

# Quantum computation: a tutorial



[Samuel L. Braunstein](#)

## Abstract:

*Imagine a computer whose memory is exponentially larger than its apparent physical size; a computer that can manipulate an exponential set of inputs simultaneously; a computer that computes in the twilight zone of Hilbert space. You would be thinking of a quantum computer. Relatively few and simple concepts from quantum mechanics are needed to make quantum computers a possibility. The subtlety has been in learning to manipulate these concepts. Is such a computer an inevitability or will it be too difficult to build?*

In this paper we give a tutorial on how quantum mechanics can be used to improve computation. Our challenge: solving an exponentially difficult problem for a conventional computer---that of factoring a large number. As a prelude, we review the standard tools of computation, universal gates and machines. These ideas are then applied first to classical, dissipationless computers and then to quantum computers. A schematic model of a quantum computer is described as well as some of the subtleties in its programming. The Shor algorithm [1,2] for efficiently factoring numbers on a quantum computer is presented in two parts: the quantum procedure within the algorithm and the classical algorithm that calls the quantum procedure. The mathematical structure in factoring which makes the Shor algorithm possible is discussed. We conclude with an outlook to the feasibility and prospects for quantum computation in the coming years.

Let us start by describing the problem at hand: factoring a number  $N$  into its prime factors (e.g., the number **51688** may be decomposed

$$2^3 \times 7 \times 13 \times 71$$

as  $2^3 \times 7 \times 13 \times 71$ ). A convenient way to quantify how quickly a particular algorithm may solve a problem is to ask how the number of steps to complete the algorithm scales with the size of the "input" the algorithm is fed. For the factoring problem, this input is just the number  $N$  we wish to factor;

$$\log N$$

hence the length of the input is  $\log N$ . (The base of the logarithm is determined by our numbering system. Thus a base of **2** gives the length in

binary; a base of **10** in decimal.) 'Reasonable' algorithms are ones which scale as some small-degree polynomial in the input size (with a degree of perhaps **2** or **3**).

On conventional computers the best known factoring algorithm runs

$$O(\exp[(64/9)^{1/3}(\ln N)^{1/3}(\ln \ln N)^{2/3}])$$

in

steps [3]. This

$$\log N$$

algorithm, therefore, scales exponentially with the input size. For instance, in 1994 a 129 digit number (known as RSA129 [3]) was successfully factored using this algorithm on approximately 1600 workstations scattered around the world; the entire factorization took eight months [4]. Using this to estimate the prefactor of the above exponential scaling, we find that it would take roughly 800,000 years to factor a 250 digit number with the same computer power; similarly, a 1000 digit number would require  $10^{25}$  years (significantly longer than the age of the universe). The difficulty of factoring large numbers is crucial for public-key cryptosystems, such as ones used by banks. There, such codes rely on the difficulty of factoring numbers with around 250 digits.

Recently, an algorithm was developed for factoring numbers on a quantum

$$O((\log N)^{2+\epsilon})$$

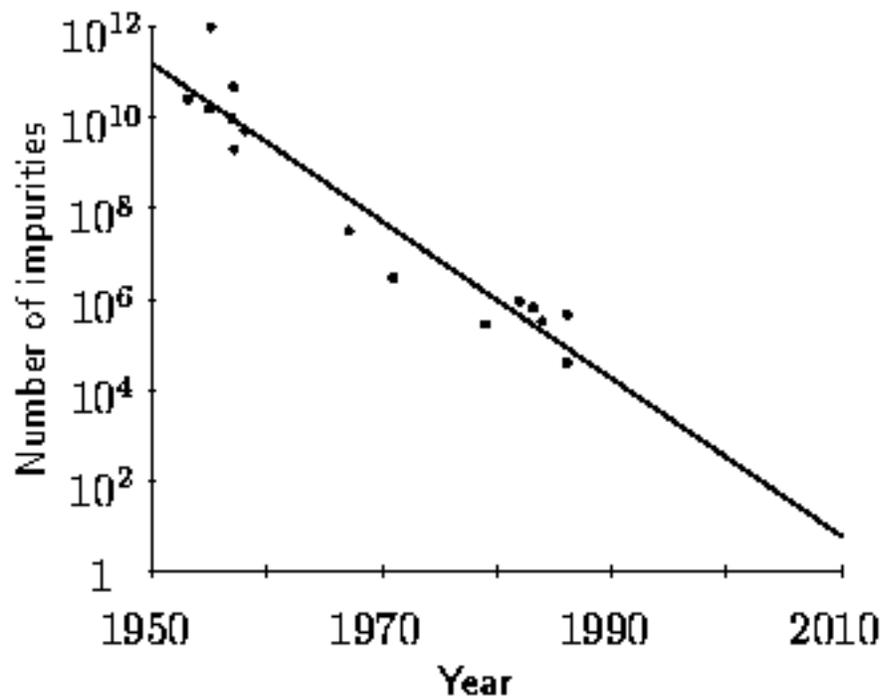
computer which runs in steps where  $\epsilon$  is small [1]. This is roughly quadratic in the input size, so factoring a **1000** digit number with such an algorithm would require only a few million steps. The implication is that public key cryptosystems based on factoring may be breakable.

To give you an idea of how this exponential improvement might be possible, we review an elementary quantum mechanical experiment that demonstrates where such power may lie hidden [5]. The two-slit experiment is prototypic for observing quantum mechanical behavior: A source emits photons, electrons or other particles that arrive at a pair of slits. These particles undergo unitary evolution and finally measurement. We see an interference pattern, with both slits open, which wholly vanishes if either slit is covered. In some sense, the particles pass through both slits in parallel. If such unitary evolution were to represent a calculation (or an operation within a calculation) then the quantum system would be performing computations in parallel. Quantum parallelism comes for free. The output of this system would be given by the constructive interference among the parallel computations.

---

## Computing at the atomic scale:

Quantum computers will perform computations at the atomic scale [5,6]. Fig. 1 shows a survey made by Keyes in 1988 [7]: The number of dopant impurities required for logic in the bases of bipolar transistors is plotted against the year. This plot may be thought of as showing the number of electrons required to store a single bit of information. An extrapolation of the plot suggests that we might be within the reach of atomic-scale computations within the next two decades.



**Fig. 1** Plot from Ref. [7] showing the number of dopant impurities involved in logic in bipolar transistors with year. (Copyright 1988 by International Business Machines Corporation, reprinted with permission.)

# Reversible computation:

What are the difficulties in trying to build a classical computing machine on such a small scale? One of the biggest problems with the program of miniaturizing conventional computers is the difficulty of dissipated heat. As early as 1961 Landauer studied the physical limitations placed on computation from dissipation [8]. Surprisingly, he was able to show that almost all operations required in computation could be performed in a reversible manner, thus dissipating no heat! The first condition for any deterministic device to be reversible is that its input and output be uniquely retrievable from each other. This is called logical reversibility. If, in addition to being logically reversible, a device can actually run backwards then it is called physically reversible and the second law of thermodynamics guarantees that it dissipates no heat.

The work on classical, reversible computation has laid the foundation for the development of quantum mechanical computers. On a quantum computer, programs are executed by unitary evolution of an input that is given by the state of the system. Since all unitary operators  $U$  are invertible with  $U^{-1} = U^\dagger$ , we can always "uncompute" (reverse the computation) on a quantum computer.

---

5.1.3

## Classical universal machines and logic gates:

We now review the basic logic elements used in computation and explain how conventional computers may be used for any "reasonable" computation. A reasonable computation is one that may be written in terms of some (possibly large) Boolean expression, and any Boolean expression may be constructed out of a fixed set of logic gates. Such a set (e.g., AND, OR and NOT) is called universal. In fact we can get by with only two gates, such as AND and NOT or OR and NOT. Alternatively, we may replace some of these primitive gates by others, such as the exclusive-OR (called XOR); then AND and XOR form a universal set. The truth tables for these gates are displayed in Table 1. Any

machine which can build up arbitrary combinations of logic gates from a universal set is then a universal computer.

$A$	$B$	AND	OR	XOR	NOT $B$
0	0	0	0	0	1
0	1	0	1	1	0
1	0	0	1	1	1
1	1	1	1	0	0

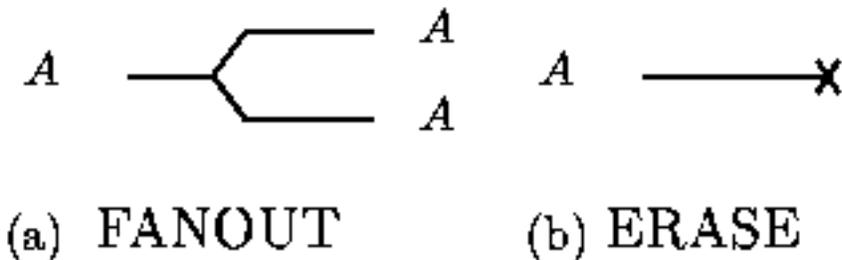
**Table 1** Truth table defining the operation of some elementary logic gates. Each row shows two input values  $A$  and  $B$  and the corresponding output values for gates AND, OR and XOR. The output for the NOT gate is shown only for input  $B$ .

Which of the above gates is reversible? Since AND, OR, and XOR are many-to-one operations they are not, as they stand, logically reversible. Before we discuss how these logic gates may be made reversible we consider some non-standard gates that we shall require.

#### 5.1.4

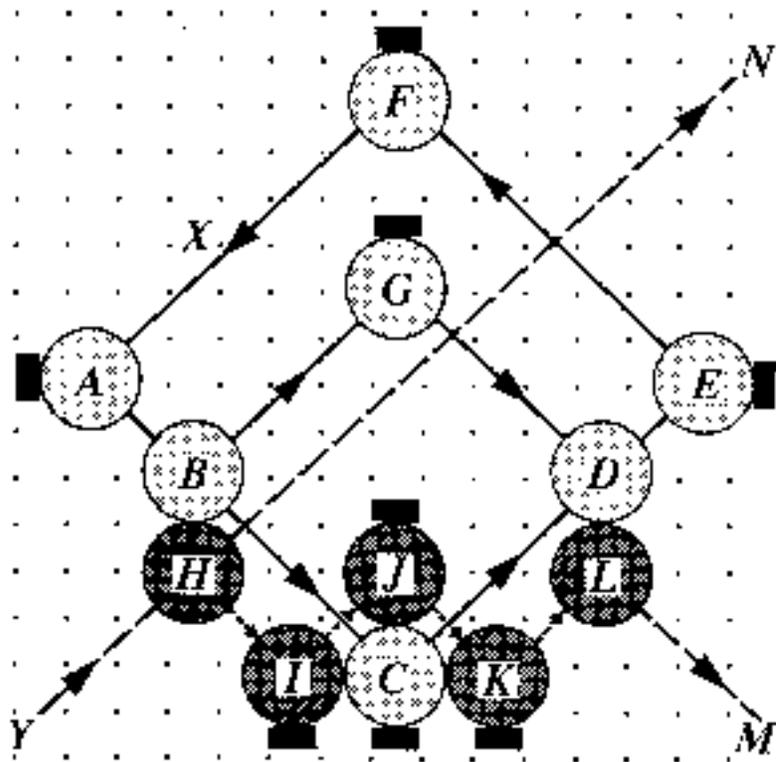
## FANOUT and ERASE:

Although the above gates are sufficient for the mathematics of logic, they are not sufficient to build a practical machine. A useful computer will also require the FANOUT and ERASE gates (Fig. 2).



**Fig. 2** Two non-standard gates that are required to build a computer, in addition to a universal set of logic gates, are: (a) the FANOUT gate which duplicates an input **A** and (b) the ERASE gate which deletes its input.

First consider the FANOUT gate: Is it reversible? Certainly no information has been destroyed so it is at least logically reversible. Landauer showed that it could also be physically reversible [8]. Let us describe a simple model for FANOUT based on Bennett's scheme for a reversible measurement (Fig. 3) [9]. Here a dark ball is used to determine the presence or absence of a second (light) ball inside a trap. The trap consists of a set of mirrors and may be thought of as a one-bit memory register. If the trap is occupied then the dark ball is reflected and leaves along direction **M** (with the light ball continuing along its original trajectory); otherwise it passes unhindered towards **N**. Upon leaving the trap, the dark ball's direction is used to populate, or not, another trap.



**Fig. 3** A reversible measurement of the existence of a (light) ball in a trap of mirrors (dark rectangles) [9]. A (dark) ball enters the trap from **Y**. In the

absence of a light ball in the trap the dark ball will follow the path **HN**. In presence of a light ball (timed to start at **X**) the dark ball will deflect the light one from its unhindered trajectory **ABCDEF** to **ABGDEF** and will follow the path **HIJKLM** itself. (Copyright 1988 by International Business Machines Corporation, reprinted with permission.)

Let us now consider the ERASE operation which is required to "clean out" the computer's memory periodically. One type of erasure can be performed reversibly: If we have a backup copy of some information, we can erase further copies by uncomputing the FANOUT gate. The difficulty arises when we wish to erase our last copy, referred to here as the primitive ERASE.

Consider a single bit represented by a pair of equally probable classical states of some particle. To erase the information about the particle's state we must irreversibly compress phase-space by a factor of two. If we allowed this compressed phase-space to adiabatically expand, at temperature **T**, to its original size, we could obtain an amount of work equal

to  $k_B T \ln 2$  (where  $k_B$  is Boltzmann's constant). Landauer concluded, based on simple models and more general arguments about the compression of phase-space, that the erasure of a bit of information at temperature **T** requires the dissipation of at least  $k_B T \ln 2$  heat (a result known as Landauer's principle) [8].