Quantum Information and Computation, Vol. 2, No. 2 (2002) 97–116

SUPERPOSITION, ENTANGLEMENT AND QUANTUM COMPUTATION

T.M. FORCER
Department of Electronics and Computer Science, University of Southampton
Southampton,SO17 1BJ,UK

A.J.G. HEY
Department of Electronics and Computer Science, University of Southampton
Southampton,SO17 1BJ,UK

D.A. ROSS
Department of Physics and Astronomy, University of Southampton
Southampton,SO17 1BJ,UK

P.G.R. SMITH
Department of Electronics and Computer Science, University of Southampton
Southampton,SO17 1BJ,UK

Received January 17, 2001
Revised November 26, 2001

The paper examines the roles played by superposition and entanglement in quantum computing. The analysis is illustrated by discussion of a 'classical' electronic implementation of Grover's quantum search algorithm. It is shown explicitly that the absence of multi-particle entanglement leads to exponentially rising resources for implementing such quantum algorithms.

Keywords: superposition, entanglement, quantum computing, Grover's search
Communicated by: S Braunstein & G Milburn

1. Introduction

In 1982 Feynman observed that N-particle quantum systems cannot be simulated on a classical computer whose resources do not grow exponentially with N, the number of particles [1]. He went on to speculate that such a simulation would be possible with a new kind of computer 'a quantum computer', which he claimed was not a Turing machine, but a machine of 'a different kind'. Both these insights have subsequently been shown to be correct. Several people have demonstrated that a quantum computer can be used to efficiently simulate a quantum system, in the sense that the required scaling of resources for the simulation of an N-particle quantum system grows only polynomially with N [2, 3, 4]. The second suggestion of Feynman (and of Benioff [5]) that quantum computers were distinct from Turing machines was proved by Deutsch in 1985 [6].

Quantum computers can exist in non-classical quantum superpositions of states and each state can simultaneously follow a distinct computational path. Such 'quantum parallelism' does not immediately lead to any dramatic computational speed-up compared to computation

with a classical computer since quantum measurement on the final state selects only one of these paths. It was not until a decade later that Shor showed how a small amount of information about the totality of these final quantum states could usefully be extracted [7]. Shor was able to construct a quantum algorithm that can factorize large integers in polynomial time. Since this problem is believed to be intractable on any classical computer, and the difficulty of factorising large integers is the basis of several popular cryptographic systems, Shor's result has stimulated great interest in quantum computing. In 1997, Grover published a quantum algorithm that can speed-up the search of an unsorted database containing N items [8]. The time to find one item that satisfies a given condition on a classical computer is $\mathcal{O}(N)$ steps: Grover's algorithm for a quantum computer can accomplish this in $\mathcal{O}(\sqrt{N})$ steps.

The power of quantum computers is thus remarkable. Of course, it may be that the difficulty of protecting delicate quantum states from the effects of decoherence will mean that practical quantum computing will be unfeasible. Nevertheless, quantum error correction schemes have shown that error correction for quantum computers is possible in principle [9, 10]. However, such schemes increase the size of quantum systems required to solve a given problem considerably. For example, Preskill estimates that, without error correction, factorising a 130 digit number (432 bits) would require a quantum computer with $\mathcal{O}(10^3)$ qubits and $\mathcal{O}(10^9)$ quantum gates [11]. Using a concatenated quantum error correction scheme to reduce errors down to an acceptable level would require an $\mathcal{O}(10^6)$ qubit system. Such systems are many orders of magnitude away from present-day quantum computing systems. Nonetheless, it is still of interest for us understand better precisely which feature of quantum mechanics is responsible for the power of quantum algorithms.

2. Superposition and Quantum Entanglement

We have seen that the ability of quantum systems to exist in coherent superpositions of eigenstates has been identified as a key feature of quantum algorithms. However, superposition of eigensolutions is a common property of many classical wave equations and is not, of itself, a specifically quantum feature. A discussion of this point by Ekert and Jozsa is particularly helpful concerning this point [12]. These authors consider a multi-qubit system and apply a fixed two qubit quantum gate U to any pair (i, j) of qubits sequentially. After n steps, qubit 1 is measured and 0 or 1 obtained according to a probability distribution $P_n = \{p_n(0), p_n(1)\}$. Implementing this process on an actual physical quantum system allows us to obtain the result in time $\mathcal{O}(n)$. They then consider how this process could be mimicked by classical means to sample the same probability distribution P_n. They show that a classical simulation of this process requires the storage of $\mathcal{O}(2^n)$ coefficients for its description. They conclude that 'a classical simulation will slow down exponentially in time under the weight of this exponentially growing information that needs to be processed in each step.' Thus a classical simulation requires $\mathcal{O}(2^n)$ time steps compared to $\mathcal{O}(n)$ time steps required by the quantum simulation.

Can one mimic the quantum process using classical waves? Classical waves allow superposition of modes and each qubit could be represented by the lowest two modes of a vibrating elastic string with fixed end points. Ekert and Jozsa then make the following point. No matter how much the strings interact with each other, in their subsequent vibrational evolution

each combined state may always be written as a product state of separate classical vibrations. The total space available to the classical system is the so-called Cartesian product of the individual state spaces of the classical sub-systems. Contrast this situation with the state space of a quantum many particle system where the total Hilbert space is a tensor product of the individual Hilbert sub-spaces. A simple example will clarify the point at issue. Consider a two qubit register with the qubits implemented as electron spins. Each of the spin $\frac{1}{2}$ particles has two eigenstates which can be denote as $\uparrow$ ('spin up') and $\downarrow$ ('spin down'). The four possible states of the two qubit system may be written as

$$|J\rangle, \quad (J \; = \; 0 \cdots 3) \tag{1}$$

A general state of the two qubit register is the superposition

$$|\Psi\rangle \; = \; \sum_{J=0}^{3} a_J |J\rangle \tag{2}$$

Mathematically, this state is a vector in the tensor product space $\mathcal{C}_2 \otimes \mathcal{C}_2$. However, only some of the states in this space may be written as the product of states for each individual qubit. For example, the state

$$\frac{1}{2}\left(|0\rangle \; + \; |1\rangle + \; |2\rangle + \; |3\rangle\right) \tag{3}$$

may be factorized into the product state

$$\frac{1}{\sqrt{2}}\left(|\downarrow\rangle + |\uparrow\rangle\right) \otimes \left(|\downarrow\rangle + |\uparrow\rangle\right) \tag{4}$$

whereas the state

$$\frac{1}{\sqrt{2}}\left(|0\rangle \; + \; |1\rangle\right) \tag{5}$$

cannot be factorised. Similarly, the state

$$\frac{1}{2}\left(|0\rangle \; + \; |1\rangle - \; |2\rangle + \; |3\rangle\right) \tag{6}$$

with one phase reversed is non-factorizable into product states. These non-factorizable quantum states are said to be 'entangled'. In general, the superposition

$$a_0|0\rangle \; + \; a_1|1\rangle \; + \; a_2|2\rangle \; + \; a_3|3\rangle \tag{7}$$

is an entangled state unless

$$det \left| \begin{array}{cc} a_0 & a_1 \\ a_2 & a_3 \end{array} \right| \; = \; 0 \tag{8}$$

The concept of entanglement extends easily to L qubits: a state is entangled unless it can be written as a product of states for each of the L qubits. Regarding the state as a vector in the tensor product space $\mathcal{C}_2 \otimes \mathcal{C}_2 \otimes \cdots \otimes \mathcal{C}_2$, a state is said to be entangled if it cannot be expressed as a single product of vectors in each $\mathcal{C}_2$ space, but only as a linear superposition

of such products. This distinction between Cartesian and tensor product states is precisely the phenomenon of quantum entanglement.

The term 'entanglement' is not new: it was first used by Schroedinger in 1935 [13]. In this paper, inspired by the challenge of Einstein, Podolsky and Rosen [14]. Schroedinger says 'that he would not call [quantum entanglement] *one* but rather *the* characteristic trait of quantum mechanics, the one that enforces its entire departure from classical lines of thought'. In the case of the EPR experiment, it is just these entangled states that introduce the spooky, non-local action-at-a-distance effects of quantum mechanics that Einstein so detested, but experiment apparently insists upon [15].

However, whether or not a given quantum state is deemed to be entangled depends on the basis used to describe the state. In the two qubit example considered above, it is natural to use the Hilbert space $\mathcal{C}_2 \otimes \mathcal{C}_2$ with each $\mathcal{C}_2$ being the Hilbert space for each qubit. In this basis the most general state is indeed 'entangled'. On the other hand, one is equally at liberty to consider the Hilbert space as $\mathcal{C}_4$ (e.g. the states of a spin-$\frac{3}{2}$ particle). In this space the most general state is a simple vector (a general superposition of the four basis vectors). One can extend this to the description of L-qubits at the price of describing the states in a 2^L-dimensional Hilbert space. In their paper [12], Ekert and Jozsa point out that entanglement can be represented using classical states in the following manner. The state of n qubits is a 2^n-dimensional space and can be viewed as isomorphic to the state space of a single particle with 2^n levels. Using such a representation, certain states of this one-particle system may be regarded in some sense as 'entangled' via their correspondence with the states of the n-particle qubit system. However, they state that: 'physical implementation of this correspondence appears always to involve an exponential overhead in some physical resource so that the isomorphism is not a valid correspondence for considerations of complexity.' As an example, they suppose that the 2^n levels of the one-particle quantum system, or corresponding classical system, are equally spaced energy levels. A general state of n qubits requires at most an energy that grows linearly with n for each qubit to be excited. The general state of the 2^n-level quantum or classical system requires an amount of energy that grows exponentially with n. This is the trade-off that genuine multi-particle quantum entanglement delivers: linear versus exponential resources. To explore this correspondence in detail we shall describe the implementation of a classical version of Grover's 'quantum' algorithm for fast database searching.

3. Grover's Quantum Search Algorithm

Consider the problem of finding someone's name in a telephone directory knowing only their telephone number. If there are N entries in the directory, the fastest classical algorithm takes $\mathcal{O}(N)$ time steps. Grover showed that a quantum computer searching (a quantum version of) such a directory could successfully perform the search in $\mathcal{O}(\sqrt{N})$ steps [8]. The problem may be stated more formally as follows. Each of the $N = 2^L$ different states are labelled $S_0,\ S_1,\ S_2, \cdots S_{N-1}$. Only one state fulfils the search condition C_J so that $C_J(S_J) = 1$ and $C_J(S_K) = 0,\ K \neq J$. The goal is to identify the state S_J with the minimum number of evaluations of the condition C_J.

Grover's solution was as follows:

1. Start with an L qubit register in the state $|0\rangle$. Every possible state S_K is represented

by the state $|K\rangle$.

2. Apply an L qubit Hadamard transformation H to the register. This yields the equally weighted superposition

$$\frac{1}{\sqrt{2^L}} \sum_{K=0}^{2^L-1} |K\rangle \tag{9}$$

3. Repeat $\mathcal{O}(\sqrt{N})$ times, the following operation:

   (a) Apply the operator U_J defined by

$$U_J|J\rangle = -|J\rangle \tag{10}$$

$$U_J|K\rangle = |K\rangle, \ K \neq J \tag{11}$$

   This 'black box' or 'oracle' applies a π phase shift only to the required element of the database. In a quantum implementation, this involves a portion of the quantum system sensing the state and performing a π phase rotation on the selected state. It does not involve a quantum measurement, nor can we 'open' the black box to see directly the value of the parameter J.

   (b) Apply Grover's 'diffusion' operator, D, defined as

$$D \ = \ H\,U_0\,H \tag{12}$$

   where U_0 is given by

$$U_0|0\rangle \ = \ -|0\rangle \tag{13}$$

$$U_0|K\rangle = |K\rangle, \ K \neq 0 \tag{14}$$

4. After $\mathcal{O}(\sqrt{N})$ iterations, the outcome of a meaurement on the register is the state $|J\rangle$ with high probability.

Grover has given an explanation of the effects of his operator D in terms of an 'inversion about the average' of the coefficients of the quantum state.

Consider the example of a two qubit system corresponding to a 4-entry database. Using our previous notation we start in the state $|0\rangle$ and apply the Hadamard operator H

$$H|0\rangle \ = \ |\Psi\rangle \ = \ \frac{1}{2}\left(|0\rangle + |1\rangle + |2\rangle + |3\rangle\right) \tag{15}$$

Now apply the oracle U_J

$$U_J|\Psi\rangle \ = \ |\Psi\rangle - 2\,\langle J\ |\Psi\rangle|J\rangle \tag{16}$$

Applying Grover's diffusion operator D it can be shown that

$$DU_J H|0\rangle \;=\; -|J\rangle \tag{17}$$

i.e. this sequence of operators forces the initial state $|0\rangle$ into the required state $|J\rangle$ (up to an overall sign) in a single pass.

For our classical implementation of Grover's algorithm we shall reformulate the problem in terms of inverting a function. A quantum system can be used to map an integer between 0 and $2^L - 1$ to another integer in the same range as follows. We define L qubits to be the 'control' register $|J\rangle$ and another L qubits to be the 'target' register $|K\rangle$. We can then devise a function V_f that performs the mapping

$$|J\rangle \otimes |K\rangle \;\rightarrow\; |J\rangle \otimes |K \oplus f(J)\rangle \tag{18}$$

If we start with the state $|K\rangle = |0\rangle$ we generate the state $|f(J)\rangle$. We can now define a modified version of Grover's search problem as follows [16]. Consider a function $f(M)$ that maps an integer M to an integer $F = f(M)$. (For simplicity, we assume the mapping is one-to-one.) The objective is to force the initial state into the state $|M\rangle \otimes |F(M)\rangle$ so that measurement of the control register yields $f^{-1}(F) = M$. Define an operation W that is a Hadamard gate H_C acting on the control register only, followed by the operator V_f

$$W \;=\; V_f \, H_C \tag{19}$$

Acting with this operator on an initial state in which *both* registers are in the state $|0\rangle$, we find

$$W|0\rangle_C \otimes |0\rangle_T \;=\; \frac{1}{\sqrt{2^L}} \sum_{K=0}^{2^L-1} V_f|K\rangle_C \otimes |0\rangle_T \;=\; \frac{1}{2^L} \sum_{K=0}^{2^L-1} |K\rangle_C \otimes |f(K)\rangle_T \tag{20}$$

Since the map is one-to-one we can rewrite the last expression as

$$\frac{1}{2^L} \sum_{K=0}^{2^L-1} |K\rangle_C \otimes |F\rangle_T \tag{21}$$

Now let U_F be an oracle that flips the sign of the quantum state if and only if the target register is in the state $|F\rangle$. From the preceding discussion of Grover's algorithm it follows that

$$\left(W \, U_0 W^{-1} U_F\right)^{\sqrt{N}} W|0\rangle \otimes |0\rangle \;\sim\; |f^{-1}(F)\rangle \otimes |F\rangle \tag{22}$$

where $N = 2^L$.

4. An Electronic Implementation of a two qubit Hadamard Gate.

We consider first the implementation of an electronic analogue of a two qubit Hadamard gate. For a single qubit Hadamard we have

$$H|0\rangle \;=\; \frac{1}{\sqrt{2}} \left(|1\rangle \,+\, |0\rangle\right) \tag{23}$$

$$H|1\rangle \;=\; \frac{1}{\sqrt{2}}\left(-\,|1\rangle \,+\, |0\rangle\right) \tag{24}$$

For a two qubit state we have

$$\begin{aligned} H|0\rangle \otimes |0\rangle \;&=\; \frac{1}{\sqrt{2}}\left(|1\rangle \,+\, |0\rangle\right) \otimes \frac{1}{\sqrt{2}}\left(|1\rangle \,+\, |0\rangle\right) \\ &=\; \frac{1}{2}\left(|1\rangle \otimes |1\rangle \,+\, |1\rangle \otimes |0\rangle \,+\, |0\rangle \otimes |1\rangle \,+\, |0\rangle \otimes |0\rangle\right) \end{aligned} \tag{25}$$

or

$$H|0\rangle \;=\; \frac{1}{2}\left(|0\rangle \,+\, |1\rangle \,+\, |2\rangle \,+\, |3\rangle\right) \tag{26}$$

in our previous notation. In matrix notation, where the state vectors are written as the column vectors

$$|3\rangle \;=\; \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |2\rangle \;=\; \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |1\rangle \;=\; \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |0\rangle \;=\; \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

the two qubit Hadamard operator becomes

$$H \;=\; \frac{1}{2}\begin{pmatrix} 1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{27}$$

At this point we emphasize that the notation used above can now refer to either classical or quantum states since no reference is made to entanglement. Thus, in a classical implementation, one requires four signal paths, each assigned to one of the four states. Superposition is now effected simply by superposing signals on different signal paths.

It is easy to design an electronic implementation of an L qubit Hadamard device using op-amps with 2^L input and output signal paths. Our electronic implementation of a 2-qubit Hadamard gate uses operational amplifiers and resistor networks to implement a set of inverters and summers to achieve the required transformation of a vector of input voltages to the vector of output voltages as per the 2-qubit Hadamard operator defined above. In our circuit implementation bi-color light emitting diodes (LEDs) are used to indicate the signal state.

The Hadamard gate circuitry was realised as a simple plug-in circuit board. A number of these modules were produced and fitted to both the dual-gate demonstrator system of Figure 3 and the full Grover search system of Figure 10. LEDs were not fitted to the Hadamard gates themselves, only to the motherboards of the respective demonstrator systems. Simple switches on the motherboards allowed a stimulus of +1, 0 or -1 to be applied to any input signal path. This set of four switches with their associated LEDs is clearly visible at the left side of Figure 3. The set of sixteen switches on the full Grover search motherboard is identifiable at the left side of Figure 10.

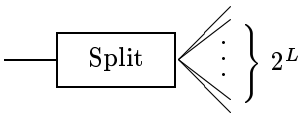
In our classical analogue, electrical signals take the place of probability amplitudes and we need only 0 and π phase shifts. After these technical preliminaries we can summarize the implementation of the electronic algorithm for an L qubit Hadamard gate as follows:

1. The general L qubit Hadamard operator may be written as a $2^L \times 2^L$ matrix of the form

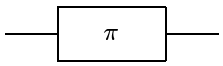
$$H_{ij} \;=\; \frac{1}{\sqrt{2^L}}\eta_{ij}, \quad i,j = 0 \cdots 2^L - 1 \tag{28}$$

   where $\eta_{ij} = \pm 1$

2. Each of the 2^L input signals is split into 2^L separate signals each with amplitude $1/\sqrt{2^L}$ of the input signal. This guarantees our probability amplitude in term of the signals and guarantees unitarity of the device.

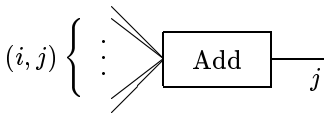


3. Label each of the 2^{2L} signals i, j to indicate which is the parent and which is the daughter. If $\eta_{ij} = -1$ pass the signal through a phase reversal device



   This phase reversal is carried out by an inverting amplifier.

4. For each j pass the signals from all parents, i, through an adder



The 2^L signal paths marked j now each contain the signal

$$H_{ji}|i\rangle$$

Figures 1 and 2 show the circuits for single qubit ($L = 1$) and two qubit ($L = 2$) devices respectively. Note that the Hadamard operator is idempotent (logically reversible) and a second application will regenerate the initial state, i.e.

$$H\,H|\Psi\rangle \;=\; |\Psi\rangle \tag{29}$$

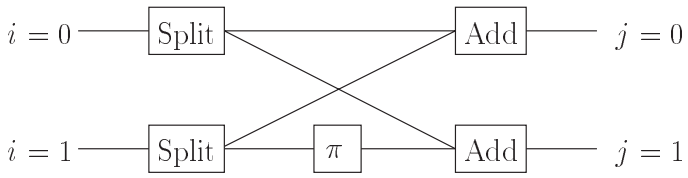


Fig. 1. Schematic diagram for the single qubit Hadamard gate.

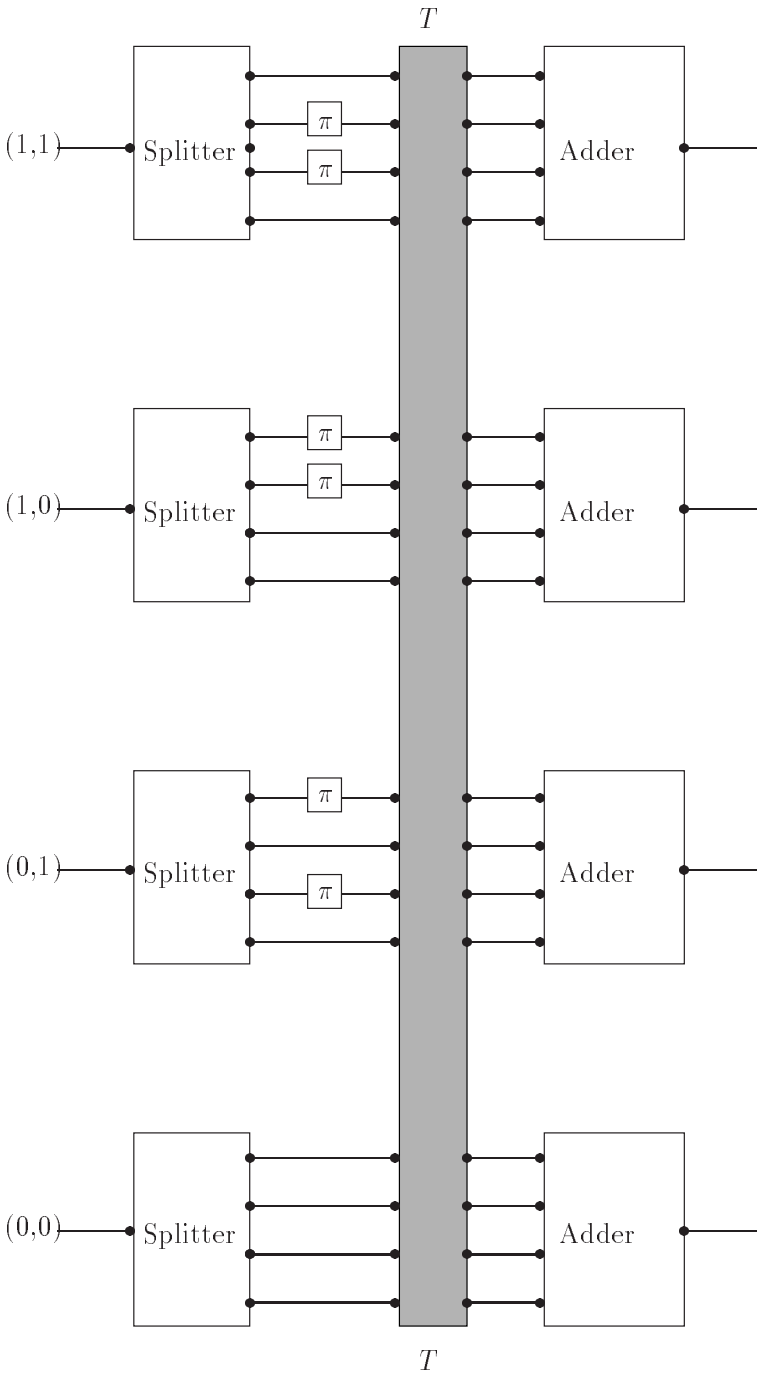


Fig. 2. Schematic diagram for the two qubit Hadamard gate. The device marked T is a transpose matrix junction box shown in Figure 6

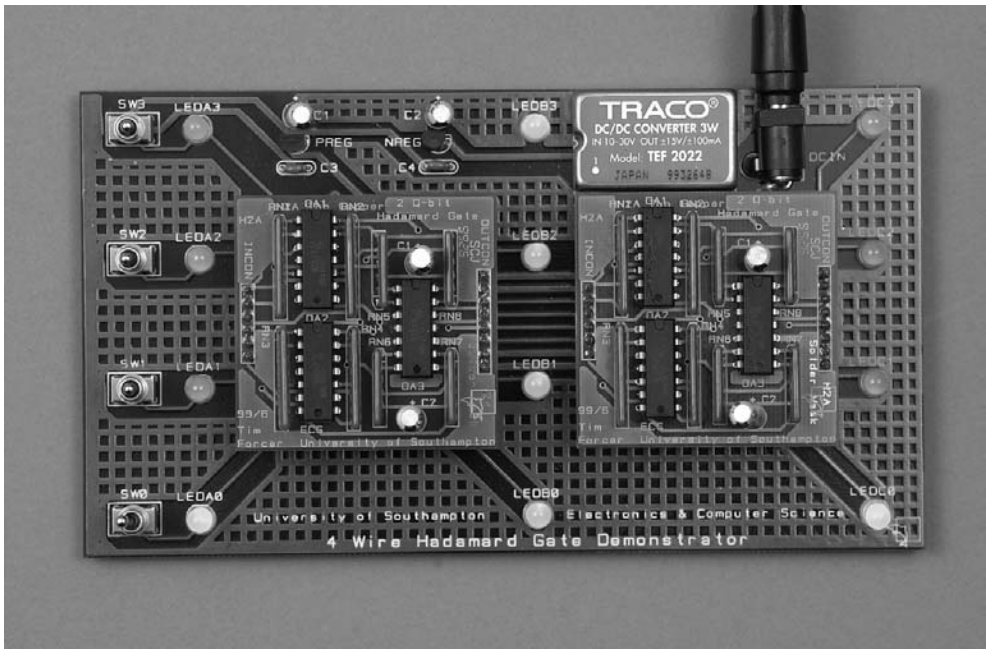


Fig. 3. Photograph of two qubit Hadamard gate

A photograph of an implementation of a two qubit version of this device is shown in Figure 3. We conclude this section with five comments:

1. The device acts like a 4-input reversible gate and there is clearly nothing specifically quantum mechanical about such a device. As can be seen on our Hadamard demonstrator board in Figure 3, the gate module is 'reversible' in the sense defined by Feynman, Bennett and Fredkin, namely 'from what comes out we can deduce what went in' (see p35 of the 'Feynman Lectures on Computation' [17]). The twin-gate system of Figure 3 demonstrates the idempotent property of such reversible gates - that a repeated application of the transformation through a second reversible gate of the same type restores the original input vector - since the pattern displayed on the right-hand set of four LEDs (the output vector of the second gate) is always identical to that on the left-hand set (input vector of the first gate). Note that this is true for all possible values and phases of the input vector (although the implementation shown here is limited to 81 possible combinations). This behaviour illustrates a further relevant property of this 2-qubit electronic Hadamard gate, namely that linear superposition applies - the output vector for an input consisting of superposed vectors A and B is the linear sum of the output vectors produced by A and B in isolation. This should be expected from inspection of the 2-qubit Hadamard operator above but might not be immediately obvious to those more familiar with conventional digital gates.

2. Although the device described here represents a unitary idempotent matrix, it is *not*

reversible in the usual sense in that it cannot be run backwards with the same effect because the splitter and adder are not bidirectional components as realized here.

3. The use of linear elements (operational amplifiers and resistors) ensures that the module would operate correctly with an AC signal allowing multiple stimulus patterns of arbitrary frequency, amplitude and phase to be superposed and handled correctly.

4. To encode an L qubit register we need to use 2^L signal paths. However, between the splitter and the adder we require 2^{2L} signal paths. We may think of these extra signal paths as analogous to the extra universes of Deutsch's multi-universe interpretation of quantum mechanics [18] since they provide all the histories between given input and output states. In a genuine quantum device we cannot inspect these intermediate states without affecting the output state, but in our classical implementation it is possible to do so.

5. It should be remembered that these electronic emulations of quantum gates and quantum algorithms are intended as demonstration systems. They therefore incorporate features and characteristics intended specifically to assist the function of demonstration. An obvious example is the use of LEDs to monitor intermediate state vectors of the algorithm. These would not be present in any 'real' system so that the energy-consuming LEDs would be absent. The easiest stimulus for a system in which something simple like an LED is to be used to indicate the state of a signal, is a DC voltage. This requires active (i.e. operational amplifier based) circuitry to achieve inversion or π-phase reversal. Using AC stimuli one could implement almost lossless passive inversion. The net result would be a system with negligible energy dissipation in the computation.

5. An Electronic Implementation of Grover's 'Quantum' Search Algorithm

We have constructed an electronic operational amplifier implementation of the modified version of Grover's quantum search algorithm for inverting a two bit function, as discussed in Section 3. The device simulates a quantum circuit with two qubit control and target registers. Since this is not implemented on quantum states, each 2-qubit system requires four signal paths to represent its four states explicitly. The combined system therefore requires 16 input and 16 output signals which we label (I, J) with $(I, J) = 0 \cdots 3$. The individual values of I and J represent the control register state (argument of the function) and the target register state (value of the function), respectively.

The sixteen signals are arranged in four quartets with each quartet labelled by I and J denoting the particular member of that quartet. For convenience, in describing the Hadamard gate acting on the control register we have introduced a transposition matrix T (shown in Figure 6) that re-arranges the signals so that they emerge labelled (J, I) and are grouped into quartets labelled by J. This transposition matrix is just a junction box and is only introduced to minimize the complexity of the schematic diagrams.

The demonstration system shown in Figure 10 implements the transposition matrices implicitly in the nodal interconnections to and from the columns of 2-qubit Hadamard gates. These columns of gates are clearly visible in the photograph (Figure 10) but the transposition

matrices connecting each set of four Hadamard gates to preceding and succeeding stages of the system are not directly visible, being copper tracks in the motherboard.

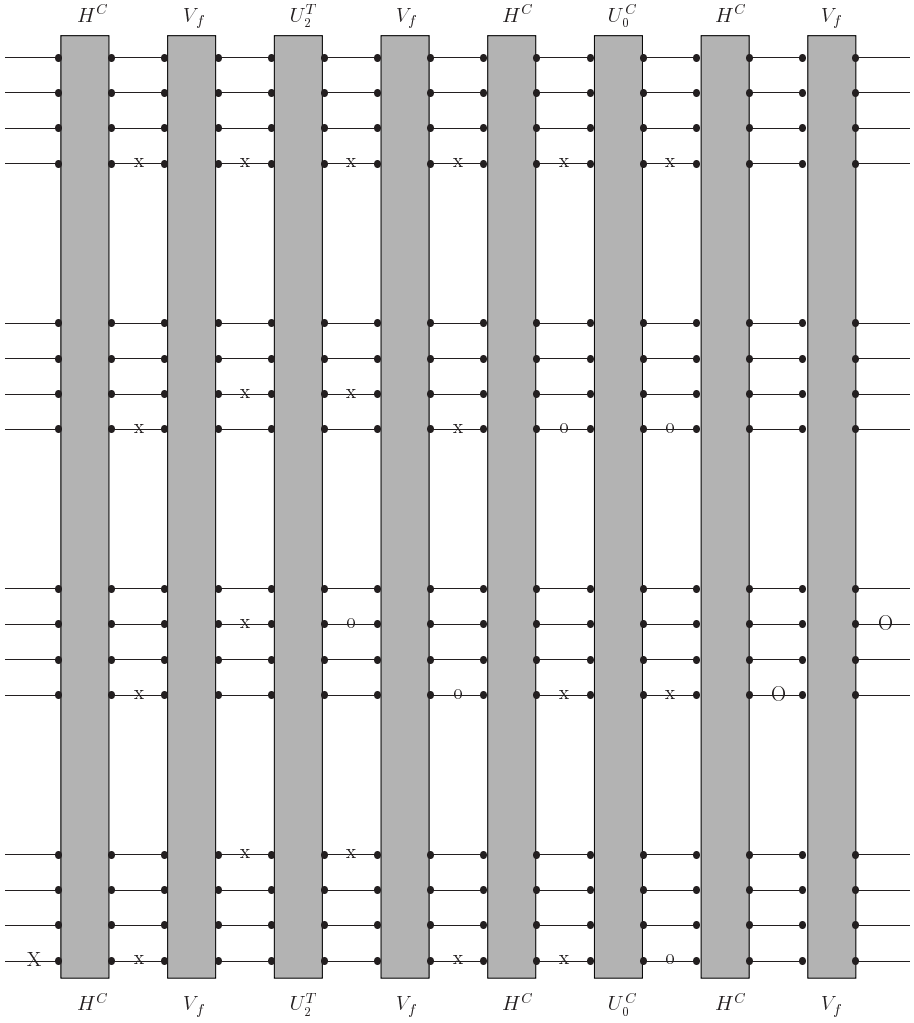


Fig. 4. The complete Grover search algorithm circuit, with $F = 2$, starting in the state $(0, 0)$

The example function $f(I)$ that we have chosen to model is

$$f(I) \;=\; 3 \;-\; I \tag{30}$$

although other one-to-one mappings could easily be implemented. Given the value $F = f(I)$ we search for $I = f^{-1}(F)$. The value of F may be manually selected from the range 0 to 3. The function is encoded by the module denoted by V_f in Figures 4 and 7. This device is implemented as a removeable plug-in, re-wireable, module that allows other functions to

be encoded and investigated. The implementation uses wire-wrap technology so that the implemented wiring visible in Figure 10 has a direct visual equivalence to Figure 7. This device translates its input vector, observable as a column of LEDs, to an output vector similarly visualized.

Figure 4 shows the full working of the algorithm, starting with a stimulus only on the (0,0) input and with F = 2. At subsequent stages, the signal paths with voltages are marked with a lower case 'x' for a positive voltage or 'o' for a negative (phase-reversed) voltage. Upper case 'X' and 'O' symbols are used when only a single signal path has a voltage present at that stage. The lower case letters are used when the signal is shared between four paths and has a voltage half of that associated with the upper case X or O. Note that upon exit, only output (1,2) has a (negative) voltage on it, indicating that $f^{-1}(2) = 1$. The first two stages of our device (reading from left to right) consists of the 'preparation' of the desired superposition state

$$\sum_I |I\rangle \otimes |f(I)\rangle \tag{31}$$

The remaining stages constitute the modified Grover diffusion operator. In this 2-qubit case, this consists of the oracle marking the desired state followed by one Grover iteration $W\,U_0 W$ and, in this special case, leads exactly to the exact solution. The individual components are detailed in Figures 5 - 9. A photograph of our electronic implementation clearly showing the different stages of Grover's algorithm is shown in Figure 10. Our implementation makes explicit use of our electronic version of 2-qubit Hadamard gates shown in Figure 2. The Oracle, U_2^T, of Figure 8 achieves phase inversion using an inverting operational amplifier circuit. As implemented for demonstration the device includes a switch that allows the signal on any one of the four signal paths to be inverted. This is how the value of F is selected. In the case of the Oracle U_0^C shown in Figure 9 the same implementation technique has been used as for U_2^T.

6. Conclusions

We have seen that it is perfectly possible to implement Grover's quantum search algorithm in terms of classical electronic circuitry as an analogue computer. Where then are the quantum aspects of the problem? Superposition is not peculiar to quantum systems and although measurement clearly is a quantum phenomenon, the probabilistic nature of the algorithm for larger numbers of qubits would show up as signals of differing intensity on the output.

An inspiration for our work was the optical interferometric implementation of Grover's algorithm by Kwiat et al. [19]. However, there is an important difference between this electronic version and Kwiat's optical implementation using optical paths and polarization as qubits. In an ideal classical implementation we can measure the signal amplitudes at any stage during the algorithm. However in the optical system measuring the light intensity to determine the path or polarization will terminate the algorithm. At very low input powers the optical implementation becomes truly quantum mechanical as there is only a single photon in the interferometer and it must interfere with itself. Interestingly, in the electronic system, at very low input signal strengths, any intermediate state measurement would introduce extra noise and would similarly invalidate the calculation.

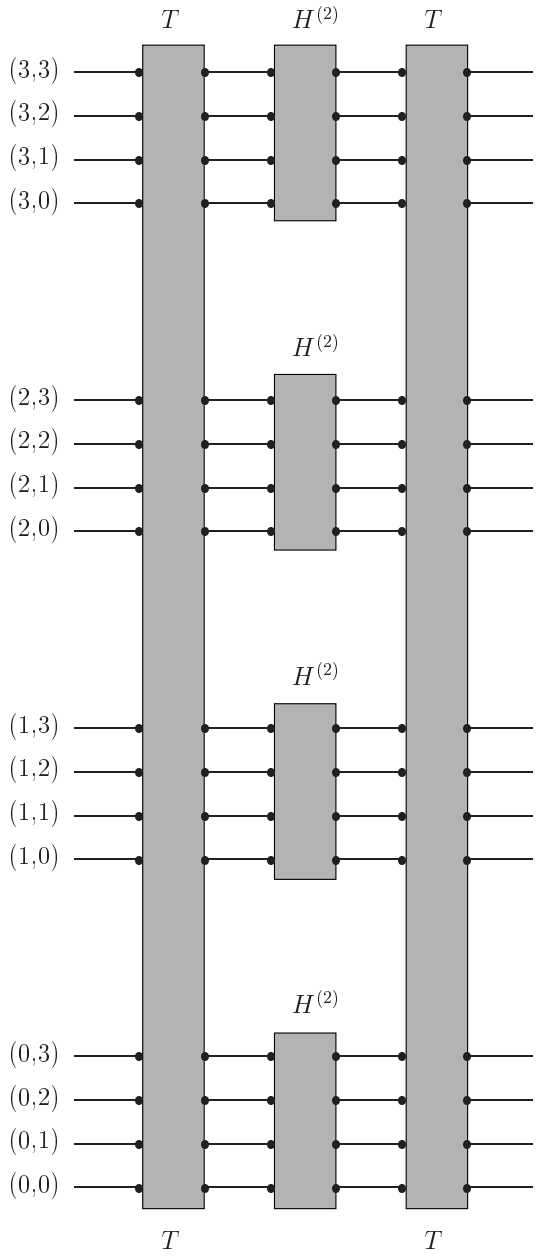


Fig. 5. The device H^C, which is a Hademard gate acting on the control register, consists of a transposition matrix, T (shown in figure 6), followed by four 2-qubit Hadamard gates, $H^{(2)}$, (shown in Figure 2) followed by another transposition matrix, T.

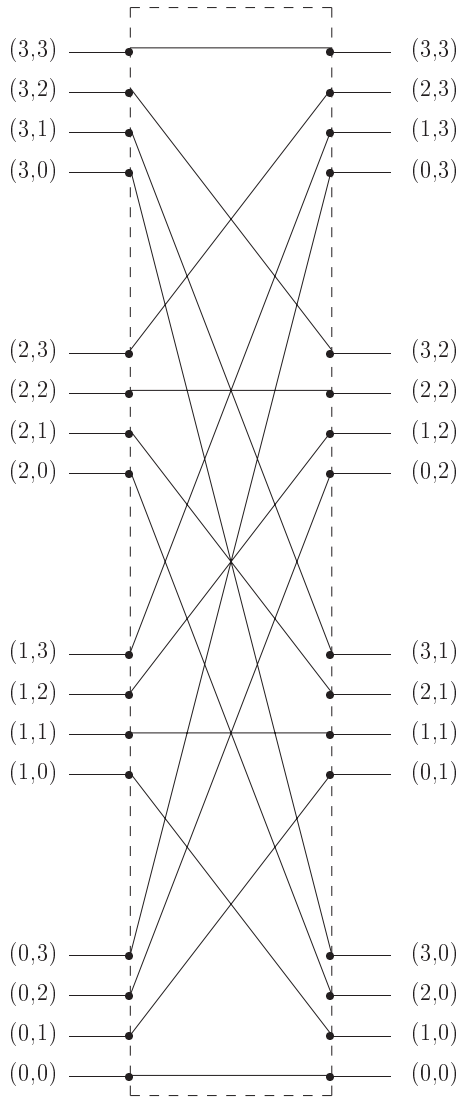


Fig. 6. The transposition matrix T. This is just a junction box.

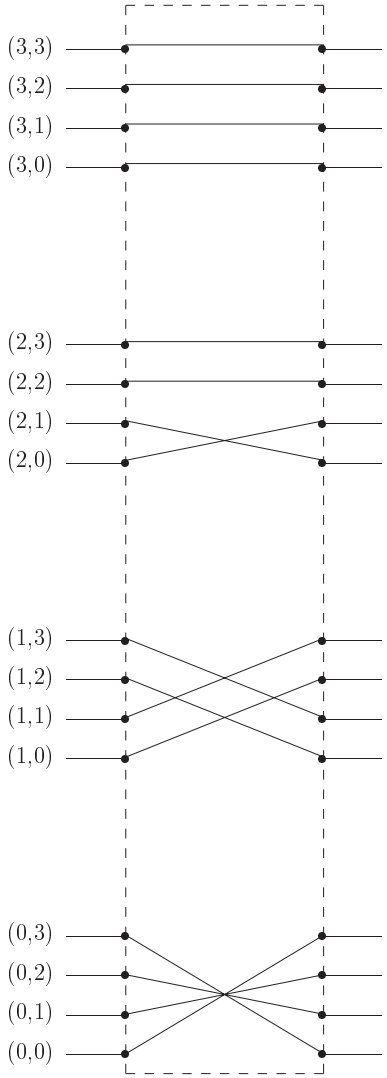


Fig. 7. The function encoding device V_f. This is also a junction box, which interchanges the wire $(I,0)$ with the wire $(I,f(I))$. It also interchanges the remaining two wires if $f(I) > 1$ (in this way it takes the state $|I\rangle \otimes |J\rangle \rightarrow |I\rangle \otimes |f(I) \oplus J\rangle$). In this example we have taken the function $f(I) = 3 - I$.

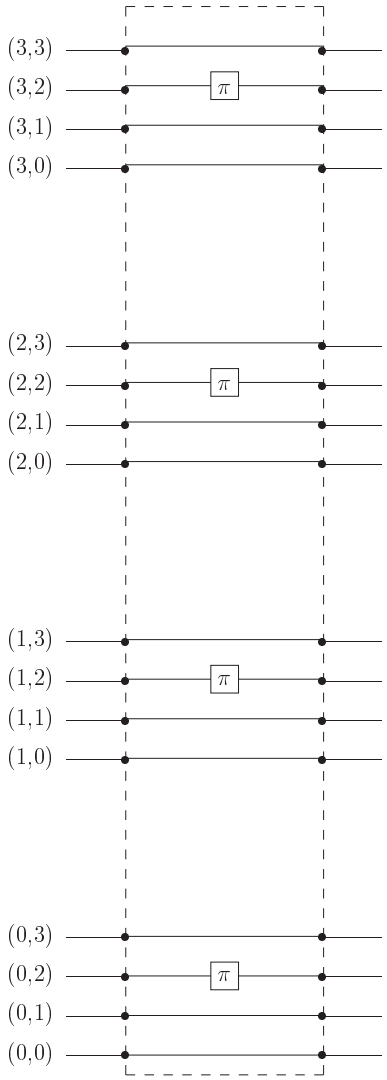


Fig. 8. The 'oracle' U_2^T which reverses the phase if (and only if) the target register is in the state $|2\rangle$

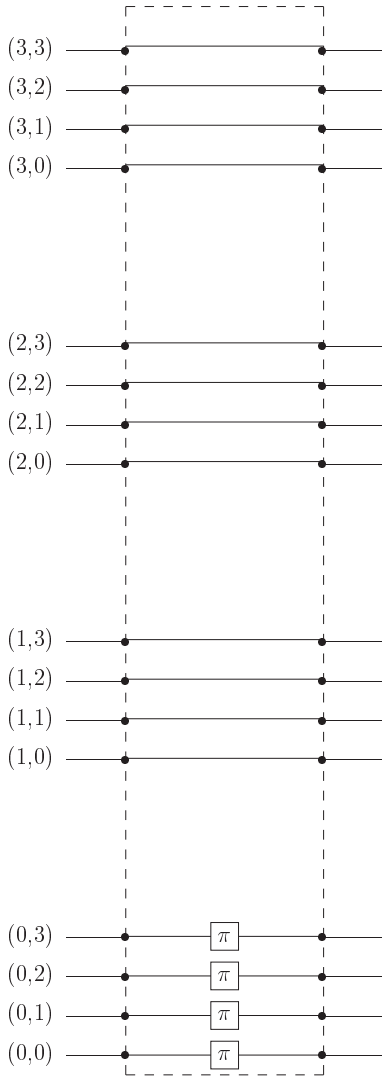


Fig. 9. The device U_0^C which reverses the phase if (and only if) the control register is in the state $|0\rangle$

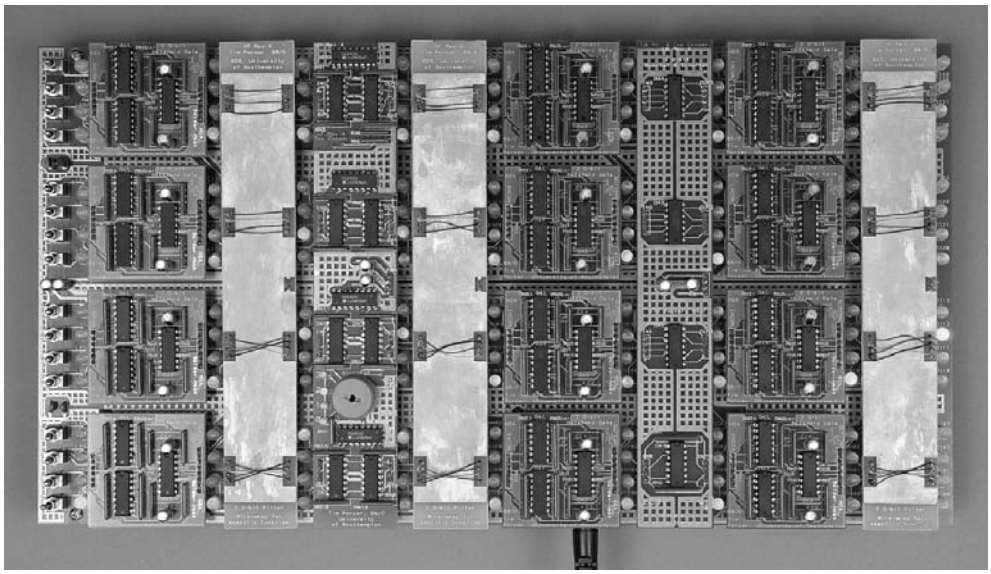


Fig. 10. Photograph of the two qubit Grover search device with two qubit control register

In the case of the device described in this paper, the only phase that is required is a π inversion. However, it is perfectly possible to construct initial states which are superpositions with arbitrary complex coefficients and build phase rotators to simulate the quantum operations in any quantum computational algorithm. An electronic device is under construction which is designed to carry out Shor's algorithm [7] for factorisation by using phase rotators to effect a fast Fourier transform on any given incoming state.

What of entanglement? As we have seen, entanglement is representation dependent. A two qubit system using two electron spins can exhibit entanglement: the same 2-qubit system implemented using the four states of a single spin $\frac{3}{2}$ particle or as four signal paths cannot show genuine entanglement. However, as our electronic implementations show, any implementation without multi-particle entanglement requires resources that grow exponentially with the number of qubits. This result is a concrete realisation of the discussion of Ekert and Jozsa [12]. In their example, the energy required rose exponentially with the number of qubits: in our case the number of signal paths increases exponentially and makes electronic implementations of large numbers of qubits impracticable. We conclude that multi-particle entanglement is the key property of quantum systems that gives rise to the remarkable power of quantum computers.

Acknowledgements

We thank Joanne Baggett and Jonathan Hey for their initial implementation of the operational amplifier simulation of Grover's algorithm and we also thank Graham Bowden, and other members of the Southampton Quantum Technology Centre for helpful discussions. We also thank Juri Papay for valuable help in the preparation of the manuscript.

References

1. R.P.Feynman (1982), *Int. J. of Theor. Phys.*, 21, 410/11/200067.
2. C.Zalka (1996), quant-ph/9603026.
3. S.Wiesner (1996), quant-ph/9603028.
4. D.S.Abrams and S.Lloyd (1997), *Phys. Rev. Lett.*, 79, pp.2586.
5. P.Benioff (1982), *J. Stat. Phys.*, 29, pp.515.
6. D.Deutsch (1985), *Proc. Roy. Soc. London*, A400, pp.97.
7. P.W.Shor (1994), *Proc. of 35th Annual Symposium on the Foundations of Computer Science*, edited by S.Goldwasser (IEEE, Los Alamitos, California), pp.124.
8. L.V.Grover (1997), *Phys. Rev. Lett.*, 79, pp.325.
9. P.W. Shor (1995), *Phys. Rev.*, A52 R2493.
10. A.Steane (1996), *Proc. Roy. Soc. London*, A452, pp.2551.
11. J.Preskill (1998), *Introduction to Quantum Computation and Information* edited by H-K Lo, S.Popescu and T.Spiller (World Scientific, Singapore), pp.213.
12. A.Ekert and R.Jozsa (1998), *Phil. Trans. Roy. Soc. London*, A356, pp.1769.
13. E.Schroedinger (1935), *Proc. Camb. Phil. Soc.*, 31, pp.555.
14. A.Einstein, B.Podolsky and N.Rosen (1935), *Phys. Rev.*, 47, pp.777.
15. A.Aspect, J.Dalibard and G.Roger (1982), *Phys. Rev. Lett.*, 49, pp.1804.
16. D.A.Ross (1998), quant-ph/9807078.
17. R.P.Feynman (1996), *The Feynman Lectures on Computation*, edited by Anthony J.G. Hey and Robin W. Allen, p35 (Addison-Wesley).
18. D.Deutsch (1998), *The Fabric of Reality*, Penguin books, ISBN 0140146903.
19. P.G.Kwiat, J.R.Michell, P.D.D.Schwindt and A.G.White (2000), *J. Mod. Opt.*, 47, pp.257.