What Are Nature's "Natural" Ways of Computing?

Tommaso Toffoli
Laboratory for Computer Science
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Cambridge, MA 02139

Abstract

Today's circuit elements are built out of physical materials, effectively characterized by macroscopic properties. *To maintain the current rate of progress in miniaturization, we'll eventually have to design computing elements based directly on* microscopic physical effects.

In order best to exploit Nature's potential for computation, we'll have to ask a lot of questions before starting to give orders. What are Nature's predispositions when it comes to computing? What types of computational processes are more naturally and directly supported by physics? What classes of computing tasks stand favored by Nature's predispositions?

1 Introduction

If you want to get rid of a person, you can take a burglar and try to turn him into a murderer by offering him a lot of money. That's not easy, and might not work. The right way, as Mario Puzo tells us in *The Godfather*, is to look for somebody with a strong predisposition—somebody who's already dying to kill—and just nudge him the right way.

What are Nature's predispositions when it comes to computing? What types of computational processes are more naturally and directly supported by physics? What classes of computing tasks stand favored by Nature's predispositions?

2 How computers are really made

Suppose we want to compute the function $y = x^2$. We can take a sheet of very smooth material and with a jigsaw carefully cut out a piece in the form of a parabola (Fig. 1). We'll then fit this parabolic cam in a jig consisting of a sliding rod and a spring-mounted finger (Fig. 2).

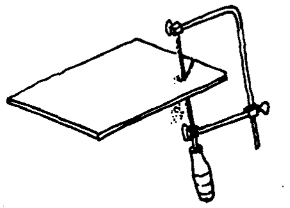

Figure 1: In the macroscopic world, one can cut an arbitrary shape out of a piece of smooth material.

As we push the rod along the x axis, the finger will move up and down along the y axis as it follows the cam, thus returning the result $y = x^2$. Do we want to compute $y = \sin x$? Easy! We just cut a new cam in the shape of Fig. 3. In a similar way, we can program our hardware to compute any reasonably well-behaved function $y = f(x)$. One may imagine computing a *composite* function such as $\sin x^2$ by cascading two such devices, one for 'square' and one for 'sine', rather than cutting a custom cam for the purpose.

When I show the above slides (Figs. 1–3), the audience gets a good laugh at this naive perversion of a computer. But—the point is—this is exactly how today's computers are made.

In any current implementation, a NAND gate is in effect a two-dimensional cam shaped as in Fig. 4. The

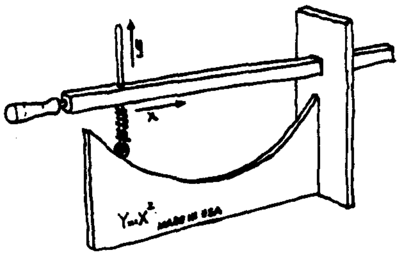

Figure 2: A spring-mounted finger traces the cam and translates the x motion of the sliding rod into a y motion realizing the function $y = x2$.

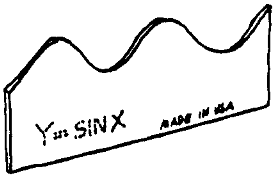

Figure 3: A cam for $\sin x$ can be cut just as easily as one for x^2.

inputs a and b and the output y are represented by voltages instead of mechanical positions, and the cam by some sort of charge transfer device, but the principle is the same. The combination of input voltages a and b probes a particular point of the cam's surface; the cam's height at this point is returned as an output voltage, y. Though this is an analog mechanism, it is designed to be usable as a *digital* mechanism: if a and b come reasonably close to one of the nominal values 0 and 1, y will come at least as close (actually, much closer) to one of these values, according to the following table

a	b	y
0	0	1
1	0	1 .
0	1	1
1	1	0

In this sense, the cam of Fig. 4 implements the NAND function. More complex logic functions are realized by cascading gates of this kind.

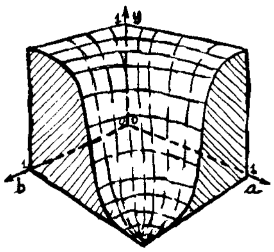

Figure 4: A two-dimensional cam implementing a two-input, one-output analog function; for distinguished input values, this function realized the Boolean NAND function.

To compute faster with a cam device, we have to make all the parts smaller: rods and fingers will then have less inertia, less distance to travel, etc. However, as we go on making everything smaller, eventually we reach a point where our "smooth" cam starts looking like Fig. 5a! The fiction that we can cut whatever shape we want out of an infinitely divisible material is gone; all we can do now is create certain specific contours by assembling discrete building blocks. The "cutting with a jigsaw" metaphor fails, and must be replaced by something like "putting together LEGO pieces" (Fig. 5b).

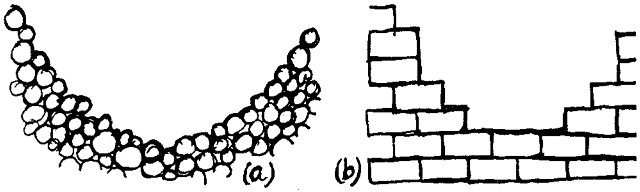

Figure 5: (a) As cams are made smaller and smaller, the discrete nature of matter starts showing. (b) Eventually, the metaphor of "cutting out an arbitrary curve out of a smooth material" must be replaced with that of "making a discrete shape by putting together LEGO pieces."

3 The macrophysics/microphysics watershed

For the last four decades, circuit technology has managed to realize ever faster and denser physical implementations of basically the same kind of sequential circuitry. That is, as linear dimensions have shrunk by a factor of about $k = 10^5$ (with an attendant increase of k in speed and a little over k^2 in density), the following features have remained by-and-large constant

- Binary digital signals.
- Logic gates having modest fan-in and fan-out (gates typically have two or three inputs, and each output typically goes to one or two places).
- Virtually complete regeneration of signals at each circuit stage.
- Overall fractal (self-similar) interconnections: block diagrams have more or less the same interconnection statistics at all levels of a circuit's hierarchy. The pattern is essentially one of random interconnections with rather rapid drop-off in the number of direct interconnections vs distance.
- Reliance on classical, deterministic, error-free behavior at the lowest logic level.

This long permanence of one design philosophy is due mainly to the fact that so far we have been using physics at the *materials* level rather than at the *particle* level. Copper, silicon, glass, and vacuum retain the same intensive properties (density, conductivity, etc.) from planetary size down to a scale of perhaps 10 nanometers (about 20 atoms across). Within this range, we can *specify* materials properties varying over a wide, continuous range of parameters (e.g., by doping), and *machine* our parts to arbitrary shapes out of these materials. In other words, in a macroscopic world one can design first, and then order custom parts to fit the design.

At the current rate of progress, in two decades we will have reached the end of that range. Beyond that point, the only parts available for making a computer will be those listed in physics' catalog of microscopic components. No orders for custom parts accepted!

For this new situation, let me sketch three possible outcomes.

Scenario 1. Just as today we can easily find a variety of materials out of which to make our gates and wires, tomorrow we'll find simple ways to make the same gates and wires directly out of discrete molecules, then of atoms, then of subatomic particles... We'll thus be able to retain the present circuit design philosophy well into the microphysics realm.

Scenario 2. After an earnest struggle, we'll conclude that the components and mechanism of microphysics are irremediably unsuitable for the construction of computing machinery.

Scenario 3. We *will* eventually discover ways to use microscopic particles and interactions for computational purposes, but only at the cost of radical innovations in our ways of thinking of computation—and perhaps also of physics.

I find the first scenario ("There is no problem!") unlikely, for the reasons mentioned above. It would be too much of a coincidence if nature always happened to have in stock just what we were about to order. The second scenario ("There is no solution!") would entail the opposite conspiracy—a catalog in which all pegs listed were round and all holes square.

The third scenario corresponds to the default assumption that Nature is *indifferent* ("Who are we, that Nature should conspire with or against us?"), and for this reason is the one I find most likely. We'll have to *learn to recognize* computational capabilities in physical activities that on first sight might look quite unrelated to anything that has to do with computing. After all, the human brain, the machinery within a cell, and an electronic computer (Fig. 6) are all capable of general-purpose computation, but they go about it in radically different ways. Computation based on microphysics may have to be based on information-processing paradigms that are just as different.

4 Making one's own the constraints of physics

In a nonlinear macroscopic system kept far away from equilibrium by a lavish energy flow, virtually anything is possible—much as in an arcade game.

However, (a) to achieve the most efficient interconnection, the machinery must be contracted into a three-dimensional lump rather than spread out in two dimensions; and (b) to reliably operate mechanisms built on a microscopic scale, the temperature must be kept low. But to keep a *three-dimensional* body *cool* throughout, energy dissipation must be kept at very low values. Thence the need to use computational mechanisms that are as far as possible *conservative* (microscopically invertible, energy and momentum conserving, etc.).

The above considerations imply that, in designing computational mechanisms at a microscopic level, the basic structural aspects of microscopic physics (symmetries and conservation laws, canonical conjugacy, quantum character, etc.) must already be present in the abstract model of computation used, rather than relegated to the implementation level. (For instance, in Fredkin's billiard-ball model of computation[4] all aspects of computation take place within a stylized version of Newtonian mechanics). As a simple example, suppose that logic 1's are physically represented by indestructible tokens; then, in the logic model for this computation, only Boolean functions that conserve the number of 1's would be admissible.

Let's go back to the LEGO metaphor introduced at the end of §2. On the "Pirates" page, the LEGO catalog lists tiny parts that look like hooks (Fig. 7a), leg-stumps, daggers, etc. and mate with more conventional LEGO blocks. With those parts, my son can synthesize a realistic Captain Hook two inches tall. Had those special parts not been available, Captain Hook's hook would have had to be made like in Fig. 7b, and the whole pirate would be as tall as me!

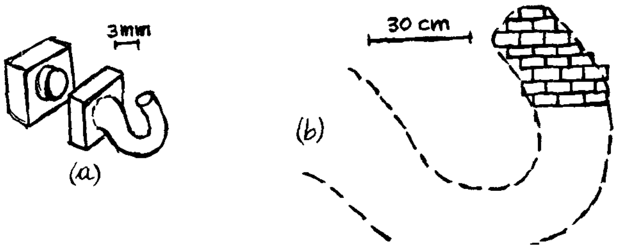

Figure 7: If LEGO didn't provide a special "pirate's hook" part (a), the hook we could make, by default, out of standard LEGO blocks would be impractically large (b).

Figure 6: (a) Neural circuitry; (b) cell machinery; and (c) computer circuitry.

On the other hand, there happen to be no "Nuns" parts—such as veils, crosses, and rosaries—in LEGO. Whether my son plays "Pirates" or "Nuns" is determined not only by his inclinations, but also by what the creators of LEGO saw fit to put in their catalog.

Coming out of the metaphor, we have to look at what is in stock in microphysics. This, of course, physicists have been doing all along. But now we are are going to do it with "computational covetousness" in our hearts; this is a new element in the picture. After all, new chapters of physics came into being when people started looking at ways of making better steam engines.

For a logician, the NAND gate is a universal Boolean primitive, and that settles the matter. Until, that is, we realize that all physical effects are microscopically reversible, while the NAND gate is not; who, then, will give us a NAND gate in the microscopic world? At that point, Boolean primitives that are logically equivalent may be revealed to be not at all equivalent from a physical viewpoint (cf. [4, 5]).

5 Is quantum mechanics a nuisance, or can it provide room for new models of computation?

Today's integrated circuits are the pride of solid-state physics—and solid-state physics is among the prides of quantum mechanics. In spite of that, the nonclassical features of quantum mechanics today play no essential role in the way we organize a computation: they are relevant to the properties of materials used to make logic devices, but are completely invisible at the circuit level.

As devices get smaller, quantum effects will become significant at or above the device level. Will the quantum nature of our world have any relevance to computation, whether in terms of constraints imposed or of new opportunities offered? In conventional models of computation, the flow of information and the structure of correlations between different parts of a system mimic those of classical mechanics. Can we base computation on systems that display at the circuit level the nonlocal correlations peculiar to quantum mechanics? Will such a feature be felt as a handicap, or will it on the contrary open up novel modes of computation?

Work on these questions is still in its infancy (cf. [2, 6]). However, the emergence of quantum cryptography[1] shows that, at least in certain contexts, quantum effects make possible information-processing feats of a fundamentally novel nature.

6 More of the same, or a different band of the computational spectrum?

A common tacit assumption is that, if realizable, computers based on microphysics will be just like the present ones, only smaller and faster.

Suppose we manage to domesticate microphysics for computational purposes. Will all classes of computation be affected in a similar way, or will certain classes be immensely favored with respect to others? If we compare today's computation to vision with our naked eyes, will microscopic computation be like a visible-light telescope, which collects and focuses much more light in the same band of the spectrum, or will it be like the invention of an X-ray telescope, that opens up a different band?

Besides speeding up certain "old" activities (such as bank accounting) the introduction of computers gave life to a myriad of "new" activities (e.g., packet switching). Microscopic computation will certainly allow us to do much faster what we are now doing anyhow; however, I venture that its significance will lie more in what other qualitatively different things it will make possible.

7 How much room is there at the bottom?

The title of this section recalls an early manifesto by Feynman[3].

On today's computers, one can run fine-grained simulation models of physical systems having 10^8 sites. Architectures like CAM-8[7, 8] can comfortably handle 10^{12} sites; and, even with today's technology, more specialized hardware could deal with perhaps 10^{16} sites (this is already the "Avogadro number in two dimensions"). When discussing with physicists the merits of such "programmable matter" computers (cf. [8]), I sometimes hear the following objection. Yes, with an assembly of 10^{16} subsystems we can perhaps model some properties of tangible matter starting from atomic-size primitives. But many physical theories (quantum gravitation, string theory, and the like) postulate primitives on a scale astronomically smaller than that of atoms. Where will we get computers able to deal with 10^{20}, 10^{30}, 10^{40} sites?

My first temptation is to recall SCENARIO 3 of §3. If there are nontrivial physical phemomena taking place over a distance of, say, 1 Fermi (10^{-15} meters), why shouldn't we be able to make computers based on those phenomena, and consequently be able to effectively simulate physics at that level of resolution? If nontrivial physics occurs at Planck's lenghth (10^{-35} meters), why shouldn't we eventually be able to compute at that scale? In other words, why should our ability to *domesticate* physics for computational purposes not be able to track our ability to *understand* physics at finer and finer levels of detail?

Here is a possible catch. Eventually, we may well be able to *design* computers based on, say, Planck-length mechanisms. However, to *operate* such computers we'll need energy flow on the same scale. If substantial deviations from equilibrium are present in the world at that scale (or can be induced by releasing some metastable states—think of nuclear energy for an analogy), then our scheme may be made to work. But if it should turn out that the world is thoroughly and irrevocably thermalized on that scale, then the bottom will be given not by how far down we can design computers, but by how far down we can find "batteries" to operate them with!

Acknowledgements

This research was supported in part by the Defense Advanced Research Projects Agency (N00014-89-J-1988), and in part by the National Science Foundation (8618002-IRI). LEGO is a registered trademark of Interlego AG.

References

[1] BENNETT, Charles, et. al., "Experimental Quantum Cryptography," *J. Cryptology* **5** (1992), 3–28.

[2] DEUTSCH, David, "Quantum theory, the Church-Turing principle and the universal quantum computer," *Proc. Royal Soc. Lond. A* **400** (1985), 97.

[3] FEYNMAN, Richard, "There is plenty of room at the bottom," reprinted in *Microminiaturization* (H. D. GILBERT, ed.), Reinhold (1961).

[4] FREDKIN, Edward, and Tommaso TOFFOLI, "Conservative Logic," *Int. J. Theor. Phys.* **21** (1982), 219–253.

[5] MARGOLUS, Norman "Physics-like models of computation," *Physica* **10***D* (1984), 81–95.

[6] MARGOLUS, Norman, "Parallel Quantum Computation," *Complexity, Entropy, and the Physics of Information* (W. H. ZUREK ed.), Addison-Wesley (1990), 273–287.

[7] MARGOLUS, Norman, and Tommaso TOFFOLI, "Cellular Automata Machines," *Lattice-Gas Methods for Partial Differential Equations* (Gary DOOLEN et al., ed.), Addison-Wesley (1990), 219–249.

[8] TOFFOLI, Tommaso, and MARGOLUS, Norman, "Programmable matter," *Physica D* **47** (1991), 263–272.