# Quantum Mechanical Searching

**Lov K. Grover**
1D-435, Bell Labs, 600 Mountain Avenue,
Murray Hill NJ 07974,
lkgrover@bell-labs.com

**Abstract - It has recently been shown that the searching speed of computers based on quantum mechanics is far superior to their classical counterparts.**

## 1. The search problem -

**Table 1: Directory of senior Bell Labs Physics researchers**

| Name | Phone no. | email |
|---|---|---|
| Batlogg, B | 6663 | batlogg |
| Bishop, D | 3927 | djb |
| Capasso, F | 7737 | fc |
| Chen, YK | 7956 | ykchen |
| Hamann, DR | 4454 | drh |
| Hopfield, J | 7593 | jhopfield |
| Murray, CA | 5849 | camurray |
| Nalamasu, O | 6548 | nims |
| Reichmanis, E | 2504 | er |
| Slusher, RE | 4094 | res |
| Tank, D | 7058 | dwtank |
| Wiltzius, P | 4762 | wiltzius |

Consider the above table describing some of the senior members of the physics research laboratory at Bell Labs, Murray Hill. The names are arranged alphabetically and so if we need the information about any particular member, we can retrieve the information very rapidly in the same manner in which we look up names in a phone book or words in a dictionary. However, if we need some other information, that may not be so easily accessible - e.g., if we wanted to find whether or not a member had a phone number 5849, we would have to look at each and every member of the list until we find one who had the desired phone no. or else until the list terminated.

Neural networks provide the capability to design efficient associative memories where each record is stored in memory as the local minima of a non-linear circuit [Hopfield 82]. One can therefore start the circuit with initial conditions that are close to the local minima and the circuit will rapidly converge to the appropriate local minimum. One can thus give the circuit partial information about the record and the circuit rapidly reaches a state that provides complete information about the record. This technique has been very successful for the *pattern matching* class of problems. For example in commonly designed neural networks, one can give a query like - ***hopfield*** without even specifying which part of the record we were referring to. The circuit, if well designed, would successfully return the appropriate record. In most cases, the circuit would tolerate small amounts of errors, e.g. given a query like ***rechmanas***, the network would probably again converge to the local optimum corresponding to the appropriate record, even though the name was misspelled.

The limitation of the neural network technique is that the network only recognizes proximity to the local optima in certain obvious ways. For complicated queries, e.g. find the person, the number of letters in his email address, multiplied by his phone no. gives $10,027$. That may sound artificially complicated, unfortunately most interesting applications in computer science, especially cryptography, fall into this class.

For this type of application, there is no alternative but to examine each item in the database, one by one, until we find the desired item. There are no short cuts. In the worst case, we will have to examine all $N$ items in the database.

No short cuts, only if we are designing systems based on classical physics. In quantum physics, the same system can be in multiple states at the same time, and by proper design, can be made to examine multiple items at the same time. As this article shows, it is indeed possible to design a quantum mechanical system that

0-7803-5536-9/99/$10.00 ©1999 IEEE

will exhaustively search a database with $N$ records in only about $\sqrt{N}$ steps [Grover 97].

## 2. The laws of the microcosm

Quantum computation is the design of computers in the regime of quantum mechanical phenomena. An $N = 2^n$ state quantum system is implemented as $n$ two state quantum systems. Each two state quantum system is referred to as a qubit. For those with an engineering background, the structure of quantum mechanics is most easily grasped as an extension of classical probabilistic processes. The rules themselves are relatively simple, it is the consequences of those that are mind-boggling, and about which Niels Bohr is said to have remarked, "If after thinking of it for some time, it does not make you dizzy, you have not grasped quantum mechanics."

In order to describe the behavior of a classical probabilistic system, we need to specify probabilities of each state. Quantum mechanical systems have a deeper structure, and as a result, in addition to having a certain probability of being in each state, they also have a phase associated with each state. This leads to wavelike interference. The result of any quantum mechanical phenomena can be calculated by the following rules:

*(i) Amplitudes:* Just as the complete description of a probabilistic system requires the probability of each state, the complete description of a quantum mechanical system is given by specifying the amplitude of each state, which, in general, is a complex number. The amplitude of each state can be written as a magnitude portion and a phase portion. The specification of amplitudes in all of the states is called an amplitude vector ($\bar{\psi}$) or a *superposition*.

*(ii) Probability:* The absolute square of the magnitude of the amplitude of a state gives the probability of the system being in that state.

*(iii) Time-evolution* ($\bar{\psi}(t+1) = U\bar{\psi}(t)$) : Just like the evolution of a probabilistic system is described by premultiplying the probability vector by a state transition matrix, the evolution of the quantum system is obtained by premultiplying the amplitude vector by the state transition matrix ($U$).

*(iv) State transition matrix ($U$) has to be unitary:* In order to conserve probabilities, the classical state transition matrix must follow the constraints that each matrix element must be non-negative and the sum of the matrix elements in each row should be unity. The equivalent constraints the quantum mechanical state transition matrix must follow are that the various columns must be orthonormal, i.e. the dot product of any two distinct column vectors must be zero and the sums of the squares of the magnitudes of the entries in each column vector must be unity (see sections 3 and 5 for examples of state transition matrices).

*(v) Measurement & Collapse* The four rules described above lead to the wave-behavior of particles at the microscopic level. The other characteristic of quantum mechanics that leads to most of its puzzling effects is - the *collapse* of the wavefunction. Whenever any component of the wavefunction is observed, the rest of the wavefunction readjusts itself instantaneously so as to be consistent with the observation. One of the best known consequences of this is the EPR paradox that was presented in 1935 by Einstein, Podolsky and Rosen.

In principle, it is possible to deduce much of the structure of quantum mechanics, such as the uncertainty principle, just from these rules.

$$\bar{p}(t) = (0.4, 0.3, 0.1, 0.2)$$

$$\sum_\alpha p_\alpha = 1$$

$$\bar{p}(t+1) = \left[\ M\ \right] \bar{p}(t)$$

$$\sum_\alpha M_{\beta\alpha} = 1$$

---

$$\bar{\psi}(t) = (0.63, -0.55, 0.32, 0.45)$$

$$\sum_\alpha |\psi_\alpha(t)|^2 = 1$$

$$\bar{\psi}(t+1) = \left[\ U\ \right] \bar{\psi}(t)$$

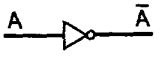
$$\sum_\alpha |U_{\beta\alpha}|^2 = 1;\ \sum_\alpha U_{\beta\alpha} U^*_{\gamma\alpha} = 1 \quad (\beta \neq \gamma)$$

**Figure 1 - A quantum system is described by its *amplitude* vector, its evolution is obtained by premultiplying this by a *unitary* matrix. This is analogous to classical probabilistic systems which are described by a probability vector and whose evolution is obtained by premultiplying this vector by a Markov state transition matrix.**

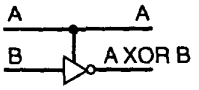
## 3. Basic Devices

Basic building blocks of digital circuits are usually the NAND and the NOR gates. It is easily seen that these are not unitary, e.g. we can think of the NAND gate as the following 2 input, 2 output gate which sends $(A, B)$ to $(A, \overline{AB})$. If we encode the 4 states as 00, 01, 10, 11 respectively, the transformation matrix corresponding to the NAND gate is: $\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$. This is not unitary since the first two columns are not orthogonal and thus it cannot be realized quantum mechanically. So is there no way of synthesizing boolean functions?

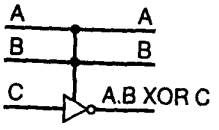
Any Boolean function can indeed be synthesized quantum mechanically but this has to be done carefully with primitives that are unitary. The following is a set of three such primitives:




NOT $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ **A -> (NOT A)**




CNTRL NOT $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ **(A, B) -> (A, A XOR B)**




CNTRL CNTRL NOT — 8X8 unitary matrix — **(A, B, C) -> (A, B, A.B XOR C)**

Each of these primitives can be implemented quantum mechanically (for a nice description of implementation in terms of atoms and photons see [1]).

It is possible to synthesize NAND and NOR gates from these primitives and thus any function, $f(x)$, that can be evaluated classically can be evaluated quantum mechanically. For example the following figure shows the actual design of a full adder in terms of the gates mentioned above.

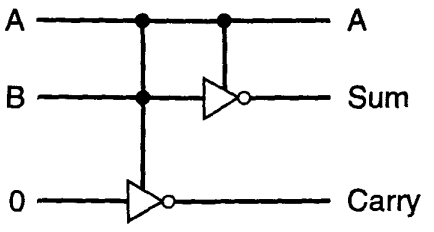
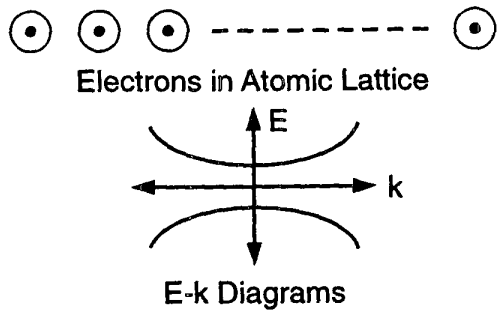



**Figure 2 - Any function *f(x)* that can be synthesized classically, can be efficiently synthesized using NOT, CNTRL-NOT and CNTRL-CNTRL-NOT gates. The above figure shows the design of an adder.**

## 4. Algorithms

The previous section mentioned that any function that can be evaluated classically can equally well be evaluated quantum mechanically with comparable hardware, e.g., the adder of figure 2. Equivalently, any calculation that can be done classically can be done quantum mechanically with comparable hardware. This was of interest as it showed that solid state devices would still work when they got down to atomic scales (although with very different designs). Then, in 1994 Peter Shor showed that a quantum mechanical algorithm could solve a problem no known classical algorithm could.

The problems of efficiently finding the factors of large numbers has been studied for several decades, yet no efficient classical algorithms have been found - it is not known whether or not such an algorithm exists. The problem is vitally important in cryptography since the security of well known codes, such as RSA, is based on the difficulty of this problem. If someone were to find an efficient algorithm for factorization, RSA would become insecure. Shor's insight was to recall the observation that the factorization problem can be converted into one of estimating the periodicity of a sequence, something that quantum systems are very good at.




$\{f(x) = a^x \bmod N, x \text{ integral}\}$ (a is an arbitrary integer < $N$)
$f(x)$ has a periodicity of $p$ ; from the value of $p$, the factors of $N$ can be deduced.

$(7^1 \text{ modulo } 10), (7^2 \text{ modulo } 10) \ldots (7^x \text{ modulo } 10) \ldots$ has a periodicity of 4. From this, the factors of 10 can be deduced.

**Figure 3 - The propagation of electrons in periodic structures is a wonder of quantum mechanics. Shor's factorization algorithm uses quantum mechanics to estimate the periodicity of a sequence of numbers. From this periodicity, the factors of a given number are deduced.**

Peter Shor's discovery generated a lot of excitement in the field [Shor 94]. It raised hopes that this would soon be followed by efficient quantum mechanical algorithms for other important problems. However the next significant quantum mechanical algorithm had to wait until 1996 when I discovered the quantum search algorithm.

## 5. Two more quantum mechanical gates

In order to develop powerful quantum algorithms such as that for search and factorization, two further operations are needed - the Walsh-Hadamard (WH) transformation and the selective inversion operation (figure 4). Both of these operations are carried out on the qubits in a quantum computer.

$$W \equiv \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & \pm 1 & \cdots & \pm 1 \\ 1 & \pm 1 & & \vdots \\ \vdots & \vdots & & \vdots \\ 1 & \pm 1 & \cdots & \pm 1 \end{bmatrix}$$

$$\text{Selective Inversion} \equiv \begin{bmatrix} 1 & & & \\ & -1 & & \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \end{bmatrix}$$

**Figure 4 - In addition to the NOT, CNTRL-NOT & CNTRL-CNTRL-NOT operations, two additional operations are needed to design powerful quantum algorithms.These are the Walsh-Hadamard transform (W) and the selective inversion operation.**

*W-H transformation* - A basic operation in quantum computing is that of a "fair coin flip" performed on a single qubit whose states are 0 and 1. This operation is represented by the following matrix: $M = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. A qubit in the state 0 is transformed into a superposition in the two states: $\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$. Similarly a qubit in the state 1 is transformed into $\left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)$, i.e., the magnitude of the amplitude in each state is $\frac{1}{\sqrt{2}}$ but the *phase* of the amplitude in the state 1 is inverted. Note that the two columns are orthonormal and so the matrices are unitary and can be implemented quantum mechanically. The phase does not have an analog in classical probabilistic algorithms. It comes about in quantum mechanics since the amplitudes are in general complex. This results in *interference* of different possibilities as in wave mechanics; this is what distinguishes quantum mechanical systems from classical probabilistic systems.

When the states are described by $n$ qubits (the system has $N \equiv 2^n$ possible states), we can perform the transformation $M$ on each qubit independently in sequence. The state transition matrix representing this operation will be of dimension $2^n \times 2^n$. If the initial

state had all $n$ qubits in the 0 state, the resultant configuration will have an identical amplitude in each of the $2^n$ states. This is a way of creating a superposition with the same amplitude in all $2^n$ states. In case when the starting state is another one of the $2^n$ states, i.e. a state described by an $n$ bit binary string with some 0s and some 1s. The result of performing the transformation $M$ on each qubit will be a superposition of states described by all possible $n$ bit binary strings with amplitude of each state having an equal magnitude and sign either + or −. This transformation is the WH transformation. It is one of the features that makes quantum mechanical algorithms more powerful than classical algorithms. Either this or a closely related transform called the Fourier Transform, forms the basis for most significant quantum mechanical algorithms.

*Selective inversion* - The other transformation that we need is the *selective inversion* operation. The transformation matrix describing this for a 4-state system is of the form: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, i.e. phases of certain states are inverted. Unlike the WH transformation and other state transition matrices, the probability in each state stays the same since the square of the absolute value of the amplitude in each state stays the same. The application of this phase rotation transform which inverts the amplitude in the j[th] state will be represented as $I_j$. When interpreted in terms of qubits, this selective inversion requires a conditional phase shift - this is the most difficult to implement. It is responsible for much of the power and mystery of quantum computing. A circuit that accomplishes this is shown in figure 5. Keep in mind that it does not need prior knowledge as to which values of $x$ make the function $f(x)$ non-zero. All that is needed is to be able to evaluate $f(x)$ for any given $x$

Note that in general, as mentioned in section 2, the amplitudes can be complex quantities; however, in this paper, we will only need real amplitudes - with either positive and negative signs.

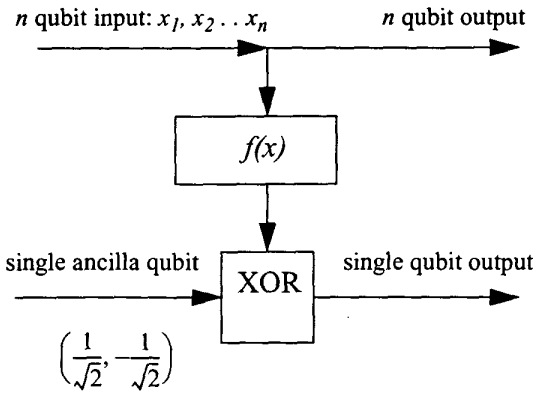



**Figure 5: The above quantum mechanical circuit inverts the amplitudes of precisely those states for which the function *f(x)* is 1.**

## 6. Quantum search

The exhaustive search problem corresponds to the situation described in the introductory section where we had to find the person whose phone no. was specified - we had to exhaustively search the table until we came to a person with the right phone number. In the general case, in the exhaustive search problem, a function $f(x), x = 1, 2 \ldots N$, is given which is known to be non-zero at a single (unknown) value of $x$, say $t$ ($t$ for target) - the goal is to find $t$. If there was no other information about $f(x)$ and one were using a classical computer, it is easy to see that on the average it would take about $\frac{N}{2}$ function evaluations to solve this problem successfully. However, quantum mechanical systems can explore multiple states simultaneously and there is no clear lower bound on how fast this could be done. It was shown in 1994, by using subtle arguments about unitary transforms, that it could not be done in fewer than $O(\sqrt{N})$ steps - subsequently in 1996, I invented an algorithm that took precisely $O(\sqrt{N})$ steps [Grover 97], [Brassard 97]. This problem occurs in two types of contexts - first, where the function $f(x)$ depends on data in memory (this is the database search problem); second, where there is an algorithmic formula known to compute $f(x)$ for any given $x$, yet there is no known way to invert $f(x)$ (e.g. solution of NP-complete problems.)

The circuits of section 3, such as the adder, operate by setting the input to a well defined state and using a sequence of digital gates (NOT, CONTR.NOT and CONTR.CONTR.NOT) to evaluate a Boolean function.

In contrast, the quantum search algorithm starts by setting the system to a superposition of $N$ states corresponding to the $N$ points in the domain to be searched. It can then simultaneously examine all $N$ points, one of which is the desired one. However, if it is programmed to immediately print out the point examined, it will only print out the right one with a probability of $\frac{1}{N}$ since only one of the $N$ points examined is the desired point. It will thus need $N$ such experiments before getting a single observation of the state corresponding to the desired point. Instead, by carrying out a set of quantum mechanical operations, it is possible to amplify the amplitude, and hence the probability, in the desired state at the expense of other states. After this it will indeed print out the desired point with a high probability.

In order to carry out this amplification, we need an *inversion about average* operation. The average is defined as the sum of the amplitudes in all states, divided by the number of states. This operation can be shown to be a unitary operation and may be synthesized as a composite of the following three elementary operations that were discussed in section 5: $WI_{\bar{0}}W$. Here $W$ is the W-H transform operation and $I_{\bar{0}}$ the operation that selectively inverts the amplitude in one of the $N \equiv 2^n$ states in which all qubits are 0. The inversion about average leaves amplitudes whose value is equal to that of the average, unchanged; it increases (decreases) the other amplitudes so that they are as much below (above) the average as they were initially above (below) the average.

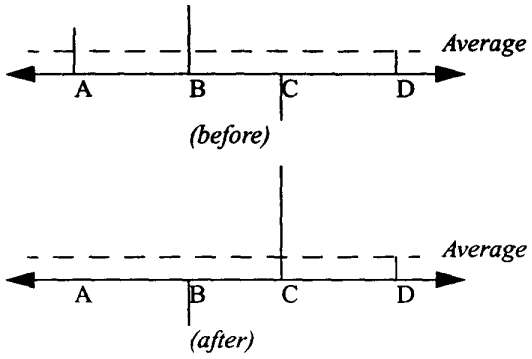
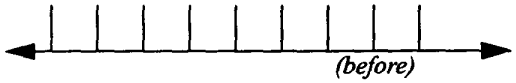
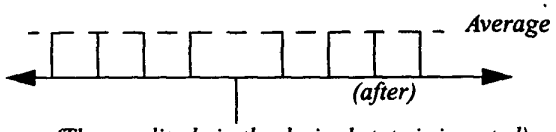



**Figure 6. *Inversion about average* is a unitary operation that can be synthesized in terms of well known quantum mechanical operations.**

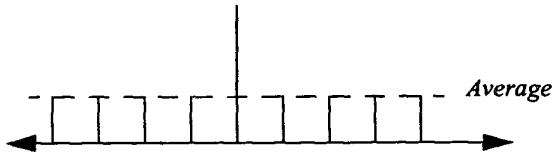
In the quantum search algorithm, the system is first placed in the state with all $n$ qubits in the $\bar{0}$ state, i.e., with all qubits in the 0 state. A W-H transformation operation then creates a superposition with an amplitude of $\frac{1}{\sqrt{N}}$ in each of the $N$ states (note that the amplitudes have to be $\frac{1}{\sqrt{N}}$ since the sum of the squares of the amplitudes, i.e. the probabilities, in all $N$ states should be unity. After that, as shown in figure 7, the amplitude in the state with $f(x) = 1$ is inverted by the circuit of figure 5 by the circuit of figure 5. This is followed by an *inversion about average* operation. This process is successively repeated. The amplitude in the desired state increases by approximately $\frac{2}{\sqrt{N}}$ in each repetition and in approximately $O(\sqrt{N})$ repetitions, the amplitude gets entirely concentrated in the desired state.




*(The system is initialized with equal amplitudes in each state)*




*(The amplitude in the desired state is inverted)*




*(An inversion about average operation boosts the magnitude of the amplitude in the state in which it is opposite of the average)*

**Figure 7: The amplitude in the desired state is boosted by alternately inverting the amplitude in this state and doing an *inversion about average* as in figure 6. After $O(\sqrt{N})$ repetitions the amplitude is entirely in the desired state.**

## 8. Can it be improved?

The quantum search algorithm evoked a lot of interest. It had generally been assumed that one needed to make use of the structure of a problem in order to make use of algorithmic techniques to expedite its solution. This algorithm demonstrated that this was not the case in quantum computing where the well known counter intuitive nature of quantum mechanics could be used to get fast algorithms for problems even without any structure.

Could this be improved? The quantum search algorithm was just a first attempt and surely there would be other algorithms that would improve this beyond $\sqrt{N}$ steps. The surprising result was that it could not be improved beyond $O(\sqrt{N})$ steps. Even more surprisingly, it was later proved that it could not even be improved by a constant factor [Zalka 97]. In the classical case, it is more or less obvious that any algorithm, whether probabilistic or deterministic, cannot search a list of size $N$ in less than $O(N)$ steps. In contrast, in the quantum mechanical case there is no simple explanation why it is not possible to search in fewer than $O(\sqrt{N})$ steps. The proof that this is not possible is based on subtle properties of unitary transformations.

## 9. Extensions

The quantum search algorithm has been shown to be very broadly applicable. In fact it has recently been shown to be a particular case of a large class of algorithms based on the amplitude amplification idea, i.e. by means of a sequence of quantum operations, the amplitude in the desired state is amplified at the expense of the amplitude in other states. In fact, any algorithm that gives a probabilistic solution can be used as part of a quantum mechanical algorithm. The number of steps required by the quantum mechanical algorithm will is the square root of the number of steps required just as in quantum search. One example of such an application is search in the presence of partial information, e.g. retrieving an image from a noisy data transmission [Grover 98].

Another natural application of quantum mechanical algorithms is in the field of statistics. Since a quantum mechanical system can simultaneously be in multiple states and thus simultaneously examine multiple pieces of data, it seems plausible that statistical applications might constitute a natural niche for quantum algorithms. Indeed, the amplitude amplification technique of quantum search adapts well to these applications as well. Efficient algorithms for estimating the mean and median of a population have recently been invented. The number of steps required by these quantum algorithms is the square-root of the best possible classical algorithm.

## Acknowledgments

This material is based upon work supported in part by, the U. S. Army Research Office under contract no. DAAG55-98-C-0040.

## Bibliography

[Hopfield 82] Neural networks and physical systems with emergent collective computational properties, J. Hopfield, *Proceedings, National Academy of Sciences, Vol. 79, p. 2554, 1982.*

[Grover 96] A fast quantum mechanical algorithm for database search, L. K. Grover, *Proceedings, STOC*, 1996, p. 212-218.

[Grover 98] A framework for fast quantum mechanical algorithms, L. K. Grover, *Proceedings STOC*, 1998, p. 53-62, http://xxx.lanl.gov/abs/quant-ph/9711043.

[Brassard 97] Searching a quantum phone book, G. Brassard, *Science*, January 31, 1997, p. 627-629.

[Zalka 97] Grover's quantum searching is optimal, C. Zalka, http://xxx.lanl.gov/abs/quant-ph/9711070.

[Shor 94] Algorithms for quantum computing, discrete log and factoring, P. W. Shor, *Proceedings, FOCS*, 1994, p. 124-134.

## Further reading

[1] Quantum Mechanical Computers, S. Lloyd, *Scientific American*, Oct. 1995, pp. 140-145 (a physics-oriented perspective of quantum computers).

[2] Quantum Computing - Pro & Con, John Preskill, http://xxx.lanl.gov/abs/quant-ph/9705032 (a state of the art description of the potential and limitations of quantum computers).

[3] Quantum Computing with Molecules, N. Gershenfeld & I. Chuang, *Scientific American*, June 1998, pp. 66-71 (a description of NMR based quantum computers).

[4] Quantum Information processing, cryptography, computation and teleportation, T. Spiller, *Proceedings of the IEEE*, Vol. 84, No 12, p. 1719-1746, Dec. 1996.

[5] Beyond Factorization and Search, L. K. Grover, *Science*, Vol. 281, p. 792-794, Aug. 7, 1998.