Information is Physical

Rolf Landauer

Citation: Physics Today **44**, 5, 23 (1991); doi: 10.1063/1.881299
View online: http://dx.doi.org/10.1063/1.881299
View Table of Contents: http://physicstoday.scitation.org/toc/pto/44/5
Published by the American Institute of Physics

INFORMATION IS PHYSICAL

There are no unavoidable energy consumption requirements per step in a computer. Related analysis has provided insights into the measurement process and the communications channel, and has prompted speculations about the nature of physical laws.

Rolf Landauer

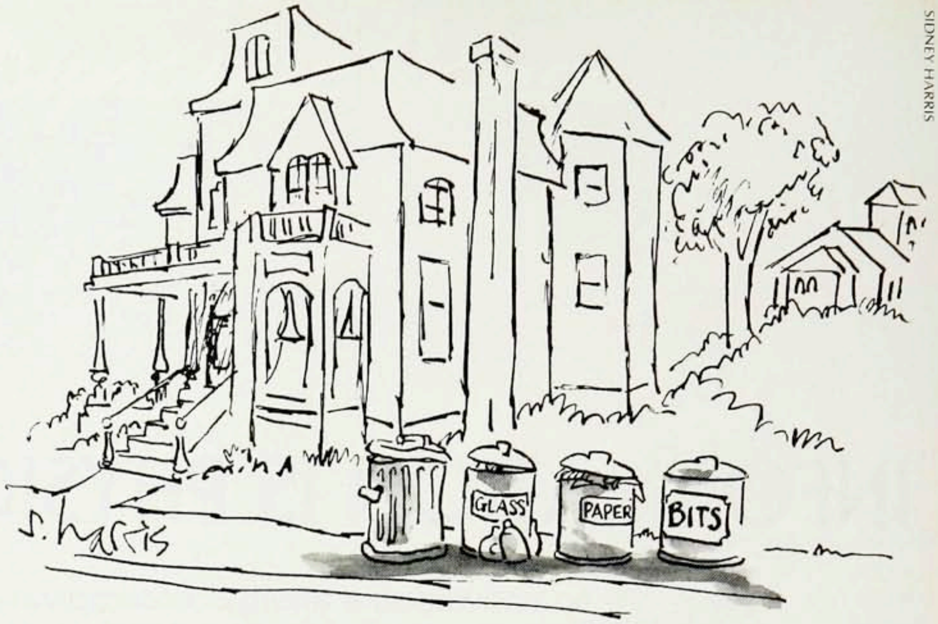
Thermodynamics arose in the 19th century out of the attempt to understand the performance limits of steam engines in a way that would anticipate all further inventions. Claude Shannon,[1] after World War II, analyzed the limits of the communications channel. It is no surprise, then, that shortly after the emergence of modern digital computing, similar questions appeared in that field. It was not hard to associate a logic gate with a degree of freedom, then to associate kT with that, and presume that this energy has to be dissipated at every step. Similarly, it seemed obvious to many that the uncertainty principle, $\Delta E \Delta t \sim \hbar$, could be used to calculate a required minimal energy involvement, and therefore energy loss, for very short Δt.

A long journey led to the understanding that these back-of-the-envelope estimates are not really unavoidable limits. In the process, we also learned to take a new look at the minimum energy requirements of the communications channel and the measurement process.

Computation is inevitably done with real physical degrees of freedom, obeying the laws of physics, and using parts available in our actual physical universe. How does that restrict the process? The interface of physics and computation, viewed from a very fundamental level, has given rise not only to this question but also to a number of other subjects, which will *not* be explored here. For example, cellular automata (spatially periodic arrays of interacting logic elements) are used to model a variety of physical systems.[2] A good many investigators have studied measures of complexity, attempting to quantify that intuitive notion. Much of this enterprise is motivated by the hope that the physical scientist can find an easy route to profound insights concerning the origin of life and the progress of evolution. Concern with complexity replaces, to some extent, an earlier concern with self-organization. There is also a view, originally presented by the great computer pioneer Konrad Zuse[3] and later elaborated by Edward Fredkin,[4] that the world itself is a computer. The particle passing by you is really a bit, or group of bits, moving along a set of interlinked logic units, much like a cellular automata machine. These interlinked logic units operate, of course, on a very fine scale of time and space. Quantum cryptography[5] and neural networks are two further fields; we need not list them all. I mention these subjects only to make it clear that they are not our concern.

What is a computer? It is basically an array of bits—0's and 1's—with machinery that maps one configuration of such bits into another configuration. A *universal* computer can simulate any other computer and can execute any specifiable set of successive transformations on bit patterns. The Turing machine is the archetype for fundamentally oriented computer discussions. (The box on page 26 explains how a Turing machine operates). The Turing machine preceded the modern electronic computer, and for practical purposes is too slow and too hard to program, but it has a remarkable advantage. The actual devices in the "head," doing all the work, can connect to an unlimited array of information, without the need for unlimited registers or unlimited memory-addressing machinery. At any one step of a Turing machine computation, only a very limited number of bits in close functional and spatial relationship, are subject to change. The Turing machine embodies, in a striking manner, a requisite of a reasonable computer: The designer of the machine needs to understand only the function carried out

Rolf Landauer is an IBM Fellow at the Thomas J. Watson Research Center, in Yorktown Heights, New York



SIDNEY HARRIS

Must information be discarded in computation, communication and the measurement process? This question has physical importance because discarding a bit of information requires energy dissipation of order kT. **Figure 1**

by the head, and not the whole computational trajectory. The designer need not anticipate all the possible computations carried out by the machinery; that is what makes a computer more than the mechanical equivalent of looking things up in a table.

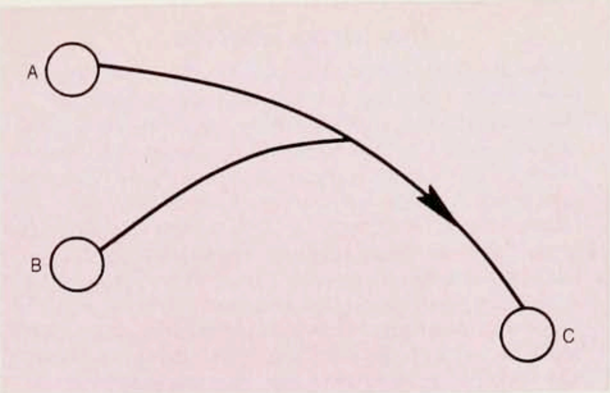
Reversible computation

Normally, in computation, we throw away information with great frequency (see figure 1). We do that, for example, when we erase an entry in memory or use a typical elementary logic operation such as "and" or "or," with two inputs and one output. Figure 2 illustrates, somewhat symbolically, the process in which differing initial states are mapped into the same final state and information is discarded. Figure 2 shows the compression in phase space of the degrees of freedom bearing the information. Total phase space cannot be compressed; the compression of the computer's information-bearing degrees of freedom requires an expansion of other degrees of freedom. That corresponds to an increase in their entropy. Thus throwing away information requires dissipation. Erasing a bit that was initially equally likely to be in a 0 or 1 state turns out, from the elementary formula $\Delta Q = T\Delta S$, to need an energy dissipation of $kT \ln 2$. In ordinary computers erasure of information occurs at almost every step.

Erasure of information, however, is not really essential, and computation can be carried out as shown schematically in figure 3a. The letters A_0, B_0, C_0 . . . represent different possible initial states, that is, different initial programs or different initial data. Each initial state is the beginning of a succession of states. Each step along the way results from a 1:1 mapping and allows identification of the preceding state. A merging of different tracks as shown in figure 3b corresponds to erasure and need not be invoked. Computation that preserves information at every step along the way (and not just by trivially storing the initial data) is called *reversible computation*, and was first described correctly and completely as a physical process by Charles Bennett.[6] Computers can easily be designed so that the energy at each of the successive steps in figure 3a is the same. We could, for example, use spin up and spin down, charge on the left electrode or the right, or superconducting current flowing one way or the other to denote 0 and 1.

Now, figure 3a is still symbolic; we have not yet described the actual machinery carrying out the logic operation that takes us from one state to the next. Let us assume, however, that the "motion" along the tracks of figure 3a is subject not to static friction but to viscous frictional forces proportional to the velocity of motion, as in electricity and hydrodynamics. Let us also assume that noise, at least thermal equilibrium noise resulting from the ambient temperature, is present. Then motion of the system along the chain of figure 3a is akin to an electron moving along a one-dimensional lattice. Noise will cause diffusive motion, but a very small forward force will serve to give the system a predictable average forward velocity. With a very small applied forward force we will encounter a very small dissipation, much less than kT per step, if desired. Remember, also, that one step of the whole computer, shown as a step along the chains of figure 3a, can consist of many logic operations done simultaneously, not just the operation of a single gate. Reversible computation is not computation using small components. The use of reversible computation is only aimed at minimizing energy consumption.

Reversible computation does not allow us to use most of the typical logic functions. All the commonly used multiple input gates, which have only a single output, throw away information. If we need to use these we have to embed them in more complex functions that preserve information, and thereafter save the extra outputs, which we call *history* or, perhaps more honestly, *garbage*, as symbolized in figure 1. We arrive at the end of the computation with the desired output, as in any computer, in designated output registers. Additionally we have a good deal of history that could not be discarded. We can copy the output registers with arbitrarily little dissipation if we do so slowly enough, as an example will later show. But we cannot erase the unneeded history; that would void the precautions we took to save it. After copying the intended output, however, we can reverse the computation and unwind it, returning to the initial state. Just as we were able to push the system forward, slowly, along the chains of figure 3, we can push it backward. I will not take the space here to discuss the possible alternatives, and their respective energy costs, once we return to the initial state. But, at worst, if we simply erase the initial program we incur an energy cost proportional to its size and not proportional to the possibly much larger number of steps in the program. While I will not discuss input–output operations in detail, there is a central point: Information transfer from one apparatus to another need not be more dissipative than information transfer within the reversible computer. We can, of course, invoke much more dissipative operations, for example, those using physiological machinery such as eyes, ears and brains. But those are hardly optimal processes.



Phase space of a computer, sketched symbolically showing information loss in the transition from A or B to C. **Figure 2**

Time-modulated potentials

What is the actual machinery that can take us from one state in figure 3a to the next? There are a number of proposals, but I will not list all of them. Some of the proposals come in two versions: either with viscous friction or in a form presumed to be frictionless. One proposal uses the Fredkin gate, in which balls are pushed through pipes and, in turn, control switches.[7] Bennett has described two reversible Turing machines.[8] One is based on genetic code machinery; the other involves machinery without springs, in which movable hard pieces can block the motion of other parts. I will describe none of these and will only discuss some aspects of an additional classical proposal using particles in time-varying potentials. Later we will consider quantum mechanical computers.

Figure 4 illustrates a classical potential whose time variation is externally imposed. A heavily damped well is taken from a narrow monostable state through a flat bifurcation stage (or second-order transition) to a deeply bistable state, and then back again. In the flat state, before going on to the deeply bistable state, the particle in the well is very susceptible to external influences. These biasing forces are provided by coupling to particles in other wells that are already in their deeply bistable state, and which subsequently will be restored to the monostable state. Later, the particle in the well of figure 4 will control the motion of further particles. This potential-well scheme is adapted from independent inventions by John von Neumann and Eiichi Goto.[9] Their proposals invoked microwave excitation of nonlinear circuits, and demonstrated that all the logic in a computer can be executed with such a scheme. A further variation on the theme is due to Konstantin Likharev,[10] who used Josephson-junction circuits, and who first pointed out that an appropriate choice of logic functions would allow reversible computation.

In such schemes it is possible that as a result of noise, the particle will be left unintentionally in the wrong well, on the uphill side of the applied biasing force coming from other wells. It can be shown that if we use strong enough forces, and if we modulate the wells sufficiently slowly that the particle's probability distribution is never far from the Boltzmann distribution, the probability of error can be made as small as we wish. Nevertheless, for a given design and a given speed, there will be a residual, nonvanishing error probability. This is typical of all reversible computer proposals. They assume, somewhere, the equivalent of our large forces, for example, by invoking hard and impenetrable parts.[8] Thus reversible computation can be made as immune to error as we wish, preventing jumping between the tracks of figure 3a.

In some schemes an alternative to minimizing errors is to recognize and correct errors. For example, we can restore the computation to the intended track if particles deviate gradually from intended trajectories or if we carry out the computation in three simultaneous systems and intermittently compare results. Throwing away the error is a dissipative event. It leads to an energy cost per step that is proportional to the error rate, but not directly dependent on the computational velocity.

The interaction of a time-dependent potential with a particle at a fixed position is not a source of dissipation. Dissipation occurs only through the motion of the particle against frictional forces. Thus slow motion of the particle insures minimal dissipation.

Figure 5 illustrates a particularly simple use of this time-modulated potential-well scheme. In figure 5a we start with information in the left-hand well, and in figure 5b it has been transferred to the right-hand well. In figure 5c the left-hand well has been restored to a monostable state. The transition from figures 5a to 5b represents the production of a copy and shows that this can be done with as little dissipation as desired, a fact we have already mentioned. The transfer of information in the step leading from figures 5a to 5b is essentially part of a measurement cycle: The right-hand well has acquired information about the left-hand well. We can see that it is not the information transfer step that requires dissipation in measurement.

The transition from figures 5b to 5c is called *uncopying*. It is the inverse of copying. We start with two copies of a bit, guaranteed to be identical. We end up with only one copy of the bit; the other bit is now in a previously designated standardized state. Uncopying is not equivalent to erasure and, just like copying, can be done with a dissipation per step proportional to speed. When we reverse a computer after completion of a program and return to the initial state, uncopying can be used to clear out the initial program with minimal dissipation, if a second copy of that program is available.

The transfer of a bit, shown in the total sequence in figure 5, can be iterated. The bit can be passed on to further wells. This is then a communications channel, and we have shown that a bit can be moved along a chain with a dissipation proportional to its speed of motion, avoiding any minimum energy requirement of order kT.

The time-modulated potential-well scheme was originally conceived as one in which the time variation was externally imposed. Thus the process was clocked, as real computers are. We can conceive of the machinery which controls the time variation of the potential as diffusive, and itself subject to a very small bias force. Thus, the potential-well scheme can also become one of the schemes illustrated by figure 3a.

The Turing Machine

Originally proposed as a device for discussing logical executability, the Turing machine is also suitable for a discussion of physical executability. The Turing machine consists, in part, of an infinite tape as shown in the figure below. One of several designated states occurs at each tape position representing the information on the tape. A binary choice of states, 0 or 1, is adequate, though a larger "alphabet" may be more convenient. The tape is initially in a standardized state, say 0, except for a limited number of positions that are prepared as a program and determine what the machine does subsequently. The machine also includes the "head," which does the work. The head has an internal memory that serves to define its state. The head reads the tape information at its current location and this, together with the internal memory state, determines the subsequent action of the head. The subsequent action consists of

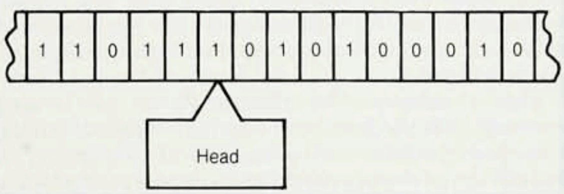
▷ setting the information state at the tape element in question, that is, either leaving it alone or changing it

▷ resetting the internal memory state of the head

▷ moving the head one position to the right or to the left.

Then the whole cycle starts over again.

In modern terminology we can think of the head as a small processor, or logic block, whose inputs are its existing internal memory state and the bit at the current tape position. The three actions highlighted above are the outputs. If the head is equipped with a suitably chosen logic function, the Turing machine can, with this one head, execute all computer programs or equivalently, all executable algorithms. If the machine is given a terminating program (for example, to calulate π to 25 places, in contrast to a continuing calculation of π) it will come to a halt upon completion of the program. The Turing machine embodies the key ingredient of the stored-program computer, whose development followed some years later: Data and instructions are presented and handled in the same format, in the same storage space, and thus instructions can be modified by the program.



Reversible computation has been described and elaborated by a good many authors with differing viewpoints. (See references 11 and 12 for a start to the citation trail.) Nevertheless, objections continue to appear. On one side there are pessimists who believe that more energy has to be expended.[13] On the other side there are optimists who believe that even if information is discarded, we can still minimize dissipation to any desired extent.[14]

Measurement and communication

The energy requirements of the measurement process have been of interest for over a century as a result of concern with Maxwell's demon. Maxwell pointed out that if we knew the locations and motions of individual molecules, we could get them to do work, even though they come from a thermal equilibrium state. Leo Szilard,[15] in a pioneering analysis in 1929, pointed to the need for concern with the bit (or bits) that provides information about the molecule and controls the subsequent behavior of the apparatus extracting the energy. The prevailing wisdom, until recently, was that the transfer of information from an object to be measured (in this case the molecule) to a meter or register requires energy dissipation. Leon Brillouin and Dennis Gabor[16] found different dissipative ways of measuring the location of a molecule. They invoked a photon that was used up in the process of "seeing" the molecule, and which had to be of high enough energy to be distinguishable from blackbody radiation. Neither they nor the authors of many later papers asked the obvious question: How do we know that this is the least dissipative information transfer process? The discussion associated with figure 5 indicates that information transfer can be done with arbitrarily little dissipation. Today, as a result of the work by Bennett,[17] we know that this also holds for the molecular measurements needed to operate Maxwell's demon. The dissipation required to save the second law and to prevent us from making molecules in thermal equilibrium do work comes not from information transfer to the meter or control apparatus but from the subsequent resetting of that apparatus. An early version of this viewpoint was given by Oliver Penrose,[18] and a recent elaboration by Wojciech Zurek.[19] A charming and scholarly review of the demon's long history and the many viewpoints it has generated is given in reference 20.

A somewhat similar history has beset the question, how much energy is needed to move a bit along a communications link? Shannon[1] showed that in a linear transmission line with thermal equilibrium noise at least $kT \ln 2$ per bit is required, assuming that the energy in the message has to be dissipated at the receiving end.[1] Unfortunately, later authors ascribed a universal applicability to Shannon's conclusion, which he had presented as arising from an analysis of a *special case*. Information does not have to be sent by waves; we can use the postal service to mail a letter or a floppy disk. Information need not use linear transmission media in which noise added to the signal can easily cause it to be confused with another signal. As we have seen in connection with figures 4 and 5, information can be handled in nonlinear systems with local states of stability, where small noise signals introduce no error at all. By iterating the process shown in figure 5, information can be passed along a chain of time-modulated wells with a dissipation proportional to the speed of transmission. Elsewhere I have analyzed this and several other communications links that demonstrate this possibility.[12]

Quantum models

Our discussion up to now has focused on dissipative classical systems with noise. We will skip classical dissipationless models and turn directly to quantum mechanical Hamiltonian systems. The discussion of such systems commenced with the work of Paul Benioff[21] and has been elaborated by him and others.[22] These theories specify Hamiltonians that cause an interacting set of bits (which can be considered to be spins up and down instead of 0's and 1's) to evolve in time, just as we would want them to do in a computer. The Hamiltonians are Hermitian operators, but the theories do not tell us how to assemble fundamental particles, or parts in the stockroom, to build a computer. They are not patent disclosures. Furthermore, the Hamiltonian includes only the information-bearing degrees of freedom. The parts holding these bits in place, and their lattice vibrations, are not included: The descriptions assume that there is no noise and no friction.

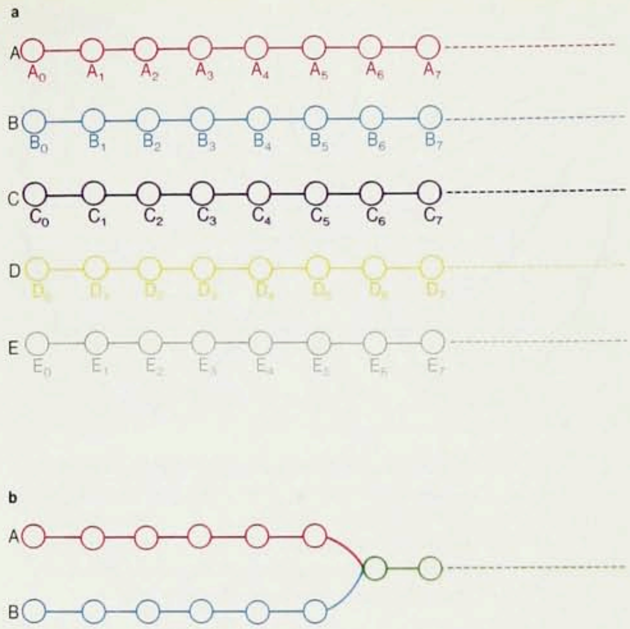
We first consider Hamiltonians that include an explicit externally imposed time dependence. Some investigators consider such a time dependence a blemish, but as long as we do not really explain how to achieve the invoked Hamiltonian, the time dependence does not—in my view—constitute a serious additional fault. The time-dependent scheme described here is based on Benioff's work but is simplified for the purposes of this account, partly on the basis of suggestions by Charles Bennett.

Consider the symmetrical bistable potential well shown in figure 6a. The particle is initially in the left-hand well. This represents a linear superposition of the symmetric ground state and the antisymmetric first excited state, chosen so as to interfere destructively in the right-hand well (as shown in figure 6b), at the initial time. Now if the energy splitting between these two states is ΔE, then at a time $\Delta t = \pi\hbar/\Delta E$ later, the states will interfere destructively in the left-hand well, and the particle will have tunneled to the right-hand well (as shown in figure 6c). (Actually the wavefunction in the initial well will not vanish exactly; the particle is not really transferred completely. We can avoid this difficulty by using an abstract two-state system or a combination of spin and projection operators, as invoked by Benioff.[21] But wavefunctions in potential wells are easier to draw and more suggestive.) When tunneling to the right-hand well of figure 6c is completed, a wall is erected at the top of the barrier, as shown in figure 6d. This will prevent the particle from returning to its original well during the next time step. The particle is now coupled to a third well, as shown in figure 6e, where an impenetrable wall, shown by a dashed line, has just been removed. The particle can then tunnel into this third well. Note, incidentally, that we have presented the basis of a quantum mechanical communications link, akin to the classical one mentioned in connection with figure 5.

The excitation energy ΔE above the ground state can be very small in a computation if we are willing to accept slow information transfer. The initial left-hand well in figure 6a can be considered to represent a computational state of the computer. The computer (rather than a particle) is then transferred to the state represented by the right-hand well in figure 6c. This represents the next state of the computer, except perhaps for a special clocking bit that labels it as an interim state. Then in the motion to the third state, at the right in figure 6e, we reset this special status bit and arrive at the full next state of the computer, ready to restart the whole two-phase cycle.

Our discussion focuses on the execution of logic in a computer, but I pause here for a comment about storage density. Figure 6 demonstrates that we can store information in a bistable well with arbitrarily little energy above the ground state, if the barrier between the two valleys is made sufficiently impenetrable (this decreases the gap ΔE). That holds whether we use occupation of the left and right wells, respectively, for 0 and 1, or use the symmetric and antisymmetric states for that purpose. Note that we can use *very deep* and *very narrow* wells to store information. Quantum mechanics imposes no obvious limits on the spatial density of storage.

We can view the time-dependent potentials of figure 6 as forming "pipes" connecting successive states. We

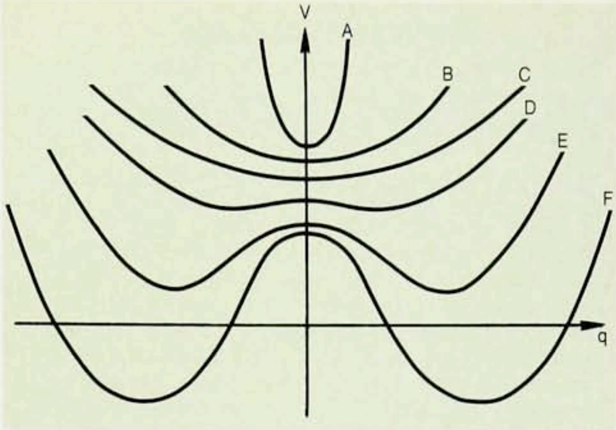


One-to-one mapping in computation. a: The left-hand end of a horizontal chain is the initial state. Motion to the right yields forward steps through a sequence of states represented by successive labeled circles. Different letters correspond to different initial states. **b:** When two distinguishable computational paths merge into one, information is lost. **Figure 3**

assume that high barriers prevent lateral tunneling out of the pipes to other computational paths, much as we invoked barriers between the different tracks of figure 3a. The space of computational states will then have a structure much like that shown in figure 3a. Figure 3a places successive states next to each other; in a space where adjacent states differ by only a single bit, the "pipes" would have a much more complex structure.

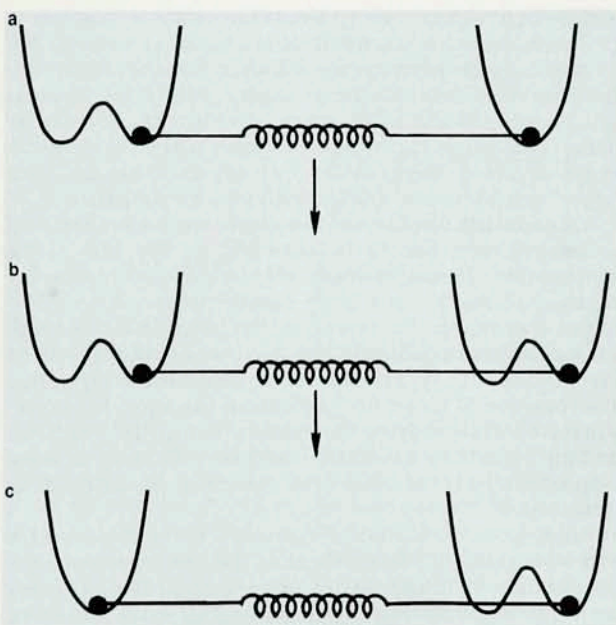
A quantum mechanical reversible computer, just like a classical one, has to be reversed at the end of its computation. It can, however, stay in the state representing the end result for a large number of cycles, to allow output operations. Furthermore, the information-bearing bits (or spins) are guaranteed to be in either the 0 state or the 1 state. They are not in a quantum mechanical superposition of these; no "collapse of the wave-function" is involved while copying the output. The initial program-loading operation, however, could benefit from a more detailed analysis than has been presented in the existing literature.

A proposal by Richard Feynman[23] avoids the need for a time-dependent Hamiltonian. Feynman views the computation as illustrated in figure 3a and motion along the sequence of states as being analogous to motion of an electronic wave packet along a periodic lattice. In this case it is the initial state, representing a moving packet, that assures the direction of the computation, without the need for externally imposed time dependence. The analogy to motion in a periodic potential immediately alerts us to a problem. In one dimensional lattices which have some disorder, an incident wave packet suffers reflection, and its transmission decreases exponentially with the length of the sample. This is known as *localization*. In our computational case, if, for example,



Time-dependent potential well going from single minimum at A to a deeply bistable state at F, and later returning to A. The curves are displaced vertically relative to one another for clarity. The variable q gives the position of the particle in the well. **Figure 4**

the energy of a state depends slightly on the exact bit pattern, we can expect similar problems. Thus, dissipationless and completely coherent quantum computation, even if it were feasible, is unlikely to be desirable. The problem of localization is avoided by the presence of inelastic (dissipative) events that disrupt the coherence of the reflections that cause localization. In the case of the time-dependent potential of figure 6, the presence of imperfections will give us some probability that the state



'Copying' and 'uncopying' using particles in time-dependent potential wells coupled through a spring. In the transition from **a** to **b**, information in the bistable well on the left determines the state of the one on the right (copying). In the transition from **b** to **c** the well on the left is brought back to a monostable state (uncopying), ready to receive new information. **Figure 5**

will stay in its original well. In that case that component of the computer's state will move along the computational track in the wrong direction. Whether the overall adverse consequences of this are as severe as for Feynman's time-independent case is not yet clear.

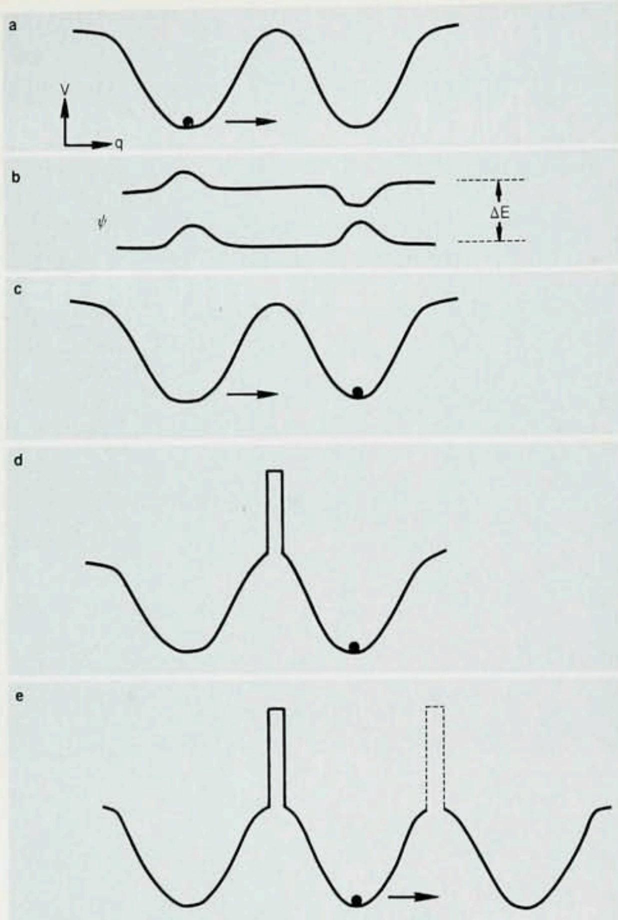
Coherent quantum mechanical computation may be unachievable in practice and even may be undesirable. Nevertheless these theories demonstrate that the uncertainty principle does not imply an energy *dissipation* requirement per computer step.

The nature of physical law

At this point the reader deserves a warning: We are entering the genuinely speculative part of this discussion. We have seen that neither kT nor the uncertainty principle leads to unavoidable minimum energy dissipation requirements for computation. Are there, then, no limits imposed by physics? Undoubtedly there are such limits, but we will have to work harder to understand them. A deeper question: How large a memory can we supply for our computer? Quite likely we are in a finite universe. In any case, nature is unlikely to be so cooperative as to enable us to bring together an unlimited memory.

The finiteness of our universe, and the resulting implication for memory limits, is not the only problem of a cosmological nature. Computers are full of degradation phenomena. Corrosion, evaporation, diffusion, electromigration and earthquakes cause problems. Alpha particles, cosmic rays, spilled coffee, and lightning can also be deleterious. Can we offset these problems to any required degree by using sufficiently massive parts or by the use of such well-known schemes as triple modular redundancy? Perhaps, but then we aggravate the problem already mentioned: We will run out of parts more quickly if we make them more massive or use redundant circuitry.

In contrast to this physical situation, mathematics has taught us to think in terms of an unlimited sequence of operations. We have all grown up with the sense of values of the mathematician: "Given any ϵ, there exists an N, such that. . . ." We can calculate π to any required number of places. But that requires an unlimited memory, unlikely to be available in our real physical universe. Therefore all of classical continuum mathematics, normally invoked in our formulation of the laws of physics, is not really physically executable. The reader may object. Can we not define the real numbers within a formal mathematical postulate system? Within that system, can we not prove that $\cos^2\theta + \sin^2\theta = 1$ *exactly* and not just to a great many decimal places? Undoubtedly we can. But physics demands more than that; it requires us to go beyond a closed formal system and to calculate actual numbers. If we cannot distinguish π from a terribly close neighbor, then all the differential equations that constitute the laws of physics are only suggestive; they are not really algorithms that allow us to calculate to the advertised arbitrary precision. I am proposing that the ultimate form of the implementable laws of physics requires only operations available (in principle) in our actual universe. Whether the inevitable limit on precision is simply a limit on the number of bits that can be invoked in physics or is



Controlled tunneling through a sequence of states. Initially, in **a**, the system is in the left-hand well, in a superposition of the symmetric ground state and the antisymmetric state above it (**b**). The energy splitting ΔE between those states determines the time needed to reach **c**, where the system is in the right-hand well. At that time an impenetrable barrier between the two wells is erected (**d**), preventing the return of the system. Removal of the barrier, shown in **e** by dashed lines, will then cause the particle to tunnel into the third well. **Figure 6**

more complex and statistical is unclear. But the universe was not constructed by my firm or one of its competitors, and therefore the more complex, statistical possibility, resembling a universal source of noise, seems more likely.

Others have, in a variety of ways, suggested that space and time in the universe are not really described by a continuum and that there is some sort of discretization, or some limit on the information associated with a limited range of space and time. Most of these investigators, however, consider that to be a description of the *physical* universe and are still willing to invoke continuum mathematics to describe their picture. My suggestion is for a more self-consistent formulation: Information handling is limited by the laws of physics and the number of parts available in the universe; the laws of physics are, in turn, limited by the range of information processing available. Among the authors who have made proposals that have some relation to the view propounded here, I need to single out John Wheeler,[24] who has told us, "No continuum," and also that the laws of physics were not necessarily there at the beginning of the universe. Wheeler's suggestion, that the laws of physics are interlinked with the evolution of the universe and our observation of it, is not equivalent to my proposal, but both suggestions deviate from the more prevalent notion that the laws of physics are independent of the contents and history of the universe.

Earlier centuries gave us clockwork models of the universe. A similar, but more modern, orientation leads to the position of Zuse[3] and Fredkin[4] that the universe *is* a computer. Without going quite that far, I do suggest that there is a strong two-way relationship between physics and information handling.

References

1. C. E. Shannon, Bell Syst. Tech. J. **27**, 379 (1948); *ibid*, p. 623.
2. T. Toffoli, N. Margolus, *Cellular Automata Machines*, MIT P., Cambridge, Mass. (1987).
3. K. Zuse, Int. J. Theor. Phys. **21**, 589 (1982); *Rechnender Raum*, Friedrich Vieweg und Sohn, Braunschweig (1969).
4. R. Wright, *Three Scientists and Their Gods*, Times Books, New York (1988).
5. See The Economist, 22 April 1989, p. 81.
6. C. H. Bennett, IBM J. Res. Dev. **17**, 525 (1973).
7. E. Fredkin, T. Toffoli, Int. J. Theor. Phys. **21**, 219 (1982).
8. C. H. Bennett, Int. J. Theor. Phys. **21**, 905 (1982).
9. J. von Neumann, US Patent 2 815 488, filed 28 April 1954. E. Goto, J. Electr. Commun. Eng., Japan, **38**, 770 (1955).
10. K. K. Likharev, Int. J. Theor. Phys. **21**, 311 (1982). K. K. Likharev, S. V. Rylov, V. K. Semenov, IEEE Trans. Magn. **21**,947 (1985).
11. C. H. Bennett, IBM J. Res. Dev. **32**, 16 (1988).
12. R. Landauer, in *Selected Topics in Signal Processing*, S. Haykin, ed., Prentice–Hall, Englewood Cliffs, N. J. (1989), p. 18. (Printed version has some figures oriented incorrectly.)
13. E. Biedermann, PHYSICS TODAY November 1990, p. 122.
14. E. Goto, N. Yoshida, K. F. Loe, W. Hioe, in *Proc. 3rd Int. Symp. Foundations of Quantum Mechanics, Tokyo 1989*, S. Kobayashi, H. Ezawa, Y. Murayama, S. Nomura, eds., Phys. Soc. Japan, Tokyo (1990) p. 412.
15. L. Szilard, Z. Phys. **53** 840 (1929); English translation in J. A. Wheeler and W. H. Zurek, eds., *Quantum Theory and Measurement*, Princeton U. P., Princeton 1983, p. 539.
16. L. Brillouin in *Science and Information Theory*, Academic Press, New York (1956), p. 162. D. Gabor, in *Progress in Optics*, vol. I, E. Wolf, ed. North–Holland, Amsterdam (1961) p. 109.
17. C. H. Bennett, Sci. Am., November, 1987, p. 108.
18. O. Penrose, *Foundations of Statistical Mechanics*, Pergamon, Oxford (1970).
19. W. H. Zurek, Nature **341**, 119 (1989). See also C. M. Caves, W. G. Unruh, W. H. Zurek, Phys. Rev. Lett. **65**, 1387 (1990).
20. H. S. Leff and A. F. Rex, *Maxwell's Demon: Entropy, Information, Computing*, Princeton U. P., Princeton (1990).
21. P. Benioff, J. Stat. Phys. **22**, 563 (1980); *ibid.* **29**, 515 (1982); Phys. Rev. Lett. **48**, 1581 (1982).
22. D. Deutsch, Proc. R. Soc. London, Ser. A **425**, 73 (1989). P. Benioff, in *New Techniques and Ideas in Quantum Measurement Theory*, Ann. N. Y. Acad. Sci., vol. 480, D. M. Greenberger, ed., (1986) p. 475; N. Margolus, *ibid.*, p. 487. A. Peres, Phys. Rev. A **32**, 3266 (1985); W. H. Zurek, Phys. Rev. Lett. **53**, 391 (1984).
23. R. Feynman, Opt. News **11**, 11 (1985); reprinted in Found. Phys. **16**, 507 (1986).
24. J. A. Wheeler in *Proc. 3rd Int. Symp. Foundations of Quantum Mechanics, Tokyo 1989*, S. Kobayashi, H. Ezawa, Y. Murayama, S. Nomura, eds., Phys. Soc. Japan, Tokyo, (1990), p. 354; IBM J. Res. Dev. **32**, 4 (1988). ■