# Quantum Computation

David P. DiVincenzo

If the bits of computers are someday scaled down to the size of individual atoms, quantum mechanical effects may profoundly change the nature of computation itself. The wave function of such a quantum computer could consist of a superposition of many computations carried out simultaneously; this kind of parallelism could be exploited to make some important computational problems, like the prime factoring of large integers, tractable. However, building such a quantum computer would place undreamed of demands on the experimental realization of highly quantum-coherent systems; present-day experimental capabilities in atomic physics and other fields permit only the most rudimentary implementation of quantum computation.

Often in science, fruitful results come from combining two seemingly unrelated ideas into one. Here I discuss such a combination, quantum mechanics and computers, which together make for a new subject, quantum computers, which is beginning to define itself and explore a path, albeit a rough and rather long one, toward reality. The idea of a quantum computer is simple, even if its realization is not. In a properly functioning ordinary computer, all of the bits always have a definite state at any instant in time, say 011100101 ... In a quantum computer, however, we will say that the state of the bits can be described by a wave function, which might look like

$$\Psi = a|011100101\ldots\rangle + b|111010001\ldots\rangle + \ldots \quad (1)$$

The coefficients $a$, $b$, ... are complex numbers, and the probability that the computer is in the state 011100101 ... is $|a|^2$, that it is in the state 111010001 ... is $|b|^2$, and so on. However, describing the state of the computer by a wave function does not merely imply the ordinary uncertainties of life that we use probabilities to describe. For instance, the phases of the complex coefficients $a$, $b$, ... have genuine significance: These coefficients can describe interference among different states of the computer, a very useful process for computation, as it turns out. The quantum wave function declares that the computer exists in all of its states simultaneously so long as that state is not measured; when we do choose to measure it, a particular state will be observed with the prescribed probability.

No computer now is very well described by such a wave function; our present-day machines accurately obey the laws of classical physics. But if someday the bits of a computer are shrunk to atomic scale, then a quantum description of the bit state and the dynamics of a computer may become plausible. Feynman considered this possibility in 1985 (*1*) and concluded optimistically, "it seems that the laws of physics present no barrier to reducing the size of computers until the bits are the size of atoms, and quantum behavior holds dominant sway." In this article, I will first discuss the main basis of Feynman's optimism, which is that the analog of computer "gates" can be implemented within the realm of some very well understood (but difficult) experimental physics. Then I will go on to discuss what Feynman did not know, that by cleverly using quantum dynamics to design computations that interfere constructively or destructively, remarkably powerful computations like Shor's prime factoring algorithm (*2*) become possible. The seeds of this idea also appeared in 1985, in a paper by Deutsch (*3*). Deutsch realized then that quantum mechanics strikes down one of the most cherished principles of theoretical computer science, that of a unique computational complexity for every mathematical problem. Going back to the work of Turing (*4*), it was believed that the answer to the question of whether any given problem could be solved in a time that was polynomial in the size of its inputs, or greater than polynomial, was independent of the physical apparatus used to perform the computations. This indeed seems to be true for all computers operating on the principles of classical physics, but quantum computers can solve in polynomial time problems that have no polynomial-time solution on any classical machine.

The author is with the IBM Research Division, Thomas J. Watson Research Center, Post Office Box 218, Yorktown Heights, NY 10598, USA.

## Building Blocks of Quantum Logic

In this section I offer a bottom-up view of how a quantum computation might be reduced to practice, emphasizing that, at least in its first few steps, the required operations correspond to very well known procedures in experimental physics. At the very base of this construction is the qubit (or quantum bit) (*5*), a quantum system that, like an ordinary computer bit, has two accessible states but can, unlike an ordinary computer bit, exist in any superposition of those two states. Many two-state systems are known in physics, but throughout this article I will use as an example of this the spin-up (labeled $|1\rangle$) and spin-down (labeled $|0\rangle$) states of a spin-½ elementary particle like an electron or a proton. As in Boolean logic, we will build up operations in quantum logic using a small collection of logic gates, in which the states of input qubits (one or two qubits in the examples given below) are transformed in a specified fashion, leaving the qubits in a particular output state. In accordance with the laws of the quantum mechanics of isolated systems, we will take the allowable transformations to be unitary operations describing the time evolution of the input quantum state.

As an example, the quantum analog of the one-bit NOT or inverter gate can be implemented with spectroscopic techniques that have been well known in physics for over 50 years. As almost any elementary textbook of quantum mechanics shows (*6*), the time evolution of a spin-½ state can be accurately controlled by the judicious application of time-dependent magnetic fields. An inversion of the state, in which spin-up evolves to spin-down and vice versa, is accomplished by what is known as a tipping pulse. Suppose that we have an isolated spin in the presence of a combination of a stationary and a time-dependent magnetic field, described by the Hamiltonian

$$\mathbf{H} = \frac{1}{2} g\mu[H_0\sigma_z + H_1\sigma_y P(t)\sin(\omega t)] \quad (2)$$

Here, $g\mu$ is the magnetic dipole moment of the particle ($\mu = e\hbar/mc$, in centimeter-gram-second units, where $e$ is the electron charge, $\hbar$ is Planck's constant divided by $2\pi$, $m$ is the particle mass, and $c$ is the speed of light), the static magnetic field $H_0$ is along the $z$ axis, and the ac magnetic field pulse with amplitude $H_1$ is along the $y$ axis; $\sigma_y$ and $\sigma_z$ are the Pauli spin matrices, and $P(t)$ is the pulse envelope function, shown as a square pulse in Fig. 1. The time ($t$) evolution under this Hamiltonian is discussed fully in many places [for example, (*6*)]. During a tipping pulse, the ac field is in resonance with the energy difference between the two spin states: $\hbar\omega = g\mu H_0$. Under this condition, the $2 \times 2$ unitary matrix describing the time evolution of the spin in the spin-up–spin-down

basis, from the beginning $t = 0$ to the end $t = T$ of the pulse, simply has the form of a two-dimensional rotation matrix (except for phase factors)

$$U = \begin{pmatrix} e^{i\omega T/2} & 0 \\ 0 & e^{-i\omega T/2} \end{pmatrix} \cdot \begin{pmatrix} \cos\Omega T/2 & -\sin\Omega T/2 \\ \sin\Omega T/2 & \cos\Omega T/2 \end{pmatrix} \quad (3)$$

Here, $\Omega = g\mu H_1/4\hbar$ is the Rabi frequency; because both $\Omega$ and $T$ are at the disposal of the experimentalist conducting the tipping-pulse procedure, any angle of rotation may be obtained. For a 180° tipping pulse, when $\Omega T = \pi$, this time evolution accomplishes the NOT operation: If the system is initially in the $|0\rangle$ state, it ends up in the $|1\rangle$ state, and vice versa. Of course this classical operation has the nonclassical feature that there are definite phase factors associated with the time evolution. They can in general be chosen to be unity, although because usually $\omega \gg \Omega$, setting these phases is probably the most difficult feature of the tipping-pulse unitary transformation to control accurately.

There is nothing special in this spin-resonance operation about the tipping angle $\pi$; a whole continuous (three-parameter) family of operations, corresponding to any SU(2) matrix (*7*), can be performed. It is this generalization that is the essence of quantum computing and gives it its great potential power.

For a coupled two-spin system, there is a similar spin-resonance protocol (*8–10*), familiar to the physics of double resonance, which can perform the exclusive-or (XOR) function (*11, 12*). The XOR of two bits is simply the sum of their two Boolean values, modulo 2. The only new ingredient that is needed to accomplish the XOR by spin-resonance techniques is a nonzero Hamiltonian coupling together the two spins. The protocol is easiest to explain if this coupling has the form of an Ising interaction (*9*), so that the Hamiltonian takes the form

$$H = \frac{1}{2} g_a \mu H_0 \sigma_{az} + \frac{1}{2} g_b \mu H_0 \sigma_{bz} + J\sigma_{az}\sigma_{bz} + \mathcal{H}(t) \quad (4)$$

although an XOR protocol can be constructed no matter what the form of the coupling term between the two spins *a* and *b*. Here, $\mathcal{H}(t)$ is the time-dependent Hamiltonian to be prescribed by a tipping-pulse protocol. Without the application of these tipping pulses, this Hamiltonian simply describes a stationary quantum system with exactly four energy eigenstates (Fig. 2A). Because of the spin-spin interaction, the energy spacing between every pair of levels in this four-level spectrum will generically be distinct. This permits a tipping-pulse protocol in which specific individual resonances can be selected. Thus, if a pulse is applied at time $t_1$ whose ac frequency is tuned to $\omega_1$ [the energy spacing between the first and third energy levels in this spectrum (Fig. 2A)] and the tipping angle is chosen again to be $\pi$, then at the end of the pulse at time $t_2$, the desired XOR will be complete. That is, by flipping the state of the *a* spin if the *b* spin is $|1\rangle$, and doing nothing otherwise, this pulse leaves the final state of spin *a* in the XOR of the initial states of *a* and *b*, while leaving *b* in its original state, as summarized by the first two columns of the truth table in Fig. 2C. A gate symbol for this XOR operation is shown in Fig. 2D.

The XOR protocol is very closely related to procedures invented long ago in the field of resonance spectroscopies (*13*). In 1956, Feher introduced a procedure for polarization transfer in electron-nucleus double resonance (ENDOR), which contains the XOR protocol just discussed. In Feher's initial experiments, the *a* spin was carried by the outermost unpaired electron of a P dopant in crystalline Si, and the *b* spin was carried by a nearby $^{29}$Si nucleus (hence the name of the technique). The ENDOR and XOR protocols differ only in that Feher's procedure used a second $\pi$ pulse applied at time $t_2$ at a different frequency $\omega_2$ resonant with the transition between the first and second energy levels in the spectrum in Fig. 2A. At the end of the second pulse, at time $t_3$, the ENDOR operation is complete. The truth table for the ENDOR protocol is the first and third columns of Fig. 2C; like the one-pulse protocol, it leaves the *a* spin (the P electron spin in Feher's experiment) in the XOR of the initial states of *a* and *b*. In addition, it leaves *b* in the initial state of *a*, which is the polarization transfer that was of interest to Feher; for many purposes in physics, chemistry, and biology, it is highly desirable to move the spin state of an electron onto a nearby nucleus. The fact that this procedure also performs an interesting logical function, XOR, was not previously noted by ENDOR spectroscopists.

In either the one- or two-qubit gates, high-precision methods from experimental physics are required. It is necessary that the timing of the tipping pulses be precisely controlled, in order that the accumulated phase $\omega T$ be precisely zero (or some other chosen value). For the two-qubit operations, it is also necessary that the interaction Hamiltonian that determines the energy level splittings in the four-level spectrum be precisely known and controlled. In addition, the frequency content of the $\pi$ pulses should be tailored in such a way that a pulse that nominally has ac frequency $\omega_1$ has no small residual undesirable component at $\omega_2$. This requires a careful choice of the pulse shape (in general, the square-pulse form in Fig. 1A would be undesirable). Many of these issues, especially those of pulse shaping and frequency stability, have been considered extensively in the science of magnetic resonance (*14*).

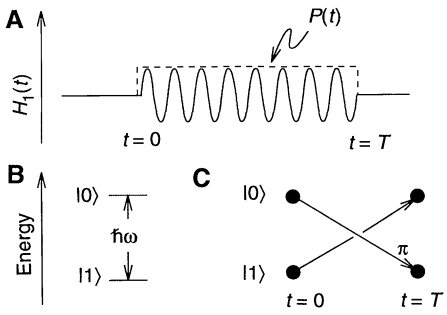


**Fig. 1.** The action of the NOT or inverter gate. The Hamiltonian describing the magnetic-resonance manipulation that results in the NOT operation is $\mathbf{H} = g\mu[H_0\sigma_z + H_1(t)\sigma_y]$. (**A**) The time dependence of the magnetic field of the tipping pulse, in this example a sinusoid at frequency $\omega$ multiplied by a square function $P(t)$ going from time $t = 0$ to $t = T$. (**B**) Energy level diagram for the qubit. The tipping pulse is tuned to be in resonance with the energy gap between the two stationary energy eigenstates $|0\rangle$ and $|1\rangle$. (**C**) State evolution diagram, showing the evolution paths of the two computational basis states. The $\pi$ in this diagram denotes that on the path indicated, the state acquires a 180° phase shift (assuming the parameters are chosen such that $\omega T = 0$ and $\Omega T = \pi$).

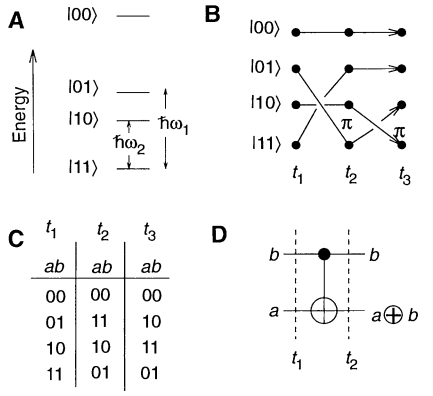


**Fig. 2.** The action of the two-qubit XOR gate. (**A**) Energy level diagram for the two qubits, showing the four stationary states of the Hamiltonian in Eq. 4. The states are labeled by the two qubit values of the two spins $|ab\rangle$. (**B**) The time evolution pathways of the quantum states under the action of the tipping-pulse protocol described in the text. Again, the $\pi$'s denote 180° phase shifts along the indicated pathways. (**C**) The truth table summarizing the result of the time evolution of the gate from the initial state (time $t_1$) to after the first (time $t_2$) and second (time $t_3$) tipping pulses. (**D**) The gate notation used for the XOR operation, obtained by using just the first of the two pulses of the ENDOR protocol. The resulting gate leaves qubit *b* unchanged and leaves *a* in the state given by the sum of *a* and *b*, modulo 2.

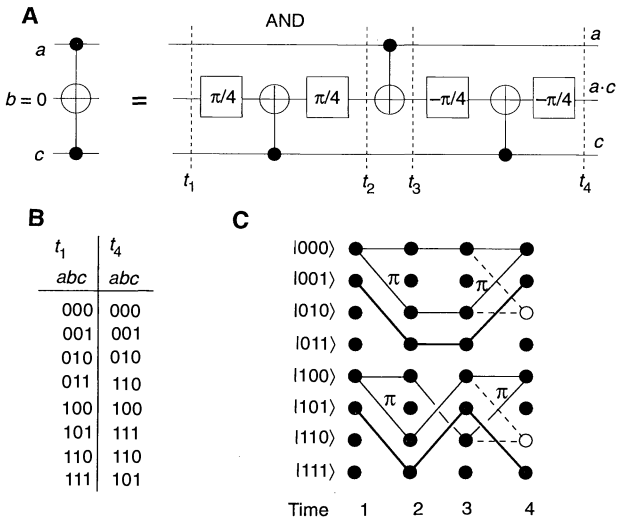


**Fig. 3.** Construction of the AND gate. (**A**) A notation for the three-qubit AND operation, and a gate construction of AND using three XOR gates and four single-qubit rotations. The $\pi/4$ gate corresponds to the operation in Eq. 3, with $\omega T = 0$ and $\Omega T = \pi/4$. When the work qubit $b$ is initially set to $|0\rangle$, it ends up in the state $|a{\cdot}c\rangle$. (**B**) The full truth table of the three-qubit AND gate. (**C**) The state evolution diagram for the AND gate, showing the intermediate state along selected pathways at the times shown in (A). A new feature appears here: For some input states, the intermediate state is a superposition of two different computational pathways. The final state is definite again because constructive interference permits only one of the possible outcomes (the pathways that interfere destructively at the last step are dashed).

## Quantum Circuits

Virtually any unitary operations on sets of qubits can be thought of as the universal gates of quantum computation (*15, 16*). What this means is that any unitary transformation in the $2^n$-dimensional Hilbert space spanned by $n$ qubits can be decomposed exactly (*12*) into a set of these universal operations applied in sequence to the $n$ qubits. The two operations introduced above, one-bit rotations and the two-bit XOR, possess this universal property (*12*). Thus, even though it is beyond present-day experimental capabilities, we could build up any quantum computation (which includes all ordinary Boolean computations, and more) by applying these basic operations in sequence to selected qubits or pairs of qubits to build up a "circuit" of arbitrary complexity.

As an example of the use of this repertoire to efficiently construct a useful quantum computation, the construction of an AND gate is shown in Fig. 3 (*12, 17*). It involves three bits because the input bits $a$ and $c$ are left unchanged during the operation; the work bit $b$ is set to $|0\rangle$ initially and is left in the state ($a$ AND $c$) at the end. (The AND is the product of the two bit values.) It is well known in "reversible" logic (*18, 19*) that it is necessary to introduce a work bit because the AND operation by itself is irreversible; the same is true in quantum computing because all unitary operations are reversible (that is, have an inverse). The AND gate in Fig. 3A requires three XOR gates, in each case with the result placed in the $b$ bit, along with four one-bit gates, all of which are just $\pm 45°$ tipping pulses. This particular implementation of the AND has phase factors that are all unity except for one: The state $|110\rangle$ is transformed to the state $-|110\rangle$. In many cases this change of phase may be acceptable for the operation of the gate (for example, if it is known that the input qubit $b$ will always be set to $|0\rangle$). If it is necessary that all the phase factors be unity, then the implementation is somewhat more complicated, requiring six XORs and eight one-bit gates (*12*).

Diagrams such as Fig. 3A give a deceptively simple impression of the ease with which elementary quantum-mechanical manipulations might be assembled to perform a quantum computation. In the implementation of the AND gate, it is implied that we know how to "wire up" three XORs and a number of other gates. But consider what this "wiring up" means: While the XOR connecting qubits $b$ and $c$ is in operation, spins $b$ and $c$ should have some prescribed interaction (say, the Ising coupling of Eq. 4), whereas the couplings between $a$ and $b$ and between $a$ and $c$ should be zero. When the second XOR is in operation, the microscopic couplings should be rearranged, with the $a$-$b$ coupling being nonzero. This is not a commonplace happening, and it certainly was not envisioned by Feher or any magnetic resonance experimentalists in the 1950s.

This "interconnection" problem can probably be solved, but it is one whose solution involves the most speculative and uncertain features of the quantum computer implementations suggested to date. The gedanken apparatus in Fig. 4 shows a possible future device that might solve the interconnection problem for a quantum computer. It depicts the tip of a specifically designed atomic force microscope (AFM) (*20*) approaching the surface of a crystal from above. It is imagined that both the tip and the surface are constructed with the following criteria in mind: (i) The spins of the H-atom nuclei, one of them placed at the very end of the tip and the others placed periodically on the surface, serve as the qubits. (ii) All the electrons are tied up in bonds, both in the bulk (crystalline Si is an insulator) and on the surface. This is done so that flipping an electron spin, or transporting it, is not an available quantum degree of freedom because such excitations require too much energy. (iii) Likewise, all other nuclei in the system have a spin of zero, so that only the H-atom proton spins are available for interaction. (I will say more at the end of this article about the undesirable consequences of stray quantum degrees of freedom in a quantum computer, principally concerning their role in the loss of quantum phase coherence.) (iv) Like any AFM, this one should be capable of moving the atom at the tip into contact with any atom on the surface at will. This property is

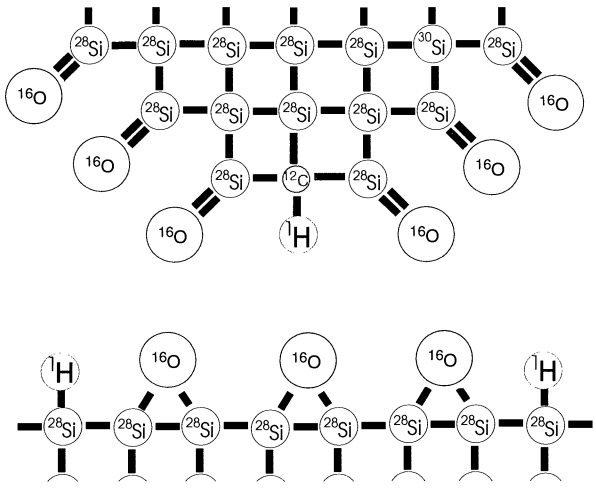


**Fig. 4.** Cartoon illustrating the kind of atomic-scale engineering that would be required to implement quantum computation with an AFM. It is imagined that an undoped crystalline Si tip is approaching a crystalline Si surface. The qubits are carried by the proton spin of the H atom at the very end of the tip and the H atoms arranged periodically along the surface. Interactions between the tip qubit and the other qubits can be turned on and off by the physical approach of the tip to various sites on the surface, permitting a gate protocol like the one of Fig. 3A to be carried out. By arranging for all the surface dangling bonds to be saturated, one can eliminate undesirable qubits carried by stray electron spins. Stray qubits carried by nuclear spins are likewise avoided by permitting only spin-zero isotopes in the vicinity of the H atoms. The tip qubit can be made spectroscopically distinct by bonding it to a different atom producing a chemical shift, which can be useful in devising selective magnetic resonance protocols.

the one that accomplishes the desired interconnection action: When the first XOR in Fig. 3A is to be performed, the microscope tip should be parked in contact with the H atom to the right on the surface in Fig. 4; for the second XOR, it should be parked in contact with the H atom on the left, and so on. Present-day AFMs cannot yet satisfy all of these design criteria. However, incredible strides have been made in the last few years in using AFMs to do spin-resonance manipulations and measurements on small groups of spins (*20, 21*).

## Peter Shor's Prime Factorization

The AND gate constructed in Fig. 3A performs a classical Boolean operation. Still, its implementation is very nonclassical, in a way that is a prototype for the really powerful procedures (for prime factorization and the like) that are unique to quantum computation. Figure 3C shows how the various input states of the AND construction of Fig. 3A evolve in time through three of its stages to the final answer (we only follow the computations for which the work qubit $b$ is set to $|0\rangle$). In any classical computation, each of these computations would follow a single, definite pathway in time from the beginning of the computation to the end, but in quantum computation, the computation can be split up into several (in this case just two) pathways that, by the quantum-mechanical principle of superposition, evolve in time in parallel. Because these pathways carry definite phases (note in the figure the points at which a 180° phase shift is introduced by the time evolution), these computations can, upon recombining at the end, interfere either constructively or destructively to produce definite outcomes. In the example shown, $|000\rangle$ evolves in time to itself, as required by the AND truth table, because the two computational pathways arriving at $|000\rangle$ at the end have the same phase (0 along one path, $2\pi$ along the other) and so interfere constructively. The incorrect outcome, $|010\rangle$, which the computational pathways also reach, is prevented because the phases are opposite (0 and $\pi$) and the interference is destructive. If the phases were not carefully controlled, then the incorrect outcome would occur half the time. It is by this means that a quantum computation, despite being in a very complex, indeterminate computational state in its intermediate stages, can be in a definite, computationally useful state at the end.

It is this general schema for quantum computation, first introduced by Deutsch (*3*), that is used to great effect in Shor's algorithm (*2, 4*) for efficiently solving a large-scale computational problem. Figure 5 illustrates a kind of architecture of the quantum computation that Shor uses to perform prime factorization. This diagram is a generalization of the one used to illustrate the operation of the AND in Fig. 3A, in that it shows the evolution pathways of the computational states as a function of time though the factoring procedure (time runs downward now, rather than to the right). As Deutsch and Jozsa envisioned in earlier work (*22*) [see also (*23*)], Shor divides the qubits of the computer into two registers labeled (somewhat misleadingly) input and output. The number of bits in each register needs to be of order the number of bits in the integer to be factored; for argument's sake (to be very ambitious), suppose that both registers contain $k = 1000$ bits. The rectangles in Fig. 5 depict the members of the entire Hilbert space of the input and output bits. This diagram is of necessity highly schematic because the dimension of this Hilbert space is huge: $2^{1000}$ for both the input and output registers. In fact, it is this exponential scaling of the size of the Hilbert space with respect to the number of particles in the system that is one of the reasons for the great potential power of quantum computing; unitary matrices can easily be constructed and multiplied on an ordinary digital computer, but their size cannot be exponential in the number of components of that computer (*24*).

The shading in Fig. 5 indicates the instantaneous state vector throughout the three main stages of Shor's computation. The first few steps are very simple: The starting state is fixed to be all zeros (all spins down). (The classical input, that is, the integer $N$ to be factored, does not enter the procedure yet.) In stage 1, the computation is split up into $2^{1000}$ pathways, so that the wave function of the system becomes a linear superposition of all possible states, with equal phases, of the input register $x$. This highly nonclassical computation is very easy to prescribe spectroscopically: A 90° tipping pulse applied to each input spin places the wave function of the system in the desired state.

Stage 2 of computation is less trivial, requiring a single evaluation of a classical Boolean function

$$f(x) = c^x (\mathrm{mod}\ N) \qquad (5)$$

The value of this function is placed in the output register y. Here $x$ is the value of the input register considered as an integer in binary representation, $N$ is the integer to be factored, and the constant $c$ is any other integer that has no prime factors in common with $N$; mod $N$ indicates modular arithmetic, in which the result is the remainder after division by $N$. Because of the superposition principle, a single evaluation of $f(x)$ obtains every value of the output, given that the input is a superposition of all possible values. To evaluate $f(x)$ on a quantum computer, it should first be "compiled," in the usual classical sense, so that $f(x)$ is written as a sequence of operations of primitive Boolean functions like NOT and AND. Then one would implement this sequence of NOTs and ANDs as a quantum procedure, for example, in a sequence of magnetic-resonance tipping pulses.

I will not give a complete explanation of why the parallel evaluation of this particular $f(x)$ is useful for prime factorization. This requires some straightforward technicalities of number theory; good discussions may be found in the original literature (*2*) and in some recent reviews (*4*). In a nutshell, the important property of $f(x)$ is its periodicity with respect to $x$. If $N$ is a prime number, then the period of $f(x)$ is $N - 1$, but if $N$ is composite, the period of $f(x)$ is shorter, and knowledge of this period leads, after a straightforward (classical) calculation, to one of the prime factors of $N$.

Shor noted that a quantum computer is very well adapted to finding the periodicity

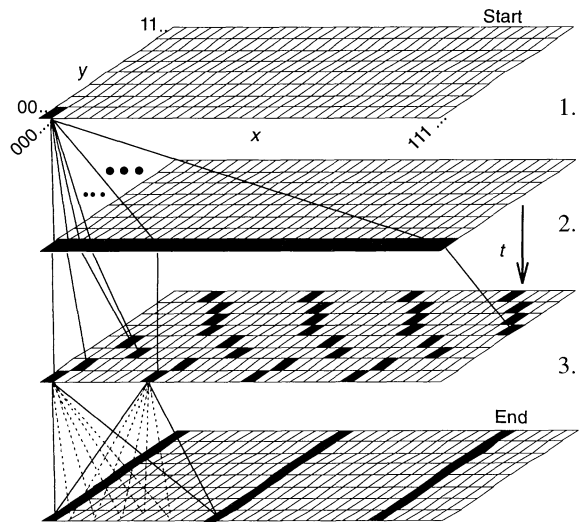


**Fig. 5.** A schematic depiction of the time evolution pathways in Shor's prime factoring procedure. The computational states appearing in the wave function at each selected instant in time are indicated by the filled rectangles. A few of the pathways are sketched out. Most of the pathways in the final step (dotted lines) interfere destructively, with only a few (solid lines) interfering constructively.

of $f(x)$, by means of the execution of a Fourier transform on the input register $x$ (not the output register $y$); this is the third and final stage of computation depicted in Fig. 5. To be precise, the Fourier transform takes a wave function of the form

$$\Psi_i = \sum_{x=00\ldots0}^{11\ldots1} c_x |x\rangle \quad (6)$$

and evolves it in time so that it ends up as

$$\Psi_f = \sum_{x=00\ldots0}^{11\ldots1} \left( 2^{-k/2} \sum_{x'=00\ldots0}^{11\ldots1} e^{2\pi i x x'/2^k} c_{x'} \right) |x\rangle \quad (7)$$

or in words, the final wave function coefficients are the discrete Fourier transform of their initial values. Shor observed that this transformation is a unitary operation and showed that it could be performed in a number of steps polynomial in $k$, the number of bits in the input register (which is in turn of order the number of bits needed to represent $N$, the number to be factored). Coppersmith (*25*) found a simple, explicit, robust gate construction (Fig. 6) for implementing the radix-2 Fourier transform of Eq. 7. It is a straightforward transcription of the steps involved in performing the Cooley-Tukey fast Fourier transform (FFT), with the individual "twiddle factors" of the FFT implemented by the two-qubit $X_i$ gates shown. This procedure is very similar to the first step of Shor's computation, which just consists of the 90° tipping pulses applied to each bit in turn, without the twiddle-factor gates. Coppersmith noted that this sequence of operations takes on the order of $k^2$ steps.

This final FFT step is a very efficient way of obtaining the period of $f(x)$, in the same way that the scattering of x-rays from a crystal is a good way of obtaining its periodicity; a final measurement of the value of the register $x$ is a measurement of the position of one of the "Bragg peaks" of this scattering process [although this measurement only obtains some unknown multiple of the fundamental period of $f(x)$, there are again some straightforward number-theoretic considerations that permit the fundamental period itself to be deduced reliably from this measurement]. Shor's procedure manages to be useful in the same way as the AND gate implementation; despite being in an indefinite, superposed computational state through the middle parts of the computation, destructive interference forbids almost all possible outcomes (just as in Bragg scattering, diffraction into almost every direction is forbidden), leaving the system in an (almost) definite, and thus computationally useful, state. Note that the diffraction process described here differs in a crucial respect from the diffraction of classical waves: the size of the diffraction grating used in Fig. 5 grows as an exponential of the size of the number to be factored. It is for this reason that classical wave optics cannot efficiently solve this problem.

A final tally of the number of steps required to perform Shor factoring reveals why this result has caused a stir in the computer science community. The result is polynomial in $k$, going as $k^3$ for small $k$ and asymptotically approaching $k^2$ for large $k$. After many decades of effort, the best algorithms for factoring on an ordinary Boolean computer are nonpolynomial, scaling like $\exp(ak^{1/3})$, where $a$ is some constant (*4*, *26*). Although it is not known whether this particular problem might ultimately yield to a polynomial-time solution on an ordinary computer, Shor's result has made computer scientists realize that the algorithmic approaches available on a quantum computer are much more powerful than ordinary Boolean logic, and active work is under way in more fully defining the power of quantum mechanics to solve important mathematical problems.

## The Decoherence Problem

Even though the formal results on the great capabilities of quantum computation are perfectly in accord with the laws of quantum physics, there are still several very fundamental physical obstacles that need to be overcome before quantum computation can be performed in the laboratory. These obstacles will make the path to constructing a quantum computer a long and arduous one, and one that will not be traversed before many years have elapsed. Two principal obstacles have been identified: the error correction problem and the decoherence problem. I will not discuss error correction, which may ultimately be very difficult in quantum computing because it seems that slight imperfections in the implementation of $\pi$ pulses and other elements would ultimately lead a calculation off track (*27*). Other authors have explored the difficulties here (*28*, *29*) and are beginning to define the rudiments of an error correction scheme (*30*). For the Shor algorithm, small errors can be defeated simply by running the computation repeatedly until the correct answer is obtained—prime factors can easily be confirmed by multiplication. (Actually, Shor's algorithm, like many other useful ones, is not guaranteed to give a correct answer in one run, even in the absence of errors.)

It seems to be the decoherence problem that makes even the initial investigation of modest-sized quantum computations difficult. Decoherence is this: If the quantum system is not perfectly isolated from its environment, the quantum dynamics of the surrounding apparatus will also be relevant to the operation of the quantum computer, and its effect will be to make the computer's evolution nonunitary. Because computational pathways separated at the beginning of the computation only recombine at the very end (Fig. 5), loss of phase coherence along these paths will spoil the constructive and destructive interference that is essential for quantum computing; therefore, the decoherence time $t_\Phi$ needs to be much longer than the expected running time of the computation. Fortunately, the decoherence problem is one for which continuing advancement in the experimental art is likely to make a difference. Improving the isolation between the quantum system and its environment, which accompanies the steady advance of the technology used in high-precision quantum physics experiments, results in a lengthening of $t_\Phi$ and a growing possibility to perform useful quantum computations.

Table 1 gives a survey of the current state of the art for $t_\Phi$'s in a wide variety of two-state quantum systems (*16*). Because of the great disparity of energy scales, the available speeds range over 16 orders of magnitude. There is also a great disparity in the present technological capability for applying tipping pulses to each of these systems; the gamma-ray spectroscopy that would be needed for manipulating the Mössbauer nucleus does not exist, whereas the high-precision radio-frequency technol-

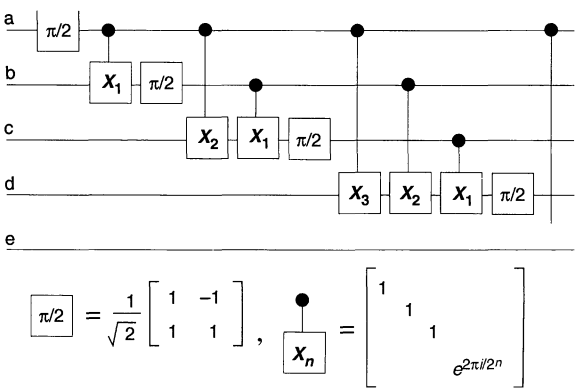


**Fig. 6.** The gate array introduced by Coppersmith (*25*) for performing the Fourier transform (step 3 of the Shor procedure in Fig. 5). The matrix unitary operators corresponding to the two types of quantum gates used in the figure are shown. The two-qubit $X_n$ gate may be implemented by a simple combination of XORs and one-qubit gates (*12*). The $X_n$ gate acts symmetrically on its two qubits. The process can be extended for inputs beyond a through e.

ogy for doing tipping pulses in nuclear magnetic resonance (NMR) is very mature. Also, the technology may not allow the potential speed of any given qubit to be fully utilized: For example, in the recent proposal for implementing quantum computation with a linear ion trap (*31*), the switching time would be $10^{-5}$ s rather than $10^{-14}$ s because it is limited by the qubit encoded in the quantized vibrations of the ions in the trap.

There is also a great disparity in the decoherence times available in these systems (Table 1). The ratio of the switching times to the decoherence times is an important figure of merit for quantum computation, being the number of steps of computation that might be performed before phase coherence is spoilt. Unruh's (*32*) calculations indicate that to perform computations like Shor factoring, this figure of merit should be something like the cube of the number of bits in the integer to be factored. If the object is to factor a $10^4$-bit number (a task believed to be beyond the capability of any conceivable classical computer), it is evident that most of the present-day qubits are inadequately phase coherent to do the job. Nevertheless, several experiments have now been done in which actual, rudimentary quantum logic gates have been constructed, one involving optical microcavities (*33*) and another using trapped ions (*34*). The technology used in this last example is the one that is under development for the next generation of atomic clocks (*35*). This is significant because quantum computers require long dephasing times in addition to long decoherence times. Dephasing, loss of the accuracy of the phase factors of Eq. 3 because of the drift of the clock, must also be kept very small in quantum computation.

**Table 1.** Important times for various two-level systems in quantum mechanics that might be used as quantum bits, including prospective qubits ranging from nuclear physics, through atomic, electronic, and photonic systems, to electron and nuclear spins. The time $t_{switch}$ is the minimum time required to execute one quantum gate; it is estimated as $\hbar/\Delta E$, where $\Delta E$ is the typical energy splitting in the two-level system; the duration of a $\pi$ tipping pulse cannot be shorter than this uncertainty time for each system. The phase coherence time as seen experimentally, $t_\phi$, is the upper bound on the length of time over which a complete quantum computation can be executed accurately. The ratio of these two times gives the largest number of steps permitted in a quantum computation using these quantum bits. See (*16*) for the original references.

| Quantum system | $t_{switch}$ (s) | $t_\phi$ (s) | Ratio |
|---|---|---|---|
| Mössbauer nucleus | $10^{-19}$ | $10^{-10}$ | $10^9$ |
| Electrons: GaAs | $10^{-13}$ | $10^{-10}$ | $10^3$ |
| Electrons: Au | $10^{-14}$ | $10^{-8}$ | $10^6$ |
| Trapped ions: In | $10^{-14}$ | $10^{-1}$ | $10^{13}$ |
| Optical microcavity | $10^{-14}$ | $10^{-5}$ | $10^9$ |
| Electron spin | $10^{-7}$ | $10^{-3}$ | $10^4$ |
| Electron quantum dot | $10^{-6}$ | $10^{-3}$ | $10^3$ |
| Nuclear spin | $10^{-3}$ | $10^4$ | $10^7$ |

## Outlook

It is evident from this survey of the current state of the art in quantum experimental physics that the construction of quantum computers is presently in the most rudimentary stage, and that to even think about a procedure like Shor factorization, which might require millions of operations (*14*) on thousands of qubits, might be absurdly premature. However, even a much more modest quantum computer will permit the study of effects that are of great scientific interest. For example (*10*), even a few bits of quantum computation will be very useful in performing so-called Bell measurements, which could be used to implement quantum teleportation (*36*), in which an unknown quantum state can be transported to a remote location. At perhaps the 10-qubit level, a quantum computer becomes capable of performing Schumacher's quantum coding (*5*), which would be of interest in the implementation of efficient quantum cryptography (*37*). And at perhaps the 100-qubit level, a quantum computer becomes an efficient repeater for a noisy (that is, partially decohered) quantum cryptographic link (*38*). In this application, it might become possible to create Einstein-Podolsky-Rosen pairs (*36*) at very remote locations, permitting new, stringent tests of the validity of the quantum theory. At this time, both physicists and computer scientists are actively searching for new ways to use quantum computers.

The quantum-gate approach outlined here appears to be a very arduous one for the ultimate implementation of a quantum computer, but other paradigms, which might ultimately provide an easier path to implementation, are being explored. For example, it might be that the natural time evolution of some simple quantum system, like the quantum states of a crystal, might itself perform some useful computation; preliminary work on this sort of "quantum cellular automaton" has been done (*9*, *39*). Perhaps our present understanding that stringent isolation is a requirement for quantum computation is untrue; it may be that the time evolution of the density matrix of an open quantum system is also a powerful computational tool, or that new approaches to error correction (*30*) will let noisy qubits compute. In any case, the next few years should be interesting ones for the quantum computer.

## REFERENCES AND NOTES


1. R. P. Feynman, *Opt. News* **11**, 11 (February 1985).
2. P. W. Shor, in *Proceedings of the 35th Annual Symposium on the Foundations of Computer Science* (IEEE Computer Society, Los Alamitos, CA, 1994), p. 124 [this paper, "Algorithms for Quantum Computation: Discrete Log and Factoring," is available on the World Wide Web (*40*)].
3. D. Deutsch, *Proc. R. Soc. London Ser. A* **400**, 97 (1985).
4. For a review, see A. Ekert and R. Jozsa, in preparation; see (*41*); see also J. Brown, *New Sci.* **133** (no. 1944), 21 (1994).
5. The term "qubit" has been in circulation for the last several years: see B. Schumacher, *Phys. Rev. A* **51**, 2738 (1995) [available on the Web ("Quantum Coding") (*40*)].
6. G. Baym, *Lectures on Quantum Mechanics* (Benjamin-Cummings, Reading, MA, 1969), pp. 140 and 317–324.
7. See J. Mathews and R. L. Walker, *Mathematical Methods of Physics* (Benjamin, Menlo Park, CA, ed. 2, 1970), p. 464, for details of the SU(2) group.
8. A. Ekert, in *Atomic Physics 14: 14th International Conference on Atomic Physics, Boulder, CO, 1994* (AIP Conference Proceedings 323, AIP Press, New York, 1995), p. 450; see (*41*).
9. S. Lloyd, *Science* **261**, 1569 (1993); *ibid.* **263**, 695 (1994).
10. A. Barenco, D. Deutsch, A. Ekert, R. Jozsa, *Phys. Rev. Lett.* **74**, 4083 (1995) (available from the Los Alamos preprint archive as quant-ph/9503017); also see (*41*).
11. D. Deutsch, *Proc. R. Soc. London Ser. A* **425**, 73 (1989).
12. A. Barenco *et al.*, *Phys. Rev. A*, in press [available from Los Alamos preprint archive (quant-ph/9503016) and on the Web ("Elementary Gates for Quantum Computation") (*40*)]; T. Sleator and H. Weinfurter, *Phys. Rev. Lett.* **74**, 4087 (1995).
13. C. P. Slichter, *Principles of Magnetic Resonance* (Springer-Verlag, Berlin, ed. 3, 1992).
14. W. S. Warren and M. S. Silver, *Adv. Magn. Reson.* **12**, 247 (1988). These authors report the record pulse-sequence length in NMR to be 4096.
15. S. Lloyd, *Phys. Rev. Lett.* **75**, 346 (1995); D. Deutsch, A. Barenco, A. Ekert, *Proc. R. Soc. London Ser. A* **449**, 669 (1995) [available from Los Alamos preprint archive (quant-ph/9505018)]; see (*41*).
16. D. P. DiVincenzo, *Phys. Rev. A* **50**, 1015 (1995) [available from Los Alamos preprint archive (cond-mat/9407022) and on the Web ("Two-Bit Gates are Universal for Quantum Computation") (*40*)]. The sources of the numbers in the table are given here.
17. A related implementation of the so-called "Fredkin gate" (*19*) has been considered previously. See Y. Yamamoto, M. Kitegawa, K. Igeta, in *Proceedings of the 3rd Asia-Pacific Physics Conference* (World Scientific, Singapore, 1988); G. J. Milburn, *Phys. Rev. Lett.* **62**, 2124 (1989).
18. R. Landauer, *IBM J. Res. Dev.* **5**, 183 (1961); T. Toffoli, in *Automata, Languages and Programming*, J. W. de Bakker and J. van Leeuwen, Eds. (Springer, New York, 1980), p. 632.
19. E. Fredkin and T. Toffoli, *Int. J. Theor. Phys.* **21**, 219 (1982).
20. J. A. Sidles *et al.*, *Rev. Mod. Phys.* **67**, 249 (1995).
21. D. Rugar, C. S. Yannoni, J. A. Sidles, *Nature* **360**, 563 (1992); O. Züger and D. Rugar, *Appl. Phys. Lett.* **63**, 2496 (1993).
22. D. Deutsch and R. Jozsa, *Proc. R. Soc. London Ser. A* **439**, 554 (1992).
23. D. R. Simon, in (*2*), p. 116 [available on the Web ("On the Power of Quantum Computation") (*40*)].
24. This viewpoint on quantum computing was explored by R. P. Feynman [*Int. J. Theor. Phys.* **21**, 467 (1982)].
25. D. Coppersmith, *IBM Res. Rep. RC19642* (1994), and unpublished material; see also R. Cleve, unpublished material.
26. A. K. Lenstra, H. W. Lenstra Jr., M. S. Manasse, J. M. Pollard, in *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing* (ACM Press, New York, 1990), p. 564; an extended version appears in A. K. Lenstra and H. W. Lenstra Jr., Eds., *The Development of the Number Field Sieve*, vol. 1554 of *Lecture Notes in Mathematics* (Springer-Verlag, Berlin, 1993), p. 11.
27. However, see I. L. Chuang, R. Laflamme, P. Shor, W. H. Zurek, in preparation [available from Los Alamos

preprint archive (quant-ph/9503007) and on the Web ("Quantum Computers, Factoring, and Decoherence") (*40*)].
28. R. Landauer, in *Proceedings of the Drexel-4 Symposium on Quantum Nonintegrability—Quantum Classical Correspondence*, D. H. Feng and B.-L. Hu, Eds. (International Press, Boston, in press).
29. R. Landauer, *Philos. Trans. R. Soc. London Ser. A*, in press.
30. A. Berthiaume, D. Deutsch, R. Jozsa, *Proceedings of the Workshop on Physics and Computation, PhysComp '94* (IEEE Computer Society Press, Los Alamitos, CA, 1994), p. 60; A. Barenco, D. Deutsch, A. Ekert, C. Machiavello, unpublished material.
31. J. I. Cirac and P. Zoller, *Phys. Rev. Lett.* **74**, 4091 (1995).
32. W. G. Unruh, *Phys. Rev. A* **51**, 992 (1995) [available from Los Alamos preprint archive (hep-th/9406058)].
33. Q. A. Turchette, C. J. Hood, W. Lange, H. Mabuchi, H. J. Kimble, "Measurement of Conditional Phase Shifts for Quantum Logic," preprint (June 1995).
34. S. R. Jefferts, C. Monroe, E. W. Bell, D. J. Wineland, *Phys. Rev. A* **51**, 3112 (1995); C. Monroe, private communication.
35. R. S. Van Dyck Jr., in *Physics News in 1994*, April 1995 supplement of *APS News*, P. W. Schewe, Ed. (American Institute of Physics, College Park, MD, 1995), p. S6.
36. C. H. Bennett *et al.*, *Phys. Rev. Lett.* **70**, 1895 (1993).
37. C. H. Bennett, G. Brassard, A. Ekert, *Sci. Am.* **267** (no. 4), 50 (1992).
38. C. H. Bennett *et al.*, "Purification of Noisy Entanglement, and Faithful Teleportation via Noisy Channels," preprint (May 1995).
39. N. Margolus, in *Complexity, Entropy, and the Physics of Information*, vol. VIII of *Santa Fe Institute Studies in the Sciences of Complexity*, W. H. Zurek, Ed. (Addison-Wesley, Reading, MA, 1990), p. 273.
40. URL http://vesta.physics.ucla.edu/~smolin/(Quantum Information Page, Center for Advanced Accelerators).
41. URL http://eve.physics.ox.ac.uk/QChome.html (Quantum Computation and Cryptography page, Clarendon Laboratory, University of Oxford).
42. I am grateful to my colleagues N. Amer, C. H. Bennett, D. Coppersmith, N. Gershenfeld, R. Landauer, S. Lloyd, A. Peres, J. Smolin, and W. Wootters for many helpful discussions about this work. I thank the various audiences who heard me speak on this topic recently; they all helped me hone the story that I have written here.