Adiabatic Switching, Low Energy Computing, and the Physics of Storing and Erasing Information

Jeffrey G. Koller & William C. Athas
koller@isi.edu & athas@isi.edu
USC Information Sciences Institute
4676 Admiralty Way, Marina Del Rey, CA 90292

Abstract

A new CMOS logic family allows the design of digital computing circuits that are more energy efficient than conventional CMOS circuits, and that become increasingly energy efficient the slower they are operated. The properties of the new logic family support Landauer's thermodynamically motivated conjecture that the only necessarily dissipative opertion in compution is the erasure of information. Also, they suggest that there is an analogous result in switching theory, which bounds below the energy dissipation in circuits containing feedback. In this paper, we sketch the principles of the new logic family, and discuss some intuitive insights which might be useful in constructing a rigorous proof of a switching-theoretic analog of Landuaer's principle.

1 Introduction

Work on the physics of computation [1,2] has led to a better understanding of the physical limits of computer technology, but has not yet influenced the way computers are built. The Adiabatic Switching project at ISI is developing digital CMOS computer circuits that use much less energy than conventional circuits, by applying fundamental principles of physics. Conversely, by dealing with real VLSI chips and real computer circuits, we have gained added insights into the energetics of computational processes such as information storage and erasure and combinational logic.

Conventional CMOS digital circuits represent information as charges stored on capacitors[4]. They dissipate energy when actively computing, because changing the value of a bit of information requires converting the signal energy into heat. The idea of adiabatic switching is to instead recycle the signal energy, save it, and later reuse it to represent other information. It turns out that this can indeed be done, but that a small fraction of the signal energy is still dissipated during the recycling process. However, the slower we operate the circuit, the smaller this fraction becomes. In fact, the characteristics of adiabatic CMOS circuits confirm the theoretical arguments of Landauer[7,8]:

1. The energy dissipation of combinational logic can be made arbitrarily small by operating the circuit slowly enough,
2. Information can be loaded into memory circuits, dissipating only an arbitrarily small amount of energy, and
3. Information can be copied with arbitrarily small energy dissipation, but
4. Erasing the last copy of a piece of information inevitably dissipates an irreducible finite amount of energy.

What is interesting is that these properties follow directly from the properties of networks of real CMOS devices, and do not rely in any way on thermodynamics arguments. In particular, the size of the irreducible energy dissipation associated with erasure is determined by the sensitivity of the CMOS switches, and there is no mention of kT, the fundamental limit predicted by thermodynamics[3,7,8].

In this paper, we will sketch the principles of adiabatic switching, and explain what the connection is between kT and the sensitivity of the switches.

2 Conventional CMOS Energetics

CMOS circuits are built from two types of transistors, pfets and nfets, which are three-terminal devices used as switches to create networks of logic gates. The control terminal (gate) of a fet is essentially a capacitor, and by putting charge on the capacitor, one can control charge flow between the other two terminals. The nfet is a "normally off" switch, in the sense that when there is no charge on the gate, there is no connection between the other two terminals (the source and the drain). When charge is placed on the gate, the source is connected to the drain, and the nfet turns on. In contrast, the pfet is "normally on": with

0-8186-3420-0/93 $03.00

no charge on the gate, the source and drain are connected, and this connection can be broken by placing charge on the gate.

How much charge is required to operate these devices? The answer is expressed in terms of a "threshold voltage" V_{th}. to turn an nfet on, the gate voltage must be raised more than V_{th} above at least one of the other terminals. Similarly, to turn a pfet on, the gate voltage must be lowered more than V_{th} below the voltage of at least one of the other terminals. The value of V_{th} depends on variables like the thickness of the insulating layer in the chip and the amount of doping, and in present-day VLSI circuits, which operate on a 5V supply, these are adjusted so that V_{th} is around 1V, allowing pfets and nfets to be turned on and off without requiring very precise voltage control.

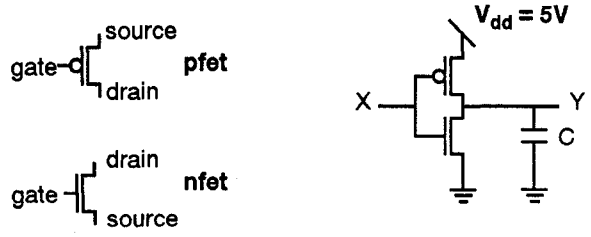

Figure 1: CMOS Inverter

Figure 1 shows the simplest CMOS switching circuit, an inverter. The load capacitance C represents the gate capacitance of the devices to which the inverter is connected. Suppose the input X to the inverter is a logical 1, i.e., 5 volts. The nfet is then on, and the pfet is off, so the output Y is connected to ground through the nfet switch, and is at 0 volts, i.e., logical 0, the negation of the input. Next, suppose the input changes to logical 0. The nfet now turns off, and the pfet turns on. Charge flows from the power supply, through the pfet, into the load capacitance. From conservation of charge and energy, one can easily calculate the amount of energy dissipated in the process: when the output is at voltage V, the capacitor has a charge Q=CV, and is storing a "signal energy" $E_s = {}^1/_2CV^2$. However, the power supply delivered an amount of energy $E_{dd} = QV = CV^2$, so the difference, $E_h = {}^1/_2CV^2$ must have been dissipated as heat in the pfet during the charging process. Note that the amount of energy dissipated is always ${}^1/_2CV^2$, independent of the on-resistance of the pfet, or its linearity, or the threshold voltage, or the charging time, or anything else. When the input changes back to 1, the load capacitor is discharged to ground, dissipating the signal energy once more, this time in the nfet. This is a characteristic of digital CMOS: a switching event always dissipates an amount of energy equal to the signal energy, because charge, initially at 5 volts, is being pumped to ground. Since any given computation involves a fixed number of switching events, it dissipates a fixed amount of energy[3].

3 Adiabatic Switching

The adiabatic charging principle provides a way to charge capacitors through a resistance without dissipating the ${}^1/_2CV^2$ of energy. Consider figure 2, where we are

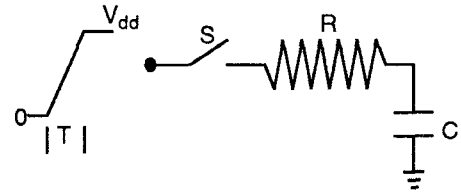

Figure 2: Adiabatic charging

charging a load capacitance through a switch. If we turn on the switch and then slowly ramp up the power supply over a time T, the energy dissipated can be calculated as follows: let the "on resistance" of the switch be R, and suppose the charging time T is much greater than RC. Then the voltage on the capacitor will follow the supply voltage closely. The charge transferred will still be Q=CV, but the average current will be small: I = CV/T. The dissipated energy can then be computed as $I^2RT = (CV/T)^2RT = (2RC/T).({}^1/_2CV^2)$, which is down from ${}^1/_2CV^2$ by a factor of 2RC/T. By making T as large as we like, we can make the dissipation as small as we like.

This result is an instance of the very general "adiabatic principle:" very slow changes to a system dissipate less energy than fast ones, because dissipation rates are proportional to the rates of change. In the limit of infinitely slow changes, the total dissipation is zero, and no heat is generated. Bennett has called computers based on this general principle "Brownian Computers," so the devices we are building are practical instantiations of Brownian Computers[6, and references in 1].

We have developed a number of ways to build logic circuits that use adiabatic charging. They have in common the use of a three-valued logic: each signal can be true, false, or in a "de-energized" state. A block of logic is normally in the powered-down state, so the outputs are de-energized, and the inputs can change without causing the block to dissipate any energy. To perform a computation, the inputs are stabilized at true or false, and the logic block is powered up adiabatically by ramping up the supply voltage. Charge flows from the supply into the outputs, causing them to move towards valid true or false values, at which point they are available for use as inputs to the next logic stage. When the succeeding stage has finished using

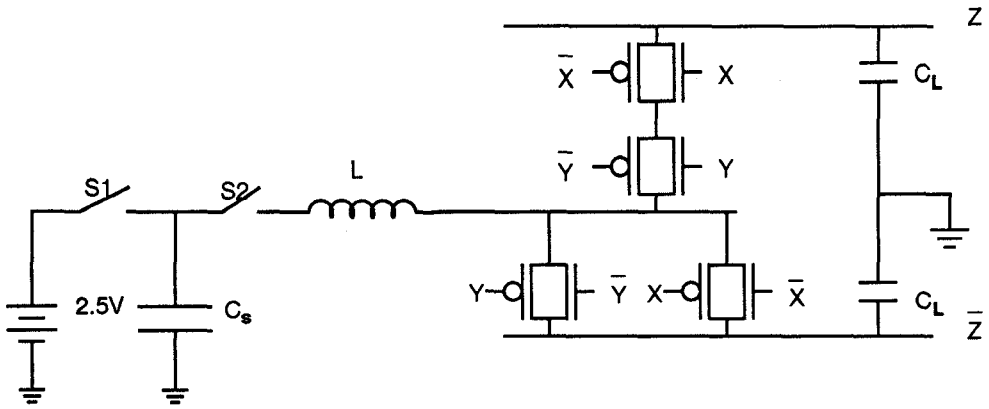

Figure 3: Adiabatic AND gate and power supply

them, the supply voltage is ramped down again, and the logic block and its outputs return to the deenergized state. The inputs are then free to change.

Figure 3 shows a logic block, complete with power supply, for one such scheme. This approach uses "dual-rail logic," where each signal X is represented by two wires, labeled $(X,\overline{X})$. The deenergized state is (0,0), and true and false are represented by (1,0) and (0,1) respectively. The circuit we have drawn is an AND gate, which in the dual rail scheme has 4 inputs $(X,\overline{X},Y,\overline{Y})$ and two outputs $(Z,\overline{Z})$. The basic element in the circuit is the transmission gate, a parallel combination of a pfet and an nfet, which acts as a better switch than either the pfet or nfet alone.

The circuit is arranged in such a way that if X and Y are both true, Z is connected to the central power rail, whereas for any other combination, $\overline{Z}$ is connected to the central power rail. Thus, to compute X AND Y, we merely allow the inputs to stabilize, then close switch S2. Charge flows through the inductor into either Z or $\overline{Z}$, correctly representing the answer. When Z or $\overline{Z}$ is fully charged to 5V, we open S2 again, and the block remains in this state. To discharge, we simply close S2 again, and let the charge drain off the output capacitors back into the power supply. This type of power supply is familiar to designers of "resonant clock-driver circuits[5]."

The energy dissipated during the charging and discharging is, up to a small numerical constant, (RC/T) times the signal energy, where $T \sim \sqrt{(LC)}$ is the charging time, and R is the resistance of the transmission gate when on. Thus, by making L large enough, we can make the energy dissipation arbitrarily small.

4 Memory

The circuit above is an example of combinational logic. The output of the block is a logical function of the inputs, and one can make this function as complicated as desired. However, to build more flexible computers, it is desirable to have memory elements, which have an internal state that can store information. Memory element are characterized by having feedback loops in their switching circuits, and the simplest one in the dual rail scheme, a one-bit latch, is shown in figure 4.

To use the latch to store the output of a block of combinational logic, we operate it as follows. Initially, the latch and the combinational logic block are both deenergized. The switches between the two are now closed. Next, both the combinational logic block and the latch are powered up with separate but synchronized voltage ramps. The outputs of the latch enter a true or false state, reflecting the output of the combinational logic block. The switches are then opened, and the combinational logic block can now be powered down. However, the outputs of the latch maintain the stored logic value as long as the latch is energized, because the outputs are fed back to two pfets to create a stable state. The energy dissipated during the loading of the latch is again inversely proportional to the charging time T, and can be made as small as desired.

To load a new value into the latch, we first have to erase the existing value in it, by returning it to the deenergized state. However, consider what happens when we ramp down the power supply: initially, charge flows off the output load capacitances through the pfets, and back into the power supply, dissipating an energy proportional to 1/T as usual. But this process stops when the output gets down to V_{th}, because the pfet turns off. In other words, a small amount of charge QV_{th} and energy $^1/_2CV_{th}^2$ remains in the

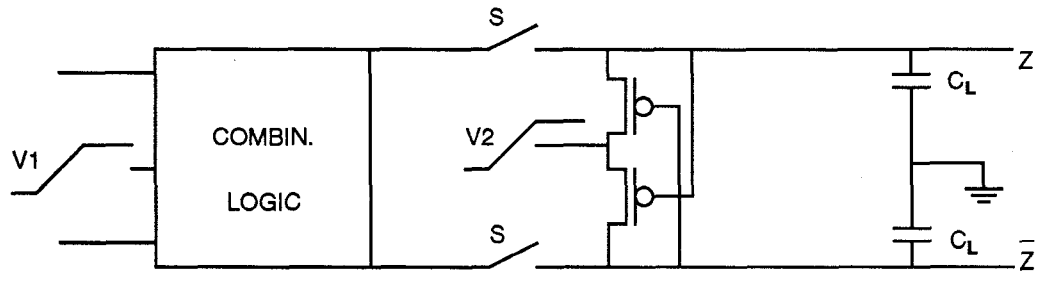

Figure 4: Adiabatic latch circuit

latch after the deenergize step. This energy is inevitably dissipated when we open the switches prior to performing the next store. Moreover, since this energy is independent of the charging time, it forms an irreducible energy penalty for the erase operation.

5 Comments

1. The irreducible energy loss is not specific to this circuit, and cannot be avoided by adding nfets, extra feedback etc. It arises because, to return the energy to the power supply without dissipating it, we must remember which output line we are discharging. However, the only record we have of that is the very piece of information we are erasing. Therefore, the system reaches the point where it has "forgotten what it is trying to forget," and cannot progress. This, in an intuitive sense, is the origin of the irreducible energy loss.

2. Because of the feedback, the smallest the load capacitance can be is the gate capacitance C_g of the pfet. Therefore, the smallest the energy loss can be is $^1/_2 C_g V_{th}^2$. We recognize this as the switch sensitivity: the amount of energy required to change the switch from an on state to an off state or vice versa. We conjecture that this is a general result: a switching circuit with internal state must dissipate an energy at least equal to the switch sensitivity for each bit of information erased.

3. What does this have to do with kT? Why can't we build circuits with very sensitive switches, and get below the kT barrier? Answer: because if kT is bigger than the switch sensitivity, thermal noise can turn switches on and off randomly, and the circuit can no longer compute reliably. Now, there are ways to build computers with redundant circuits that can operate in the presence of noise, so we have a second conjecture: the redundancy always adds enough extra switches to the circuit so that the total energy dissipation for an erase operation is greater than kT log2. In other words, if a computer dissipates an energy E_{sw} on each erase operation, it necessarily stops being a computer if the ambient temperature is raised above E_{sw} / (k log 2).

The two conjectures above should be provable entirely within the context of switching theory and information theory, and thus provide an interesting insight into the relationship between computing and physical processes.

The ideas sketched above also have very practical applications, and we are currently testing an adiabatic shift register, implemented on a 2 micron CMOS MOSIS TinyChip.

6 Acknowledgments

The research described in this paper was supported in part by the Defense Advanced Research Projects Agency under contract number DABT63-92-C-0052.

7 References

[1] Leff, H.S. and A.F. Rex (Eds.), "Maxwell's Demon, Entropy, Information, Computing," Princeton University Press, Princeton, 1990.

[2] Zurek, W.H. (Ed.), "Complexity, Entropy and the Physics of Information," Addison-Wesley, Redwood City, 1990.

[3] Mead, C. and L. Conway, "Introduction to VLSI Systems," Addison-Wesley, Reading, 1980.

[4] Weste, N.H.E. and K. Eshraghian, "Principles of CMOS VLSI Design," Addison-Wesley, Reading, 1985.

[5] Seitz, C., "Hot-Clock nMos," in Proceedings of the 1985 Chapel Hill Conference on Very Large Scale Integration, Computer Science Press, 1985.

[6] Bennett, C.H., IBM J. Res. Develop. 17, 525-532 (1973).

[7] Landauer, R., IBM J. Res. Develop. 3, 183-191 (1961).

[8] Landauer, R., Physics Today, May 1991, pp23-29.