) in the examples in Section 2.3. This is an entangled state since it cannot be factorised. The bits are the same with a 50% probability of being 00 or 11. Measuring one constrains the other to be the same. In Section 2.3 it was defined as one of the four maximally entangled Bell States.

Now consider:

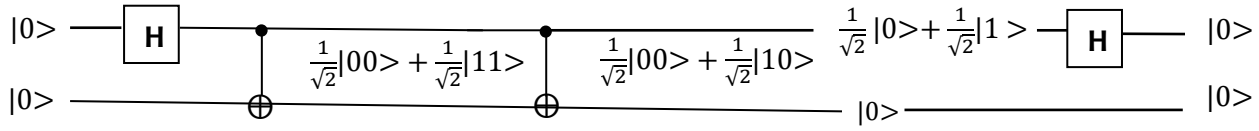


In this circuit, the input to the C-NOT gate is $\frac{1}{\sqrt{2}}(|01> + |11>)$ and the output $\frac{1}{\sqrt{2}}(|01> + |10>)$, state (4) in Section 2.3. This is another of the maximally entangled Bell states. This time, measuring one qubit constrains the other to be the opposite. We can follow this procedure for inputs |0> |1> and |1> |1> to obtain the other two Bell states $\frac{1}{\sqrt{2}}(|00> - |11>)$ and $\frac{1}{\sqrt{2}}(|01> - |10>)$.

The C-NOT gate is therefore able to create entangled states. As mentioned in Section 2.3, entanglement yields the multiple states that give the potential for speedup in quantum computing.

4.2.7 Unentangling two qubits: Bell State Measurement (BSM)

The gates used above for entangling two qubits can be used in reverse order for separating (unentangling) two qubits in one of the four Bell States. Quantum gates are defined to be reversible, but following through the detail confirms this, for example:



The output of the RHS C-NOT is $\frac{1}{\sqrt{2}}(|00> + |10>)$ where the second qubit is now separable as |0>. The H gate applied to the first qubit cancels out the |1> terms, yielding |0>. And similarly for the other three Bell States. Again, notice that this process shows the reversibility of quantum gates.

Measuring the output of the circuit on the right can determine which Bell State two qubits were in, in this case |00>. This is a simple example of Bell State Measurement (BSM). In Section 5 we show how the circuit is used more generally in superdense coding and so-called quantum teleportation. An application is to create repeaters and routers for quantum networks, as mentioned in Section 3.

4.2.8 Toffoli gates (3 qubit gate)

Toffoli gates operate on 3 qubits. They are also called **CCNOT** gates.

Toffoli truth table for 3 classical bits

000 -> 000
001 -> 001
010 -> 010
011 -> 011
100 -> 100
101 -> 101
110 -> 111
111 -> 110

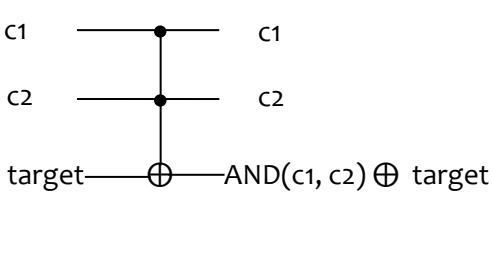
3 qubits as ket vectors (8 states)

$$\begin{pmatrix}1\\0\\0\\0\\0\\0\\0\\0\end{pmatrix}\begin{pmatrix}0\\1\\0\\0\\0\\0\\0\\0\end{pmatrix} \cdots\cdots \begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\1\end{pmatrix}$$

Toffoli operator in matrix form:

$$\begin{pmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&0\end{pmatrix}$$

Toffoli operator as circuit:



If we consider inputs of only |0> or |1> then if the first two bits are in state |1> it applies a NOT to the third (least significant) bit, else it does nothing.

The purpose is to provide a quantum, reversible AND gate. The fact that a logical AND of the first two bits yields a 1 is indicated by a flip in the third (least significant) bit.

4.2.9 Transforming the basis of an operator

Section 2.4.4 described how to translate a vector's representation from one computational basis to another by applying an operator, say B, to the vector. Suppose a matrix U represents an operator in the standard computational basis |0> |1> and the matrix B translates any vector represented in that basis (|0>, |1>) to some other basis. For example, we saw in Section 2.4.4 that B=H= $\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$ for translating a vector from the basis |0>, |1> to the basis |+>, |->.

It can be shown that the transformation BUB^{-1} translates an operator in |0>, |1> to its equivalent in the new basis. For example, X=$\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$ in |0>, |1> becomes $\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$ in |+>, |->

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} = \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}\begin{pmatrix}2 & 0\\ 0 & -2\end{pmatrix} = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$$ i.e. HXH = Z

4.3 Quantum computing programs

We saw in Section 2.4 that although in theory a qubit and system of qubits can exist in infinitely many states, measurements result in eigenstates. The above definitions introduce that a quantum computing program transforms a system from some initial state to others according to a quantum algorithm to achieve a result (with some probability). However, no measurement (or copy) can be taken at any intermediate state since this would collapse the system into one irrevocable value.

Putting all this together, quantum gates (the circuits of a quantum computer) are only allowed to carry out unitary operations, without directly measuring (and therefore collapsing) the superposed distribution of states. In other words, a program combines state distributions without knowing them. Therefore, what programs do effectively is to "steer" a collection of qubit states (in physics, this might be called an ensemble) through a sequence of such distributions from one allowed ensemble to the next one, until the desired solution is present. When this is detected (by some means) it can be "read out" by measurement, collapsing the last set of qubit superposed states, into one classical reading. Depending on the algorithm, this may have to be done many times because of the probabilistic nature of some results. The job of the quantum computer "programmer" is therefore to design/choose circuits (comprising gates) to achieve this steering .

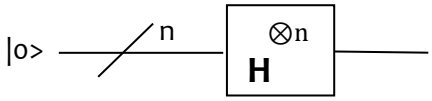
Note that this approach returns us to the early Colossus-type model of a computer where the circuits are designed to solve one problem. A quantum computer is not a general stored-program computer.

As mentioned in Section 1.2, there are similarities with Bayesian programming in classical systems, in which we combine probabilities within event-chains to build up the relative probabilities associated with different outcomes. But in Bayesian programming there is no notion of the interference (leading to cancellations due to phase differences) or entanglement that we see in quantum computing.

4.3.1 Initialisation The starting point for a computation will need to be set up. In classical computing we might read a particular number (bit sequence) into a register. In quantum computing, achieving this depends on the physical realisation of the qubits. For example, a photon is typically created with a specific polarisation. For other realisations, cooling might put the qubit into a ground state, typically zero. Gates can transform the qubits to the required values. We have introduced some quantum gates above; Section 6 describes some current technology for implementing qubits.

In practice, qubits will be prepared for reuse after being measured. Any that measured to zero can be reused as they are, starting from |0>, any that measured to 1 can be put through a NOT (X) gate to create |0>. Initialisation can therefore be abstracted by introducing a non-reversible CLEAR gate that clears a qubit to |0>. The usual starting procedure is to initialise each qubit to state |0>. The next step is to prepare the system to be in a superposition of all states, see Sections 2.2 and 4.2.3. That is, instead of n bits being in state |0000……000>, they are transformed to an equal superposition. This is achieved by applying a Hadamard transformation to the system. There is a Hadamard (H) gate for any number of bits, or we can see it as applying the H gate we saw above to each bit.

In circuit form this can be represented as:
(See Figure 16 for circuit notation)



We can see the effect for a single qubit from the 2x2 matrix representation of H.
As above: $H|0\rangle = H\begin{pmatrix}1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$ i.e. the state |0> is transformed to an equal superposition of states $|0\rangle \text{ and } |1\rangle$, each with amplitude $\frac{1}{\sqrt{2}}$, i.e. $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$

For two qubits we have $H|00\rangle = H\begin{pmatrix}1\\0\\0\\0\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\\1 & -1 & 1 & -1\\1 & 1 & -1 & -1\\1 & -1 & -1 & 1\end{pmatrix}\begin{pmatrix}1\\0\\0\\0\end{pmatrix} = ½\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$, an equal superposition of states |00>, |01>, |10>, |11>, each with amplitude ½ (and the squares of the amplitudes (4 x¼) summing to 1). Generalising, 3 bits have 8 states, each with amplitude $\frac{1}{\sqrt{8}}$, 4 bits 16 states, each with amplitude ¼ , 5 bits 32 states, and so on.

4.3.2 Example of a type of program:

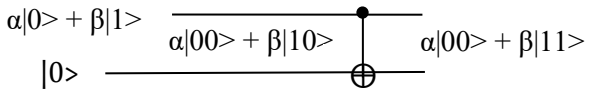
A classical approach to solving a problem could be, for example, to iteratively search a large space until arriving at the desired answer. Moving from one step to the next can be hugely expensive and it is sometimes impossible to solve problems to a feasible, useable time-frame, even when heuristics or metaheuristics are used. The main difference of a quantum computer is that its circuits act together in one "clock cycle". Each step moves the system on towards the required answer which can finally be measured. The next section presents some quantum computing algorithms and considers their possible impact.

4.4 Quantum error correction (QEC)

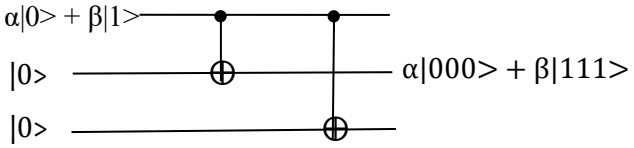
An underlying problem for quantum computing is that of quantum error correction (QEC). In a classical digital system it is relatively easy to remove noise that can lead to errors e.g., a binary pattern sent as a signal along a wire can be cleaned (regenerated) periodically by repeaters into 0s and 1s. In a computer, the circuits that operate on classical digits can be made to output clean representations of 1 and 0, and bit patterns in memory can be kept correct. That is, a pattern can be seen as close to a 1 or a 0 and reconstructed.

This is not possible in a quantum system since a qubit is in a complex superposition of states and does not collapse to 1 or 0 until measured. Quantum gates apply fields to qubits and gates themselves can add small errors. Also, over time, a qubit tends to be slightly perturbed by its environment (Hamiltonian), eventually losing coherence. It is the entanglement of qubits that leads to the multiplicity of states that give quantum computing its potential. The fabrication of qubits must therefore make interaction (coupling) possible. If, for example, neutrinos were used they would be free from being corrupted by their environment but would not be capable of the interactions required for quantum algorithms.

Without QEC, the number of operations that comprise an algorithm would be severely limited; it is only the possibility of QEC that makes quantum computation feasible. However, the non-cloning theorem means that it is not possible to take copies of quantum state. The no cloning theorem was outlined in Section 2.5, making use of the linearity and unitarity properties of quantum operators. Instead, a C-NOT can create an entangled pair that can be tested, separated and measured.



A natural approach might be to attempt to encode every computational bit as three qubits, so state |0> would become |000> and |1> would become |111>. A general qubit state α|0>+β|1> could then be encoded as the entangled state α|000>+β|111>, achieved by the following circuit:



This seems promising for allowing an error in one bit to be corrected, by taking the value of the majority e.g. |010> could have a NOT (X) gate applied to the middle bit to reconstruct |000>. More recent work has used 5 and 7 qubits to encode each computational qubit. However, unlike classical computing, measurement in quantum computing destroys the measured state so cannot be used to monitor the values.

To solve the problem of not being able to measure the replicated qubits, two ancillary qubits, associated with the three qubits for each state, are used and measured. In outline, the ancillary bits are initialized to |0> and are combined using C-NOT gates with α|000> + β|111>. If α|000> + β|111> has not suffered a bit flip, the ancillary bits are measured as 0 and 0. Otherwise, the ancillary bits' measured values are used to determine the necessary correction that needs to be applied. The details are left for further reading.

The outline above has only considered a bit flip. A phase change might also be problematic when the algorithm being computed is designed to rely on cancellation, or when an unwanted phase change causes an unwanted cancellation. These situations must also be detected and corrected without measuring the qubits comprising the calculation.

5 Quantum algorithms and protocols

In two-state quantum computers, every bit is a qubit which is capable of representing any value between 0 and 1 with some probability. Due to quantum properties, a qubit can be thought of as representing all these values simultaneously. Only when a qubit is measured does it adopt the value 0 or 1 and after this is done it cannot change Section 2.1 and the values in the pre-measurement superposition are lost.

The fact that a sequence of qubits can represent every possible result of applying an algorithm to some data, as outlined in Section 4.1, has led to the interest in quantum computing. Algorithms that are so complex that classical computers would take many thousands of years to compute the result, might become solvable on a quantum computer. Possible applications of quantum computing include the following:

1. Random number generation; classical systems use pseudo-random numbers.
2. An atomic clock on a chip for accurate timestamping, e.g. for sensor readings.
3. Modelling molecules for chemical and materials science.

4. Searching a (large) data set for a given value.
5. Finding the prime factors of very large numbers.

Application 4 is discussed further in Section 5.4 on Grover's algorithm. Application 5 has caused substantial investment in the development of quantum computing, following the publication of a quantum algorithm for prime factorization by P. W. Shor in the 1990s, see Sections 5.5 and 5.6. We now introduce some quantum algorithms to give a flavour of what is possible. We first consider how entanglement can be used in quantum networks.

5.1 Quantum teleportation

In Section 3.2 we introduced how entanglement can be used in transporting a secret key. We have now defined a number of quantum gates and can show a similar protocol in more detail. In 1993 it was shown by Bennet et al. that an unknown quantum state can be transported by exploiting entanglement. Figure 19 shows the circuits used to achieve the transfer (note, not a copy) of Alice's qubit to Bob.

As shown in Figure 19, Alice and Bob each have a bit from an entangled pair, for example $\frac{1}{\sqrt{2}}|00> + \frac{1}{\sqrt{2}}|11>$ (state 3 in Section 2.3). The state of the three bits is:

$$\frac{\alpha}{\sqrt{2}}|000> + \frac{\alpha}{\sqrt{2}}|011> + \frac{\beta}{\sqrt{2}}|100> + \frac{\beta}{\sqrt{2}}|111>$$

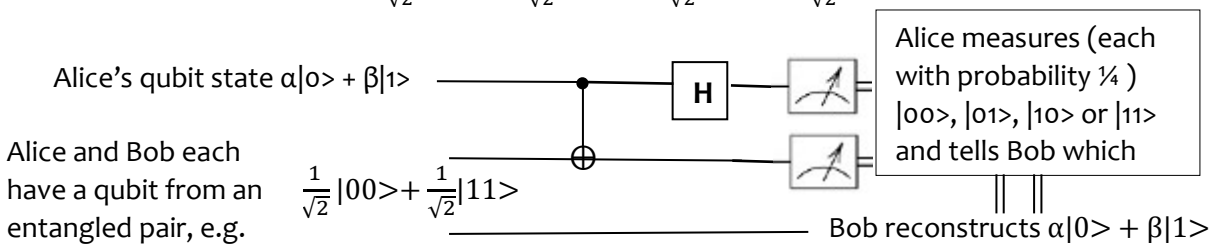


Fig. 19 Quantum teleportation

Alice inputs the qubit she wishes to transfer to Bob and her entangled qubit into a C-NOT circuit as shown in Figure 20. After the C-NOT the state is:

$$\frac{\alpha}{\sqrt{2}}|000> + \frac{\alpha}{\sqrt{2}}|011> + \frac{\beta}{\sqrt{2}}|110> + \frac{\beta}{\sqrt{2}}|101>$$

Then the H gate is applied to the first bit

$$\frac{\alpha}{\sqrt{2}}(H|0>(|00> + |11>)) + \frac{\beta}{\sqrt{2}}(H|1>(|10> + |01>))$$

After which the state is:

$$\frac{\alpha}{\sqrt{2}}(\frac{1}{\sqrt{2}}(|0> + |1>)(|00> + |11>)) + \frac{\beta}{\sqrt{2}}(\frac{1}{\sqrt{2}}(|0> - |1>)(|10> + |01>))$$

Rearranging this expression in preparation for measuring the first two bits gives:

$$\frac{1}{2}|00>(\alpha|0> + \beta|1>) + \frac{1}{2}|01>(\beta|0> + \alpha|1>) + \frac{1}{2}|10>(\alpha|0> - \beta|1>) + \frac{1}{2}|11>(-\beta|0> + \alpha|1>)$$

This form also makes it clear how the third bit (Bob's) is affected by the entanglement on measurement. Alice measures the first two bits as |00>, |01>, |10> or |11> each with probability ¼. Because the second bit is entangled with Bob's (the third bit) measuring it has affected Bob's bit. The expression above gives the four possible forms that Bob's bit can take when Alice measures the first two bits. Alice then communicates these two bits to Bob using a regular channel.

If Alice sends |00> Bob knows his qubit is already in the required state $\alpha|0> + \beta|1>$.
If Alice send |01> Bob applies a NOT (X) gate to his qubit $\beta|0> + \alpha|1>$ to get $\alpha|0> + \beta|1>$.

If Alice send |10> Bob applies a Pauli-Z gate $\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$ to his qubit α|0> - β|1> to get α|0>+ β|1>.

If Alice send |11> Bob applies Z X= $\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix} = \begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix}$ to his -β|0> + α|1> to get α|0>+ β|1>.

These actions are called *quantum state reconstruction.*

Note that Alice's bit has been transferred, not copied, since by measuring her two bits she has collapsed their states. As discussed in Section 3.3, repeaters and routers are needed for multi-hop quantum networks. This protocol can form the basis for their design. Redrawing the above scheme as in Figure 20, the operations can be extended to create a repeater.

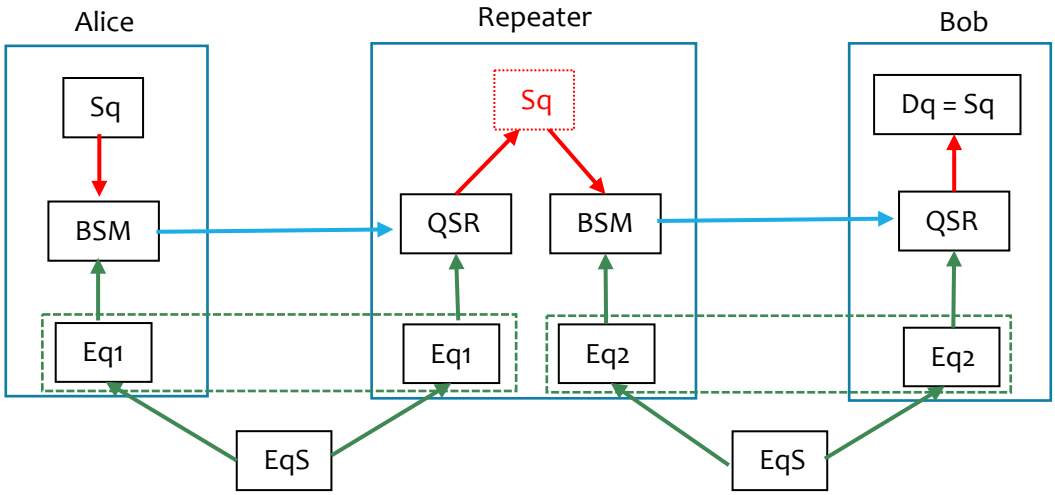


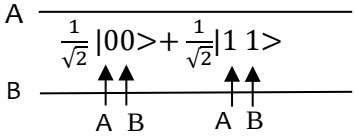
Fig. 20 A quantum repeater

Sq and Dq are source and destination qubits. BSM and QSR are Bell state measurement and quantum state reconstruction as in Figure 20. EqS is entangled qubit source; Eq is a qubit of an entangled pair.

5.2 Superdense coding

Above we saw how a qubit quantum state can be transferred without copying. Superdense coding allows Alice to transfer two measured bits (00, 01, 10, 11) to Bob by means of a single communication. The requirement is that Alice and Bob, share an entangled qubit pair:

Alice and Bob each have a qubit from an entangled pair



Alice measures her two bits then carries out the following operations on her entangled qubit:

Recall that X = $\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$ and Z = $\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$ are used in quantum state reconstruction during quantum teleportation, depending on the values of the two measured bits as follows:

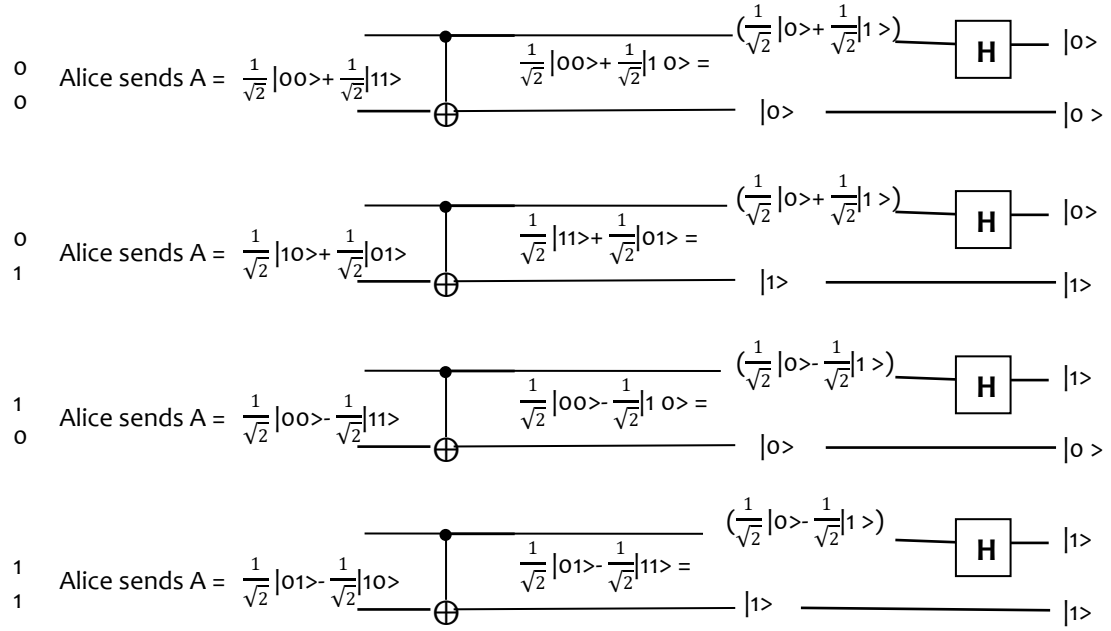
= 1
= 0 Alice applies Z to A and sends $\frac{1}{\sqrt{2}}|00>-\frac{1}{\sqrt{2}}|11>$ to Bob

= 1
= 1 Alice applies ZX to A and sends $\frac{1}{\sqrt{2}}|01>-\frac{1}{\sqrt{2}}|10>$ to Bob

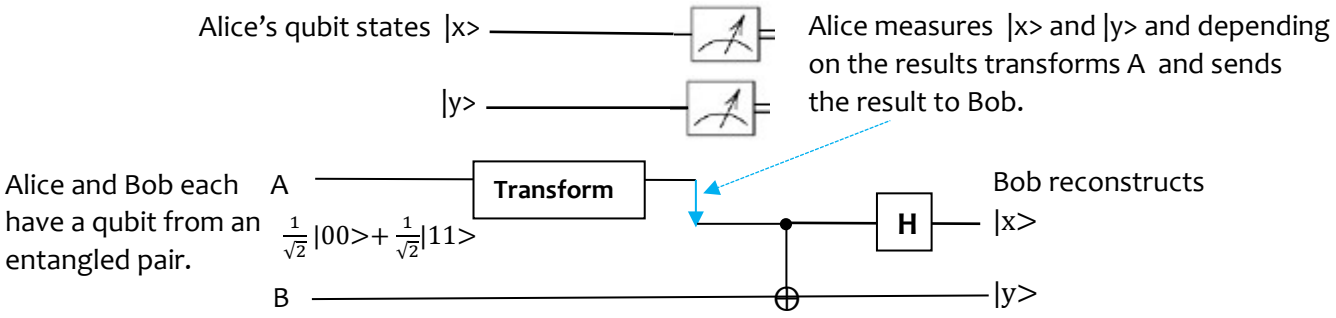
0
0 Alice sends her entangled qubit $\frac{1}{\sqrt{2}}|00\rangle+\frac{1}{\sqrt{2}}|11\rangle$ to Bob

0
1 Alice applies X to A (the first bit of each part of her qubit) and sends $\frac{1}{\sqrt{2}}|10\rangle+\frac{1}{\sqrt{2}}|01\rangle$ to Bob

On receiving the entangled qubit state from Alice, Bob applies a C-NOT gate to the incoming pair A, B, then a Hadamard gate to the resulting A, as we saw in Section 4.2.5 for unentanglement and BSM.



Bob can therefore measure the two bits that Alice holds, resulting from a communication of one qubit. An outline of the circuit is:



5.3 Deutsch and Deutsch-Jozsa algorithms

The possibility of devising algorithms to demonstrate quantum speedup compared with classical algorithms was investigated by David Deutsch and Richard Jozsa in the 1990s. They explored functions to map binary strings to binary strings. The starting point was a function to map |x> to |f(x)>, where both are a single qubit |0> or |1>. This was generalized to strings of any length n where the function is applied separately to each bit of the string.

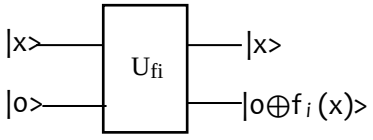
The possibilities for single-bit input and output are (omitting brackets for conciseness):

x	f_o	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

We shall below distinguish functions f_o and f_3 (called *constant*) from functions f_1 and f_2 (called *balanced*, yielding 0 for half of the input domain and 1 for the other half).

It is not possible to devise a single-qubit quantum gate of a form that achieves any of the functions, since the associated transformation matrices operating on $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ are not all unitary ($U^\dagger U = UU^\dagger = I$), being for $f_o \begin{pmatrix} 11 \\ 00 \end{pmatrix}$, $f_1 \begin{pmatrix} 10 \\ 01 \end{pmatrix}$, $f_2 \begin{pmatrix} 01 \\ 10 \end{pmatrix}$, $f_3 \begin{pmatrix} 00 \\ 11 \end{pmatrix}$

We can instead design a two-bit quantum circuit in the form we used for C-NOT, using an ancillary bit $|0\rangle$.



$\oplus$ is XOR and the second output is the required function applied to the input $|x\rangle$. The following unitary matrices (so valid transformations) are associated with the functions f_o, f_1, f_2, f_3:

$$f_o \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{pmatrix} \quad f_1 \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{pmatrix} \quad f_2 \begin{pmatrix} 0&1&0&0 \\ 1&0&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{pmatrix} \quad f_3 \begin{pmatrix} 0&1&0&0 \\ 1&0&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{pmatrix}$$

These are called permutation matrices. Notice that they include $f_1 = I$ and f_0 = C-NOT; all can be created from C-NOT and X gates.

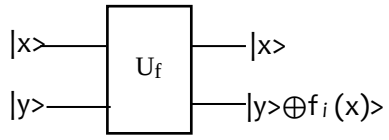
Recall that in vector representation $|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ $|10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$ to check the operators.

Summarising the operation of the circuit with ancillary bit $|0\rangle$ we have:

x, f_o, $0 \oplus f_o$ U_f	x, f_1, $0 \oplus f_1$ U_f	x, f_2, $0 \oplus f_2$ U_f	x, f_3, $0 \oplus f_3$ U_f
0, 0, 0, $\|00\rangle \rightarrow \|00\rangle$	0, 0, 0, $\|00\rangle \rightarrow \|00\rangle$	0, 1, 1, $\|00\rangle \rightarrow \|01\rangle$	0, 1, 1, $\|00\rangle \rightarrow \|01\rangle$
1, 0, 0, $\|10\rangle \rightarrow \|10\rangle$	1, 1, 1, $\|10\rangle \rightarrow \|11\rangle$	1, 0, 0, $\|10\rangle \rightarrow \|10\rangle$	1, 1, 1, $\|10\rangle \rightarrow \|11\rangle$

We could instead use $|1\rangle$ as ancillary bit and $|01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$ $|11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$, with a rearrangement of the f_i

We have, somewhat circuitously, arrived at a widely used equation in quantum algorithms for binary functions:



$U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle$

We used it for a single input bit $|x\rangle$ but it generalises to n input bits $|x\rangle$ plus an ancillary bit $|y\rangle$.

U_f as a "black box." Now suppose that the circuit U_f is a "black box", i.e. we don't know which of the f_i it implements. In order to tell, we would have to test the circuit with both $|0\rangle$ and $|1\rangle$. Deutsch and Jozsa showed that it's possible to tell from a single operation of a similar quantum circuit whether it implements a constant function (as f_o and f_3) or a balanced function (as f_1 and f_2), given that the function is one or the other. Here, the inputs are prepared as equal superpositions, as we saw in Section 4.5.2. Figure 21 shows the

form of the circuit for a 1-bit string input as |x>. Figure 22 shows intermediate values.

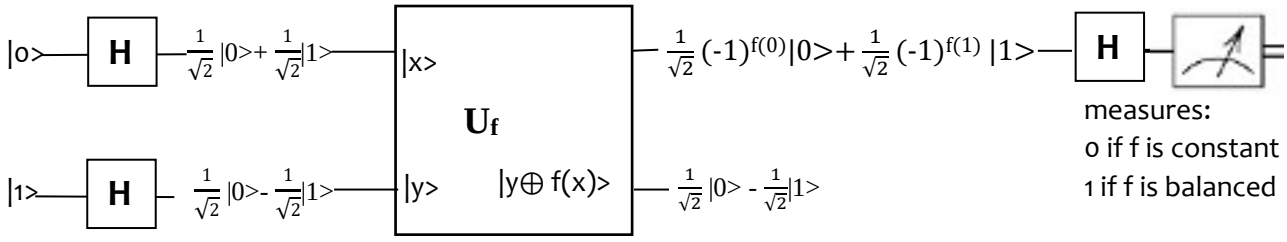


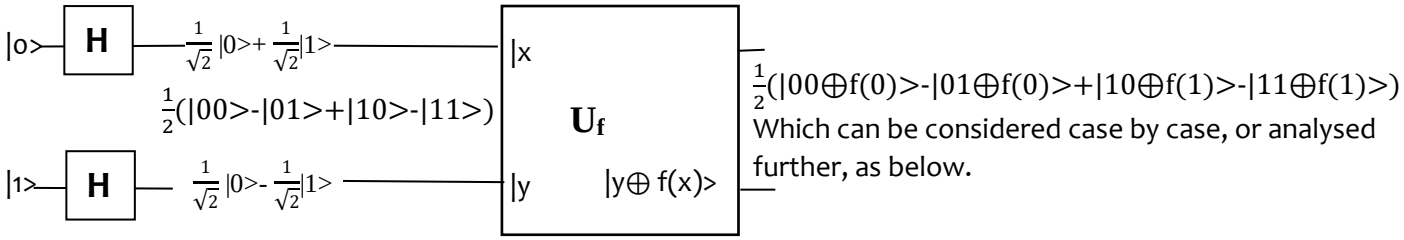
Fig. 21 Deutsch-Jozsa algorithm for a 1-bit constant or balanced function - overview

|1> is used as the second input in order to selectively change the sign of components of the first input, as we first saw in Section 4.2.5.

Let's consider the |x> output of U_f in Figure 21, i.e., $\frac{1}{\sqrt{2}}(-1)^{f(0)}|0>+\frac{1}{\sqrt{2}}(-1)^{f(1)}|1>$, for all cases, then show where it comes from.

f(0) f(1)	$\frac{1}{\sqrt{2}}(-1)^{f(0)}\|0>+\frac{1}{\sqrt{2}}(-1)^{f(1)}\|1>$	apply H	measured as
0 0	$\frac{1}{\sqrt{2}}(-\|0> - \|1>)$	$-\|0>$	0
1 1	$\frac{1}{\sqrt{2}}(\|0> + \|1>)$	$\|0>$	0
0 1	$\frac{1}{\sqrt{2}}(-\|0> + \|1>)$	$-\|1>$	1
1 0	$\frac{1}{\sqrt{2}}(\|0> - \|1>)$	$\|1>$	1

So if f is *constant* the first output is measured as o, if f is *balanced*, the first output is measured as 1.
Let's see where the first output comes from:



$\frac{1}{2}(|00\oplus f(0)>-|01\oplus f(0)>+|10\oplus f(1)>-|11\oplus f(1)>)$
Which can be considered case by case, or analysed further, as below.

Fig. 22 Deutsch-Jozsa algorithm for a 1-bit constant or balanced function - detail

The input to U_f is : $\frac{1}{2}(|00>-|01>+|10>-|11>) = \frac{1}{2}|0>(|0>-|1>)+\frac{1}{2}|1>(|0>-|1>)$

The output from U_f is : $\frac{1}{2}|0>(|0\oplus f(0)>-|1\oplus f(0)>)+\frac{1}{2}|1>(|0\oplus f(1)>-|1\oplus f(1))$

For f(0) = f(1) = 0, this is: $\frac{1}{2}|0>(|0>-|1>)+\frac{1}{2}|1>(|0>-|1>)=\frac{1}{\sqrt{2}}(|0>+|1>)\frac{1}{\sqrt{2}}(|0>-|1>)$

For f(0) = f(1) = 1, this is: $\frac{1}{2}|0>(|1>-|0>)+\frac{1}{2}|1>(|1>-|0>)=-\frac{1}{\sqrt{2}}(|0>+|1>)\frac{1}{\sqrt{2}}(|0>-|1>)$

For f(0) =0, f(1) = 1, this is: $\frac{1}{2}|0>(|0>-|1>)+\frac{1}{2}|1>(|1>-|0>)=\frac{1}{\sqrt{2}}(|0>-|1>)\frac{1}{\sqrt{2}}(|0>-|1>)$

For f(0) =1, f(1) = 0, this is: $\frac{1}{2}|0>(|0>-|1>)+\frac{1}{2}|1>(|1>-|0>)=-\frac{1}{\sqrt{2}}(|0>-|1>)\frac{1}{\sqrt{2}}(|0>-|$

Transforming the first output using a Hadamard gate gives ±|o> for a constant f which both measure as o, and ±|1> for a balanced f which both measure as 1.

For a more formal analysis we make use of the formula: $H|x>=\frac{1}{\sqrt{2}}|0>\pm\frac{1}{\sqrt{2}}|1>=\frac{1}{\sqrt{2}}|0>+\frac{1}{\sqrt{2}}(-1)^x|1>$

Continuing from above, the output from U_f

$$= \frac{1}{2}(-1)^{f(0)}|0\rangle(|0\rangle - |1\rangle) + \frac{1}{2}(-1)^{f(1)}|1\rangle(|0\rangle - |1\rangle)$$
$$= \frac{1}{2}(-1)^{f(0)}|0\rangle + \frac{1}{2}(-1)^{f(1)}|1\rangle)(|0\rangle - |1\rangle)$$
$$= (\frac{1}{\sqrt{2}}(-1)^{f(0)}|0\rangle + \frac{1}{\sqrt{2}}(-1)^{f(1)}|1\rangle)\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

The second output can be discarded. The first output, as shown in Figure 22, and the associated table, is transformed by H

$(\frac{1}{\sqrt{2}}(-1)^{f(0)}|0\rangle + \frac{1}{\sqrt{2}}(-1)^{f(1)}|1\rangle) = \pm|0\rangle$ if $f(0) = f(1)$ (either both are $|1\rangle$ or both are $|0\rangle$)

$= \pm|1\rangle$ if $f(0) \neq f(1)$

On measuring, the value 0 indicates a constant function and 1 a balanced function.
Alternatively, the first output can be rewritten as:

$$\frac{1}{\sqrt{2}}(-1)^{f(0)}(|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle)$$

The final H application takes this state to: $(-1)^{f(0)}\, f(0)\oplus f(1)$
The measurement therefore results in the value of $f(0)\oplus f(1)$ which is 0 if f is constant and 1 if f is balanced.

The result generalises to a binary string of any length n, in which case $(2^n/2 + 1)$ applications of the function are needed for a classical system, one for a quantum system. Figure 23 shows the circuit.

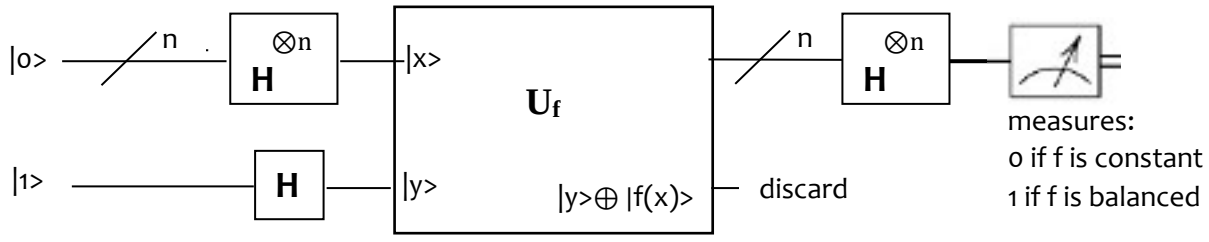


Figure 23 Deutsch-Jozsa algorithm for n bits

The intuition here is that all possibilities for the input values |x> are input in parallel. These can start off as |0> because the Hadamard gate transforms them into equal superpositions of |0> and |1>, that is, all possible inputs are being considered in parallel. The analysis uses similar techniques to those used in the analysis of the single bit algorithm.

5.4 Grover's algorithm for unstructured search

Grover's algorithm (1996) is often described as "database search" which implies that a very large set of data can be searched. In general, you might be comparing a value you have for a "key". If you find the item in the data list with that key you can read the information stored with the key. An example is that you have a phone number and want the associated name and address by searching a telephone directory. Large data sets are beyond the scope of quantum computers, even those envisaged for the long term. However, Grover's algorithm could be considered for breaking certain forms of cryptography in timescales similar to those envisaged for Shor's algorithm. For example, the set of possible passwords, always stored in encrypted form, might become a feasible search target. Simplifying, if passwords may only contain alphanumeric characters, the search field is quite small. If the encrypted (hashed) value is available to query the algorithm, the result would be the password in-clear.

The assumption is that the data items to be searched are in random order, unlike an alphabetically sorted indexed list like a telephone directory. Applications that are being considered for Grover's algorithm include inverting hash functions of certain short items, finding hash collisions and proof-of-work computations in some implementations of blockchain technology.

Searching for a value in an unordered list of length N takes on average N/2 comparisons, with maximum N comparisons if you don't know whether the item is in the list. Grover's algorithm takes $\sqrt{N}$ steps. We have seen the types of gate that are proposed to implement Grover's algorithm in Section 4 and proof of concept implementations exist for small numbers of bits.

We assume the items to be searched are represented in *n* qubits which can have 2^n values (states). As outlined in Section 4.3, the bits are first initialized to state 0, then a Hadamard transformation is applied so that the system is in a superposition of all possible states. A simplified circuit of the overall procedure is given below and Figure 24 visualises the operation of the algorithm.

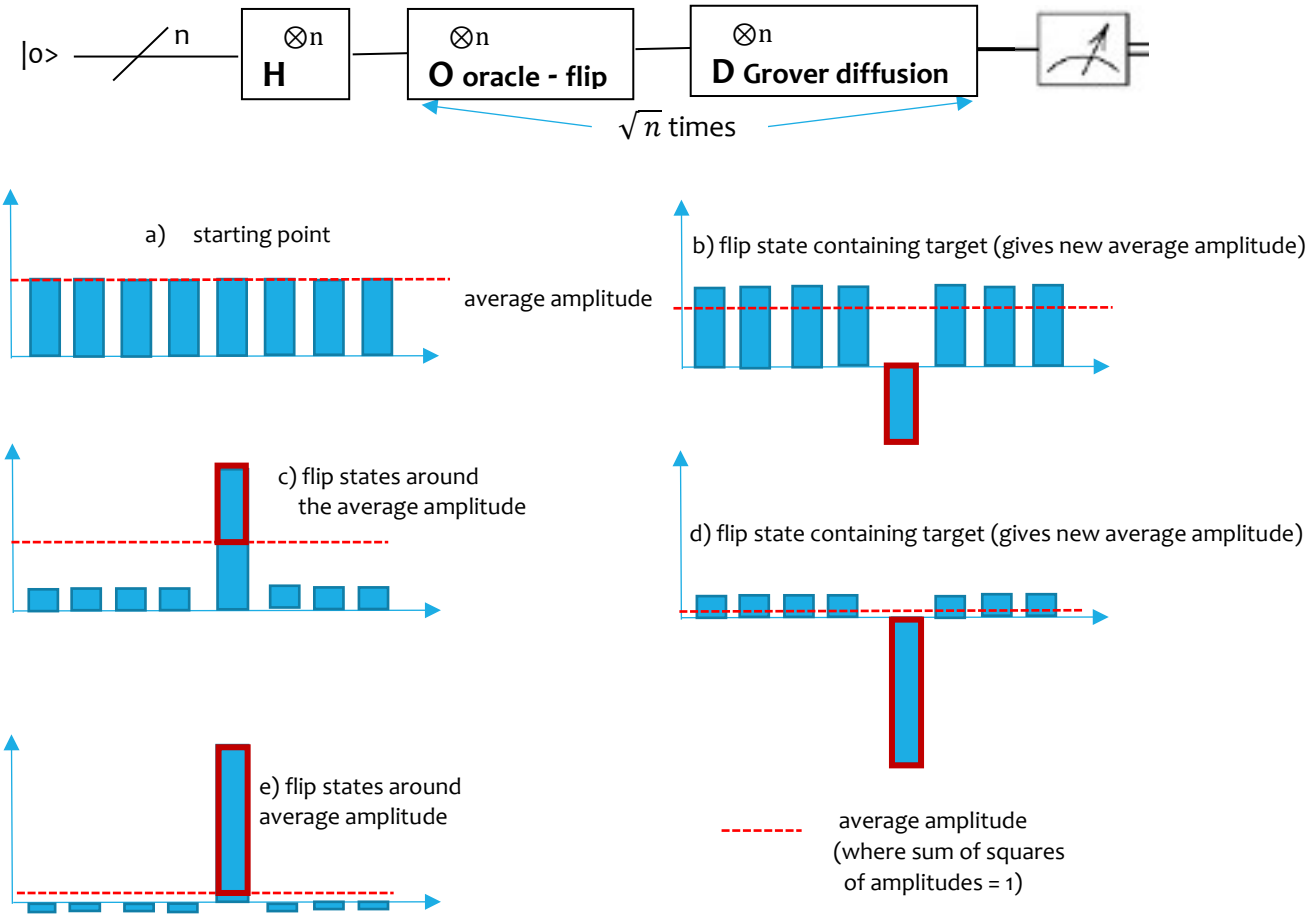


Fig. 24 Grover's algorithm visualisation

We start in a) with 8 equally likely states, each with amplitude $\frac{1}{\sqrt{8}}$ since ($8 = 2^3$). The next operation assumes that we have some knowledge of the sought state, e.g. in the form of a query, and that a gate (called the Oracle) has been encoded for this query to operate on the whole system (all the system states). The Oracle flips the sign of the amplitude of the required state. Note that this operation reduces the average amplitude since two states now cancel out. Although the state has been detected internally, measuring would not

reveal it since amplitudes are squared in the probability calculations and the measured value would collapse to some random value since all states are equally likely as in the original superposition. The next transformation takes us to c) where each state has been flipped about the average amplitude. Our required solution is beginning to emerge. Another 2-step iteration can be carried out, again increasing the relative amplitude of the required state. After $\sqrt{N}$ iterations the required state can be read with a high probability of being correct. The amplitude amplification step can be thought of as a Fourier transform.

5.5 Outline implementation of PKI and implications of breaking it

The creation of a public-private key-pair to be used for PKI requires two large prime numbers, say p and q (which are kept private), to be multiplied together. Let's say p x q = N. Simplifying to give the general idea, N can be published as part of a public key and p and q kept secret as part of the corresponding private key, to be used in the process of encrypting data for transmission.

The scheme is known as RSA after its authors (1977) Rivest, Shamir and Adleman of MIT. It was independently invented by Clifford Cocks of UK GCHQ in 1973 but this was not publicly revealed until 1997. Showing more detail:

- Bob finds two large prime numbers, say p and q and calculates m = p x q
- Bob then finds two numbers e and d such that d x e = 1 modulo (p - 1)(q - 1).
  There are efficient algorithms for this, given p and q.
- The public key is m and e, the private key is d.
- Alice uses Bob's public key to transform her message x into cyphertext using
  $x \rightarrow x^e$ modulo m
- It can be shown that $(x^e \text{ modulo } m)^d$ modulo m = x modulo m, the original message which Bob can recover using his private key d.

Computing the prime factors of a very large number is a process that is highly complex and would take a classical computer many thousands of years. This complexity is what we rely on to make PKI effective. If a quantum computer could be used to carry out this factorisation the implications would be substantial:

1 If private keys can be computed from public keys, data encrypted with private keys can be read. Databases stored long-term become readable.
2 Anyone who can compute the private key corresponding to a public key can decrypt the sensitive data encrypted with the private key and/or claim ownership of it.
3 Since a private key is someone's key to ownership of digital currency on a blockchain, anyone who could obtain the private key could claim the money.
4 Data integrity could no longer be assured by signature. New versions of data could be rehashed and re-encrypted.
5 Secret keys could no longer be securely transmitted using PKI.

It is this last point (5) that gives the impetus for quantum key distribution, see section 3. Also, post-quantum cryptography standards are already being defined and evaluated. We now outline the progress that has been made towards prime factorisation on a quantum computer.

5.6 Shor's quantum algorithm for prime factorisation

Shor's algorithm for prime factorisation was formulated in 1994, archived in 1995 and published in 1997 (P. W. Shor 1997 "*Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum*

computer" SIAM Journal on Computing. 26(5) pp 1484 – 1509). This work has provided a major incentive to develop quantum computing, if only because others worldwide might be doing it and already breaking PKI.

Although Shor's algorithm contains a classical part, the core relies on application of the quantum Fourier Transform (FT). FTs convert data representing multiple combined waveforms, such as audio representing a piece of music, into its component frequencies, as shown in Figure 9. The graphs representing the data input to FTs show a superposition of all the component waveforms in time e.g. base frequencies and harmonics. The output of FT represents the frequencies present and their intensities/amplitudes.

A full description of Shor's algorithm is beyond the scope of this paper. Some intuition on the approach is as follows, using small numbers and simplifying the process. Like current PKI, Shor's algorithm uses modular arithmetic. For example, consider two coprime numbers 7 and 15:

$7 = 7 \pmod{15}$

$7^2 = 49 = 45 + 4 = 4 \pmod{15}$

$7^3 = 7^2 \times 7 = 4 \pmod{15} \times 7 = 28 = 15 + 13 = 13 \pmod{15}$

$7^4 = 7^3 \times 7 = 13 \pmod{15} \times 7 = 91 = 1 \pmod{15}$

$7^5 = 7^4 \times 7 = 7$ and so on

The period of an integer a modulo N is the smallest positive integer r such that $a^r = 1 \pmod{N}$, so the period of 7 modulo 15 is 4. Suppose we have a period-finding machine for two co-prime integers N and a. Suppose we are given an N with only two distinct prime factors which we want to find (to break current cryptography). An algorithm to solve this goes through a number of random choices for a between 2 and

N-1. Let r be the period of a modulo N. Repeat the above steps until an even r is found. At this point we have found some pair r, a such that r is even and r is the smallest integer such that $a^r - 1$ is a multiple of N. Then

$a^r - 1 = (a^{\frac{r}{2}} + 1)(a^{\frac{r}{2}} - 1) = (7^2 + 1)(7^2 + 1) = (48)(50)$

The greatest common divider of $(15, 48) = 3$ and the gcd of $(15, 50) = 5$. These are the prime factors of 15.

5.6.1 Quantum implementation of Shor's algorithm

Prime factorisation has been the target of several attempts to engineer and use quantum computers. To date, only small numbers have been factorised by quantum computers. In 2001 IBM's implementation of a quantum computer with 7 qubits factored 15 = 5x3. In 2007 an implementation using 4 photonic qubits also factored 15, see Physical Review Letters 99(25). In 2012 a solid-state qubit quantum computer also achieved 15 = 5x3, see Nature Physics 8(10). In 2012 photonic qubits were used to achieve 21 = 7x3, see Nature Photonics 6(11). In 2012 the D-Wave computer factorised 143 = 11 x 13 (Physical Review letters 108 (13)) and in 2014 it achieved 56153 = 233 x 241. The technology used, adiabatic quantum computation, does not use gates and is designed for optimisation problems and not Shor's algorithm.

6 Fabrication technology for qubits and quantum computers

Ref National Academies of Sciences, Engineering and Medicine 2018. *Quantum Computing: progress and prospects*. Washington DC: National Academies Press: https://doi.org/10.17226/25196

There are two main approaches to the quantum computing process which can be thought of as analogue and digital. Although the state is still represented as qubits, in the analogue model the state is initialised then caused to evolve to the required result with high probability. This is achieved by changing its energy

environment (called its Hamiltonian), smoothly and slowly. The quantum operations are therefore analogue in nature. This approach includes adiabatic quantum computing (AQC), quantum annealing (QA) and direct quantum simulation.

Digital or gate-based quantum computing is similar in approach to classical computing, breaking down computations into small units operating on small numbers of bits. We have seen a selection of quantum gates in Section 4 and their use in some algorithms in Section 5.

The fabrication technology of qubits must allow for their interaction. As we have seen, the qubits must become entangled to yield quantum speedup and quantum gates must effect changes. Neutrinos would be an unsuitable choice because they do not interact, even though this property would protect them from unwanted effects. Possible implementations of qubits include:

- Photon, using the polarisation of light as horizontal or vertical, see Section 3.
- Electron, using spin as up or down.
- Nucleus, using spin as up or down
- D-Wave's qubits – "little magnets"? "superconducting flux"

In a classical system, a bit can be the state of a transistor in a processor, the presence or absence of a current in a cable or the magnetisation of a surface on a hard disc. In a quantum system, any two-level quantum mechanical system can in theory be used as a qubit. Current practice is to use trapped ion (silicon) qubits or superconducting qubits.
Most silicon-based qubits have been made from the electron or the nucleus of a single phosphorus atom to create a single qubit inside a layer of silicon. Qubits in silicon systems interact through electric fields. Both silicon and superconducting quantum systems only work in temperatures close to absolute zero.

6.1 An atom in a laser field

The light field from a laser interacts with an atom. The effect of the light is to cause transitions between different energy levels in the atom. These transitions will normally occur only if the frequency of the light is tuned to match the energy gap between the levels, so that the light is *resonant* with the transitions.

Measuring the quantum state of an atom or ion is in principle relatively straightforward, since it is possible to make transitions in states separated in energy by quite large energies (several eV), and these transitions can be detected, for example by detecting photons emitted by fluorescence. Preparing atoms in initial states is also straightforward is principle since the energy gaps involved may be large enough to allow direct cooling to the ground state at reasonable temperatures.

6.2 Spins in magnetic fields for atoms or electrons

Spins in magnetic fields provide one of the simplest and most natural physical systems for implementing qubits. Experimental spin physics is one of the simplest examples of coherent quantum control. The treatment is essentially that of two-level atoms in laser fields. The transitions between nuclear spin energy levels occur at very much lower frequencies than atoms in a laser field. It is also possible to use electron spins as qubits, with similar techniques to those for atoms. Electron transitions usually occur at much higher frequencies than nuclear transitions since electrons have a much larger magnetic moment, but the frequencies are very low compared with direct atomic transitions.

This approach is known as Nuclear Magnetic resonance (NMR). Radio-frequencies (RF) are used, which makes radiation easier to control than with lasers and coherence easier to achieve. The disadvantages are

severe, in that RF wavelengths are too large to allow spatially selective excitation and it is virtually impossible to detect single photons. Measurement and initialisation are also extremely challenging.

6.2.1 Quantum dots

In 1998, collaborators with IBM at the University of Basel, Switzerland devised quantum dot technology, based on electron spin, for qubit fabrication, see Daniel Loss (Univ Basel) and David DiVincenzo (IBM) *Quantum computation with quantum dots*, Phys Rev A 57, 129 – January 1998. Quantum dots are currently being investigated at the Cavendish Laboratory, Cambridge and at the Tyndall Institute, Cork, Ireland.

6.3 Photon techniques

The approach used for photons differs greatly from those for atoms and spins. The qubit is not mapped onto two distinct energy levels but onto two polarisation degrees of freedom. We saw the approach in use in Section 3.

7 Quantum computers under development

A related paper ***<insert URL>*** summarises what we have found on current quantum computing projects. Much is made of achieving "*quantum supremacy*", that is, demonstrating the execution of an algorithm beyond the capabilities of modern classical computers, but only very small steps have been made to date. Potential application areas claimed for the various projects are summarised in the paper.

It is often unclear which specific technology is being deployed and how many qubits are being used for computation and how many for error correction. Projects by Microsoft, IBM, Google and others are summarised. D-Wave uses a process called quantum annealing instead of quantum gates although, like most quantum computing technologies, it needs to be kept close to absolute zero to minimise decoherence. There is controversy about whether DWave is a true quantum computer and whether it can ever achieve quantum supremacy. However, DWave is already commercialised.

8 Challenges

8.1 Noise and precision

It is relatively easy to remove noise from classical digital systems, and therefore prevent or correct errors, since it is known that a value must be either 0 or 1. In both classical and quantum computers, gate operations contribute a small amount of noise. This can be removed in classical computers (as we know the input and outputs should be 0 or 1) but not in quantum computers. The number of iterations needed in a quantum computing algorithm may therefore be low to minimize the accumulation of errors. The fabrication technology for qubits has to be chosen so that they can interact, becoming entangled, rather than being isolated. This property also implies unwanted interaction with the external environment over time, known as decoherence. For this reason, qubits have to be kept in highly controlled conditions, such as by super-cooling.

In Section 4.4 we outlined the problems of unwanted coupling between qubits and their environment that causes errors. We considered briefly how to use replicated qubits, e.g. 3, 5 or 7 qubits to encode each qubit, to introduce redundancy, and ancillary bits that can be measured to detect errors. Quantum error correction (QEC) is a hard problem and requires of the order of 11 qubits to keep one qubit correct.

In order to monitor progress towards quantum computers a short-term class has been defined as NISQ computers – Noisy, Intermediate-Scale, Quantum Computers. Long term, an error-corrected quantum

computer is necessary to solve many problems. Algorithms suitable for NISQCs are required for the short to medium term.

8.2 Cost

The most recent D-Wave computer costs $15 million. All require some form of environmental control such as super-cooling to minimize decoherence over time. In the conventional computing industry Moore's Law resulted in ever-increasing productivity. This generated huge amounts of wealth, some of which could be fed back into research and development. It seems likely that quantum computing will be funded publicly or by very few wealthy companies.

8.3 Algorithms

As mentioned in Section 8.1, a first generation of algorithms is needed that are fault-tolerant to run on NISQ computers. That is, results given to some probability should be tolerable, rather than requiring an exact answer to some specified precision.

As we saw in Section 4, computer programs are circuits. At present, there is no sign of a move towards a universal quantum computer, capable of executing a variety of stored programs. The universality of classical computers comes from the fact that programs as well as data are stored in binary, see Figures 1 and 2.
It seems that a quantum computer might only ever be a special-purpose add-on, similar to current graphical accelerator hardware. Classical computers would be needed for software development and all their current diverse uses.

8.4 Performance.

Performance was the prime motivator for quantum computers. Current encryption systems rely on problems that are too hard to solve to any reasonable timescale on classical computers, due to the complexity of the solutions. This has been the main drive towards developing quantum computing.

In parallel with quantum computing development, we are already seeing **post-quantum cryptography.** In Section 3 we saw quantum key distribution (QKD) that uses quantum effects to distribute secret encryption keys, thus bypassing the need to use PKI for key distribution. In the next few years new encryption standards will be developed that are not vulnerable to quantum algorithms. The timescale of the development of the quantum algorithms themselves to useful scale is likely to be much longer.
It seems therefore that we can start producing encrypted data that is not vulnerable to quantum computing algorithms. Also, as time passes we can selectively re-encrypt stored data that was encrypted by the potentially vulnerable methods, although there are always likely to be undetected copies. In the likely development time of effective quantum computers we could have moved away from vulnerability to their algorithms.

Although the performance of classical computers may have reached a ceiling, we have data centres with millions of cores on which parallelisable algorithms can be run. For example, some blockchain systems use proof-of-work in which a random number is included in a block that is being hashed in order to link it into the chain. A competition is run in which the random number is changed by competitors (miners) to make the hash of the block start with some number of zeroes. The miner that is first to achieve this, has their version of the block included next in the chain and they receive payment.

Although proof-of-work uses an obscene amount of energy unproductively, it is part of many prominent blockchain systems. Originally it was said to be democratic, in that any computer could perform this mining

function. In practice, the test is parallelisable and carried out by consortia who can use large data centres. It is not likely that a quantum computer could ever be set up to solve this problem faster than conventional computers. Hopefully, a better more planet-aware method will soon succeed proof-of-work.

Providing refrigerated quantum computers might become the norm in future data centres, as part of a cloud computing offering. Although each requires a huge amount of energy, if quantum supremacy is ever achieved, one quantum computer will be able to carry out computations infeasible for large numbers of conventional computers.

9 Conclusions

In spite of the challenges noted in Section 8, it seems likely that quantum computing will continue to be funded by governments and rich companies, if only as a safety net for the eventuality of foes or competitors succeeding. If quantum computing succeeds at useful scale with appropriate precision and manageable decoherence, certain currently intractable problems (indeed, their intractability is their essential property) will become solvable such as prime factorisation. However, these eventualities are being foreseen and corrective actions taken, for example in quantum key distribution and post-quantum cryptography. Quantum communication is becoming established in practice.

Acknowledgement This paper is an attempt by Jean Bacon to explain to herself and others with very little physics and maths, a presentation "QC for QCs" given by Jon Crowcroft to our MCCRC project : www.cl.cam.ac.uk/~jac22/talks/qc-for-qc.pdf.
Thanks to Jon for suggestions and to Richard Jozsa for corrections and clarifications. Any remaining errors and misconceptions are entirely those of Jean.
This paper is at:
www.cam.ac.uk/~jmb25/QC-introTech-withMathsandGates-2019-08-12.docx
www.cam.ac.uk/~jmb25/QC-introTech-withMathsandGates-2019-08-12.pdf.
The version numbers (dates) in the titles may change.

A companion paper on quantum computing projects and proposed application areas is referenced where appropriate in this paper and is available at ***<insert URL>***.

Appendix: Some definitions from linear algebra

Norm of a vector

The norm (length) of a vector $|a\rangle = \sqrt{\langle a|a\rangle}$

Inner product of two vectors

The inner product of two vectors $|a\rangle$ and $|b\rangle = \langle a|b\rangle$

$|a\rangle = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}$, $\langle a| = (\alpha_1^*, \alpha_2^*)$ and $|b\rangle = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}$, $\langle b| = (\beta_1^*, \beta_2^*)$

$\langle a|a\rangle = (\alpha_1^*, \alpha_2^*) \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = |\alpha_1|^2 + |\alpha_2|^2$ and $\langle a|b\rangle = (\alpha_1^*, \alpha_2^*) \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = (\alpha_1^* \beta_1 + \alpha_2^* \beta_2)$.

Commutative matrices

Two square matrices A, B commute if $AB = BA$

Inverse of a matrix

The inverse A^{-1} of a square matrix A is such that $AA^{-1}=I$

Transpose of a matrix

The transpose of a matrix A is obtained by exchanging rows and columns, denoted A^T

Conjugate transpose of a matrix

The conjugate transpose of a complex matrix A is obtained by taking the complex conjugate of each element and exchanging rows and columns, denoted $A^\dagger$ (sometimes A^*).

Note that if A is real then $A^\dagger = A^T$

Normal matrix

A complex square matrix A is normal if it commutes with its conjugate transpose, $AA^\dagger = A^\dagger A$

An eigenvector v of a linear transformation A is a non-zero vector that changes by only a scalar factor when that linear transformation is applied to it, i.e. Av = cv, where (scalar) c is the **eigenvalue** associated with the eigenvector v.

Hermitian matrix

A complex matrix A is Hermitian if $A^\dagger = A$

If A is real, $A^\dagger = A^T$. Note that $A^T = A$ is just the definition of a symmetric real matrix.

Hermitian matrices have real eigenvalues and their eigenvectors are orthogonal for different eigenvalues, so they form a basis for the whole space.

Unitary matrix

A complex matrix U is unitary if $UU^\dagger = U^\dagger U = I$, i.e. $U^\dagger = U^{-1}$ and $U^\dagger$ is also unitary.

The importance of unitary matrices in quantum mechanics is that they preserve norms, and thus probability amplitudes. U is a normal matrix with eigenvalues lying on the unit circle.

Unitary matrices are Hermitian and are those matrices with a complete set of orthonormal eigenvectors such that the corresponding eigenvalues are ±1.

The rows (and columns) of a Unitary matrix form a unitary basis, that is, each row (or column) is of length 1.

Given two complex vectors a and b, multiplication by U preserves their inner product:

$\langle Ua|Ub\rangle = \langle a|U^\dagger U|b\rangle = \langle a|I|b\rangle = \langle a|b\rangle$