R. Landauer

Irreversibility and Heat Generation in the Computing Process

Abstract: It is argued that computing machines inevitably involve devices which perform logical functions that do not have a single-valued inverse. This logical irreversibility is associated with physical irreversibility and requires a minimal heat generation, per machine cycle, typically of the order of kT for each irreversible function. This dissipation serves the purpose of standardizing signals and making them independent of their exact logical history. Two simple, but representative, models of bistable devices are subjected to a more detailed analysis of switching kinetics to yield the relationship between speed and energy dissipation, and to estimate the effects of errors induced by thermal fluctuations.

1. Introduction

The search for faster and more compact computing circuits leads directly to the question: What are the ultimate physical limitations on the progress in this direction? In practice the limitations are likely to be set by the need for access to each logical element. At this time, however, it is still hard to understand what physical requirements this puts on the degrees of freedom which bear information. The existence of a storage medium as compact as the genetic one indicates that one can go very far in the direction of compactness, at least if we are prepared to make sacrifices in the way of speed and random access.

Without considering the question of access, however, we can show, or at least very strongly suggest, that information processing is inevitably accompanied by a certain minimum amount of heat generation. In a general way this is not surprising. Computing, like all processes proceeding at a finite rate, must involve some dissipation. Our arguments, however, are more basic than this, and show that there is a minimum heat generation, independent of the rate of the process. Naturally the amount of heat generation involved is many orders of magnitude smaller than the heat dissipation in any practically conceivable device. The relevant point, however, is that the dissipation has a real function and is not just an unnecessary nuisance. The much larger amounts of dissipation in practical devices may be serving the same function.

Our conclusion about dissipation can be anticipated in several ways, and our major contribution will be a tightening of the concepts involved, in a fashion which will give some insight into the physical requirements for logical devices. The simplest way of anticipating our conclusion is to note that a binary device must have at least one degree of freedom associated with the information. Classically a degree of freedom is associated with kT of thermal energy. Any switching signals passing between devices must therefore have this much energy to override the noise. This argument does not make it clear that the signal energy must actually be dissipated. An alternative way of anticipating our conclusions is to refer to the arguments by Brillouin and earlier authors, as summarized by Brillouin in his book, *Science and Information Theory,*[1] to the effect that the measurement process requires a dissipation of the order of kT. The computing process, where the setting of various elements depends upon the setting of other elements at previous times, is closely akin to a measurement. It is difficult, however, to argue out this connection in a more exact fashion. Furthermore, the arguments concerning the measurement process are based on the analysis of specific models (as will some of our arguments about computing), and the specific models involved in the measurement analysis are rather far from the kind of mechanisms involved in data processing. In fact the arguments dealing with the measurement process do not define *measurement* very well, and avoid the very essential question: When is a system A coupled to a system B performing a measurement? The mere fact that two physical systems are coupled does not in itself require dissipation.

Our main argument will be a refinement of the following line of thought. A simple binary device consists of a particle in a bistable potential well shown in Fig. 1. Let us arbitrarily label the particle in the left-hand well as the ZERO state. When the particle is in the right-hand well, the device is in the ONE state. Now consider the operation   **183**

RESTORE TO ONE, which leaves the particle in the ONE state, regardless of its initial location. If we are told that the particle is in the ONE state, then it is easy to leave it in the ONE state, without spending energy. If on the other hand we are told that the particle is in the ZERO state, we can apply a force to it, which will push it over the barrier, and then, when it has passed the maximum, we can apply a retarding force, so that when the particle arrives at ONE, it will have no excess kinetic energy, and we will not have expended any energy in the whole process, since we extracted energy from the particle in its downhill motion. Thus at first sight it seems possible to RESTORE TO ONE without any expenditure of energy. Note, however, that in order to avoid energy expenditure we have used two different routines, depending on the initial state of the device. This is not how a computer operates. In most instances a computer pushes information around in a manner that is independent of the exact data which are being handled, and is only a function of the physical circuit connections.

Can we then construct a single time-varying force, $F(t)$, which when applied to the conservative system of Fig. 1 will cause the particle to end up in the ONE state, if it was initially in either the ONE state or the ZERO state? Since the system is conservative, its whole history can be reversed in time, and we will still have a system satisfying the laws of motion. In the time-reversed system we then have the possibility that for a single initial condition (position in the ONE state, zero velocity) we can end up in at least two places: the ZERO state or the ONE state. This, however, is impossible. The laws of mechanics are completely deterministic and a trajectory is determined by an initial position and velocity. (An initially unstable position can, in a sense, constitute an exception. We can roll away from the unstable point in one of at least two directions. Our initial point ONE is, however, a point of stable equilibrium.) Reverting to the original direction of time development, we see then that it is not possible to invent a single $F(t)$ which causes the particle to arrive at ONE regardless of its initial state.

If, however, we permit the potential well to be lossy, this becomes easy. A very strong positive initial force applied slowly enough so that the damping prevents oscillations will push the particle to the right, past ONE, regardless of the particle's initial state. Then if the force is taken away slowly enough, so that the damping has a chance to prevent appreciable oscillations, the particle is bound to arrive at ONE. This example also illustrates a point argued elsewhere[2] in more detail: While a heavily overdamped system is obviously undesirable, since it is made sluggish, an extremely underdamped one is also not desirable for switching, since then the system may bounce back into the wrong state if the switching force is applied and removed too quickly.

2. Classification

Before proceeding to the more detailed arguments we will need to classify data processing equipment by the means used to hold information, when it is not interacting
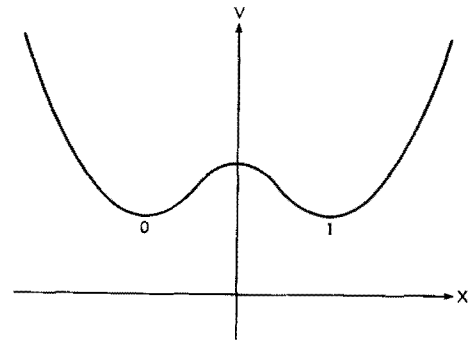

Figure 1 **Bistable potential well.**
x is a generalized coordinate representing quantity which is switched.
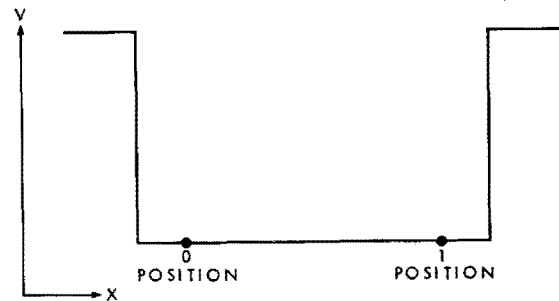

Figure 2 **Potential well in which ZERO and ONE state are not separated by barrier.**
Information is preserved because random motion is slow.

or being processed. The simplest class and the one to which all the arguments of subsequent sections will be addressed consists of devices which can hold information without dissipating energy. The system illustrated in Fig. 1 is in this class. Closely related to the mechanical example of Fig. 1 are ferrites, ferroelectrics and thin magnetic films. The latter, which can switch without domain wall motion, are particularly close to the one-dimensional device shown in Fig. 1. Cryotrons are also devices which show dissipation only when switching. They do differ, however, from the device of Fig. 1 because the ZERO and ONE states are not particularly favored energetically. A cryotron is somewhat like the mechanical device illustrated in Fig. 2, showing a particle in a box. Two particular positions in the box are chosen to represent ZERO and ONE, and the preservation of information depends on the fact that Brownian motion in the box is very slow. The reliance on the slowness of Brownian motion rather than on restoring forces is not only characteristic of cryotrons, but of most of the more familiar forms of information storage: Writing, punched cards, microgroove recording, etc. It is clear from the literature that all essential logical functions can be performed by
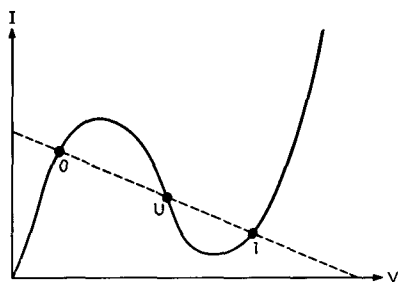
Figure 3 **Negative resistance characteristic (solid line) with load line (dashed).**
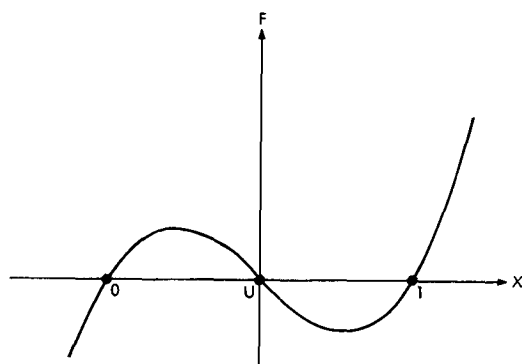ZERO *and* ONE *are stable states, U is unstable.*

Figure 4 **Force versus distance for the bistable well of Fig. 1.**
ZERO *and* ONE *are the stable states, U the unstable one.*

devices in this first class. Computers can be built that contain either only cryotrons, or only magnetic cores.[3,4]

The second class of devices consists of structures which are in a steady (time invariant) state, but in a dissipative one, while holding on to information. Electronic flip-flop circuits, relays, and tunnel diodes are in this class. The latter, whose characteristic with load line is shown in Fig. 3, typifies the behavior. Two stable points of operation are separated by an unstable position, just as for the device in Fig. 1. It is noteworthy that this class has no known representatives analogous to Fig. 2. All the active bistable devices (latches) have built-in means for restoration to the desired state. The similarity between Fig. 3 and the device of Fig. 1 becomes more conspicuous if we represent the bistable well of Fig. 1 by a diagram plotting force against distance. This is shown in Fig. 4. The line $F=0$ intersects the curve in three positions, much like the load line (or a line of constant current), in Fig. 3. This analogy leads us to expect that in the case of the dissipative device there will be transitions from the desired state, to the other stable state, resulting from thermal agitation or quantum mechanical tunneling, much like for the dissipationless case, and as has been

discussed for the latter in detail by Swanson.[5] The dissipative device, such as the single tunnel diode, will in general be an analog, strictly speaking, to an unsymmetrical potential well, rather than the symmetrical well shown in Fig. 1. We can therefore expect that of the two possible states for the negative resistance device only one is really stable, the other is metastable. An assembly of bistable tunnel diodes left alone for a sufficiently long period would eventually almost all arrive at the same state of absolute stability.

In general when using such latching devices in computing circuits one tries hard to make the dissipation in the two allowed states small, by pushing these states as closely as possible to the voltage or current axis. If one were successful in eliminating this dissipation almost completely during the steady state, the device would become a member of our first class. Our intuitive expectation is, therefore, that in the steady state dissipative device the dissipation per switching event is at least as high as in the devices of the first class, and that this dissipation per switching event is supplemented by the steady state dissipation.

The third and remaining class is a "catch-all"; namely, those devices where time variation is essential to the recognition of information. This includes delay lines, and also carrier schemes, such as the phase-bistable system of von Neumann.[6] The latter affords us a very nice illustration of the need for dissipative effects; most other members of this third class seem too complex to permit discussion in simple physical terms.

In the von Neumann scheme, which we shall not attempt to describe here in complete detail, one uses a "pump" signal of frequency ω_0, which when applied to a circuit tuned to $\omega_0/2$, containing a nonlinear reactance, will cause the spontaneous build-up of a signal at the lower frequency. The lower frequency signal has a choice of two possible phases (180° apart at the lower frequency) and this is the source of the bistability. In the von Neumann scheme the pump is turned off after the subharmonic has developed, and the subharmonic subsequently permitted to decay through circuit losses. This decay is an essential part of the scheme and controls the direction in which information is passed. Thus at first sight the circuit losses perform an essential function. It can be shown, however, that the signal reduction can be produced in a lossless nonlinear circuit, by a suitably phased pump signal. Hence it would seem adequate to use lossless nonlinear circuits, and instead of turning the pump off, change the pump phase so that it causes signal decay instead of signal growth. The directionality of information flow therefore does not really depend on the existence of losses. The losses do, however, perform another essential function.

The von Neumann system depends largely on a coupling scheme called *majority logic*, in which one couples to three subharmonic oscillators and uses the sum of their oscillations to synchronize a subharmonic oscillator whose pump will cause it to build up at a later time than the initial three. Each of the three signals which are

185

added together can have one of two possible phases. At most two of the signals can cancel, one will always survive, and thus there will always be a phase determined for the build-up of the next oscillation. The synchronization signal can, therefore, have two possible magnitudes. If all three of the inputs agree we get a synchronization signal three times as big as in the case where only two inputs have a given phase. If the subharmonic circuit is lossless the subsequent build-up will then result in two different amplitudes, depending on the size of the initial synchronization signal. This, however, will interfere with the basic operation of the scheme at the next stage, where we will want to combine outputs of three oscillators again, and will want all three to be of equal amplitude. We thus see that the absence of the losses gives us an output amplitude from each oscillator which is too dependent on inputs at an earlier stage. While perhaps the deviation from the desired amplitudes might still be tolerable after one cycle, these deviations could build up, through a period of several machine cycles. The losses, therefore, are needed so that the unnecessary details of a signal's history will be obliterated. The losses are essential for the standardization of signals, a function which in past theoretical discussions has perhaps not received adequate recognition, but has been very explicitly described in a recent paper by A. W. Lo.[7]

3. Logical irreversibility

In the Introduction we analyzed Fig. 1 in connection with the command RESTORE TO ONE and argued that this required energy dissipation. We shall now attempt to generalize this train of thought. RESTORE TO ONE is an example of a logical truth function which we shall call *irreversible*. We shall call a device *logically irreversible* if the output of a device does not uniquely define the inputs. We believe that devices exhibiting logical irreversibility are essential to computing. Logical irreversibility, we believe, in turn implies physical irreversibility, and the latter is accompanied by dissipative effects.

We shall think of a computer as a distinctly finite array of N binary elements which can hold information, without dissipation. We will take our machine to be synchronous, so that there is a well-defined machine cycle and at the end of each cycle the N elements are a complicated function of their state at the beginning of the cycle.

Our arguments for logical irreversibility will proceed on three distinct levels. The first-level argument consists simply in the assertion that present machines do depend largely on logically irreversible steps, and that therefore any machine which copies the logical organization of present machines will exhibit logical irreversibility, and therefore by the argument of the next Section, also physical irreversibility.

The second level of our argument considers a particular class of computers, namely those using logical functions of only one or two variables. After a machine cycle each of our N binary elements is a function of the state of at most two of the binary elements before the machine cycle. Now assume that the computer is logically reversi-

ble. Then the machine cycle maps the 2^N possible initial states of the machine onto the same space of 2^N states, rather than just a subspace thereof. In the 2^N possible states each bit has a ONE and a ZERO appearing with equal frequency. Hence the reversible computer can utilize only those truth functions whose truth table exhibits equal numbers of ONES and ZEROS. The admissible truth functions then are the identity and negation, the EXCLUSIVE OR and its negation. These, however, are not a complete set[8] and do not permit a synthesis of all other truth functions.

In the third level of our argument we permit more general devices. Consider, for example, a particular three-input, three-output device, i.e., a small special purpose computer with three bit positions. Let p, q, and r be the variables before the machine cycle. The particular truth function under consideration is the one which replaces r by $p \cdot q$ if $r = 0$, and replaces r by $\overline{p \cdot q}$ if $r = 1$. The variables p and q are left unchanged during the machine cycle. We can consider r as giving us a choice of program, and p, q as the variables on which the selected program operates. This is a logically reversible device, its output always defines its input uniquely. Nevertheless it is capable of performing an operation such as AND which is not, in itself, reversible. The computer, however, saves enough of the input information so that it supplements the desired result to allow reversibility. It is interesting to note, however, that we did not "save" the program; we can only deduce what it was.

Now consider a more general purpose computer, which usually has to go through many machine cycles to carry out a program. At first sight it may seem that logical reversibility is simply obtained by saving the input in some corner of the machine. We shall, however, label a machine as being logically reversible, if and only if all its individual steps are logically reversible. This means that every single time a truth function of two variables is evaluated we must save some additional information about the quantities being operated on, whether we need it or not. Erasure, which is equivalent to RESTORE TO ONE, discussed in the Introduction, is not permitted. We will, therefore, in a long program clutter up our machine bit positions with unnecessary information about intermediate results. Furthermore if we wish to use the reversible function of three variables, which was just discussed, as an AND, then we must supply in the initial programming a separate ZERO for every AND operation which is subsequently required, since the "bias" which programs the device is not saved, when the AND is performed. The machine must therefore have a great deal of extra capacity to store both the extra "bias" bits and the extra outputs. Can it be given adequate capacity to make all intermediate steps reversible? If our machine is capable, as machines are generally understood to be, of a non-terminating program, then it is clear that the capacity for preserving all the information about all the intermediate steps cannot be there.

Let us, however, not take quite such an easy way out. Perhaps it is just possible to devise a machine, useful in the normal sense, but not capable of embarking on a

186

nonterminating program. Let us take such a machine as it normally comes, involving logically irreversible truth functions. An irreversible truth function can be made into a reversible one, as we have illustrated, by "embedding" it in a truth function of a large number of variables. The larger truth function, however, requires extra inputs to bias it, and extra outputs to hold the information which provides the reversibility. What we now contend is that this larger machine, while it is reversible, is not a useful computing machine in the normally accepted sense of the word.

First of all, in order to provide space for the extra inputs and outputs, the embedding requires knowledge of the number of times each of the operations of the original (irreversible) machine will be required. The usefulness of a computer stems, however, from the fact that it is more than just a table look-up device; it can do many programs which were not anticipated in full detail by the designer. Our enlarged machine must have a number of bit positions, for every embedded device of the order of the number of program steps and requires a number of switching events during program loading comparable to the number that occur during the program itself. The setting of bias during program loading, which would typically consist of restoring a long row of bits to say ZERO, is just the type of nonreversible logical operation we are trying to avoid. Our unwieldy machine has therefore avoided the irreversible operations during the running of the program, only at the expense of added comparable irreversibility during the loading of the program.

4. Logical irreversibility and entropy generation

The detailed connection between logical irreversibility and entropy changes remains to be made. Consider again, as an example, the operation RESTORE TO ONE. The generalization to more complicated logical operations will be trivial.

Imagine first a situation in which the RESTORE operation has already been carried out on each member of an assembly of such bits. This is somewhat equivalent to an assembly of spins, all aligned with the positive z-axis. In thermal equilibrium the bits (or spins) have two equally favored positions. Our specially prepared collections show much more order, and therefore a lower temperature and entropy than is characteristic of the equilibrium state. In the adiabatic demagnetization method we use such a prepared spin state, and as the spins become disoriented they take up entropy from the surroundings and thereby cool off the lattice in which the spins are embedded. An assembly of ordered bits would act similarly. As the assembly thermalizes and forgets its initial state the environment would be cooled off. Note that the important point here is not that all bits in the assembly initially agree with each other, but only that there is a single, well-defined initial state for the collection of bits. The well-defined initial state corresponds, by the usual statistical mechanical definition of entropy, $S = k \log_e W$, to zero entropy. The degrees of freedom associated with the information can, through thermal relaxation, go to any

one of 2^N states (for N bits in the assembly) and therefore the entropy can increase by $kN \log_e 2$ as the initial information becomes thermalized.

Note that our argument here does not necessarily depend upon connections, frequently made in other writings, between entropy and information. We simply think of each bit as being located in a physical system, with perhaps a great many degrees of freedom, in addition to the relevant one. However, for each possible physical state which will be interpreted as a ZERO, there is a very similar possible physical state in which the physical system represents a ONE. Hence a system which is in a ONE state has only half as many physical states available to it as a system which can be in a ONE or ZERO state. (We shall ignore in this Section and in the subsequent considerations the case in which the ONE and ZERO are represented by states with different entropy. This case requires arguments of considerably greater complexity but leads to similar physical conclusions.)

In carrying out the RESTORE TO ONE operation we are doing the opposite of the thermalization. We start with each bit in one of two states and end up with a well-defined state. Let us view this operation in some detail.

Consider a statistical ensemble of bits in thermal equilibrium. If these are all reset to ONE, the number of states covered in the ensemble has been cut in half. The entropy therefore has been reduced by $k \log_e 2 = 0.6931 k$ per bit. The entropy of a closed system, e.g., a computer with its own batteries, cannot decrease; hence this entropy must appear elsewhere as a heating effect, supplying $0.6931 kT$ per restored bit to the surroundings. This is, of course, a minimum heating effect, and our method of reasoning gives no guarantee that this minimum is in fact achievable.

Our reset operation, in the preceding discussion, was applied to a thermal equilibrium ensemble. In actuality we would like to know what happens in a particular computing circuit which will work on information which has not yet been thermalized, but at any one time consists of a well-defined ZERO or a well-defined ONE. Take first the case where, as time goes on, the reset operation is applied to a random chain of ONES and ZEROS. We can, in the usual fashion, take the statistical ensemble equivalent to a time average and therefore conclude that the dissipation per reset operation is the same for the timewise succession as for the thermalized ensemble.

A computer, however, is seldom likely to operate on random data. One of the two bit possibilities may occur more often than the other, or even if the frequencies are equal, there may be a correlation between successive bits. In other words the digits which are reset may not carry the maximum possible information. Consider the extreme case, where the inputs are all ONE, and there is no need to carry out any operation. Clearly then no entropy changes occur and no heat dissipation is involved. Alternatively if the initial states are all ZERO they also carry no information, and no entropy change is involved in resetting them all to ONE. Note, however, that the reset operation which sufficed when the inputs were all ONE (doing

187

nothing) will not suffice when the inputs are all ZERO. When the initial states are ZERO, and we wish to go to ONE, this is analogous to a phase transformation between two phases in equilibrium, and can, presumably, be done reversibly and without an entropy increase in the universe, but only by a procedure specifically designed for that task. We thus see that when the initial states do not have their fullest possible diversity, the necessary entropy increase in the RESET operation can be reduced, but only by taking advantage of our knowledge about the inputs, and tailoring the reset operation accordingly.

The generalization to other logically irreversible operations is apparent, and will be illustrated by only one additional example. Consider a very small special-purpose computer, with three binary elements p, q, and r. A machine cycle replaces p by r, replaces q by r, and replaces r by $p \cdot q$. There are eight possible initial states, and in thermal equilibrium they will occur with equal probability. How much entropy reduction will occur in a machine cycle? The initial and final machine states are shown in Fig. 5. States α and β occur with a probability of $\frac{1}{8}$ each: states γ and δ have a probability of occurrence of $\frac{3}{8}$ each. The initial entropy was

$$S_i = k \log_e W = -k\Sigma\rho \log_e \rho$$

$$= -k\Sigma\tfrac{1}{8} \log_e \tfrac{1}{8} = 3k \log_e 2 \ .$$

The final entropy is

$$S_f = -k\Sigma\rho \log_e \rho$$

$$= -k(\tfrac{1}{8} \log \tfrac{1}{8} + \tfrac{1}{8} \log \tfrac{1}{8} + \tfrac{3}{8} \log \tfrac{3}{8} + \tfrac{3}{8} \log \tfrac{3}{8}) \ .$$

The difference $S_i - S_f$ is $1.18\ k$. The minimum dissipation, if the initial state has no useful information, is therefore $1.18\ kT$.

The question arises whether the entropy is really reduced by the logically irreversible operation. If we really map the possible initial ZERO states and the possible initial ONE states into the same space, i.e., the space of ONE states, there can be no question involved. But, perhaps, after we have performed the operation there can be some small remaining difference between the systems which were originally in the ONE state already and those that had to be switched into it. There is no harm in such differences persisting for some time, but as we saw in the discussion of the dissipationless subharmonic oscillator, we cannot tolerate a cumulative process, in which differences between various possible ONE states become larger and larger according to their detailed past histories. Hence the physical "many into one" mapping, which is the source of the entropy change, need not happen in full detail during the machine cycle which performed the logical function. But it must eventually take place, and this is all that is relevant for the heat generation argument.

5. Detailed analysis of bistable well

To supplement our preceding general discussion we shall give a more detailed analysis of switching for a system representable by a bistable potential well, as illustrated, one-dimensionally, in Fig. 1, with a barrier large com-

BEFORE CYCLE				AFTER CYCLE			FINAL STATE
p	q	r		p_1	q_1	r_1	
1	1	1	⟶	1	1	1	α
1	1	0	⟶	0	0	1	β
1	0	1	⟶	1	1	0	γ
1	0	0	⟶	0	0	0	δ
0	1	1	⟶	1	1	0	γ
0	1	0	⟶	0	0	0	δ
0	0	1	⟶	1	1	0	γ
0	0	0	⟶	0	0	0	δ

Figure 5 **Three input - three output device which maps eight possible states onto only four different states.**

pared to kT. Let us, furthermore, assume that switching is accomplished by the addition of a force which raises the energy of one well with respect to the other, but still leaves a barrier which has to be surmounted by thermal activation. (A sufficiently large force will simply eliminate one of the minima completely. Our switching forces are presumed to be smaller.) Let us now consider a statistical ensemble of double well systems with a non-equilibrium distribution and ask how rapidly equilibrium will be approached. This question has been analyzed in detail in an earlier paper,[2] and we shall therefore be satisfied here with a very simple kinetic analysis which leads to the same answer. Let n_A and n_B be the number of ensemble members in Well A and Well B respectively. Let U_A and U_B be the energies at the bottom of each well and U that of the barrier which has to be surmounted. Then the rate at which particles leave Well A to go to Well B will be of the form $vn_A \exp[-(U-U_A)/kT]$. The flow from B to A will be $vn_B \exp[-(U-U_B)/kT]$. The two frequency factors have been taken to be identical. Their differences are, at best, unimportant compared to the differences in exponents. This yields

$$\frac{dn_A}{dt} = -n_A v \exp[-(U-U_A)/kT]$$
$$+ n_B v \exp[-(U-U_B)/kT] \ ,$$

$$\frac{dn_B}{dt} = n_A v \exp[-(U-U_A)/kT]$$
$$- n_B v \exp[-(U-U_B)/kT] \ . \tag{5.1}$$

We can view Eqs. (5.1) as representing a linear transformation on (n_A, n_B), which yields $\left(\dfrac{dn_A}{dt}, \dfrac{dn_B}{dt}\right)$. What are the characteristic values of the transformation? They are:

$$\lambda_1 = 0, \quad \lambda_2 = -v \exp[(U-U_A)/kT]$$
$$- v \exp[-(U-U_B)/kT] \ .$$

The eigenvalue $\lambda_1 = 0$ corresponds to a time-independent well population. This is the equilibrium distribution

$$n_A = n_B \exp \frac{1}{kT} [U_B - U_A] .$$

The remaining negative eigenvalue must then be associated with deviations from equilibrium, and $\exp(-\lambda_2 t)$ gives the rate at which these deviations disappear. The relaxation time τ is therefore in terms of a quantity U_0, which is the average of U_A and U_B

$$\frac{1}{\tau} = \lambda_2 = \nu \exp[-(U-U_0)/kT$$
$$\cdot \{\exp[-(U_0-U_A)kT] + \exp[(U_0-U_B)kT]\} .$$

(5.2)

The quantity U_0 in Eq. (5.2) cancels out, therefore the validity of Eq. (5.2) does not depend on the definition of U_0. Letting $\Delta = \frac{1}{2}(U_A - U_B)$, Eq. (5.2) then becomes

$$\frac{1}{\tau} = 2\nu \exp[-(U-U_0)/kT] \cosh \Delta/kT .$$

(5.3)

To first order in the switching force which causes U_A and U_B to differ, $(U-U_0)$ will remain unaffected, and therefore Eq. (5.3) can be written

$$\frac{1}{\tau} = \frac{1}{\tau_0} \cosh \Delta/kT ,$$

(5.4)

where τ_0 is the relaxation time for the symmetrical potential well, when $\Delta = 0$. This equation demonstrates that the device is usable. The relaxation time τ_0 is the length of time required by the bistable device to thermalize, and represents the maximum time over which the device is usable. τ on the other hand is the minimum switching time. Cosh Δ/kT therefore represents the maximum number of switching events in the lifetime of the information. Since this can be large, the device can be useful. Even if Δ is large enough so that the first-order approximation needed to keep $U-U_0$ constant breaks down, the exponential dependence of $\cosh \Delta/kT$ on Δ, in Eq. (5.3) will far outweigh the changes in $\exp[(U-U_0)kT]$, and τ_0/τ will still be a rapidly increasing function of Δ.

Note that Δ is one-half the energy which will be dissipated in the switching process. The thermal probability distribution within each well will be about the same before and after switching, the only difference is that the final well is 2Δ lower than the initial well. This energy difference is dissipated and corresponds to the one-half hysteresis loop area energy loss generally associated with switching. Equation (5.4) therefore confirms the empirically well-known fact that increases in switching speed can only be accomplished at the expense of increased dissipation per switching event. Equation (5.4) is, however, true only for a special model and has no really general significance. To show this consider an alternative model. Let us assume that information is stored by the position of a particle along a line, and that $x = \pm a$ correspond to ZERO and ONE, respectively. No barrier is assumed to exist, but the random diffusive motion of the particle is taken to be slow enough, so that positions will be preserved for an appreciable length of

time. (This model is probably closer to the behavior of ferrites and ferroelectrics, when the switching occurs by domain wall motion, than our preceding bistable well model. The energy differences between a completely switched and a partially switched ferrite are rather small and it is the existence of a low domain-wall mobility which keeps the particle near its initial state, in the absence of switching forces, and this initial state can almost equally well be a partially switched state, as a completely switched one. On the other hand if one examines the domain wall mobility on a sufficiently microscopic scale it is likely to be related again to activated motion past barriers.) In that case, particles will diffuse a typical distance s in a time $\tau \sim s^2/2D$. D is the diffusion constant. The distance which corresponds to information loss is $s \sim a$, the associated relaxation time is $\tau_0 \sim a^2/2D$. In the presence of a force F the particle moves with a velocity μF, where the mobility μ is given by the Einstein relation as D/kT. To move a particle under a switching force F through a distance $2a$ requires a time τ_s given by

$$\mu F \tau_s = 2a ,$$

(5.5)

or

$$\tau_s = 2a/\mu F .$$

(5.6)

The energy dissipation 2Δ, is a $2aF$. This gives us the equations

$$\tau_s = 2a^2/\mu \Delta ,$$

(5.7)

$$\tau_s/\tau_0 = 4kT/\Delta ,$$

(5.8)

which show the same direction of variation of τ_s with Δ as in the case with the barrier, but do not involve an exponential variation with Δ/kT. If all other considerations are ignored it is clear that the energy bistable element of Eq. (5.4) is much to be preferred to the diffusion stabilized element of Eq. (5.8).

The above examples give us some insight into the need for energy dissipation, not directly provided by the arguments involving entropy consideration. In the RESTORE TO ONE operation we want the system to settle into the ONE state regardless of its initial state. We do this by lowering the energy of the ONE state relative to the ZERO state. The particle will then go to this lowest state, and on the way dissipate any excess energy it may have had in its initial state.

6. Three sources of error

We shall in this section attempt to survey the relative importance of several possible sources of error in the computing process, all intimately connected with our preceding considerations. First of all the actual time allowed for switching is finite and the relaxation to the desired state will not have taken place completely. If T_s is the actual time during which the switching force is applied and τ_s is the relaxation time of Eq. (5.4) then $\exp(-T_s/\tau_s)$ is the probability that the switching will not have taken place. The second source of error is the one considered in detail in an earlier paper by J. A. Swanson,[5]

189

and represents the fact that τ_0 is finite and information will decay while it is supposed to be sitting quietly in its initial state. The relative importance of these two errors is a matter of design compromises. The time T_s, allowed for switching, can always be made longer, thus making the switching relaxation more complete. The total time available for a program is, however, less than τ_0, the relaxation time for stored information, and therefore increasing the time allowed for switching decreases the number of steps in the maximum possible program.

A third source of error consists of the fact that even if the system is allowed to relax completely during switching there would still be a fraction of the ensemble of the order $\exp(-2\Delta/kT)$ left in the unfavored initial state. (Assuming $\Delta \gg kT$.) For the purpose of the subsequent discussion let us call this Boltzmann error. We shall show that no matter how the design compromise between the first two kinds of errors is made, Boltzmann error will never be dominant. We shall compare the errors in a rough fashion, without becoming involved in an enumeration of the various possible exact histories of information.

To carry out this analysis, we shall overestimate Boltzmann error by assuming that switching has occurred in every machine cycle in the history of every bit. It is this upper bound on the Boltzmann error which will be shown to be negligible, when compared to other errors. The Boltzmann error probability, per switching event is $\exp(-2\Delta/kT)$. During the same switching time bits which are not being switched are decaying away at the rate $\exp(-t/\tau_0)$. In the switching time T_s, therefore, unswitched bits have a probability T_s/τ_0 of losing their information. If the Boltzmann error is to be dominant

$$T_s/\tau_0 < \exp(-2\Delta/kT) . \tag{6.1}$$

Let us specialize to the bistable well of Eq. (5.4). This latter equation takes (6.1) into the form

$$\frac{2T_s}{\tau_s} \exp(-\Delta/kT) < \exp(-2\Delta/kT) , \tag{6.2}$$

or equivalently

$$\frac{T_s}{\tau_s} < \tfrac{1}{2} \exp(-\Delta/kT) . \tag{6.3}$$

Now consider the relaxation to the switched state. The error incurred due to incomplete relaxation is $\exp(-T_s/\tau_s)$, which according to Eq. (6.3) satisfies

$$\exp(-T_s/\tau_s) > \exp[-\tfrac{1}{2}\exp(-\Delta/kT)] . \tag{6.4}$$

The right-hand side of this inequality has as its argument $\tfrac{1}{2}\exp(-\Delta/kT)$ which is less than $\tfrac{1}{2}$. Therefore the right-hand side is large compared to $\exp(-2\Delta/kT)$, the Boltzmann error, whose exponent is certainly larger than unity. We have thus shown that if the Boltzmann error dominates over the information decay, it must in turn be dominated by the incomplete relaxation during switching.

A somewhat alternate way of arguing the same point consists in showing that the accumulated Boltzmann error, due to the maximum number of switching events per-

mitted by Eq. (5.4), is small compared to unity.

Consider now, instead, the diffusion stabilized element of Eq. (5.8). For it, we can find instead of Eq. (6.4) the relationship

$$\exp(-T_s/\tau_s) > \exp[(-\Delta/4kT)\exp(-2\Delta/kT)] , \tag{6.5}$$

and the right-hand side is again large compared to the Boltzmann error, $\exp(-2\Delta/kT)$. The alternative argument in terms of the accumulated Boltzmann error exists also in this case.

When we attempt to consider a more realistic machine model, in which switching forces are applied to coupled devices, as is done for example in diodeless magnetic core logic,[4] it becomes difficult to maintain analytically a clean-cut breakdown of error types, as we have done here. Nevertheless we believe that there is still a somewhat similar separation which is manifested.

Summary

The information-bearing degrees of freedom of a computer interact with the thermal reservoir represented by the remaining degrees of freedom. This interaction plays two roles. First of all, it acts as a sink for the energy dissipation involved in the computation. This energy dissipation has an unavoidable minimum arising from the fact that the computer performs irreversible operations. Secondly, the interaction acts as a source of noise causing errors. In particular thermal fluctuations give a supposedly switched element a small probability of remaining in its initial state, even after the switching force has been applied for a long time. It is shown, in terms of two simple models, that this source of error is dominated by one of two other error sources:

1) Incomplete switching due to inadequate time allowed for switching.

2) Decay of stored information due to thermal fluctuations.

It is, of course, apparent that both the thermal noise and the requirements for energy dissipation are on a scale which is entirely negligible in present-day computer components. The dissipation as calculated, however, is an absolute minimum. Actual devices which are far from minimal in size and operate at high speeds will be likely to require a much larger energy dissipation to serve the purpose of erasing the unnecessary details of the computer's past history.

Acknowledgments

References

1. L. Brillouin, *Science and Information Theory*, Academic Press Inc., New York, New York, 1956.

2. R. Landauer and J. A. Swanson, *Phys. Rev.*, **121**, 1668 (1961).

3. K. Mendelssohn, *Progress in Cyrogenics*, Vol. 1, Academic Press Inc., New York, New York, 1959. Chapter I by D. R. Young, p. 1.

4. L. B. Russell, *IRE Convention Record*, p. 106 (1957).

5. J. A. Swanson, *IBM Journal*, **4**, 305 (1960).
   We would like to take this opportunity to amplify two points in Swanson's paper which perhaps were not adequately stressed in the published version.

   (1) The large number of particles (~ 100) in the optimum element are a result of the small energies per particle (or cell) involved in the typical cooperative phenomenon used in computer storage. There is no question that information can be stored in the position of a single particle, at room temperature, if the activation energy for its motion is sufficiently large ($\sim$ several electron volts).

   (2) Swanson's optimum volume is, generally, not very different from the common sense requirement on U, namely: $\nu t \exp(-U/kT) \ll 1$, which would be found without the use of information theory. This indicates that the use of redundancy and complicated coding methods does not permit much additional information to be stored. It is obviously preferable to eliminate these complications, since by making each element only slightly larger than the "optimum" value, the element becomes reliable enough to carry information without the use of redundancy.

6. R. L. Wigington, *Proceedings of the IRE*, **47**, 516 (1959).

7. A. W. Lo, Paper to appear in *IRE Transactions on Electronic Computers*.

8. D. Hilbert and W. Ackermann, *Principles of Mathematical Logic*, Chelsea Publishing Co., New York, 1950, p. 10.

191