Computing Without Dissipating Energy

Computers are machines and should be subject to thermodynamics; what is the minimum energy needed for computation?

The best you can do is break even (First Law of Thermodynamics). You can't even break even (Second Law of Thermodynamics).

These popular expressions pithily convey the sweeping and seemingly tyrannical sway of the laws of thermodynamics. In the case of heat engines, such as the one in your car, the first law says that the useful work produced by an engine cannot exceed the energy put in. The second law then asserts that even an ideal (reversible) engine free of friction or heat losses will see some of the input energy turned into waste heat rather than useful work. A nonideal (irreversible) engine will be even less efficient. How do these laws apply to computers?

Computers do no work in the usual sense, so there are no energy losses in a reversible machine. Nonetheless, computation as presently carried out is an irreversible or entropy-generating process and produces a great deal of waste heat, enough to require elaborate cooling strategies in some computers. Some scientists have wondered if it is possible, in principle if not in practice anytime soon, to bring the energy dissipation per logical operation arbitrarily close to zero. In the last few years, a significant number have come to believe it is.

The matter is far from settled, however, as underscored by a group of physicists at Arizona State University in Tempe, who critically analyzed previous arguments in favor of dissipationless computing in the 16 January *Physical Review Letters*. Their conclusion is that computation inherently requires the expenditure of energy of the order of kT per bit of information processed, where k is Boltzmann's constant and T is the temperature. This is about 4×10^{-21} joules at room temperature.

Rebuttal letters are being prepared for publication. For the moment, no one is budging from strongly held positions. Resolution of the debate may not be imminent. Says Robert Bate, an interested observer at Texas Instruments in Dallas, "It probably won't be settled until someone builds [a dissipationless computer]."

Over the years, engineers have made considerable progress in reducing the energy requirements of computers. In energy units of kT, transistors in today's integrated circuits dissipate about 10^8 per switching operation, whereas old mechanical relay switches lost up to a billion times more. However, there are so many transistors in a computer and these switch so often that the total heat generated remains high.

Even if dissipationless switches were devised, the electronic circuits of computers contain resistors, either as discrete elements or as an imbedded property of other circuit elements. Resistors are the electrical equivalent of mechanical friction and thereby make all circuits irreversible.

These considerations are technology-specific. Those scientists pondering the possibility of dissipation-free computers are interested in the more generic question of whether there is anything in the computation process itself that demands the dissipation of energy. As Rolf Landauer of the IBM Yorktown Heights laboratory puts it, "Are there any fundamental limitations on computer performance that we can arrive at without anticipating every possible future computer technology?"

For the moment, no one is budging from strongly held positions.

Does it matter whether such limits exist or not? The near-term future of computer technology clearly does not depend on the answer. The energy limits being discussed are many orders of magnitude lower than engineers can attain today. Ultimately, of course, it will be important to establish whether a fundamental limit exists, so engineers will know when to stop trying.

For the moment, says Landauer, the question of dissipationless computing is more of an intellectual adventure. "Information, whether it is in biological systems, in a digital computer, or handled by pencil and paper, inevitably has a physical form. As a result, manipulation of information is inevitably subject to the laws of physics." Mathematics itself, according to this line of thinking, is not excluded.

For many years, scientists believed that there was an irreducible energy loss associated with computation of $kT \ln 2$ per bit of information transferred or per logical decision made. In 1961, Landauer argued that a fundamental source of the dissipation was the erasure of information. For example, logic circuits have the property that they are noninvertible. From the output of a logic circuit, one cannot always reconstruct what the input was. This is the lost information. Landauer asserted that logical noninvertibility translates into physical irreversibility, and hence an energy loss.

His argument is based on the statistical mechanics notion of phase space. Each particle in a gas, for example, is characterized by a position and a momentum that change with time. The phase space for a gas of N particles is then a multidimensional mathematical space with $6N$ coordinate axes, one for each component of the position and velocity of each particle. The occupied volume of phase space is defined by the allowed positions and velocities of the particles. According to statistical mechanics, the amount of occupied phase space cannot be changed by an outside force but the distribution can; that is, some parts may be populated at the expense of others.

Landauer imagined a more abstract phase space, with one coordinate being the information content of a logical device. Prior to an erasure operation, for example, the logical device can have two states (0 or 1). Afterward, it can have only one (the standard state of an erased bit). Consequently, the extent of occupied phase space in the logical coordinate is reduced by two, and the occupied volume must expand in other coordinates. These coordinates represent things like thermal vibrations in whatever physical system the logic device is implemented. Excitation of thermal vibrations means heat is generated.

Twelve years later, Charles Bennett of IBM proposed that logically reversible computation is possible. Bennett's discussion centered around an abstract entity called a Turing Machine, after one of the founders of computer science, the late British mathematician Alan Turing. Computers are supposed to be universal machines in the sense that programs can be written to carry out almost any computation, or at least the number of possible types of computations greatly exceeds the number of logic circuits in the machine. In contrast, it is possible to devise networks of logic circuits that can handle only a limited number of problems that are fixed in advance.

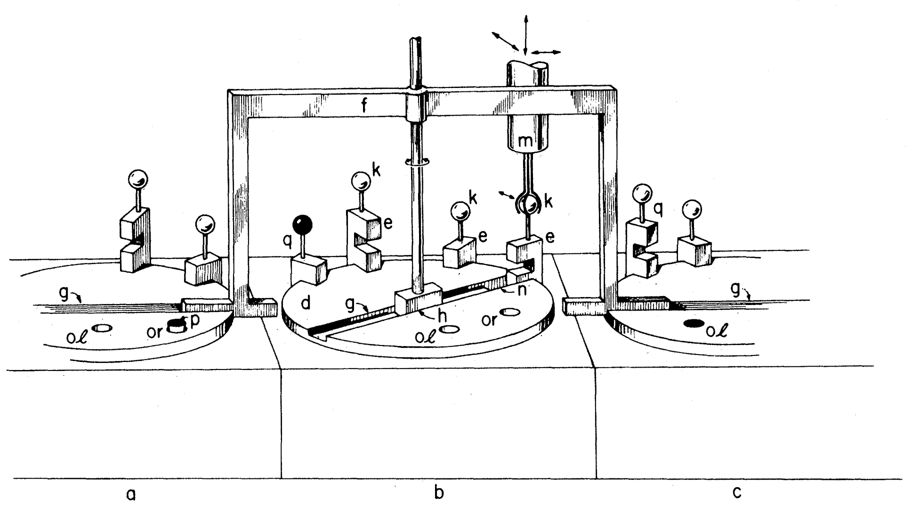
A Turing Machine is such a universal computer and consists of a tape and a head. The tape contains a sequence of symbols, such as the binary 0's and 1's of real computers. The head has several internal variables that define its state. It reads the contents of the tape. According to the contents it can change its internal state, change the symbols on the tape it has just read, and advance the tape in either direction. Changing the contents of the tape had been thought to require erasure of information. So, by Landauer's arguments, a Turing Machine should not be reversible.

Bennett maintained that a Turing Machine with three tapes could be reversible. The first tape would be read and processed as above. The second tape would be a "history" tape that retained all the input data and intermediate results of a computation. This would avoid the information loss. The third tape receives the final result of the computation and stores it. Then the machine is run in reverse with the result that the input is retained but the intermediate history of the first tape is eliminated.

A key point in the argument is that erasure of a known quantity does not distort phase space because one simply replaces one known quantity with another. Therefore, the elimination of the intermediate history of the first tape during the reverse run is not an irreversible process. Alternatively, one can think that both the forward and backward runs save all their intermediate results. It is just that the intermediate results of the backward run exactly cancel those of the forward run and leave a blank history tape.

Bennett has also devised a physical model for a reversible computer (see figure). He calls it a Brownian computer because thermal fluctuations impart a random motion to the machine. If one imagines a linear device as exemplified by the tape of the Turing Machine, then fluctuations drive it equally likely in the forward and reverse directions. Bennett therefore imposes a small driving force to steer the overall operation in the forward direction, although at any instant the motion may be forward or reverse with almost equal probabilities.

The fluctuations, which are akin to the noise in resistors in electrical circuits, make the machine operation dissipative, but the amount of dissipation can be made arbitrarily small by making the external force arbitrarily small. Then the machine is reversible in the same sense that heat engines are reversible in thermodynamics. Real heat engines are not reversible, but ideal ones can be if they are taken through their cycle in a quasistatic fashion; that is, very slowly.



Brownian Turing Machine

The parts are rigid, frictionless, and loosely fitting, so there is no dissipation of energy. The segments a, b, *and* c *represent three cells of the Turing Machine tape. The E-shaped bit-storage blocks represent binary 1's when they are up and 0's when they are down. The manipulator* m *reads the contents of the cell and changes the blocks from one position to another according to rules in a control mechanism not shown. The control unit contains a master camshaft whose motion can be forward or backward according to thermal fluctuations. A weak spring imposes an overall forward motion and is the only dissipative element in the system.* [*Courtesy of* International Journal of Theoretical Physics *21, 905 (1982)*]

When Bennett first approached Landauer with his ideas, he met with considerable skepticism. After several months of discussion, Landauer did become fully convinced and is now a proponent of reversible computation.

In the meantime, Edward Fredkin of the Massachusetts Institute of Technology has independently developed a different approach to reversible computation. Fredkin devised what he calls conservative logic gates. These do not lose information in their operation and therefore do not dissipate energy by Landauer's criterion. For example, a gate with three inputs of binary bits could work as follows: one input travels through the gate with no change. The other two either come out unperturbed or reversed, according to whether the first is a 1 or a 0.

According to Tommaso Toffoli, who works with Fredkin at MIT, one can translate any computer circuit (such as those of commercial mainframe machines) into a circuit made of conservative logic gates. However, at both the input and output of such a circuit, certain extra signals are required. To generate the extra input and to erase the extra output may require dissipating energy, but the amount of energy is independent of the complexity of the computation. Toffoli calls such a computer virtually dissipation-free. "The dissipation is not zero, but as we get smarter we can get arbitrarily close."

Fredkin has also devised a physical model for implementing conservative logic that he calls the billiard ball model. Consider a box with several entrance holes (inputs) and exit holes (output) on opposite sides of the box. Perfectly round and hard balls enter holes where the input information is to be a binary 1 and none enter where the input is a 0. Inside the box are perfectly reflecting mirrors. If a ball strikes a mirror, it changes direction 90 degrees. If two balls collide, both change directions. The placement of the mirrors and the input determine where balls appear at the outputs to be registered as 1's.

The whole affair is reversible by construction, as there is no friction and the collisions are perfectly elastic. Fredkin says that computer simulations verify that the model can do computation and therefore that in principle dissipationless computing is possible. He makes an analogy with the development of the laws of classical mechanics. Newton's laws, for example, clearly do not hold in the real world where friction is inevitably present. Nonetheless, the laws are correct. "Scientific progress is made by taking out friction to see the underlying law," says Fredkin.

The billiard ball model can be made to work in the real world, say the MIT scientists, although at the cost of introducing some dissipation. For example, if the balls are not perfectly aligned at the start, after only a small number of collisions, they will be so misaligned that computation is no longer accurate. The cure is to correct the positions from time to time. This takes energy and is the source of the dissipation. Moreover, the energy needed will be proportional to the number of logic gates, just as in today's electronic computers. The saving feature, says Toffoli, is that the proportionality constant is dictated by the state of the current technology. As technology improves, there is no lower bound to the coefficient, so that once again the billiard ball machine is virtually dissipationless.

A high point for the adherents of reversible computation came in May 1981, when Fredkin and Landauer organized an international conference on the physics of computation that was held at MIT. Researchers from Europe and the United States met to discuss the issues, and some proposed models of reversible computers. The conclusion according to Fredkin: "All serious workers in the field accept that physical models of reversible computation exist."

One noteworthy convert to the notion of reversible computation is Richard Feynman of the California Institute of Technology in Pasadena. Feynman told *Science* that he has constructed a quantum mechanical model of a computer that is reversible. The energy dissipated per operation is $kT \ln 2$ times two factors that can be made arbitrarily small: one by running the machine very slowly and the other by building it very carefully to avoid imperfections.

There are at least a few holdouts. One is Carver Mead of Caltech. Mead coauthored a 1980 book on designing very complex integrated circuits with Lynn Conway, then at the Xerox Palo Alto Research Center. One chapter contained an emphatic denial of the possibility of dissipationless computing. Mead is now working to put the assertions of that chapter on a more solid footing.

Wolfgang Porod, Robert Grondin, and David Ferry of Arizona State and Günter Porod of the University of Graz, Austria, are new participants in the discussion, but they adamantly reject the concept of reversible computation. Their basic argument is that the idea of logical reversibility has been incorrectly applied. It has yet to be proved, for example, that Bennett's logical reversibility translates to physical reversibility.

Porod discussed Bennett's three-tape Turing Machine with *Science*. From the point of view of the head of a Turing Machine, all information on a tape is random in the sense that the head has no way of knowing what is coming next. In particular, the head does not know whether it is erasing information on a single-tape machine (irreversible) or on a three-tape machine (reversible). Hence, there is no physical difference in the two processes, and the purported logical reversibility does not reflect the absence or presence of physical reversibility.

Moreover, there is, according to Ferry, a contradiction in the three-tape Turing Machine argument made by Bennett. According to Landauer's original phase space argument, one-tape machines are irreversible if they erase information. According to Bennett, three-tape machines can be reversible. But a theorem in computer science, applied to Bennett's three-tape machine, says that the three-tape machine can be emulated by a one-tape machine. Therefore, the three-tape machine cannot be reversible. (However, Bennett has argued that one-tape machines can be reversible.)

"All agree that dissipationless computing in a noisy environment is not possible. This was never a point at issue."

The Arizona State physicists assert that the true source of the irreversibility in computation is not the erasure of information but the reading operation. All such operations are done in a noisy environment characterized by thermal fluctuations. Hence, the physical system must be maintained far from thermal equilibrium in order to keep the system from thermalizaing and thereby losing its information. Physical systems maintained in a nonequilibrium state are by nature dissipative. "It is the competition between measurement and the thermalization process that requires energy to be dissipated," they wrote.

To Fredkin's assertion that one has to consider noiseless, friction-free models in order to discern the underlying physical laws of a process, Ferry retorts, "If you have to take noise out to even conceive of the problem, you have thrown out half the germane physics."

The preoccupation with noise has led some in the reversible computing camp to wonder if the Arizona State objections are as fundamental as the group believes. Says Paul Benioff of Argonne National Laboratory, who has devised a mathematical model of a reversible computer based on quantum machanics rather than classical mechanics, "All agree that dissipationless computing in a noisy environment is not possible. This was never a point at issue."

A second crucial ingredient in the Arizona State argument also revolves around the measurement process. What is it that provides a preferred direction of time in computation? Microscopic processes are time reversible; they are equally likely to proceed in either direction. Computation, however, "consists of a series of steps, which force the system from one state to its logical successor." The conclusion is that "only systems that are forced along a nonthermodynamic path by virtue of measurement, and thus dissipation, can be used for computation."

This point of view toward computation may be too narrow. Bennett, who started out his career as a chemist, makes an analogy with physical processes like crystal growth. A crystal grows approximately layer by layer. No atom can reside in the second layer until there is a first layer atom beneath it, and so on. Moreover, near the melting temperature, the probability that an atom in the second layer will stick is only slightly higher than the probability that the corresponding atom in the first layer will leave the growing surface.

The end product, the crystal, can be likened to the output of a computer. Although the correct sequence (growth of one layer after the other or completion of one logical operation after the other) is eventually followed, the individual steps leading to the output are not rigidly constrained to occur at predictable times as they are in conventional computers. Bennett's Brownian computer has built in physical constraints that guarantee that the correct sequence of steps takes place. It is like one of the puzzle cubes that can only be taken apart in one way. If it is shaken long enough, it will eventually come apart piece by piece.

Mead at Caltech says that Bennett has raised a fascinating but yet to be rigorously answered question: Is it possible to make a computing machine that is universal in the way a Turing Machine is from freely evolving systems such as Bennett's Brownian computer or Fredkin's billiard ball model? The answer to this question may be the key to reversible computing. So far, says Mead (though Toffoli disagrees), all the reversible machines demonstrated to work are actually combinational networks that can solve a limited range of problems rather than universal computers. If the answer to the question turns out to be no, then reversible universal computation is not possible.

—Arthur L. Robinson