A Technology-Independent Model for Nanoscale Logic Devices

M. Frank[*]

[*]University of Florida, Depts. of CISE and ECE
CSE Bldg., Room 301, Gainesville, FL, mpf@cise.ufl.edu

ABSTRACT

We discuss how to model nanoscale logic devices without making any assumptions about what type of physical mechanism (electrical, mechanical, optical, etc.) they are based on. Starting from core facts of quantum field theory, we review how generic physical quantities such as entropy and energy relate to computational concepts such as capacity and performance. We advocate partitioning our model into subsystems playing certain generic roles. Finally, we illustrate how our device-independent perspective lets us infer strong, general facts about any future nanocomputing technology. *E.g.*, standard irreversible logic can *never* perform more than ~10^{22} bit-ops/sec per 100 Watts of power. Furthermore, achieving logic frequencies above about 9 THz (in ~20 years) will *require* performing controlled manipulations of logical bits at generalized temperatures well above room temperature, which would also allow reversible computing to achieve sub-*kT* dissipation per bit-operation despite ambient thermal noise and decoherence.

Keywords: nanocomputing, devices, compact models, fundamental limits, reversible computing, decoherence

1 INTRODUCTION

At this time, a wide variety of different mechanisms for performing digital information processing at the nanoscale have been proposed. In the literature, one finds proposals based variously on electronic, mechanical, chemical, and optical principles, and various combinations of these. Even if we narrow our attention to the all-electronic technologies, we encounter a broad range of proposed devices, based variously on semiconductors, conductors, or superconductors; field-effect transistors, resonant tunneling transistors, or Josephson junctions; quantum dots or wires; metal, silicon crystals, carbon nanotubes, and organic molecules. For encoding information, electron position, voltage, current, or spin states could be used, or even atomic and nuclear configurations, motions, and spin states. This is not to mention all the possible permutations that also utilize photonic & electromagnetic phenomena, chemical transitions, *etc.*

We would like to provide a theoretical foundation for nanocomputing that will allow us to characterize the limits of nanocomputing, as well as to analyze, compare and optimize different candidate nanocomputer architectures. But, how are we to do this, when there is such a broad range of wildly differing technologies that have been proposed, with no clear long-term winner among them? Can we bring some order to this chaos?

One approach to this problem is to develop *technology-independent* theoretical models of nanocomputing, based not on the particular design constraints of any specific technology, but on more generic physical considerations that must apply to *any* physically possible technology.

We contend is that this technology-independent modeling effort is both feasible and useful. It is feasible because all nanotechnologies are ultimately subject to the same underlying laws of physics. At the nanoscale, the relevant "gold standard" theory is quantum electrodynamics (QED), which has stood for more than 40 years now as an extremely precise underlying model for all experimentally accessible, non-gravitational phenomena involving only photons, electrons, and stable nuclei. It thus subsumes virtually all of chemical, electrical, optical, and materials science. Within the scope of its domain of applicability, QED's predictions have been empirically confirmed to as many as 11 decimal places of precision, and no clear contradictions between the theory and experiment have been found. For processes involving very high-energy interactions, and more exotic, unstable particles, QED has been successfully extended to yield the Standard Model of particle physics, which has reigned supreme for about 30 years now as the basis of all known physical phenomena (except gravity).

Modern theories such as QED and its relatives assure us that all physical systems and processes, regardless of their makeup, can ultimately be characterized in terms of a few universal, domain-independent physical concepts, such as entropy, energy, heat, temperature, and momentum.

Meanwhile, in computer engineering, we also ultimately care only about a range of other universal, technology-independent concepts, such as operating frequency (clock speed), energy dissipation, information propagation speed, information bandwidth and bandwidth density, heat flux, throughput, latency, performance, cost, and so forth.

In the end, any particular device technology (whether it involves carbon nanotube transistors, superconducting junctions, or spintronic valves) can be viewed as just being an interfacial "glue" layer, which executes a mapping (though possibly a complex one) between one essentially technology-independent domain (that of fundamental physics) and another one (that of computer engineering).

Thus, we ought to be able to model devices in *all* nanocomputing technologies generically, by abstractly characterizing how they carry out this mapping.

What are the advantages of this unified approach, as opposed to using a different, specific model for each different device technology? (One disadvantage is that a technology-specific model might well be more accurate.)

The advantages of a generic model are that:

(1) We are not forced to make a guess (which would probably be wrong anyway) concerning which specific nanocomputing technolog(y/ies) will be commercially viable 30 years from now, and thus are worth the time of developing a detailed theory for modeling them;

(2) The model can be easily adapted to quantitatively fit whatever specific nanocomputing technology does eventually become dominant.

(3) Barring an (extremely unlikely) discovery of a huge flaw in modern fundamental physics that has eluded detection by large swarms of researchers for many decades, general qualitative results obtained from our generic model *cannot become obsolete* as new device technology concepts are developed. At most, certain quantitative predictions will need to be further refined.

(4) The model provides a framework that device physicists can use to translate from the low-level characteristics of their specific technology to system-level figures of merit (*e.g.*, performance per unit cost) that will apply to a complete, large-scale digital system design that is based on those devices. This will help technology designers to steer their efforts towards the most useful technologies.

(5) The model provides a basis for nanocomputer architecture that is by and large independent of the nanocomputing technology that is used.

The general effort to develop and explore models of computing that are based soundly on universal physical principles I call *physical computing theory*. In this document, we include a brief outline of a particular theoretical physical model of computing that we are currently developing, which we call *CORP* (Computing with Optimal, Realistic Physics). We previously described CORP in a bit more detail in [1]. Later, we will discuss some results obtained from the CORP model.

One technique that has been useful in building CORP is to start by first reinterpreting fundamental physics itself in computational terms, which allows us to identify the key physical concepts that impact computation.

2 PHYSICS AS COMPUTATION

As we previously discussed in [1], all of the received quantum field theories, such as QED and the Standard Model, can be approximated to any desired accuracy using the Q3M (quantum 3d mesh) model [2], a type of parallel quantum computer consisting of a regular 3-dimensional array of cells having only a finite number of qubits per cell (representing, *e.g.*, the number of fundamental particle quanta of each type in that cell). Each cell continuously interacts locally with its nearest neighbors, exchanging particles by means of a Hamiltonian derivable from the Schrödinger equation, while simultaneously updating its internal state according to another Hamiltonian describing the interactions between fundamental particles. Such a mesh, at a fine level of granularity, can apparently accurately capture all of Standard Model physics.

In such computational models of physics, various important physical quantities can be given precise, well-defined computational meanings. There is not space to justify and detail all of these here, so we merely summarize the most important results:

(1) The physical *entropy* in any subsystem is just the amount of incompressible (non-decomputable) information in that subsystem. Information can be effectively incompressible either when it is unknown, or when it is known but random, or even (effectively) when it is known and non-random, if its underlying pattern of order is effectively inaccessible (such as an encrypted file, when the decryption key is lost). The *maximum* entropy of a system is its total physical information content, the logarithm of its number of distinguishable states. The non-entropy physical information in a system can be called *extropy*.

(2) The physical *energy* in any subsystem is the *rate of quantum physical computation* in that subsystem. (This can be given a precise meaning based on the maximum rate of rotation of quantum state vectors in Hilbert space.) The rate of useful bit-operations is R=2E/h by the Margolus-Levitin theorem [3]; *e.g.* 1 eV is 484 Tbops (trillion bit-ops per second). *Heat* is then just the energy in that part of the information that is entropy—*i.e.* it is the rate at which the random bits of physical information are changing. For example, 1 BTU (British Thermal Unit) turns out to be a rate of 3×10^{36} random bit-flips per second.

(3) The thermodynamic *temperature* of a subsystem is then, roughly speaking, the heat per bit of entropy, that is, the "clock speed" for updating of random information [4]. *E.g.*, each degree Kelvin is a frequency $f \approx 2\mathrm{K}/(hk \ln 2)$ = 28.9 GHz of bit-updating. We can generalize temperature to *generalized temperature*, which is the total energy per bit of information, even for those bits that are not entropy. A system's generalized temperature (overall physical clock speed) can be higher than its thermal temperature, although non-thermal energy tends to degrade into heat, unless the system's high-energy extropic degrees of freedom are very well-isolated from parasitic interactions that will leech off their energy into entropic (thermal) degrees of freedom.

The above observations serve as a basis for our generic technology-independent model of computation which respects all the fundamental laws of physics.

3 CORP DEVICE MODEL

CORP (Computing with Optimal, Realistic Physics) is the theoretical physical model of computation that we are developing. CORP's device model is essentially a "lumped element model" of the underlying computational model of physics described in section 2. That is, each device is a compound subsystem that may include a large number of underlying quantum bits of state. However, as a lumped system, it still has an energy, an entropy, and thermody-

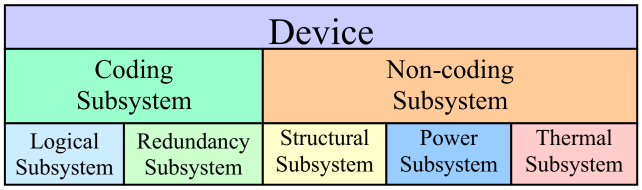

Figure 1. Conceptual hierarchy of subsystems in a CORP device. If desired, a separate *timing subsystem* can also be split out from the coding or non-coding subsystem.

namic and generalized temperatures. Further, we can still conceptualize its dynamics as decomposing into Hamiltonians for its self-interaction and its interactions with neighboring systems.

Now, not all of the degrees of freedom in a real physical device are actually used for computation. So, we break each device down into subsystems that play different roles. Figure 1 shows the conceptual structure of a device in the CORP model. The *coding subsystem* is the part of the state that is varied in a controlled way to store and manipulate logical information as part of the computation of interest. We can further divide it into the *logical subsystem*, the bits being represented, and the *redundancy subsystem*, other redundant physical bits that are included for purposes such as noise immunity and error-correction.

The rest of the device's state is its *non-coding subsystem*. We can break this into the *structural subsystem*—the part of the state that must remain unchanged if the device is to continue to operate properly—the *power subsystem*—which supplies low-entropy energy, and the *thermal subsystem*—the part of the state that is allowed to vary randomly and provides a pathway for removal of waste heat.

Each of these subsystems can be itself characterized by its energy, total physical information content (entropy plus extropy), and thermal and generalized temperatures. In addition, between each pair of subsystems within a device—as well as between it and its neighbors—there is an interaction energy and a generalized *interaction temperature* that characterize the rate at which bits of one subsystem are changed due to interactions with the other.

Ideally, all of the device's entropy is kept isolated within the thermal subsystem, although it may tend to creep into the coding subsystem (noise) or the structural subsystem (degradation) due to unwanted parasitic interactions between the thermal subsystem and the other subsystems. In general, active error correction and structural repair mechanisms (both of which are forms of refrigeration) must be used in order to keep the coding and structural subsystems clear of entropy indefinitely.

Next, we define the device's spatial geometry (region of space occupied), and identify which physical degrees of freedom within that region make up its state. This allows us to determine what assemblages of devices can exist without overlapping. Portions of the device's coding state are identified as I/O channels for communication with neighboring devices.

Finally, we summarize the device's overall computational behavior with a quantum transition function, which is a unitary map U from its input+internal logical state to its internal+output state a moment later. Such a map is necessarily invertible, so logically irreversible operations can only be implemented by extending the part of the internal state that is involved in the transformation to include not only the logical subsystem, but also the thermal subsystem. All bits discarded from the coding subsystem thus end up as entropy in the thermal subsystem.

One way to summarize the device, for technology-independent computer-engineering purposes is to specify, in addition to its transition function, also its length ℓ, area A, volume V, information capacity I, information I/O bandwidth B, maximum operation frequency f_{max}, standby power consumption P_{leak} and rate of standby entropy generation S_t, its energy and entropy coefficients E_f and S_f when doing reversible operations, its energy dissipation and entropy generation E_{i} and S_{i} for logically irreversible operations, and the maximum energy and entropy flows P_{max} and $S_{t,\mathrm{max}}$ sustainable in its power and thermal subsystems.

In previous work [2], we have used such models to show that architectures that are predominantly logically reversible are asymptotically faster and more cost-efficient than traditional irreversible architectures, for the broadest class of applications, whenever there is a fixed limit on either total system power, or on per-area entropy flux S_{At}. Irreversible machines have a fundamental limit on their performance within room-temperature environments of (100 W)/(k 300 K ln 2) = 3×10^{22} bit-operations per second, per 100 Watts of power consumption. This is only about 5 orders of magnitude beyond today's technology. Reversible computing is the only possible way to exceed this limit.

Note that these results all hold true completely independently of which domain of device technology is used (electrical, optical, mechanical, chemical). This illustrates the power of technology-independent modeling.

In the next section, we illustrate how our unified physical-computational perspective can also be applied to a more technology-specific device-level problem, of estimating the minimum entropy generation per op in field-effect devices.

4 MINIMUM ENTROPY/OP FOR FETS

For purposes of this analysis, let a device be characterized by the following independent parameters: T_{g} – Average generalized temperature for operations in the entire coding subsystem, including timing signals. E_{lb} – Energy per amount of coding-state information representing one logical bit. t_{tr} – The elapsed time of one useful logical bit-operation (transition between distinguishable states of a logical bit). t_{d} – The average time between local decoherence events for each bit within the coding subsystem. P_{lk} – Leakage power per stored logical bit. S_t – Rate of standby entropy generation per logical bit due to parasitic thermally activated transitions.

From these, we can derive the following dependent parameters: $I_{\mathrm{lb}} = E_{\mathrm{lb}}/T_{\mathrm{g}}$ – Physical information per logical bit. The dimensionless ratio $r = I_{\mathrm{lb}}$/bit is called the *redun-*

dancy factor. *E.g.*, in a voltage-coded logic circuit node, r is the number of electron states between Fermi levels at high and low voltage states. Energy per physical bit: $E_{\rm pb} = E_{\rm lb}/r = T_{\rm g}\cdot{\rm b} = kT_{\rm g}\ \ln 2$. Rate of physical computation per logical bit: $C_{\rm lb} = I_{\rm b}\cdot{\rm step}/t_{\rm tr} = I_{\rm b}\cdot({\rm op/bit})/t_{\rm tr} = r\cdot{\rm op}/t_{\rm tr} = (E_{\rm b}/T_{\rm g})$ $t_{\rm tr}$(op/bit). Rate of energy transfer (power transfer) involved in switching each bit: $P_{\rm tr} = E_{\rm b} / t_{\rm tr}$.

We are also subject to the following constraints: $I_{\rm b} \ge 1$ bit, since a bit of logical information obviously cannot be encoded in less than 1 bit of physical information. The Margolus-Levitin relation [3] tells us that the time to change each physical bit is lower-bounded by its energy, so $t_{\rm tr} \ge h/2E_{\rm pb} = h/2{\rm b}T_{\rm g}$. (The logical bit cannot change faster than its redundant physical bits can.) Thus, for example, if the generalized temperature of the coding subsystem is only 300K, then at least 0.115 ps are required to change a bit.

In a field-effect device switched over voltage V, $P_{\rm tr}/P_{\rm lk} = i_{\rm tr}V/i_{\rm lk}V = i_{\rm tr}/i_{\rm lk}$, where $i_{\rm tr}$ and $i_{\rm lk}$ are the currents during desired and leakage transitions. Now, the on/off ratio $i_{\rm tr}/i_{\rm lk} \le \exp(V/kT) \approx$[*] $\exp(E_{\rm lb}/rkT) = \exp[E_{\rm lb}/(E_{\rm lb}/T_{\rm g}\cdot{\rm bit})kT] = \exp[(\ln(2)k/k)(T_{\rm g}/T)] = 2^c$ where $c = T_{\rm g}/T$, the ratio of generalized to thermal temperature in the coding subsystem.

Decoherence will mean that that $S_t \ge I_{\rm lb}/t_{\rm d}$, and leakage will mean that $S_t \ge P_{\rm lk}/T$. Actually we can represent the total S_t as a sum of these factors, $S_t = I_{\rm lb}/t_{\rm d} + P_{\rm lk}/T$. Then, the total entropy generated over a bit-cycle is $\Delta S_{\rm lbc} = S_t t_{\rm tr} = I_{\rm lb}(t_{\rm tr}/t_{\rm d}) + P_{\rm lk}t_{\rm tr}/T$. However, from earlier we have that $P_{\rm lk} \ge P_{\rm tr}/2^c$, so $\Delta S_{\rm lbc} \ge I_{\rm lb}(t_{\rm tr}/t_{\rm d}) + P_{\rm tr}t_{\rm tr}/2^cT$. But now $P_{\rm tr} = E_{\rm b}/t_{\rm tr}$, so $\Delta S_{\rm lbc} \ge I_{\rm lb}(t_{\rm tr}/t_{\rm d}) + E_{\rm lb}/2^cT$. Since $E_{\rm b} = I_{\rm lb}T_{\rm g}$, and $T_{\rm g}/T = c$, we get:

$$\Delta S_{\rm lbc} \ge I_{\rm lb}\left(\frac{t_{\rm tr}}{t_{\rm d}}+\frac{c}{2^c}\right) \ge 1\,{\rm bit}\cdot\left(\frac{1}{q}+\frac{c}{2^c}\right) \tag{1}$$

where $q = t_{\rm d}/t_{\rm tr}$ is the quantum *quality factor* and c is the *coding speedup*. The value of q can also be expressed as $T_{\rm g}/T_{\rm d}$ where $T_{\rm d}$ is the *decoherence temperature*, which is the decoherence rate per bit, or in other words the interaction temperature between coding and non-coding subsystems.

Note that this expression for entropy generation can take on arbitrarily small values, but that this requires that both the $1/q$ and $c/2^c$ terms be made comparably small. Fortunately, both terms can be made small simultaneously by making $T_{\rm g}$—the generalized temperature of the coding system—large relative to both T and $T_{\rm d}$.

In other words, perhaps counter-intuitively, in order to minimize the entropy generated per logical bit-operation in field-effect devices, the energy per physical bit in the coding system should be made large, relative to both thermal and decoherence temperatures in the system.

Intuitively, the coding temperature $T_{\rm g}$ needs to be larger than the decoherence temperature $T_{\rm d}$ so that decoherence events don't have time to happen over the course of a logic operation. Meanwhile, it also needs to be larger than the thermal temperature, in order to suppress thermally-activated leakage of electrons over the potential energy barriers set up in the field effect devices.

We argue that a closely analogous scaling analysis ought to still hold true in any digital switching technology, since this will always involve the raising and lowering of potential energy barriers between states, activated by transitions occurring between states in other similar devices.

5 CONCLUSIONS

The laws of physics can ultimately be understood from a computational perspective. Similarly, computation can be fundamentally described in terms of physical concepts. By using universal physical concepts such as entropy, energy, and temperature, we can compose theoretical models of nanocomputing devices in a way that is independent of the particular nanotechnology that is being used, and obtain results that will apply to all future nanotechnologies.

In this document, we briefly outlined two types of results that have already been obtained using these methods. The first was a high-level computer systems engineering analysis showing that in the long run, reversible computing, if possible, is more cost-effective than irreversible computing. The second was a demonstration that reversible computing with arbitrarily little entropy generation per operation is indeed possible in a readily generalizable model of switched FET-like devices, even when accounting for decoherence effects and thermally-activated leakage, so long as the generalized temperature in the coding system—the maximum rate of transitions per bit—can be made large relative to ambient rates of decoherence and thermal transitions. At room temperature (300 K), thermal bits change at a rate of (300 K)(1 bit)/(h/2) = 8.7 THz. If we (reasonably) assume that decoherence temperatures are also at around this level, then present-day GHz-speed computers are still more than 3 orders of magnitude away from the point where sub-kT computing becomes manifestly possible according to this analysis. Historically, a factor of 1,000 in frequency takes only about 20 years to achieve. But, whenever we reach this point, reversible computing principles will be absolutely vital in order to make any further progress in nanocomputer power-performance beyond it, regardless of our choice of device technology.

REFERENCES

[1] Michael P. Frank, "Nanocomputer Systems Engineering," Nanoengineering World Forum, International Engineering Consortium, Marlborough, MA, June 23-25, 2003.

[2] Michael P. Frank, "Reversibility for Efficient Computing," Ph.D. thesis, MIT, June 1999.

[3] Norman H. Margolus and Lev B. Levitin, "The maximum speed of dynamical evolution," Physica D 120, 188-195, 1998.

[4] Seth Lloyd, "Ultimate physical limits to computation," Nature 406, 1047-1054, 2000.

[*] $V \approx E_{\rm lb}/r$ because the density of states increases with energy, so the majority of the r electron states will have energy closer to the voltage-V Fermi level than to the ground level.