Signal entropy and the thermodynamics of computation

by N. Gershenfeld

Electronic computers currently have many orders of magnitude more thermodynamic degrees of freedom than information-bearing ones (bits). Because of this, these levels of description are usually considered separately as hardware and software, but as devices approach fundamental physical limits these will become comparable and must be understood together. Using some simple test problems, I explore the connection between the information in a computation and the thermodynamic properties of a system that can implement it, and outline the features of a unified theory of the degrees of freedom in a computer.

Computers are unambiguously thermodynamic engines that do work and generate waste heat. It is hard to miss: Across the entire spectrum of machine sizes, power and heat are among the most severe limits on improving performance. Laptops can consume 10 watts (W), enough to run out of power midway during an airline flight. Desktop computers can consume on the order of 100 W, challenging the air-cooling capacity of the CPU, and adding up to a greater load in many buildings than the entire heating, ventilation, and air-conditioning system. Further, this load is poorly characterized; MIT would need to double the size of its electrical substation if it was designed to meet the listed consumption of all the computers being used on campus.[1] And supercomputers can consume 100 kilowatts (kW) in a small volume, pushing the very limits of heat transfer to prevent them from melting.

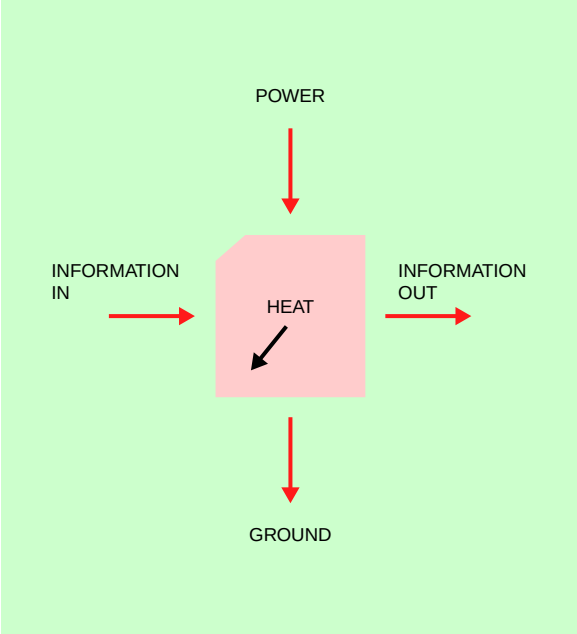
All of these problems point toward the need for computers that dissipate as little energy as possible. Current engineering practice is addressing the most obvious culprits (such as lowering the supply voltage, powering down subsystems, and developing more efficient backlights). In this paper I look ahead to consider the fundamental thermodynamic limitations that are intrinsic to the process of computation itself.

Consider a typical chip (Figure 1). Current is drawn from a power supply and returned to a ground, information enters and leaves the chip, and heat flows from the chip to a thermal reservoir. The question that I want to ask for this system is: What can be inferred about the possible thermodynamic performance of the chip from the analysis of the input and output signals? To clarify the important reasons to ask this question, this paper examines answers for some simple test cases:

- Estimating the fundamental physical limits for a given technology base, in order to understand how far it can be improved, requires insight into the expected workload.
- Short-term optimizations and long-term optimal design strategies must be based on this kind of system-dependent analysis.

The investigation of low-power computing is currently being done by two relatively disjoint camps: physicists who study the limits of single gates, but do not consider whole systems, and engineers who are making evolutionary improvements on existing

Figure 1 Fluxes into and out of a chip



designs, without considering the fundamental limits. What has been missing is attention to basic limits for practical tasks; developing low-power computing is going to require as much attention to entropy as to energy.

Bits

There has been a long-standing theoretical interest in the connection between physics and the thermodynamics of computation, reflecting the intimate connection between entropy and information. Although entropy first appeared as a way to measure heat engine efficiency, the search for a microscopic explanation helped create the fields of statistical mechanics and kinetic theory. The development of statistical mechanics raised a number of profound questions, including the paradox of Maxwell's Demon. This is a microscopic intelligent agent that apparently can violate the second law of thermodynamics by selectively opening and closing a partition between two chambers containing a gas, thereby changing the entropy of the system without doing any work on it.[2] Szilard reduced this concept to its essential features in a gas with a single molecule that can initially be on either side of the partition but then ends up on one known side.[3] Szilard's formulation introduced the notion of a bit of information, which provided the foundation for Shannon's theory of information[4] and, hence, modern coding theory. Through the study of the thermodynamics of computation, information theory is now returning to its roots in heat engines.

If all the available states of a system are equally likely (a micro-canonical ensemble) and there are Ω states, then the entropy is $k\log\Omega$. Reducing the entropy of a thermodynamic system by dS generates a heat flow of $dQ = TdS$ out of the system. Landauer[5] realized that if the binary value of a bit is unknown, erasing the bit changes the logical entropy of the system from $k\log 2$ to $k\log 1 = 0$ (shrinking the phase space available to the system has decreased the entropy by $k\log 2$). If the physical representation of the bit is part of a thermalized distribution (so that thermodynamics does apply), then this bit erasure necessarily is associated with the dissipation of energy in a heat current of $kT\log 2$ per bit. This is due solely to the act of erasure of a distinguishable bit, and is independent of any other internal energy and entropy associated with the physical representation of the bit. Bennett[6] went further and showed that it is possible to compute reversibly with arbitrarily little dissipation per operation, and that the apparent paradox of Maxwell's Demon can be explained by the erasure in the Demon's brain of previous measurements of the state of the gas (that is when the combined Demon-gas system becomes irreversible).

For many years it was assumed that no fundamental questions about thermodynamics and computation remained, and that since present and foreseeable computers dissipate so much more than $kT\log 2$ per bit, these results would have little practical significance. More recently, a number of investigators have realized that there are immediate and very important implications. A bit stored in a CMOS (complementary metal-oxide semiconductor) capacitor contains an energy of $CV^2/2$ (where V is the supply voltage and C is the gate capacitance). Erasing a bit unnecessarily wastes this energy and so should be avoided unless it is absolutely necessary. A simple calculation also shows that charging a capacitor by instantaneously connecting it to the supply voltage dissipates an additional $CV^2/2$ through the resistance of the wire carrying the current (independent of the value of the resistance), whereas charging it by linearly ramping up the supply voltage at a rate τ reduces this to approximately $CV^2\ RC/\tau$.[7] Avoiding unnecessary erasure is the subject of charge recovery and reversible logic,[8,9] and making changes

no faster than needed is the subject of adiabatic logic.[10,11] These ideas have been implemented quite successfully in conventional MOS processes, with power savings of at least an order of magnitude so far. The important point is that even though $CV^2/2$ is currently much greater than kT, by analogy many of the same concerns about erasing and moving bits apply at this much larger energy scale.

All of these results can be understood in the context of the first law of thermodynamics,

$$\begin{aligned} dU &= dW + dQ \\ &= dW + TdS \end{aligned} \tag{1}$$

The change in the internal energy of a system dU is equal to the sum of the reversible work done on it dW and the heat irreversibly exchanged with the environment $dQ = TdS$ (which is associated with a change in the entropy of the system). Integrating this gives the free energy

$$F = U - TS \tag{2}$$

which measures the fraction of the total internal energy that can reversibly be recovered to do work. Creating a bit by raising its potential energy U (as is done in charging a capacitor) stores work that remains available in the free energy of the bit and that can be recovered. Erasing a bit consumes this free energy (which charge recovery logic seeks to save) and also changes the logical entropy dS of the system; hence it is associated with dissipation (which is reduced in reversible logic).

Erasure changes the number of logical states available to the system, which has a very real thermodynamic cost. In addition, the second law of thermodynamics

$$dS \geq 0 \tag{3}$$

leads to the final principle of low-power computing. Entropy increases in spontaneous irreversible processes to the maximum value for the configuration of the system. If at any time the state of the bit is not close to being in its most probable state for the system (for example, instantaneously connecting a discharged capacitor to a power supply temporarily results in a very unlikely equilibrium state), then there will be extra dissipation as the entropy of the system increases to reach the local minimum of the free energy. Adiabatic logic reduces dissipation by always keeping the system near the maximum of the entropy. In the usual regime of linear nonequilibrium thermodynamics[12] (i.e., Ohm's law applies, and ballistic carriers can be ignored), this damping rate is given by the fluctuation-dissipation theorem in terms of the magnitude of the equilibrium fluctuations of the bit (and hence the error rate).

Fredkin and Toffoli introduced reversible gates that can be used as the basis of reversible logic families. These gates produce extra outputs, not present in ordinary logic, that enable the inputs to the gate to be deduced from the outputs (which is not possible with a conventional irreversible operation such as AND). Although this might make them appear to be useless in practice since all of these intermediate results must be stored or erased, Bennett used a pebbling argument to show that it is possible to use reversible primitives to perform an irreversible calculation with a modest space-time trade-off.[13,14] It is done by running the calculation as far forward as the intermediate results can be stored, copying the output and saving it, then reversing the calculation back to the beginning. The result is the inputs to the calculation plus the output; a longer calculation can be realized by hierarchically running subcalculations forward and then backward. This means that a steady-state calculation must pay the energetic and entropic cost of erasing the inputs and generating the outputs, but that there need be no intermediate erasure.

Therefore, an optimal computer should never need to erase its internal states. Further, the state transitions should occur no faster than they are needed, otherwise unnecessary irreversibility is incurred, and the answers should be no more reliable than they need be, otherwise the entropy is too sharply peaked and the system is over-damped. These are the principles that must guide the design of practical computers that can reach their fundamental thermodynamic limits.

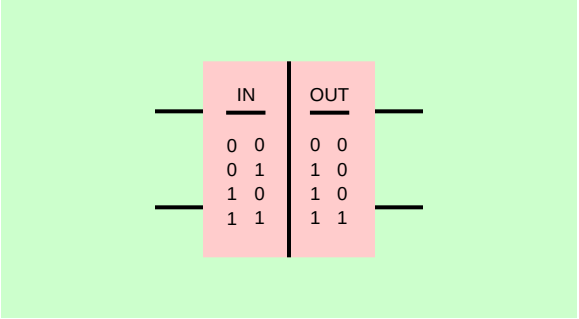
To understand the relative magnitude of these terms, let us start with a bit stored in a 5-fF capacitor (a typical CMOS number, including the gate, drain, and charging line capacitances). It has

$$\frac{(5\text{fF})(3\text{V})}{1.6\times 10^{-19} C/\text{electron}} \approx 10^5 \frac{\text{electrons}}{\text{bit}} \tag{4}$$

and at 3V this is an energy of

$$\frac{1}{2}\cdot 5\text{fF}\cdot(3\text{V})^2 = 2\times 10^{-14}\frac{\text{J}}{\text{bit}} \approx 10^5 \frac{\text{eV}}{\text{bit}} \tag{5}$$

Figure 2 A combinatorial gate that sorts its inputs, changing their entropy but not the number of zeros and ones



If 10^7 transistors (typical of a large chip) dump this energy to ground at every clock cycle (as is done in a conventional CMOS gate) at 100 MHz (a typical clock rate), the energy being dissipated is

$$2\times10^{-14}\frac{\text{J}}{\text{bit}}\cdot10^{8}\frac{\text{bits}}{\text{s}}\cdot10^{7}\text{transistors}\approx20\text{W} \tag{6}$$

This calculation is an overestimate because not all transistors switch each cycle, but the answer is of the right order of magnitude for a "hot" CPU. The electrostatic energy per bit is still seven orders of magnitude greater than the room temperature thermal energy of

$$kT = 1.38\times10^{-23}\frac{\text{J}}{\text{K}}\cdot300\text{K}\approx0.02\text{eV} \tag{7}$$

In addition to energy, entropy is associated with the distribution of electrons making up a bit because of the degrees of freedom associated with thermal excitations above the Fermi energy. It is a very good approximation to calculate this electronic entropy per bit in the Sommerfeld approximation for temperatures that are low compared to the Fermi temperature:[15]

$$S\approx\frac{N\pi^2T}{2T_f}k = \frac{10^5\pi^2300\text{K}}{2\cdot10^5\text{K}}k = 1480k\approx10^{-20}\frac{\text{J}}{\text{K}} \tag{8}$$

This entropy is not associated with a heat current if the bit is at the same temperature as the chip (remember that entropy is additive). But for comparison, if this configurational entropy was eliminated by cooling the bit, then the associated heat energy is

$$Q = TdS\approx10^{-18}\frac{\text{J}}{\text{bit}} \tag{9}$$

Finally, the dissipation associated with the logical erasure is

$$S = k\log2\approx10^{-23}\frac{\text{J}}{\text{K}} \tag{10}$$

or

$$Q = TdS\approx10^{-21}\frac{\text{J}}{\text{bit}} \tag{11}$$

We see that the heat from the thermodynamic and logical entropy associated with a bit are currently four and seven orders of magnitude lower than the electrostatic erasure energy.

Gates

Now let us turn from bits to circuits, starting first with a simple combinatorial gate (one that has no memory) with the truth table shown in Figure 2. It sorts the input bits; the output has the same energy (the same number of 0s and 1s), but the entropy is reduced by half (assuming that all inputs are equally likely):

$$H_{in} = -\sum_{states}p_{state}\log_2p_{state} = -4\times\frac{1}{4}\log_2\frac{1}{4} = 2$$

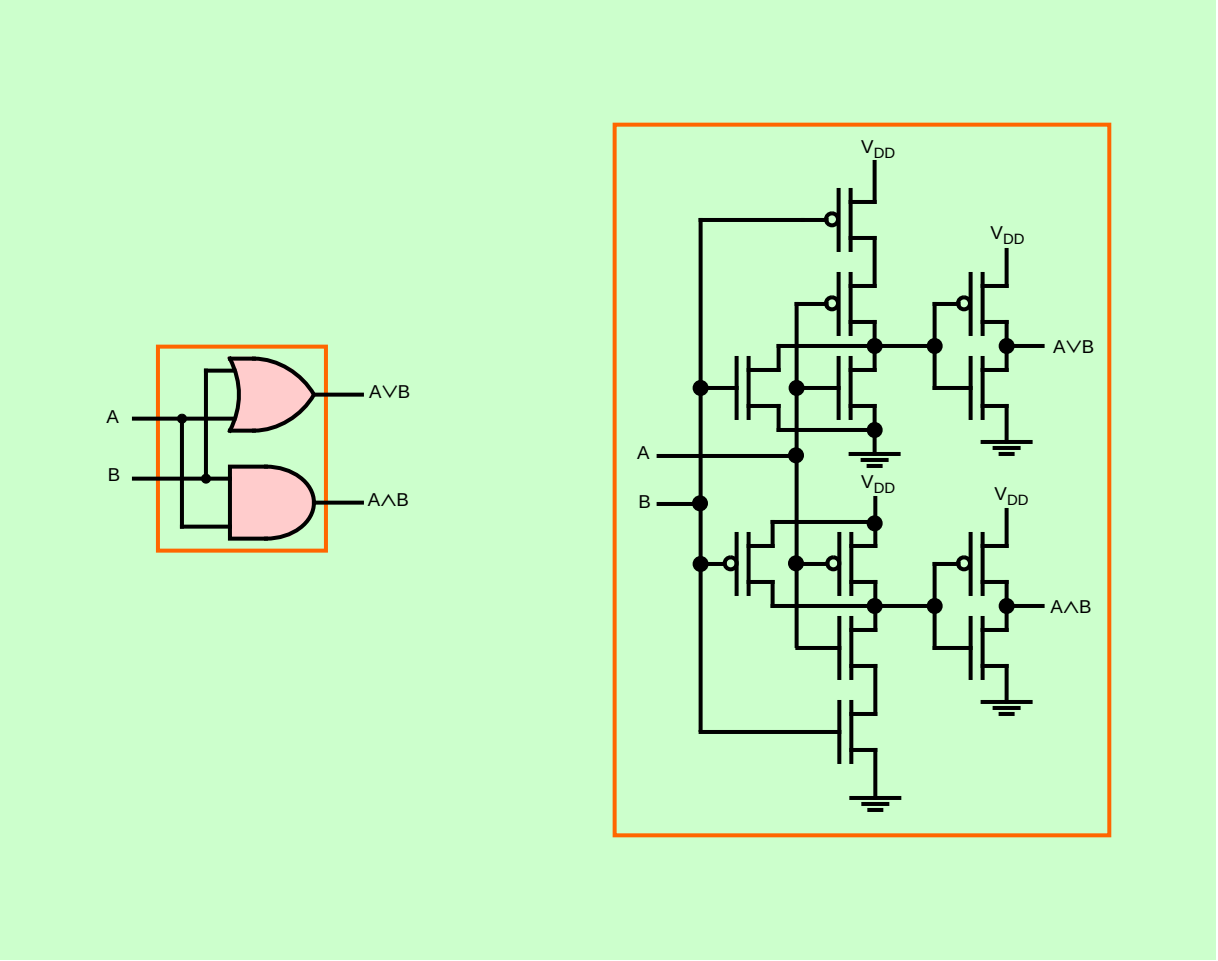
$$H_{out} = -2\times\frac{1}{4}\log_2\frac{1}{4}-\frac{1}{2}\log_2\frac{1}{2} = \frac{3}{2} \tag{12}$$

(using the convention that H measures logical entropy in bits, and S measures thermodynamic entropy in J/K). On average, at each time step the gate consumes one-half bit of entropy and so must be dissipating energy at this rate.

Where does this dissipation appear in a circuit? Figure 3 shows a conventional CMOS implementation of the two-bit sorting task by an AND and an OR gate in parallel, each of which is a combination of parallel nMOS and series pMOS FETs (field-effect transistors) or *vice versa* followed by an inverter.

This circuit dissipates energy whenever its inputs change. Any time there is a $0\rightarrow1$ transition at the input, the input capacitance (the gate electrode and the

Figure 3 A CMOS implementation of the combinatorial two-bit sort



charging line) of each FET must be charged up, and when there is a $1 \rightarrow 0$ transition, this charge is dumped to ground. The same holds true for the outputs, which charge and discharge the inputs to the following gates. Therefore, the dissipation of this circuit is given solely by the expected number of bit transitions, and the one-half bit of entropic dissipation from the logical irreversibility appears to be missing. The issue is not that it is just a small perturbation, but that it is entirely absent.

To understand how this can be, consider the possible states of the system, shown in Figure 4. The system can physically represent four possible input states (00,01,10,11), and it can represent four output states (although 01 never actually appears). When the inputs are applied to this gate, the inputs and the outputs exist simultaneously and independently (as charge stored at the input and output gates). The system is capable of representing all 16 input-output pairs, even though some of them do not occur. When it erases the inputs, regardless of the output state, it must dissipate the knowledge of which input produced the output. This merging of four possible paths into one consumes log4 = 2 bits of entropy regardless of the state. That is why the half bit is missing: each step is always destroying all of the information possible.

The real problem is that we are using irreversible primitives to do a partially reversible task. Neither

Figure 4 Possible states of the two-bit sort system

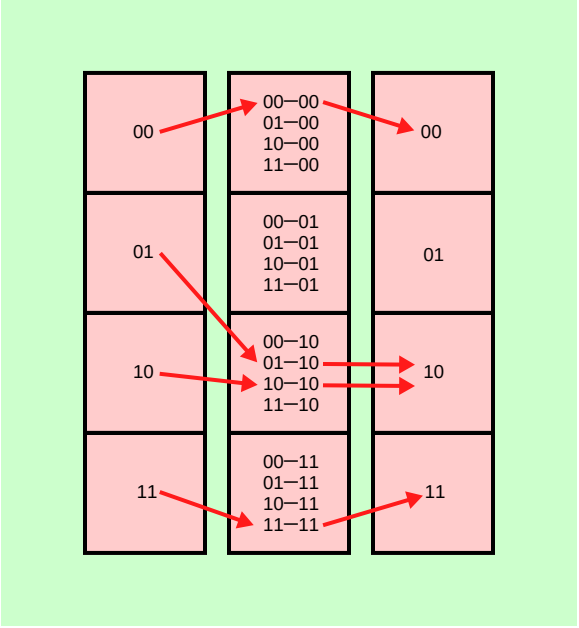
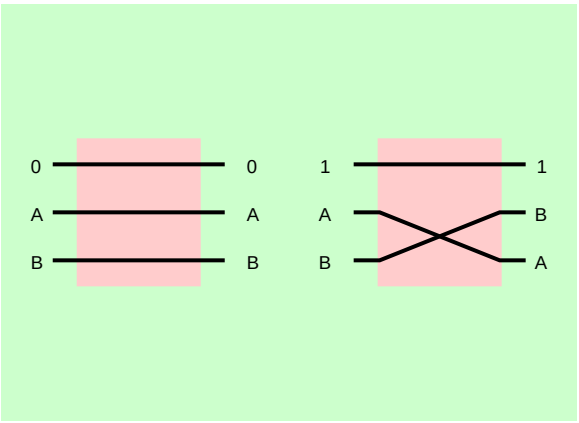


Figure 5 A Fredkin gate: the control line flips the A and B lines



gate alone can recognize when their combined actions are reversible. This naturally suggests implementing this circuit with reversible gates in order to clarify where the half bit of entropic dissipation occurs. Figure 5 shows a simple universal reversible logic element, the Fredkin gate.[16,17] Inputs that are presented to the gate (which might be packets of charge, spins, or even billiard balls in a ballistic computer) are conserved and emerge at the output, but if the control input is present, then the output lines are reversed.

Figure 6 shows our two-bit sort implemented with Fredkin gates. In addition to the AND and OR operations, there are two extra gates that are needed for the reversible version of the FANOUT operation of copying one signal to two lines. For each of these operations to be reversible, extra lines bring out the information needed to run each gate backward. Here again, the half bit of entropy from the overall logical irreversibility is not apparent. It might appear that the situation is even worse than before because of all the extra unused information that this circuit generates. However, at each time step the outputs can be copied and then everything run backwards to return to the inputs, which can then be erased.[13] This means that, as with conventional CMOS, regardless of the state we must again erase the input bits and create the output bits. The problem is now that reversible primitives are being used to implement irreversible functions, which cannot recognize the reversibility of the overall system.

The lesson to draw from these examples is that there is an entropic cost to allocating degrees of freedom, whether or not they are used. It may be known that some are not needed, but unless this knowledge is explicitly built into the system, erasure must pay the full penalty of eliminating all the available configurations.

Let us look at one final implementation of the two-bit sort (Figure 7). In this one we have recoded in a new basis with separate symbols for 00, 01, 10, 11; consider these to be balls rolling down wells with curved sides. By precomputing the input-output pairs in this way it is now obvious that two symbols pass through unchanged, and two symbols are merged. The two that are merged (01 and 10) must emerge indistinguishably at the center of the well; therefore, lateral damping is needed in that irreversible well to erase the memory of the initial condition. In the other two wells lateral damping is not needed because only one symbol will pass through it. However, it is always possible to put the ball in away from the center of the well and then damp it so that it emerges from the center. This is what has been happening in the previous examples: the system is irreversibly computing a reversible operation. This unneeded dissipation has the same maximum value for all cases, and so the logical entropy is irrelevant.

This situation is analogous to the role of entropy in a communications channel. The measured entropy gives an estimate of the average number of bits needed to communicate a symbol. However, an *a posteriori* analysis of the number of bits that were needed has no impact on the number of bits that were actually sent. An inefficient code can, of course, be used that requires many more bits; the entropic minimum is only obtained if an optimal coding is used. An example is a Fano code, a variable length code in which each extra bit distinguishes between two groups of symbols that have equal probability until a particular symbol is uniquely specified.[18] By definition, each symbol adds approximately one bit of entropy, and bits are added only as needed. Similarly, thermodynamically optimal computing should add degrees of freedom in a calculation only as needed as the calculation progresses rather than always allowing for all possibilities (this is reminiscent of using asynchronous logic for low-power design).[19] This points to the need for a computational coding theory to reduce unnecessary degrees of freedom introduced into a computation, a generalization of the algorithms such as Quine-McCluskey that are used to reduce the size of combinatorial logic.[20]

Systems

Finally, let us look at a sequential gate (one that has memory) to see how that differs from the combinatorial case. Figure 8 shows our two-bit sort implemented as a clocked CMOS gate that sorts the temporal order of pairs of bits. Now it is necessary to study the predictability of the bit strings in time: If the input to the gate is random, then 0s and 1s are equally likely at the output, but their order is no longer simply random.

It is possible to extend statistical mechanics to include the algorithmic information content in a system as well as the logical information content.[21] This beautiful theory resolves a number of statistical mechanical paradoxes but unfortunately is uncomputable because the algorithmic information in a system cannot be calculated (otherwise the halting problem could be solved). Fortunately, a natural physical constraint can make this problem tractable: if the system has a finite memory depth d over which past states can influence future ones, the conventional entropy can be estimated for a block of that length. In our example, $d = 2$.

Let us call the input at time n to the gate x_n and the output y_n. If the gate is irreversible, then there is information in x_n that cannot be predicted from y, and the

Figure 6 The two-bit sort implemented in Fredkin gates

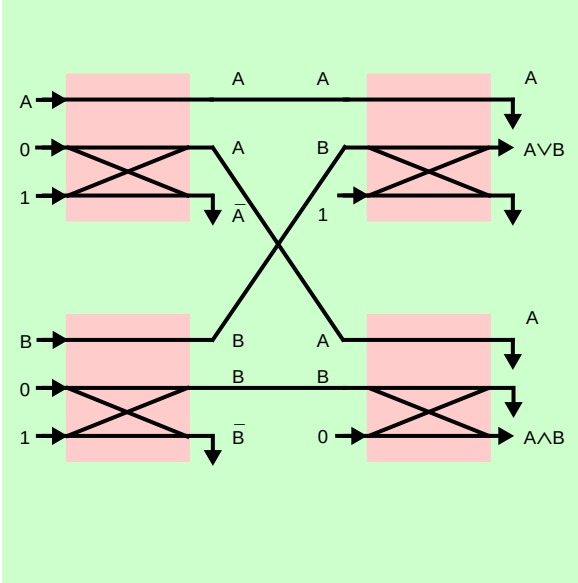
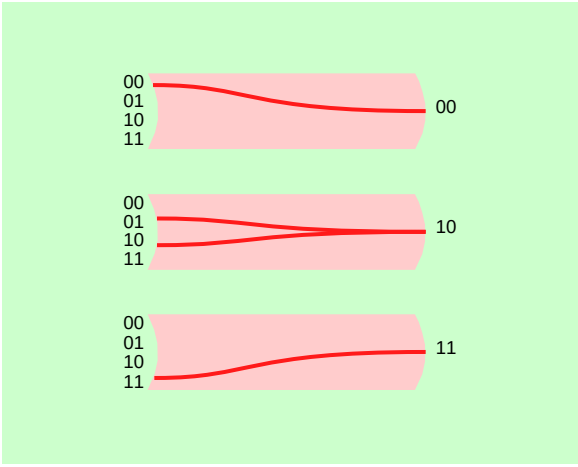
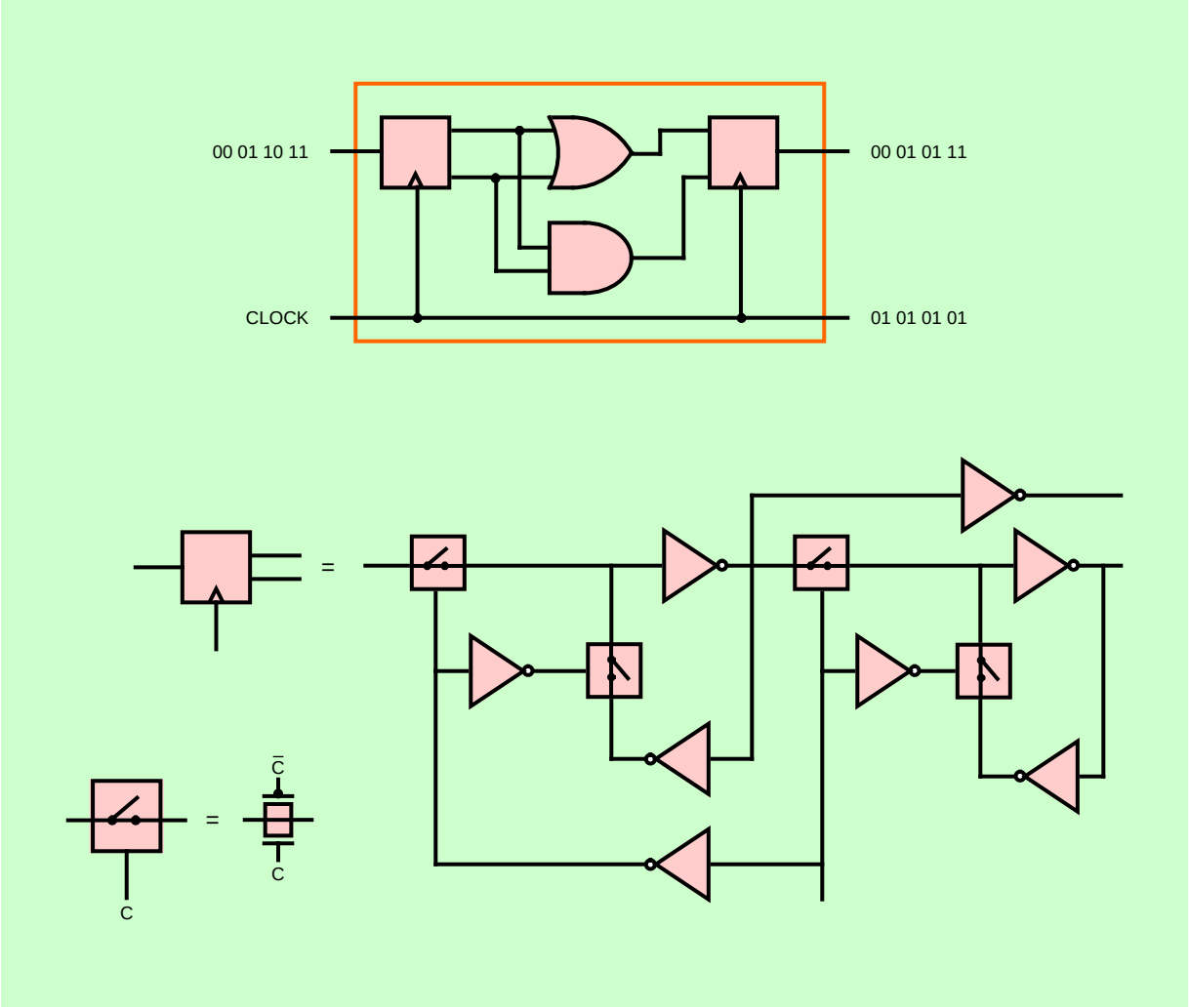


Figure 7 Alternative coding for the two-bit sort, showing the trajectories for each input



gate must dissipate this entropy decrease. Conversely, if in steady-state there is information in y_n that cannot be predicted from x or the internal initial conditions of the system, then the gate is actually a refrigerator, coupling internal thermal degrees of freedom from the heat bath to output information-bearing ones. The lat-

Figure 8 A sequential two-bit sort, using a clock and latches



ter case never happens in deterministic logic but can occur in logic that has access to a physical source of random bits.

The average information $I_{reverse}$ irreversibly destroyed per input symbol x_n can be measured by taking the difference between the entropy of a block of the d input and output samples that x_n can influence, and subtracting the entropy of the same block without x_n:

$$I_{reverse} = H(x_n, x_{n+1}, \ldots, x_{n+d}; y_n, y_{n+1}, \ldots, y_{n+d}; n) - H(x_{n+1}, \ldots, x_{n+d}; y_n, y_{n+1}, \ldots, y_{n+d}; n) \quad (13)$$

(I have also included n in case the system has an explicit time dependence). If x_n is completely predictable from the future, then these two numbers will be the same, and the mutual information $I_{reverse} = 0$. However, if x_n cannot be predicted at all, then

$$H(x_n, x_{n+1}, \ldots, x_{n+d}; y_n, y_{n+1}, \ldots, y_{n+d}; n) = H(x_n) + H(x_{n+1}, \ldots, x_{n+d}; y_n, y_{n+1}, \ldots, y_{n+d}; n) \quad (14)$$

and so

$$I_{reverse} = H(x_n) \quad (15)$$

(all of the bits in x_n are being consumed by the system). It is necessary to include the future of x as well as y in this calculation because y may not determine x, but x may not have any information content anyway (for example, the system clock is completely predictable and so carries no extra information per symbol).

In the opposite direction, the possible added information per output symbol is given by measuring its extra information relative to the symbols that can influence it:

$$I_{forward} = H(x_n, x_{n-1}, \ldots, x_{n-d}; y_n, y_{n-1}, \ldots, y_{n-d}; n) - H(x_n, x_{n-1}, \ldots, x_{n-d}; y_{n-1}, \ldots, y_{n-d}; n) \quad (16)$$

The difference between $I_{reverse}$ and $I_{forward}$ gives the net average flux of information being consumed or created by the system, and hence its informational heating (or cooling) load. It is bounded by the conventional assumption that all the input information must be destroyed. Once again, a measurement of this information difference provides only an empirical estimate of the limiting thermodynamic efficiency of a system that implements the observed transformation for the test workload and says nothing at all about any particular implementation. (The Chudnovskys[22] cannot determine if Pi is random by measuring the temperature of their computer as it prints out the digits.)

To get a feeling for these numbers, let us assume that our "hot" chip is driven by a 100 Mbit/sec network connection. RG58/U coaxial cable has a capacitance of approximately 30 pF/ft and a propagation velocity of approximately 1.5 nsec/ft, so the capacitance associated with the length of a bit is

$$\frac{1}{10^8 \frac{\text{bits}}{\text{sec}} \cdot 1.5 \times 10^{-9} \frac{\text{sec}}{\text{ft}}} \cdot 30 \frac{\text{pF}}{\text{ft}} \approx 200 \frac{\text{pF}}{\text{bit}} \quad (17)$$

and if we stay at 3V this is an energy of

$$\frac{1}{2} \cdot 200^{-12}\text{F} \cdot (3\text{V})^2 \approx 10^{-9} \frac{\text{J}}{\text{bit}} \quad (18)$$

Therefore, if the chip either consumes or generates completely random bits at this rate, the recoverable energy flux is

$$10^{-9} \frac{\text{J}}{\text{bit}} \cdot 10^8 \frac{\text{bits}}{\text{sec}} = 0.1\text{W} \quad (19)$$

for the electrostatic energy, and the irreversible heat is

$$10^{-21} \frac{\text{J}}{\text{bit}} \cdot 10^8 \frac{\text{bits}}{\text{sec}} = 10^{-13}\text{W} \quad (20)$$

for the logical entropy.

Once again, there is a huge difference between the erasure energy and entropy. However, just as the analogy between kT and $CV^2/2$ has led to the development of techniques for low-power logic, the measurement of logical entropy in computers may prove to be useful for guiding subsystem optimization.[23] Although entropy is notoriously difficult to estimate reliably in practice, there are efficient algorithms for handling the required large data sets[24] (which are easy to collect for typical digital systems).

Taken together, we have seen three kinds of terms in the overall energy/entropy budget of a chip. The first is due to the free energy of the input and output bits, which can be recovered and reused. The second is due to the relaxation of the system back to the local minimum of the free energy after a state transition, which is proportional to how many times the bits must be moved and sets a bound on how quickly and how accurately the answers become available. The final term is due to the information difference between the inputs and the outputs, which sets a bound on the thermal properties of a device that can make a given transformation for a given workload.

Concluding remarks

This paper has sketched the features of a theory of computation that can handle information-bearing and thermal degrees of freedom on an equal footing. The next step will be to extend the analysis from these simple examples to more complex systems, and to explicitly calculate the fluctuation-dissipation component associated with how sharply the entropy is peaked. This kind of explicit budgeting of entropy as well as energy is not done now but will become necessary as circuits approach their fundamental physical limits. The simple examples have shown that system-level predictability can be missed by gate-level designs, resulting in unnecessary dissipation. We have seen a very close analogy to the role of information measurements in characterizing a communications channel and then building optimal codes for it, pointing toward the possibility of "coders" that optimize chip layout for minimum dissipation over an observed

workload. Although reaching this ambitious goal still lies in the future, and may not even be of much practical importance (since in many systems power consumption is already dominated by external I/O), the required elements of measuring, modeling, and predicting the information in a system will be much more broadly applicable throughout the optimization of information processing systems.

Cited references

1. MIT Physical Plant (1994), personal communication.
2. *Maxwell's Demon: Entropy, Information, Computing*, H. S. Leff and A. F. Rex, Editors, Princeton University Press, Princeton, NJ (1990).
3. L. Szilard, *Zeitschrift für Physik* **53**, 840 (1929).
4. C. E. Shannon, "A Mathematical Theory of Communications," *Bell Systems Technical Journal* **27**, 379 (1948).
5. R. Landauer, "Irreversibility and Heat Generation in the Computing Process," *IBM Journal of Research and Development* **5**, No. 3, 183–191 (1961).
6. C. H. Bennett, "Notes on the History of Reversible Computation," *IBM Journal of Research and Development* **32**, No. 1, 16–23 (1988).
7. N. Gershenfeld, *The Physics of Information Technology*, Cambridge University Press, New York (1997), to be published.
8. R. C. Merkle, "Reversible Electronic Logic Using Switches," *Nanotechnology* **4**, 21–40 (1993).
9. S. Younis and T. Knight, "Practical Implementation of Charge Recovering Asymptotically Zero Power CMOS," *Proceedings of the 1993 Symposium on Integrated Systems*, MIT Press (1993), pp. 234–250.
10. W. C. Athas, L. J. Svensson, J. G. Koller, N. Tzartzanis, and E. Ying-Chin Chou, "Low-Power Digital Systems Based on Adiabatic-Switching Principles," *IEEE Transactions on VLSI Systems* **2**, 398–407 (1994).
11. A. G. Dickinson and J. S. Denker, "Adiabatic Dynamic Logic," *IEEE Journal of Solid-State Circuits* **30**, 311–315 (1995).
12. H. B. Callen, *Thermodynamics and an Introduction to Thermostatistics*, 2nd edition, John Wiley & Sons, Inc., New York (1985).
13. C. H. Bennett, "Time/Space Trade-offs for Reversible Computation," *SIAM Journal of Computation* **18**, 766–776 (1989).
14. R. Y. Levine and A. T. Sherman, "A Note on Bennett's Time-Space Tradeoff for Reversible Computation," *SIAM Journal of Computation* **19**, 673–677 (1990).
15. N. Aschcroft and N. D. Mermin, *Solid State Physics*, Holt, Rinehart and Winston, New York (1976).
16. R. Landauer, "Fundamental Limitations in the Computational Process," *Berichte der Bunsen Gesellschaft für Physikalische Chemie* **80**, 1048–1059 (1976).
17. E. Fredkin and T. Toffoli, "Conservative Logic," *International Journal of Theoretical Physics* **21**, 219–253 (1982).
18. R. E. Blahut, *Digital Transmission of Information*, Addison-Wesley Publishing Co., Reading MA (1990).
19. *Asynchronous Digital Circuit Design*, G. Birtwistle and A. Davis, Editors, Springer-Verlag, New York (1995).
20. F. J. Hill and G. R. Peterson, *Computer Aided Logical Design with Emphasis on VLSI*, 4th edition, John Wiley & Sons, Inc., New York (1993).
21. W. H. Zurek, "Algorithmic Randomness and Physical Entropy," *Physical Review Abstracts* **40**, 4731–4751 (1989).
22. D. Chudnovsky and G. Chudnovsky, "The Computation of Classical Constants," *Proceedings of the National Academy of Science* **86**, 8178–8182 (1989).
23. N. R. Shah and N. A. Gershenfeld, *Greedy Dynamic Page-Tuning* (1993), preprint.
24. N. A. Gershenfeld, "Information in Dynamics," Doug Matzke, Editor, *Proceedings of the Workshop on Physics of Computation*, IEEE Press, Piscataway, NJ (1993), pp. 276–280.

Accepted for publication April 25, 1996.

Neil Gershenfeld *MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: gersh@media.mit.edu).* Dr. Gershenfeld directs the Physics and Media Group at the MIT Media Lab, and is co-Principal Investigator of the Things That Think research consortium. Before joining the Massachusetts Institute of Technology faculty he received a B.A. in physics with High Honors from Swarthmore College, did research at AT&T Bell Laboratories on the use of lasers in atomic and nuclear physics, received a Ph.D. in applied physics from Cornell University studying order-disorder transitions in condensed matter systems, and was a Junior Fellow of the Harvard Society of Fellows (where he directed a Santa Fe Institute/NATO study on nonlinear time series). His research group explores the boundary between the content of information and its physical representation, including the basic physics of computation, new devices and algorithms for interface transduction, and applications in collaborations that have ranged from furniture and footwear to Yo-Yo Ma and Penn & Teller.

Reprint Order No. G321-5625.