Computers That Can Run Backwards

Reversible computations—which can, in principle, be performed without giving off heat—may be the future of computing.

Peter J. Denning and Ted G. Lewis

In the universe of computing, heat is the companion of progress—and its enemy. According to Moore's law, the number of components on a chip of a given size doubles every two years, potentially doubling its heat output at the same rate. Computer engineers have therefore constantly sought ways to reduce the energy consumption of each new generation of chips. Portable and desktop computers use circulating air and fans to cool their chips, whereas supercomputers use much more elaborate cooling systems, such as air conditioning and cooled liquid baths. These efforts have paid off handsomely: Computations per unit of energy have doubled every 1.6 years since the first electronic computers in 1945. But even with all their innovations, chip designers have found heat to be a major limitation in their quest for faster computers. Around 2005, chip makers started limiting clock speed (which controls the rate that computations are executed) to about 3 gigahertz per chip, because faster clocks generated heat too rapidly and caused chips to burn up.

To continue to keep pace with Moore's law, the computer hardware industry has needed to search for innovative new ways to make processors run cooler even as they run faster. Some look forward to quantum computers, an up-and-coming technology that will be common within a decade, if research continues at its current pace. These computers will take mere seconds to solve certain kinds of problems—such as cracking *ciphers,* or encoded messages—that would take current computers centuries. Quantum computers store information in the states of individual atoms, called quantum bits or *qubits*. Instead of wires, they use quantum-mechanical effects such as photons, superconductivity, superposition, and entanglement to store and communicate information between qubits. However, quantum circuits are exquisitely sensitive to heat: A small amount of heat can cause the atoms to vibrate too much and lose function. The initial versions of quantum computers rely on circuits cooled to within a fraction of a degree of absolute zero. We need quantum circuits that in theory give off no heat at all.

Standard computer circuits have a hidden source of heat: the loss of information when bits are erased. During a computation, components of circuits called *logic gates* most often take two bits as input and produce one bit as output. Each bit value is represented as a burst of energy that travels along wires between gates. A gate that receives two units of bit-energy at its inputs and delivers one unit of bit-energy at its output must lose a bit-unit of energy. Because energy cannot be created or destroyed, that bit-unit must go somewhere—so it appears as heat given off by the gate.

One of the earliest proposals to reduce heat from lost bits was to make computer circuits reversible. A reversible circuit has exactly as many outputs as inputs, assigning one output pattern to every input pattern and vice versa. That means each input can be reconstructed from the output; no bits are lost, so reversible circuits will not give off heat from bit loss. Furthermore, reversible circuits can simulate standard logic functions and can therefore be used in any computer.

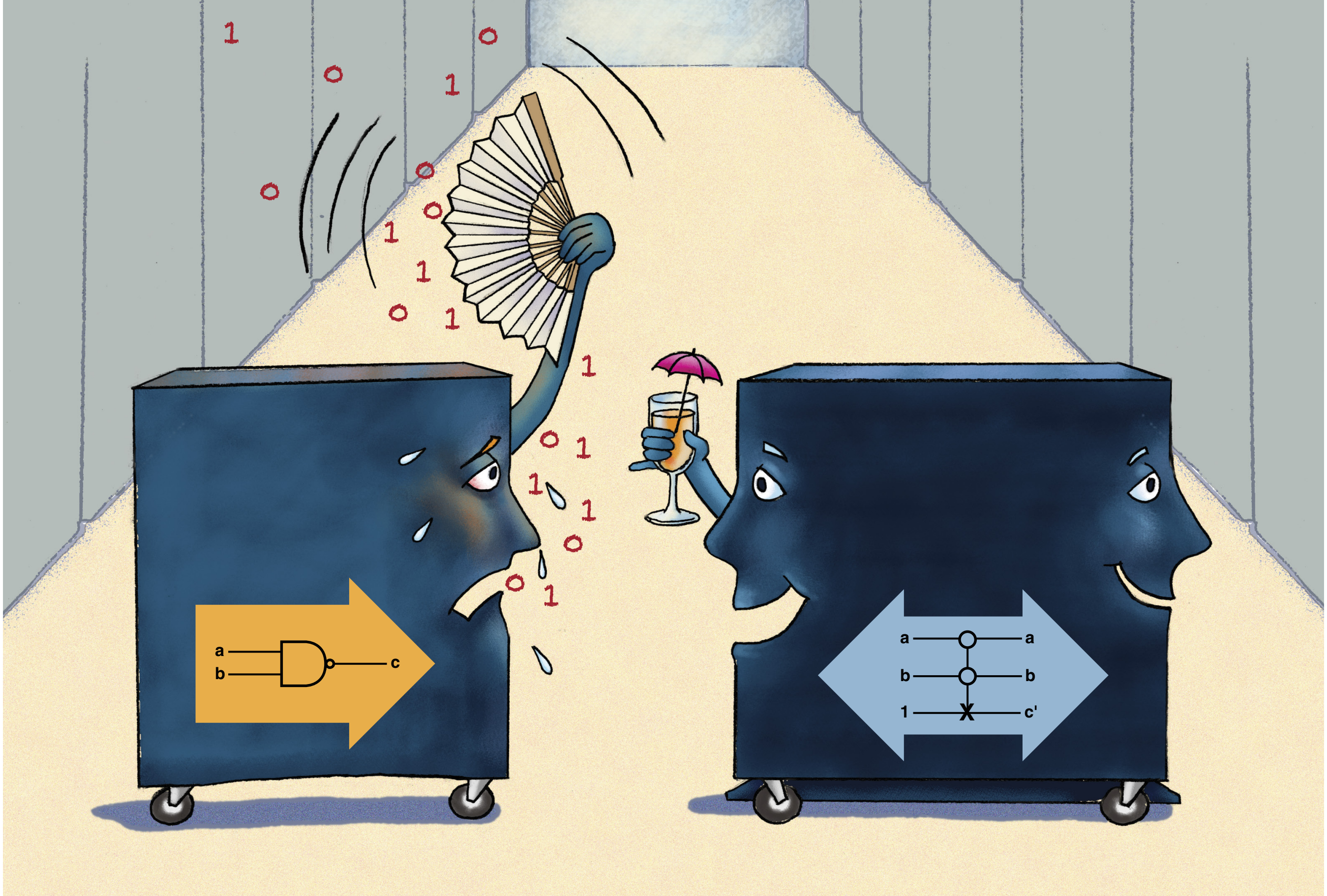
A curious feature of reversible circuits is that, as the name implies, they can actually be run in reverse! The circuits maintain the same function even if you reverse the roles of the input and output lines. Because of this ability, reversible computers are sometimes described as "computers that can run backwards." We do not actually run them backwards, although we frequently do partial reversals to recover from errors by restoring the circuit to a saved former state.

The Birth of Reversible Circuits

The question of whether computers could be built without dissipating heat was first taken seriously in the 1950s. Early researchers noted an interesting connection between information theory and thermodynamics. Both theories say that entropy increases when information about a system's state is lost. Thermodynamics says that an increase of entropy causes heat to radiate from the system. Therefore, if we can avoid losing information from a system, we can avoid giving off that form of heat.

In 1961, Rolf Landauer (1927–1999) of IBM analyzed the relationship between thermodynamic entropy and information entropy. He concluded that the erasure of a bit of information has a minimum, unavoidable energy cost. He gave the formula for the minimum energy: $kT \ln 2$, where k is the Boltzman

Peter J. Denning is Distinguished Professor of Computer Science and Director of the Cebrowski Institute for information innovation at the Naval Postgraduate School in Monterey, California, and editor of ACM Ubiquity. *The author's views expressed here are not necessarily those of his employer or the U.S. Federal Government. Ted G. Lewis is a cofounder of the Center for Homeland Defense and Security at the Naval Postgraduate School. Email for Denning: pjd@nps.edu*



Standard computers (*left, symbolized by a common logic gate used in circuits*) generate a surprising amount of waste heat from computations that have fewer outputs than inputs. If the same computations could be made reversible, with the same number of inputs and outputs (*right*), computers could use far less energy, an important consideration as more circuits are packed in.

constant (1.38 x 10^{-23} joules per kelvin) and T is the temperature in kelvins. The natural log of 2, given as ln 2, is the exponent the constant e would have to be raised to in order to equal 2, and approximately equals 0.693. In 2012 two groups experimentally confirmed Landauer's theoretical limit. According to the formula, one erased bit costs only a minuscule amount of energy; but when scaled to a computer at room temperature with 10^{12} transistors switching 10^{12} times a second, the loss becomes significant—about 3 kilowatts.

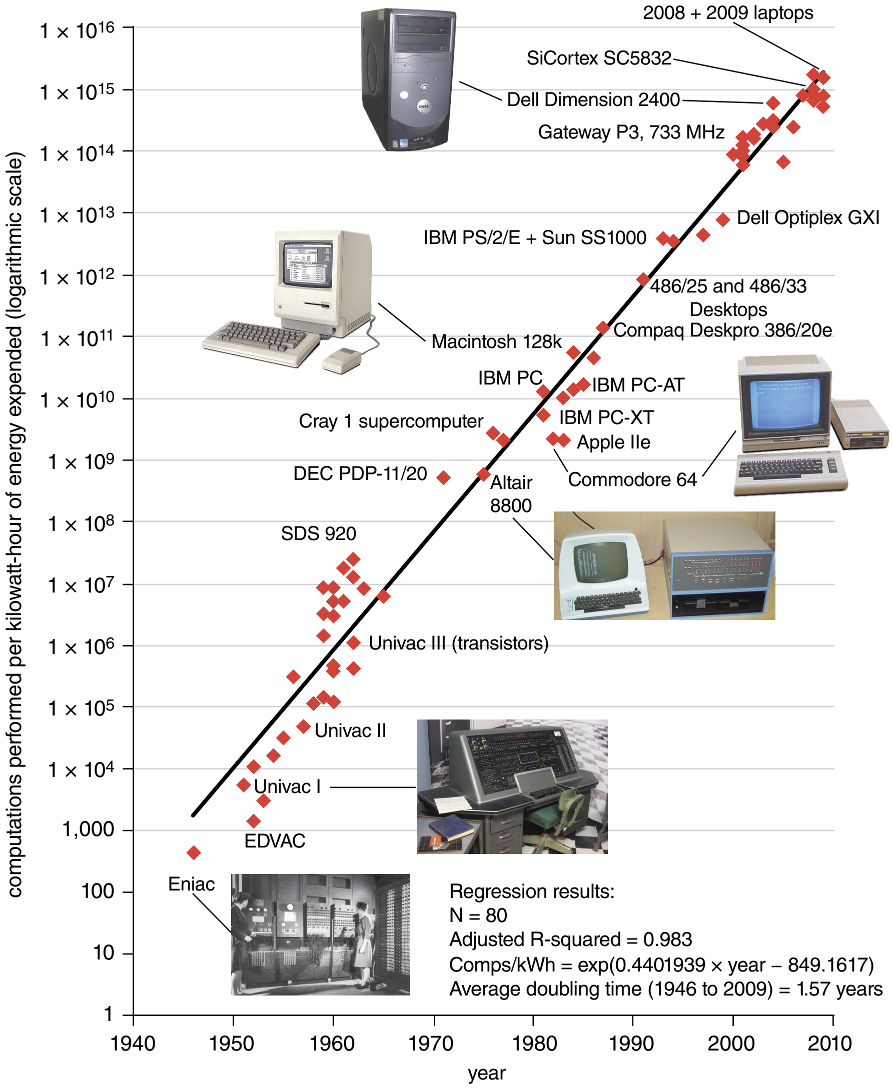
Reversible circuits could solve that problem, and if other sources of heat can be removed as well—for example, electrical resistance and the kinetic energy lost when electrons change direction—then reversible circuits would function with no heat loss. In 1973, Charles Bennett of IBM borrowed the term *adiabatic* from thermodynamics, where it means there is no heat exchange between a system and its environment. Adiabatic computers would be revolutionary because they would continue to operate indefinitely after they were initialized and started, without adding or subtracting heat energy. Adiabatic computing cannot happen without reversible circuits.

A very well insulated frictionless car is a good theoretical analog for an adiabatic computer. Once it accelerates to a cruising speed, an adiabatic car would lose no energy to friction and radiate no energy into the environment, allowing it to travel indefinitely on its momentum. It could slow down using 100 percent efficient regenerative braking and then speed up again, as long as momentum was conserved and no energy was dissipated by friction.

A reversible circuit has exactly as many outputs as inputs. Each input can be reconstructed from the output; no bits are lost, so reversible circuits will not give off heat from bit loss.

Physicist Richard Feynman showed that it is theoretically possible to create an adiabatic reversible computer. Feynman took an interest in reversible computing in the 1970s because he wanted to know whether there is a fundamental lower limit to how much energy is needed to carry out a computation. What kinds of computers attain that limit? He knew of the Landauer limit, which puts a lower bound on the amount of energy lost to heat when a single bit of information is lost. He asked: If we use reversible circuits, which lose no bits, what is the minimum energy required to get the computation done? He eventually concluded that there is no theoretical minimum.

In 1982 Edward Fredkin and Tommaso Toffoli of the Massachusetts Institute of Technology took reversible computing a step further by devising the first reversible gates. A circuit



Koomey's law, a relative of Moore's law, says that over the first 70 years of electronic computing (as represented above by typical computers of the time), computations per unit of energy have been doubling about every 1.6 years. Had this not been happening, each doubling of components would have doubled the heat to be dissipated and would have shut down Moore's law long ago. Koomey's law is also good news for mobile computing, which relies heavily on battery power.

Wikimedia Commons, Dave's Old Computers, Old Computers

made from these gates could be unambiguously backed up to its previous input. Fredkin and Toffoli also showed that their gates are universal: A computer built from them would be able to run any program that runs on a conventional computer.

Reversible computing has gathered a new following in recent years, not only because it supports energy reduction, but because it is necessary for quantum computers. In 2016 researchers at Griffith University and the University of Queensland in Australia announced they had built a quantum Fredkin-Toffoli gate using photons of light. And this year a company called D-Wave offered a 2,000-qubit quantum computer that operates at near absolute zero temperature as an adiabatic machine.

The Cost of Keeping Cool

Reversible computers may be possible in theory, but in practice, they would come at a cost. According to Feynman, zero heat dissipation is achievable only if the reversible computer operates at an infinitesimally slow speed. Even in a reversible circuit, operations that change the direction of electron flow will dissipate a small amount of heat from the change of kinetic energy of the electrons. Using slower switching speeds means less kinetic energy consumption, and is called *adiabatic switching*.

Quantum computers use photons of light to communicate signals. Because photons don't have mass, switching them does not generate kinetic energy heat loss. Feynman's caution does not apply: Quantum computers can operate much faster than circuits that switch electrons.

Although quantum Fredkin-Toffoli gates can be used to imitate classical logic gates, the first working commercial quantum computer, from D-Wave Systems, is not a general-purpose computer. It is designed to solve problems with solutions that can be represented by the minimum energy state of a system governed by a set of equations. Once the computer is given the parameter values, it settles in a few microseconds into a state representing the solution to the equations. To use the machine, the programmer has to represent the problem to be solved as a system of equations, not as a series of instructions.

To make matters more confusing, quantum computers are different from quantum dot technology. Quantum dots are a form of nanotechnology that emit resonant frequencies of light. Some researchers are investigating how to build reversible gates from quantum dots. If they succeed, they can create a reversible computer that runs at room temperature.

Some conventional uses of computers would not benefit from a reversible processor. A smartphone, for example, expends 80 percent of its energy to light the display and power the GPS and WiFi receivers. Much energy reduction research is focused on these components. However, the massive computer banks in modern cloud data centers do not need these components. A superfast, reversible quantum computer would be very valuable for saving power in these centers.

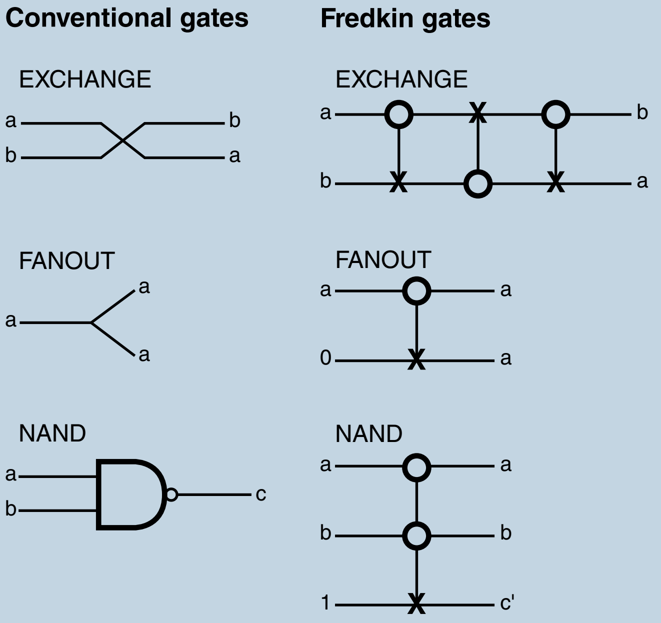
Algorithms and Energy

Algorithms pose another limit to energy that Feynman didn't consider. An algorithm specifies a series of instructions that transforms a given input into a desired output. Most algorithms expressed in standard programming languages contain many structures that are irreversible, such as conditionals, loops, jumps, and function calls. Once these structures have completed their work, you cannot determine exactly what input they had before they started. Algorithms composed of such irreversible operations can force computers to give off heat by information loss, even if the computers are made of reversible circuits. Does this put adiabatic computing out of reach?

Members of a research group at MIT, led by Nirvan Tyagi, believe they can re-

Logic Gates in Reversible Circuits

All digital computers are built of circuits using just three kinds of binary logic gates, as shown in the lefthand column at right. These circuits exist in two dimensions only. The NAND gate implements the logic function c=NOT(a AND b). It is universal because all other logic functions can be expressed solely with NANDs. Using a small network of NAND gates to avoid physical wire crossovers, the EXCHANGE gate reverses the positions of its inputs. The three corresponding Fredkin gates, which are reversible, are shown in the righthand column. A Fredkin gate must have the same number of inputs and outputs, so that there is a one-to-one map of inputs to outputs. The "X" symbol means that an incoming "1" on a vertical line inverts the horizontal signal. The "O" symbol means that a signal on a horizontal line controls another horizontal line. In the NAND, both the a and b inputs must be "1" to control the inverter on the bottom line, and the bottom line is permanently set to input "1." Tracing all the input combinations of a and b shows that the Fredkin NAND output is indeed c′=NOT(a AND b). Note that the Fredkin gates are completely reversible because if the outputs are provided at the right instead of the left, they perform the same functions.



design common irreversible algorithms into new, reversible versions, which can be expressed in a new programming language using only reversible structures. The MIT researchers have designed a new language, called EEL (Energy-Efficient Language), that restricts loops, conditionals, jumps, and function calls to reversible forms. They take a pragmatic approach and do not insist that every algorithm be made reversible; with their language they can measure how much irreversible code is generated when an EEL program is compiled into language the machine can execute. EEL can help them write programs with relatively minor amounts of irreversible code. When run on conventional computers, EEL programs will save energy because they do not require as many bit-changes as conventional algorithms.

The idea of building programs that can be reversed is not new. In the 1970s Brian Randell of Newcastle University led a group that studied how to make software more reliable. The core of their idea was to build several independent algorithms for the same function into their programs. If one algorithm failed to give the correct outcome, their system would back up to the state it was in before it started that algorithm and try a different one instead. (A similar fault-tolerance technique used in hardware is called *N-version computing*.) Interestingly, this method has the side effect of introducing the possibility of reversible computing.

In 1976, Tom Anderson, a member of Randell's team, and his colleagues built a language and operating system around units called *recovery blocks.* These modules of code take inputs, perform a computation, and test the outputs for acceptability. All inputs to all modules could be recovered by rolling recovery blocks backward.

Recovery blocks are designed to be self-contained—or, as the researchers call it, atomic. While they execute, they exchange no information with their environment. If they weren't atomic, the failure of an acceptance test would have a domino effect, causing the system to restart completely, backing up to the beginning of the program. Just before entry to a recovery block, the operating system saves a copy of the system's state in a recovery cache. This makes rollback fast and easy. The system simply pops in the previous state from the cache and restores memory to that state.

Reliability and Reversibility

Even though the EEL project comes 40 years after the recovery block project, the two systems have striking similarities. They have the same immediate goal: reversing the computation to a previous point. EEL uses the rollback mechanism to save energy; recovery blocks use it to save reliability.

This convergence raises the possibility of a relationship between reliability and reversibility. Are reversible computations inherently more reliable? Could pursuing reversibility at the algorithm and language level lead to programs of greater reliability?

These questions open new possibilities for faster, more energy-efficient, and more reliable computing. Soon we will need to find replacements for silicon chips that meet the ever-increasing demand for speed and energy conservation. Reversible computing may be one way to enjoy the benefits of both speed and low-energy consumption. But it will require a technology jump to new materials and techniques such as quantum dots, photon entanglement, or hardware and software that runs forward and backward.

Bibliography

Bennett, C. H. 1973. Logical reversibility of computation. *IBM Journal of Research and Development* 17:525–532.

Feynman, R. 2000. Lectures on Computation. Boulder, CO: Westview Press.

Frank, M. P. 2017. Foundations of generalized reversible computing. In *Proceedings of the 9th International Conference on Reversible Computation,* Kolkata, India, July 6–7, I. Phillips and H. Rahaman, eds., pp 19–34. New York, NY: Springer. DOI: 10.1007/978-3-319-59936-6 2

Fredkin, E., and T. Toffoli. 1982. Conservative Logic. *International Journal of Theoretical Physics* 21: 219–253.

Landauer, R. 1961. Irreversibility and heat generation in the computing process. *IBM Journal of Research & Development* 5:183–191.

Randell, B., P. Lee, and P. Treleaven. 1978. Reliability issues in computing system design. *ACM Computing Surveys* 10:123–165.

Tyagi, N., J. Lynch, and E. D. Demaine. 2016. Toward an energy efficient language and compiler for (partially) reversible algorithms. In *Proceedings of the 8th International Conference on Reversible Computation,* Bologna, Italy, July 7–8, S. Devitt and I. Lanese, eds., pp. 121–136. New York, NY: Springer. https://arxiv.org/abs/1605.08475