The Quantum Computing Challenge

Paul Vitányi*

CWI

Abstract. The laws of physics imposes limits on increases in computing power. Two of these limits are interconnect wires in multicomputers and thermodynamic limits to energy dissipation in conventional irreversible technology. Quantum computing is a new computational technology that promises to eliminate problems of latency and wiring associated with parallel computers and the rapidly approaching ultimate limits to computing power imposed by the fundamental thermodynamics. Moreover, a quantum computer will be able to exponentially improve known classical algorithms for factoring, and quadratically improve every classical algorithm for searching an unstructured list, as well as give various speed-ups in communication complexity, by exploiting unique quantum mechanical features. Finally, a quantum computer may be able to simulate quantum mechanical systems, something which seems out of the question for classical computers, thus reaching the ultimate goal of replacing actual quantum mechanical experiments with simulated ones. On the downside, for some problems quantum mechanical computers cannot significantly improve the performance of classical computers.

1 Introduction

Apparently, the earliest mention of quantum computing is by Paul Benioff [14] who demonstrated how to implement a classical Turing machine using quantum mechanical processes. In 1982 Richard Feynman raised the question of simulating elementary quantum mechanical systems by computer. A quantum mechanical system with 2^n basis states—for example an n-bit memory—has pure quantum states that consist of a superposition of all of the basis states. Every basis state in this superposition has a probability amplitude (a complex real) that indicates the probability of observing this basis state in an appropriate measurement. To simulate the evolution of such a system, one needs to track the evolution of the 2^n probability amplitude parameters per simulated step. It is not known how to do this classically in time less than exponential in n. To overcome this problem, Feynman [17] suggested to fight fire with fire: A quantum mechanical computer may possibly be able to simulate every quantum mechanical system in polynomial time since it operates on the same principles.In a way this boils down to the

* Partially supported by the European Union through NeuroCOLT II Working Group and the QAIP Project. Address: CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands. Email: paulv@cwi.nl. The author is also affiliated with the University of Amsterdam.

time-honored method of analogue computing, where an appropriate model of the simulated system is built—like wind tunnels to test turbulence and experimental simulation of water systems. The next step to digital quantum computing was taken by David Deutsch [16] who defined a quantum mechanical version of the classical Turing machine and in turn raised the question whether such computers could possibly speed up classical digital computations significantly over what is achievable by classical computers. The field attracted some interest but remained esotheric: Problems were formulated that benefitted from the quantum mechanical approach but they looked rather contrived. The field gained momentum in 1994 when Peter Shor proposed a fast quantum factoring algorithm [12]. This algorithm (probabilistically) factors a composite l-bit number in slightly over l^2 steps, while the best known classical algorithm, the number field sieve, takes $2^{cl^{1/3}\log^{2/3} l}$ (c constant) steps [19]. The apparent difficulty of factoring composite numbers with only large factors is the basis of almost all commonly used cryptographic systems in financial bank transactions and internet security. The fact that a quantum computer (if it can be build) will compromise such systems galvanized the physics and computer science research communities. A subsequent quantum mechanical algorithmic improvement to searching an unstructured data base excited the interest even further. Apart from some other less straightforward improvements (below), however, these remain the only genuine successes of quantum algorithmics today. Worse, it can be shown that for many problems quantum mechanical methods don't help significantly [3, 2, 9], like, for example, with binary search [1]. While the jury is still out whether quantum mechanical computing is the greatest thing in computing since the invention of personal computing, it seems evident that the advent of the quantum computer is unavoidable since improvement of computing by parallellizing and further miniaturizing of classical methods runs into problems. The purpose of this writing is to give an outline of the area of quantum computing without going into much detail—a general textbook treatment of both the theory and experimental realization is [9].

1.1 Wither Classical Computing?

In performance analysis of classical sequential computation such as performed by a Turing machine or a von Neumann architecture computer—the common computer—one can safely ignore many physical aspects of the underlying computer system and analyze the computational complexity of an algorithm or program in a purely logical fashion. One cannot always ignore the reality of the physical world we live in to such an extent. The appropriateness of the analysis may stand or fall with the account taken of physical reality: Nonclassical or nonstandard physical realizations of computers may have totally unexpected properties, such as, for example, the quantum computer. To see why quantum computing may be the natural next step in computing technology, let us analyze some problems attending the further improvement of classical computing. Two natural ways to improve classical computing are continued miniaturization and large-scale parallellization.

The Cooking Problem Computers increasingly pervade our society. This increasing influence is enabled by their ever increasing power, which has roughly doubled every 18 months for the last half-century (Moore's law). The increase in power, in turn, is primarily due to the continuing miniaturization of the elements of which computers are made, resulting in more and more elementary gates with higher and higher clock pulse per unit of silicon, accompanied by less and less energy dissipation per elementary computing event. Roughly, a linear increase in clock speed is accompanied by square increase in elements per silicon unit—so if all elements compute all of the time, then the dissipated energy per time unit rises cubicly (linear times square) in absence of energy decrease per elementary event. The continuing dramatic decrease in dissipated energy per elementary event is what has made Moore's law possible. But there is a foreseeable end to this: There is a minimum quantum of energy dissipation associated with elementary events. This puts a fundamental limit on how far we can go with miniaturization, or does it?

Both classical and quantum physics are believed to be strictly reversible at the fundamental level: A complete description of the microscopic state of the system uniquely determines the earlier and future states of the system—this holds not only in Newtonian mechanics but for example also for the unitary evolution of every quantum mechanical system. Currently, computations are commonly irreversible, even though the physical devices that execute them are fundamentally reversible. This irreversibility is due to the fact that information tends to be erased all the time: computing a function like $a + b = c$ one inputs a and b and obtains output c. From c one cannot uniquely retrieve a and b. The contrast between the physics of the computing machinery which is reversible and the executed irreversible computation is only possible at the cost of efficiency loss by generating thermal entropy into the environment. With computational device technology rapidly approaching the elementary particle level this effect gains in significance to the extent that efficient operation (or operation at all) of future computers requires them to be reversible. The 'logically irreversible' operations in a physical computer necessarily dissipate $kT \ln 2$ energy by generating a corresponding amount of entropy for every bit of information that gets irreversibly erased; the logically reversible operations can in principle be performed dissipation-free. Here k is Boltzmann's constant and T the absolute temperature in degrees Kelvin, so that $kT \approx 3 \times 10^{-21}$ Joule at room temperature.

Extrapolations of current trends show that the energy dissipation per binary logic operation needs to be reduced below kT (thermal noise) within 15 years. Even at kT level, a future laptop containing 10^{13} gates operating at 100 gigahertz dissipates 3,000 watts. For thermodynamic reasons, cooling the operating temperature of such a computing device to almost absolute zero (to get kT down) must dissipate at least as much energy in the cooling as it saves for the computing.

Especially Landauer [18] has argued that it is only the *irreversible* elementary events (like erasing information) that necessarily dissipate energy; there is no physical law that requires *reversible* events (like negation) to dissipate energy.

It has been shown that all irreversible computations can be performed logically reversibly at the cost of possibly increasing computation time and memory [4]. It remains to develop the technology to implement the physical reversible execution of the logically reversible computation in a dissipation-free manner. Reversible computers can be implemented using quantum-mechanical technologies; quantum-mechanical computers are reversible except for the observation phases. So far the development of computation machinery is mostly based on the principles of classical physics and irreversible components. At the basic level, however, matter is governed by quantum mechanics, which is reversible. Further miniaturization will very soon reach scales where quantum mechanical effects take over and classical laws cease to apply accurately. The mismatch of computing organization and reality will express itself in friction: increasingly powerful computers will dissipate increasing and unsustainable amounts of energy unless their mode of operation becomes quantum mechanical (and thus reversible). That is, harnessing quantum mechanical effects may be essential for further miniaturization and hence acceleration of classical computing methods.

There is an added bonus: once we get involved in quantum effects, it appears we can go further than just miniaturizing classical computers to the quantum scale. Quantum mechanics may actually spawn a *qualitatively new* kind of computing: a kind which profits from quantum effects to boost computation to such an extent that things are achieved that would forever be out of reach of classical computers, even if these could be miniaturized to the same level.

The Spaghetti Problem Parallel computation that allows processors to randomly access a large shared memory, or rapidly access a member of a large number of other processors, will necessarily have large latency. If we use n processing elements of, say, unit size each, then the tightest they can be packed is in a 3-dimensional sphere of volume n. Assuming that the units have no "funny" shapes, assume for example they are spherical, some units are at distance equal to the radius R from one another,

$$R = \left(\frac{3n}{4\pi}\right)^{1/3} \tag{1}$$

Because of the bounded speed of light, it is impossible to transport signals over n^α $(\alpha > 0)$ distance in $o(n)$ time. In fact, the assumption of the bounded speed of light says that the lower time bound on *any* computation using n processing elements is linear in $n^{1/3}$ outright.

The *spaghetti* problem is as follows: We illustrate the approach with a popular architecture, say the *binary d-cube*. Recall, that this is the network with $n = 2^d$ nodes, each of which is identified by a d-bit name. There is a two-way communication link between two nodes if their identifiers differ by a single bit. The network is represented by an undirected graph $C = (V, E)$, with V the set of nodes and $E \subseteq V \times V$ the set of edges, each edge corresponding with a communication link. There are $d2^{d-1}$ edges in C. Let C be embedded in 3-dimensional

Euclidean space, each node as a sphere with unit volume. The distance between two nodes is the Euclidean distance between their centers.

Lemma 1. *The average Euclidean length of the edges in the 3-space embedding of C is at least $7R/(16d)$.*

One can derive a general theorem that gives similar lower bounds that are optimal in the sense of being within a constant multiplicative factor of an upper bound for several example graphs of various diameters, [24]. At present, many popular multicomputer architectures are based on highly symmetric communication networks with small diameter. Like all networks with small diameter, also asymmetric ones like complete binary trees, such networks will suffer from the communication bottleneck above, necessarily contain *some* long interconnects (embedded edges). However, the desirable fast permutation routing properties of symmetric networks don't come free, since they require that the average of *all* interconnects is long. Then, the ratio between the volume of the combined processing elements and the required volume of the wires vanishes—even for moderate numbers the processors become needles in a haystack of wires. Here we have not yet taken into account that longer wires need larger drivers and have a larger diameter, that the larger volume will again cause the average interconnect length to increase, and so on, which explosion may make embedding altogether impossible with finite length interconnects as exhibited in a related context in [23]. It appears that the "spaghetti" problem too may be resolved—in a fashion—by the inherent parallellism of coherent quantum computation.

2 Quantum Computation

The quantum computer as first advocated in [14] is currently aimed to exploit the standard physical model that quantum evolution of an appropriate system consists in a superposition of many (potentially infinitely many) simultaneous computation paths. It is theoretically possible that through the specific quantum mechanical rules of interference of the different paths one can boost the probability associated with desirable evolutions and suppress undesirable ones. Upon observation of the system state one of the states in superposition is realized. By quantum specific algorithmic techniques the idea is to observe the desired outcome with appropriately large positive probability, or to compute the desired outcome from the observed data with high probability.

This coherent quantum computing (CQC) approach will partially alleviate the wiring problem (Section 1.1) because an exploding number of different computation paths will be simultaneously followed (with appropriate probability amplitudes, to be sure) by the same single physical apparatus requiring but a tiny amount of physical space. This may be the substance of R. Feynman's dictum "there is room at the bottom" in the context of his proposal of CQC, [17]. Of course, since the different computation paths of a quantum computation cannot communicate as is often a main feature in a parallel distributed computation, it is only a very special type of room which is available at the bottom.

Since the quantum evolution in a computation if unobstructed by observation and decoherence is reversible, the pure form of CQC, apart from the irreversible observation phase, can be energy dissipation free. CQC seems to a very large extent to achieve the optimal thermodynamic neutral computation aimed for in Section 1.1. Although there seems to be agreement that energy gets dissipated in the irreversible observation phase, to the author's knowledge it is not yet clear how much.

Through a sequence of proposals [14], [17], [16], [7], there has emerged a Turing machine model of quantum coherent computing. There is also the closely related and in often more convenient quantum circuit model, see [9]. In this introduction we only aim at outlining basic principles, algorithmic improvements, and technological problems attending the quantum computing challenge. Details can be found in the references.

2.1 Background: Probabilistic Turing Machines

The simplest way to introduce quantum computing seems by way of probabilistic computing. Suppose we consider the well known probabilistic Turing machine which is just like an ordinary Turing machine, except that at each step the machine can make a probabilistic move which consists in flipping a (say fair) coin and depending on the outcome changing its state to either one of two alternatives. This means that at each such probabilistic move the computation of the machine splits into two distinct further computations each with probability $\frac{1}{2}$. Ignoring the deterministic computation steps, a computation involving m coinflips can be viewed as a binary computation tree of depth m with 2^m leaves, where the set of nodes at level $t \leq m$ correspond to the possible states of the system after t coinflips, every state occurring with probability $1/2^t$. For convenience, we can label the edges connecting a state x directly with a state y with the probability that a state x changes into state y in a single coin flip (in this example all edges are labeled '$\frac{1}{2}$').

For instance, given an arbitrary Boolean formula containing m variables, a probabilistic machine can flip its coin m times to generate each of the 2^m possible truth assignments and subsequently check deterministically wether the assignment makes the formula true. If there are k distinct such assignments then the probabilistic machine finds that the formula is satisfiable with probability at least $k/2^m$—since there are k distinct computation paths leading to a satisfiable assignment.

Now suppose the probabilistic machine is hidden in a black box and the computation proceeds without us knowing the outcomes of the coin flips. Suppose that after m coin flips we open part of the black box and observe the bit which denotes the truth assignment for variable x_5 $(5 \leq m)$. Before we opened the black box all 2^n initial truth assignments to variables $x_1, \ldots, x_m$ were still equally possible, each with probability $1/2^m$. After we observed the state of variable x_5, say 0, the probability space of possibilities has collapsed to the truth assignments which consist of all binary vectors with a 0 in the 5th position each of which has probability renormalized to $1/2^{m-1}$.

2.2 Quantum Turing Machines

A quantum Turing machine can be viewed as a generalization of the classical probabilistic Turing machine. Consider the same computation tree. In the probabilistic computation there is a probability $p_i \geq 0$ associated with each node i (state of the system) at the same level in the tree, such that $\sum p_i = 1$, summed over the nodes at the same level. In a quantum mechanical computation there is a "probability amplitude" α_i associated with each basis state $|i\rangle$ of the system. The "bra-ket" notation is due to P. Dirac and is the standard quantum mechanics notation. The "bra" $\langle x|$ denotes a row vector with complex entries, and "ket" $|x\rangle$ is the column vector consisting of the conjugate transpose of $\langle x|$ (columns replaced by rows and the imaginary part of the entries negated, that is, $\sqrt{-1}$ is replaced by $-\sqrt{-1}$). For convenience consider the simple case (corresponding to the probabilistic example) where i runs through the classical values 0 through $2^m - 1$, so that $|i\rangle$ is an m-bit binary column vector. The amplitudes are complex numbers satisfying $\sum ||\alpha_i||^2 = 1$, where if $\alpha_i = a + b\sqrt{-1}$ then $||\alpha_i|| = \sqrt{a^2 + b^2}$, and the summation is taken over all distinct states of the observable at a particular instant. The transitions are governed by a matrix U which represents the program being executed. Such a program has to satisfy the following constraints. Denote the set of possible configurations of the Turing machine by X, where X is the set of m-bits column vectors (the basis states) for simplicity. Then U maps the column vector $\underline{\alpha} = (\alpha_x)_{x \in X}$ to $U\underline{\alpha}$. Here $\underline{\alpha}$ is a vector of amplitudes of the quantum superposition of the basis states before the step, and $U\underline{\alpha}$ the same after the step concerned. The special property which U needs to satisfy in quantum mechanics is that it is *unitary*, that is, $U^\dagger \cdot U = I$ where I is the identity matrix and $U^\dagger$ is the conjugate transpose of U (as with the bra-ket, "conjugate" means that all $\sqrt{-1}$'s are replaced by $-\sqrt{-1}$'s and 'transpose' means that the rows and columns are interchanged). In other words, U is unitary iff $U^\dagger = U^{-1}$.

The unitary constraint on the evolution of the computation enforces two facts.

1. If $U^0\underline{\alpha} = \underline{\alpha}$ and $U^t = U \cdot U^{t-1}$ then $\sum_{x \in X} ||(U^t\underline{\alpha})_x||^2 = 1$ for all t (discretizing time for convenience).
2. A quantum computation is reversible (replace U by $U^\dagger = U^{-1}$).

The quantum version of a single bit is a superposition of the two basis states a classical bit can have:

$$|\Psi\rangle = \alpha \ |0\rangle + \beta \ |1\rangle,$$

where $||\alpha||^2 + ||\beta||^2 = 1$. Such a state $|\psi\rangle$ is called a quantum bit or *qubit*. It consists of partially the basis state $|0\rangle$ and partially the basis state $|1\rangle$. The states are denoted by the column vectors of the appropriate complex probability amplitudes. For the basis states the vector notations are: $|0\rangle = \binom{1}{0}$ (that is, $\alpha = 1$ and $\beta = 0$), and $|1\rangle = \binom{0}{1}$ (that is, $\alpha = 0$ and $\beta = 1$).

Physically, for example, the state $|\psi\rangle$ can be the state of a polarized photon, and the basis states are horizontal or vertical polarization, respectively. Upon measuring according to the basis states, that is, passing the photon through a medium that is polarized either in the horizontal or vertical orientation, the photon is observed with probability $||\alpha||^2$ or probability $||\beta||^2$, respectively.

Consider a sample computation on a one-bit computer executing the unitary operator:

$$S = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}. \tag{2}$$

It is easy to verify, using common matrix calculation, that

$$S\ |0\rangle = \frac{1}{\sqrt{2}}\ |0\rangle - \frac{1}{\sqrt{2}}\ |1\rangle,\ S\ |1\rangle = \frac{1}{\sqrt{2}}\ |0\rangle + \frac{1}{\sqrt{2}}\ |1\rangle$$
$$S^2\ |0\rangle = 0\ \ |0\rangle - 1\ \ |1\rangle = -\ |1\rangle,\ S^2\ |1\rangle = 1\ \ |0\rangle + 0\ \ |1\rangle = \ |0\rangle$$

If we observe the computer in state $S\ |0\rangle$, then the probability of observing state $|0\rangle$ is $(\frac{1}{\sqrt{2}})^2 = \frac{1}{2}$, and the probability to observe $|1\rangle$ is $(-\frac{1}{\sqrt{2}})^2 = \frac{1}{2}$. However, if we observe the computer in state $S^2\ |0\rangle$, then the probability of observing state $|0\rangle$ is 0, and the probability to observe $|1\rangle$ is 1. Similarly, if we observe the computer in state $S\ |1\rangle$, then the probability of observing state $|0\rangle$ is $(\frac{1}{\sqrt{2}})^2 = \frac{1}{2}$, and the probability to observe $|1\rangle$ is $(\frac{1}{\sqrt{2}})^2 = \frac{1}{2}$. But now, if we observe the computer in state $S^2\ |1\rangle$, then the probability of observing state $|0\rangle$ is 1, and the probability to observe $|1\rangle$ is 0. Therefore, the operator S inverts a bit when it is applied twice in a row, and hence has acquired the charming name *square root of 'not'*. Note that there is no classical unconditional Boolean operator that has the effect of S; this is one difference between quantum computation and probabilistic computation.

In computing the above amplitudes, subsequent to two applications of S, according to matrix calculus we found that

$$\begin{aligned} S^2\ |1\rangle &= \tfrac{1}{\sqrt{2}} \left(\tfrac{1}{\sqrt{2}}(\ |0\rangle -\ |1\rangle) + \tfrac{1}{\sqrt{2}}(\ |0\rangle +\ |1\rangle) \right) \\ &= \tfrac{1}{2}(\ |0\rangle -\ |1\rangle +\ |0\rangle +\ |1\rangle) \ = \ |0\rangle. \end{aligned}$$

In a probabilistic calculation, flipping a coin two times in a row, we would have found that the probability of each computation path in the complete binary computation tree of depth 2 was $\frac{1}{4}$, and the states at the four leaves of the tree were $|0\rangle$, $|1\rangle$, $|0\rangle$, $|1\rangle$, resulting in a total probability of observing $|0\rangle$ being $\frac{1}{2}$, and the total probability of observing $|1\rangle$ being $\frac{1}{2}$ as well.

The principle involved is called *interference*, similar to the related light phenomenon in the seminal "two slit experiment:" If we put a screen with a single small enough hole in between a light source and a target, then we observe a gradually dimming illumination of the target screen, the brightest spot being colinear with the light source and the hole. If we put a screen with *two* small holes in between, then we observe a diffraction pattern of bright and dark stripes due to interference. Namely, the light hits all of the screen via two different routes (through the two different holes). If the two routes differ by an even number of half wave lengths, then the wave amplitudes at the target are added, resulting in twice the amplitude and a bright spot, and if they differ by an odd number of half wave lengths then the wave amplitudes are in opposite phase and are subtracted resulting in zero and a dark spot. Similarly, with quantum computation, if the quantum state is

$$|\Psi\rangle = \alpha\ |x\rangle + \beta\ |y\rangle,$$

then for $x = y$ we have a probability of observing $|x\rangle$ of $||\alpha + \beta||^2$, rather than $||\alpha||^2 + ||\beta||^2$ which it would have been in a probabilistic fashion. For example, if $\alpha = \frac{1}{\sqrt{2}}$ and $\beta = -\frac{1}{\sqrt{2}}$ then the probability of observing $|x\rangle$ is 0 rather than $\frac{1}{2}$, and with the sign of β inverted we observe $|x\rangle$ with probability 1.

2.3 Multi-Qubit states

Above, a qubit takes a value in $\mathcal{C}^2$ where $\mathcal{C}$ is the set of complex real numbers. To generalize this to strings of n qubits, we consider the quantum state space $\mathcal{C}^N$ with $N = 2^n$. The basis vectors $e_0, \ldots, e_{N-1}$ of this space are parametrized by binary strings of length n, so that e_0 is shorthand for $e_{0\ldots0}$ and e_{N-1} is shorthand for $e_{1\ldots1}$. Mathematically, $\mathcal{C}^N$ is decomposed into a tensor product of n copies of $\mathcal{C}^2$, and an n-qubit state $|a_1 \ldots a_n\rangle$ in bra-ket notation can also be written as the tensor product $|a_1\rangle \ldots |a_n\rangle$, a string of n qubits, the qubits being distinguished by position.

2.4 Input and Output

We restrict ourselves to the case where the quantum computer has a classical input. If the input x has k bits, and the number of qubits used by the computation is $n \geq k$ (input plus work space), then we pad the input with nonsignificant 0's and start the quantum computation in an initial state (which must be in $\mathcal{C}^N$) $e_{x0\ldots0}$. When the computation finishes the resulting state is a unit vector in $\mathcal{C}^N$, say $\sum_i \alpha_i e_i$ where i runs through $0, \ldots, N-1$ and the probability amplitudes α_i's satisfy $\sum_i ||\alpha_i||^2 = 1$. The output is obtained by performing a measurement with as possible outcomes the basis vectors. The observed output is probabilistic: we observe basis vector e_i, that is, the n bit string i, with probability $||\alpha_i||^2$. A typical scenario is as follows. Suppose there are "right" and "wrong" answers and we can verify whether an answer is "right", as in the case when we receive a purported factor of a composite number. Suppose further that the probability of obtaining a "right" answer is at least $\frac{1}{2}$. Then running the algorithm t times, the probability of receiving t "wrong" answers is at most 2^{-t}.

2.5 Quantum Algorithmics

A quantum algorithm corresponds to a unitary transformation U that is built up from elementary unitary transformations, every one of which only acts on one or two qubits. The algorithm applies U to an initial classical state containing the input and then makes a final measurement to extract the output from the final quantum state. The algorithm is "efficient" if the number of elementary operations is "small", which usually means at most polynomial in the length of the input. Quantum computers can do everything a classical computer can do probabilistically — and more.

3 What quantum computers can do

A readable brief introduction to quantum algorithms is [13]. The biggest success so far — and the event which ignited the current explosive growth of the field of quantum computing — was Peter Shor's 1994 discovery of an efficient quantum algorithm for finding the prime factors of large composite integers [12].

Fast Factoring All fast factoring algorithms are based on the following idea: To factor a composite integer M it suffices to obtain two integers x and y such that $x^2 \equiv y^2 \pmod{M}$ but $x \not\equiv \pm y \pmod{M}$. Then $(x-y)(x+y) \equiv 0 \pmod{M}$ but neither factor in the left-hand side is divisible by M. Hence, $x+y$ contains one factor of M and $x-y$ another one. To obtain a factor we determine the greatest common divisor of $x+y$ and M, for example by Euclid's algorithm which runs in polynomial time. In the quantum algorithm we try to find the period r of a random x modulo M with $1 < x < M$. That is, $x^r \equiv 1 \pmod{M}$. If r is even then $(x^{r/2}+1)(x^{r/2}-1) \equiv 0 \pmod{M}$. If also $x^{r/2} \not\equiv -1 \pmod{M}$ then we can determine a proper factor of M by the gcd algorithm as above. Number theory tells us that at least half of all x's yield a proper prime factor this way. Hence, repeating this process k times, the probability is at least $1-1/2^k$ that we find a proper factor of M. Thus, the problem of factoring M reduces to finding the order r. This is done by judicious use of the so-called quantum Fourier transform, the details of which can be found in Shor's articles referred to above or the textbook [9].

The trick is to first use a sequence H_n of n one-qubit unitary operations, the Hadamard transform,

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

on the successive bits of a register of n bits originally in the all–0 state $|\Psi\rangle = |00\ldots0\rangle$. The result is a superposition of

$$H_n \; |\Psi\rangle = \sum_{x \in \{0,1\}^n} 2^{-n/2} \; |x\rangle$$

of all the 2^n possible states of the register, each with amplitude $2^{-n/2}$ (and hence probability of being observed of 2^{-n}.)

The Hadamard transform is ubiquitous in quantum computing; its singlefold action is similar to that of the transform S of eq. (2) with the the roles of "0" and "1" partly interchanged. In contrast to S^2 that implements the logical "not," we have $H^2 = I$ with I the identity matrix.

Now the computation proceeds in parallel along the exponentially many computation paths in quantum coherent superposition. A sequence of tricky further unitary operations, for example the "quantum Fourier transform," and observations serves to exploit interference (and so-called entanglement) phenomena to effect a high probability of eventually observing outcomes that allow us to determine the desired period r.

One principle that is used in many quantum algorithms (and also in factoring) is as follows. If A is a classical algorithm for computing some function f, possibly even irreversible like $f(x) \equiv x \pmod 2$, then we can turn it into a unitary transformation which maps classical state $|x, 0\rangle$ to $|x, f(x)\rangle$. Note that we can apply A to a superposition of all 2^n inputs:

$$A\left(2^{-n/2}\sum_x |x,0\rangle\right) = 2^{-n/2}\sum_x |x, f(x)\rangle.$$

In some sense this state contains the results of computing f for *all* possible inputs x, but we have only applied A once to obtain it. This effect shows one of the advantages of quantum over classical computing and is called *quantum parallelism.*

By making clever use of superpositions, interference, quantum parallelism, and some classical number theory, Shor's algorithm finds a factor of a number M in time roughly the square of the length of the input (which is $l = \log M$ bits). In contrast, every known classical algorithm requires exponential time asymptotically in excess of $2^{l^{1/3}}$ (see Introduction) to factor an l-bit number. Since factoring is one of the most elementary aspects of number theory, the oldest mathematical discipline, and centuries of efforts by the greatest mathematicians have not yielded better methods, it is widely believed that such better methods either do not exist or are prohibitively difficult to find. In fact, this belief underlies most of current public-key cryptography, notably the RSA system, ubiquitously used on the Internet and in the financial world. Such crypto-systems can be broken if one can factor large numbers fast. Accordingly, the advent of quantum computing compromises all such systems: if a quantum computer can be built, then most of current cryptography becomes totally insecure, and, for example, electronic money can be forged.

Cryptography What quantum computing takes away with one hand (classical public-key crypto), it gives back in another form with the other (quantum secret-key crypto) albeit only in part. Already in 1984, Bennett and Brassard found a scheme which allowed two distant parties to obtain a shared secret key via quantum mechanical communication [5]. Their scheme was believed to be fully secure against any type of spy or eavesdropper, and this has indeed been formally proven. The security relies on the basic tennet of quantum mechanics that one cannot observe a quantum state consisting of a superposition of basic states without destroying it and a state in superposition cannot be copied—hence an eavesdropper cannot tamper with a message consisting of such states without this being detectable if the sender and receiver exchange classical information (which can be done in such a way that it gives zero information about the quantum message to the eavesdropper). On the other hand, some other parts of electronic transactions, like unforgeable signatures, appear to be beyond the power of quantum methods.

Rapid Search A third application is Grover's 1996 algorithm for searching databases [11]. Consider finding some specific record in a large unordered database of N items. Classically, there is no smarter method than just to go through all records sequentially, which will requires expected $N/2$ time steps for a record in general position. Grover's algorithm, however, uses quantum superpositions to examine all records "at the same time", and finds the desired record in roughly $\sqrt{N}$ steps. Examining 10^{12} records with unit microsecond probes, this is the difference between about two months of computing and one second of computing! His algorithm also allows to solve the widespread and notoriously hard NP-complete problems (such as the traveling salesman problem) quadratically faster than known classical methods—reducing say exponential time with exponent N to exponential time with exponent $N/2$.

Communication Complexity A fourth application deals with the setting where two separated parties, Alice and Bob, want to compute some function $f(x, y)$ depending on x (only known to Alice) and y (only known to Bob). A simple scheme would be for Alice to send her x to Bob and then let Bob do all the work by himself, but this may take a lot of bits of communication and often there are much more clever schemes requiring less communication. The field of *communication complexity* examines the optimal number of bits that have to be communicated in order to compute the function at hand. What happens if we generalize this setting to the quantum world and allow Alice and Bob the use of quantum computers and qubit-communication? It turns out that some tasks can be solved with significantly less communication if we allow such quantization [10] — and this despite a fact known as "Holevo's theorem" stating that quantum bits cannot contain more information than classical bits! It turns out there are similar advantages by sticking to classical communication, but allowing Alice and Bob the use of pre-established "entangled" qubits. Physically, such entangled qubits appear in the Einstein-Podolsky-Rosen paradox, capable of a paradoxical spooky instantaneous action at a distance. While such action can't exchange information since that would violate the speed of light bound, the mentioned results establish the unexpected fact that it can *save* on exchanged information. The first results using pre-established entangled qubits and classical communication were obtained by Cleve and Buhrman [8]. Both approaches beat the limits provable for just classical communication.

Simulating Quantum Matter Finally, some indications have been given that quantum mechanical computers may be able to simulate quantum mechanical systems at acceptable speeds, [26], just as envisioned by Feynman [17] in his original suggestion of the possibility and use of quantum mechanical computing.

Limitations The above developments suggested the vision that *all* computation can be enormously sped up by quantum computers. But not so! There are strong and general *limitations* of quantum computers as well. Recall that Grover's algorithm is quadratically faster than classical search algorithms. It was already known that such a quadratic speed-up is the best quantum computers can achieve for searching a database, so exponential speed-ups cannot be obtained for this problem. Recently, it was shown that the same holds for *all* problems in the database-setting of Grover's algorithm: for all such problems, quantum computers can be at most polynomially faster than classical computers [3]. Other lower bounds showing that for certain problems quantum mechanical methods do not improve significantly on classical methods have been given in [1] for binary search, [2, 9] for other problems. The polynomial lower bound approach has been partially extended to quantum communication complexity as well, but there have also been exhibited some new (polynomial) speed-ups, both in Grover's setting and in communication complexity.

Limiting results like the above, of course, do not preclude exponential speed-ups in different settings, like Shor's, or a clever future setting as yet unknown. Exploring this potential of quantum computation remains an exciting and important task for computer scientists and physicists alike. The current state of knowledge resembles the situation of a person in a dark room, knowing by touch some isolated objects. Somebody has to turn on the light.

4 How Quantum Computers Do It

How about actually building quantum computers that can run the fast algorithms like Shor's, Grover's, or communication complexity algorithms? To date only very small quantum algorithms (and larger quantum crypto devices) have been implemented. On the one hand, quantum cryptographical systems work fine over fiber optics interconnects and even over common co-axial TV cables over considerable distances like 40 km, sufficient for a local area network. This application essentially works by transmitting single polarized photons—in superposition of two orthogonal polarizations. Technological difficulties and system noise apparently give a very low yield in useful bits of, say, one in 10,000.

Currently, the most sophisticated experimental quantum computer is the five qubit NMR based computer built by the team of Isaac Chuang, [22], at IBM Almaden's Research Center. While there are larger qubit assemblies extant, with this five qubit computer the experimenters succeeded in executing the quantum mechanical step in Shor's fast factoring algorithm—the order-finding step—for every example in its range, while a conventional computer would require repeated steps to solve the problem.

One problem with physical realizations that we need to perform logical operations on pairs of intact pure quantum states, which means that we have to bring them together, operate on them, and transmit the result, all without destroying the superposition. Generally, physical realizations of CQC will have to struggle with the fact that the coherent states of the superposition will tend to deteriorate by interaction with each other and the universe, a phenomenon called *decoherence*, see for example [27]. The larger the scale of the coherent superposition the faster decoherence. Of course, the gain one hopes to make with quantum computing over classical computing must come out of the scale of the superposition. In [21] it is calculated that that CQC calculations using physical realizations based on spin lattices will have to be finished in an extremely short time. For example, factoring a 1000 bit number in square quantum factoring time we have to perform 10^6 steps in less than the thermal time scale $\hbar/kT$ which at 1 K is of order 10^{-9} seconds. Such a CQC computation would need to proceed at optical frequencies. Other technologies have more favorable characteristics, see [9] Chapter 7. Quantum versions of error-correcting codes have been developed recently which may solve this problem in theory, but not yet in the brittle practice of the physical lab (let alone the brittle practice of our desktops). This is related to development of Quantum Information Theory—the quantum extension of classical information theory. For a survey see [6]. Similarly, classical Kolmogorov complexity (algorithmic information theory) [20] can be extended to a quantum mechanical approach [25]. Actually building large quantum computers presents formidable problems to experimental physicists reminiscent of the initial barriers to classical computing: unreliable components, physically large components, memory, organisation, communication, programming. This has the effect that hitherto unexplored areas of quantum mechanical theory are suddenly put in the limelight. In fact, the theory of quantum mechanics experiences a renaissance, and is being extended, for example with respect to the algebraic analysis of "quantum entanglement"—a vital notion in many quantum algorithms, not yet thoroughly investigated in quantum theory, [15].

References

1. A. Ambainis, A better lower bound for quantum algorithms searching an ordered list, *Proc. 40th IEEE Symp. Foundat. Comput. Sci.*, 352–357, 1999.
2. A. Ambainis, Quantum lower bounds by quantum arguments, *Proc. 32nd ACM Symp. Theor. Comput.*, 2000.
3. R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. In *Proc. 39th IEEE Symp. Foundat. Comput. Sci.*, 1998, 352–361.
4. C.H. Bennett. Logical reversibility of computation. *IBM J. Res. Develop.*, 17(1973), 525–532.
5. C.H. Bennett, F. Bessette, G. Brassard, L. Salvail and J. Smolin, Experimental quantum cryptography, *J. Cryptology*, 5:1(1992), 3-28; C.H. Bennett, G. Brassard and A. Ekert, Quantum cryptography, *Scientific American*, Oct. 1992, 50-57.
6. C.H. Bennett and P.W. Shor, Quantum information theory, *IEEE Trans. Inform. Th.*, IT-44:6(1998), 2724–2742.
7. E. Bernstein and U. Vazirani, Quantum complexity theory, *SIAM J. Comput.*, 26:5(1997), 1411–1473.

8. R. Cleve and H. Buhrman, Substituting quantum entanglement for communication, Physical Review A, 56:2(1997),1201-1204.
9. M.A. Nielsen and I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.
10. H. Buhrman, R. Cleve, and A. Wigderson. Quantum vs. classical communication and computation. In *Proc. 30th ACM Symp. Theor. Comput.*, 1998, 63–68.
11. L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proc. 28th ACM Symp. Theor. Comput.*, 1996, 212–219.
12. P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26:5(1997), 1484–1509.
13. P.W. Shor, Introduction to quantum algorithms, http://xxx.lanl.gov/abs/quant-ph/0005003
14. P. Benioff, J. Stat. Phys., 22(1980), 563–591, also *J. Math. Phys.*, 22(1981), 495–507, *Int. J. Theoret. Phys.*, 21(1982), 177–201, *Phys. Rev. Letters*, 48(1982), 1581–1585, *J. Stat. Phys.*, 29(1982), 515–546, *Phys. Rev. Letters*, 53(1984), 1203, *Ann. New York Acad. Sci.*, 480(1986), 475–486.
15. D.P. DiVincenzo, T. Mor, P.W. Shor, J.A. Smolin, B.M. Terhal, Unextendible Product Bases, Uncompletable Product Bases and Bound Entanglement, http://xxx.lanl.gov/abs/quant-ph/9908070
16. D. Deutsch, Quantum theory, the Church-Turing principle and the universal quantum computer. *Proc. Royal Society London, Series A*, 400(1985), 97–117; see also *Proc. Royal Society London, Series A*, 425(1989), 73–90; with R. Josza, *Proc. Royal Society London, Series A*, 439(1992), 553–558.
17. R.P. Feynman, Simulating physics with computers, *Int. J. Theoret. Physics*, 21(1982), 467–488; Quantum mechanical computers. *Foundations of Physics*, 16(1986), 507–531. (Originally published in *Optics News*, February 1985); Tiny Computers Obeying Quantum Mechanical Laws. In: *New Directions in Physics: The Los Alamos 40th Anniversary Volume*,, N. Metropolis and D. M. Kerr and G. Rota, Eds.,Academic Press,, Boston, 1987, 7–25.
18. R. Landauer. Irreversibility and heat generation in the computing process. *IBM J. Res. Develop.*, 5(1961), 183–191.
19. A.K. Lenstra and H.W. Lenstra, Jr. (Eds.), *The Development of the Number Field Sieve*, Lecture Notes in Mathematics, Vol. 1554, Springer-Verlag, Berlin, 1993.
20. M. Li and P.M.B. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications*, 2nd Edition, Springer-Verlag, New York, 1997.
21. Unruh, W. G., Maintaining coherence in quantum computers, *Physical Review A*, 51(1995), 992–.
22. L.M.K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, R. Cleve, I. L. Chuang, Experimental realization of order-finding with a quantum computer, http://xxx.lanl.gov/abs/quant-ph/0008065
23. P.M.B. Vitányi, Area penalty for sublinear signal propagation delay on chip, *Proc. 26th IEEE Symp. Foundat. Comput. Sci.*, 1985, 197–207.
24. P.M.B. Vitányi, Locality, communication and interconnect length in multicomputers, *SIAM J. Computing,* 17 (1988), 659–672.
25. P.M.B. Vitányi, Quantum Kolmogorov Complexity Based on Classical Descriptions, *IEEE Trans. Inform. Th.*, To appear.
26. C. Zalka, Efficient simulations of quantum mechanical systesms by quantum computers, *Proc. Royal Soc. London, Ser. A*, 454(1998), 313–322.
27. W.H. Zurek, Decoherence and the transition from quantum to classical, *Physics Today*, 44(1991), 36–44.