Nanocomputer Systems Engineering

M. Frank*

*University of Florida, CISE Department, CSE Building
Room 301, Gainesville, FL, USA, mpf@cise.ufl.edu

ABSTRACT

In this paper, we outline how to perform an integrated systems-level analysis of nanocomputer cost-efficiency, which takes into account issues of manufacturing and operating cost, power delivery and heat removal, synchronization, error correction, communications, logic architecture, and algorithms, including a consideration of the possibility of using *reversible* and *quantum* computing techniques [1,2]. Our analysis uses a generic, technology-independent model that takes all the relevant fundamental physical considerations into account. By inserting some reasonable assumptions about future technological trends into our model, we obtain a prediction that reversible computer technologies will gradually become ~1,000 to 100,000 times more cost-efficient than ordinary irreversible technologies (depending on the application) over the course of the next 50 years. It seems that increased attention to reversible computing is warranted.

Keywords: nanocomputers, systems engineering, reversible computing, energy efficiency, cost-efficiency.

1 INTRODUCTION

Regardless of which nano-device technology ends up being most viable for general-purpose computing, the design of densely-packed computer systems comprised of nanometer-scale bit-devices offers new challenges for the field of computer systems engineering. Optimizing the cost-efficiency of high-performance nanocomputing systems requires a unified consideration of concerns in many areas, including hardware cost, power delivery and heat removal, synchronization, error correction, communications, logic architectures, and hardware and software algorithms, including the possible impact of reversible and quantum architectures and algorithms.

In this paper, we survey some of the key constraints and modeling aspects that will be essential for carrying out this important future enterprise of *nanocomputer systems engineering*. We show that useful analytical results can already be obtained regarding the nature and scalability of the optimal computer architectures and algorithms, results that are qualitatively nearly independent of the particular choice of device technology. The overall cost-efficiency of a system design can be expressed as a quantitative function of generic, abstract bit-device parameters, giving device physicists a valuable tool that they can use to compare and optimize their low-level device mechanisms so as to obtain the best overall system-level cost-efficiency.

One interesting lesson that arises from our analyses is that in a wide range of plausible future application/technology scenarios, the most cost-efficient possible system designs turn out to *require* the extensive use of almost thermodynamically reversible (adiabatic) device operation mechanisms, organized into nearly logically reversible (invertible) digital algorithms [1]. The advantages of this unusual approach are real, and furthermore, the advantages increase (albeit slowly) as the quality of individual devices and/or the total number of devices in the system are increased.

Although some related results concerning the cost-efficiency advantages of reversibility have been described in previous publications and research memos by the author [1,3,4,5], the present paper and its accompanying memo [6] are the first to integrate various factors in a single model quantifying the cost-efficiency benefits of reversibility as technology improves.

2 FUNDAMENTAL CONSTRAINTS

Any physically realistic model of computation must respect the following fundamental physical constraints. Of course, more restrictive limits than these will exist in any specific technology.

1. Information density (bits per volume) is limited, by a function of energy density and temperature [7,8].
2. The velocity of propagation of information from one point to another is limited by the speed of light [9].
3. The rate at which operations may be performed is at most proportional to the available energy [10,8].
4. The total physical information (including entropy) in any closed, constant-volume system is conserved. (Information can be moved and transformed, but not destroyed.) [11]
5. The energy loss to dispose of unwanted information has a lower bound, given the temperature of the destination thermal reservoir [12].
6. There is always a >0 rate of entropy increase in any non-equilibrium system. (2^{nd} law of thermo.)
7. There is an upper bound on size for a system with given energy density, due to general relativity. [13]

Contrary to some popular misconceptions, quantum computing and communication techniques, quantum "teleportation," *etc.*, do *not* let us exceed any of the above fundamental physical limits. Rather, the main impact of quantum information technology on computation is simply

that the use of intermediate *superposition* states opens up "short-cut" paths through state space that allow some algorithms to be carried out using many fewer total steps than seems to be possible classically. [2]

In the past, many models of computation and computer performance have been proposed that can be shown to implicitly violate one or more of the above physical limits. These include: RAM machines, PRAM (parallel RAM) machines, pointer machines, some parallel interconnection models (*e.g.*, binary trees, fat trees, hypercubes, butterfly networks), and three-dimensional meshes of irreversible devices [1]. We claim that the best-performing *physically-realistic* model of computation appears to be a 3d mesh of asymptotically reversible, quantum devices [1].

In our models, we avoid making any assumptions that might violate one or more of the above constraints.

3 GENERIC MODELS OF COMPUTING

An engineering model of a computer system, in general, may include the following components:

1. A *device model*, which specifies the physical and information-processing characteristics of the individual lowest-level information-processing functional units (bit devices) in the computer.
2. A *technology scaling model*, which specifies how the device characteristics might change as technology improves over time (for example, by making devices smaller, or increasing their energy density).
3. An *interconnection model*, which specifies how information is communicated between devices. Or, interconnects can be treated as a type of bit-device.
4. A *timing model*, which specifies how activities of different devices are to be synchronized.
5. An *architectural model*, which specifies how devices are functionally connected to form a larger unit called a *processor* that can be programmed.
6. A *capacity scaling model*, a more general sort of architecture (a.k.a. an *architecture family* [14]) that allows the machine's total capacity (in bits of storage, and ops-per-second of performance) to be scaled up via a regular transformation to ever-larger sizes. Often we use a *multiprocessor* approach, connecting many copies of a fixed-size processor.
7. An *energy transfer model*, which specifies how power is to be supplied and heat removed from all parts of a scaled processor. The energy system can be viewed from an informational perspective, as supplying known information (namely, a stable power signal) and removing unknown information (entropy). As such, it must obey the fundamental limits on information processing discussed earlier.
8. A *programming model*, which specifies how the computer is configured to carry out different types of computations. The usual programming model is a machine-language *instruction set architecture*, but a variety of rather different alternatives also exist.
9. A *performance model*, which can be used to predict how quickly any given algorithm implemented in the programming model will run on the given architecture. Performance can also be considered a special case of cost-efficiency, in which cost is considered to be directly proportional to time.
10. A *cost model*, which quantifies the *cost*, according to one or more measures, of manufacturing a required computer and/or executing a specified computation on it. In addition to execution time, we may consider other physically reasonable cost measures such as energy cost, spacetime-proportional costs, and total dollar cost. For any cost measure, (*cost-*) *efficiency* is the ratio between the minimum possible cost and the actual cost, and is maximized by minimizing the actual cost.

The various components of the model may be made as abstract or as concrete as desired for a particular purpose, or can even be left completely unspecified.

4 EXAMPLE MODEL FOR ANALYSIS

The various components of our most recent model (described further in [6]) can be summarized as follows:

1. **Device model:** Bit-devices are characterized generically by the following parameters: entropy generation per irreversible operation $S_{\text{i·op}}$, minimum time per cycle $t_{\text{cyc,min}}$, standby entropy generation rate $S_{\text{dev·t}}$, adiabatic entropy coefficient $(St)_{\text{dev·cyc}}$, minimum volume V_{dev}, maximum entropy flux $S_{\text{A·t}}$.
2. **Technology scaling model:** Discussed in §5 below. We basically follow the established Moore's Law trends until we reach known or estimated limits.
3. **Interconnection model:** For simplicity, we assume all interconnects are local; but this is not really a restriction, given the speed-of-light constraint [15].
4. **Timing model:** For simplicity, we just assume that devices can be kept synchronized via some reversible local mechanism that can operate at any desired frequency $\leq 1/t_{\text{i·cyc}}$, without dominating the costs. In §5 of [6], we discuss how our analysis can be extended to get rid of this assumption.
5. **Architectural model:** In this analysis, the only restriction placed on the irreversible architecture is that its activity factor α (transitions per bit-cycle) should be of order 1, *i.e.*, the machine's operation is highly parallel. Removing this restriction would weaken the case for reversibility a bit. Our reversible architecture is specified by applying Bennett's reversiblization algorithm [16] in parallel to all devices in the original architecture.
6. **Capacity scaling model:** We assume that capacity is scaled by replicating fixed-capacity processors in a 3d mesh, but respecting thermal constraints by allowing processors to be spread out in wider, flatter arrangements than cubical, if necessary to increase area available for cooling. This spreading, of

course, impacts the communications latency between processors in the performance model.

7. **Energy transfer model:** The entropy flux constraint $S_{A\cdot t}$ in the device model is assumed to also limit the corresponding rate of energy flow through *any* surface that may be drawn in or around the machine. But we assume the overhead to maintain such flows does not dominate the resource usage.
8. **Programming model:** Not restricted in this analysis; any programming model can be used, since we consider implementing reversibility on the hardware level. However, to obtain the best-case results described below, a reversible programming model might be needed [1], and quantum algorithms would require a quantum programming model.
9. **Performance model:** We allow the number, spacing, and frequency of processors, and the parameters of Bennett's algorithm in the reversible case, to be chosen based on the size of the problem to be solved, while respecting thermal constraints, and we compute the values of the parameters that minimize total cost in both irreversible and reversible cases.
10. **Cost model.** We include both spacetime-proportional costs and energy-proportional costs in our cost model, and combine them into an estimated total cost in currency units.

Based on the above model, in [6] we carried out a variety of analytical hand-calculations (for the easy cases), as well as numerical optimizations (for the hard cases), so as to determine the best possible cost-efficiency (in terms of computational operations per dollar) for both irreversible and reversible architectures, for the example of a 100-Watt, US$1,000 desktop or high-end laptop CPU.

5 TECHNOLOGY SCALING MODEL

Here are the specific numerical values of various inputs used in our technology and cost scaling model. Any of these can be easily changed, and by re-running our optimization software, new results obtained. The following parameters were held constant across all years:

1. **Total cost** of manufacture, of US$1,000. A typical cost for the CPU of a high-end desktop or laptop.
2. **Time constant for depreciation** of hardware of 3 years, due to obsolescence and/or breakdown.
3. **Total power** limit of 100 W, typical for a high-end desktop CPU.
4. **Power flux** limit of 100 W/cm^3, about the largest that can be handled by air cooling.
5. **Cost of energy** of ~US$0.10/kW-hr, typical for today's electric utilities in some parts of the U.S.
6. **Standby entropy generation rate** 1 knat/s/device. This specific figure is rather arbitrary, but it sets a limit to capacity scaling at fixed power. Error correction and energy leakage contribute to this.

The following independent parameters were varied as a function of year, and are plotted in figure 1 below:

1. **Entropy generated per irreversible bit-op.** Today in 0.1-μm CMOS technology, this is ~10^5 nats. The ITRS specs [17] decrease this by −28%/year; we assume this trend continues until it bottoms out at the Landauer bound [12] of 1 bit in 2038.
2. **Maximum clock frequency.** This is ~2 GHz in commercial processors today. We assume a conservative trend of 2× per 3 years applies, until we reach (by 2048) the quantum maximum of ~1 PHz (1 million GHz) for a transition of an electron excited to a potential only 1V above its ground state. If higher voltages are workable at the nanoscale, larger frequencies might be possible.
3. **Minimum device pitch.** Including space for interconnects, an average logic gate takes about a micron-square area today. We assume the average device spacing decreases at the rate of 2× per 3 years, bottoming out at 1 nm in 2033. This limit is only a few atomic diameters, which seems pretty minimal for the spacing between functional logic elements.
4. **Cost per device.** Over the history of semiconductor technology, the cost for a layer of devices patterned over a 1-cm^2 integrated circuit in leading-edge technology has remained roughly constant, at on the order of US$1,000. We assume that the implied scaling of cost per-device, of −50% per 18 months (Moore's Law [18]) continues indefinitely. Once devices reach a minimum size, this cost scaling (if it continues) will have to occur through improvements in nano-manufacturing technology.

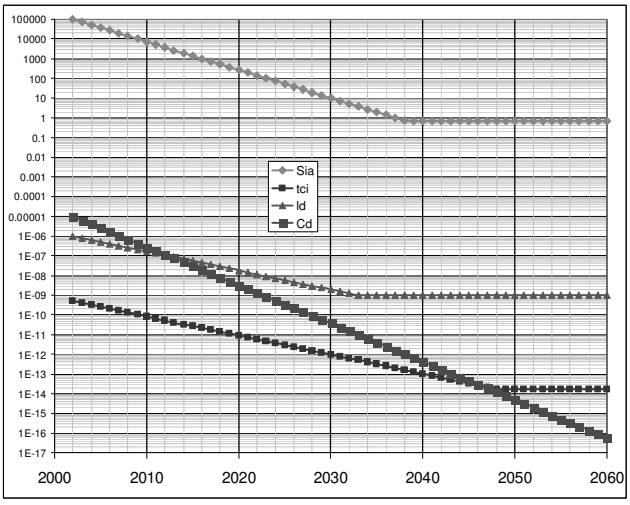

Figure 1. Independent variables for the particular technological parameter sweep we performed in this analysis. Sia = Entropy per irreversible active device transition, in nats; tci = best time per cycle for a single device operated irreversibly, in seconds; ld = minimum length between centers of neighboring devices, in meters; Cd = manufacturing cost per device, in dollars.

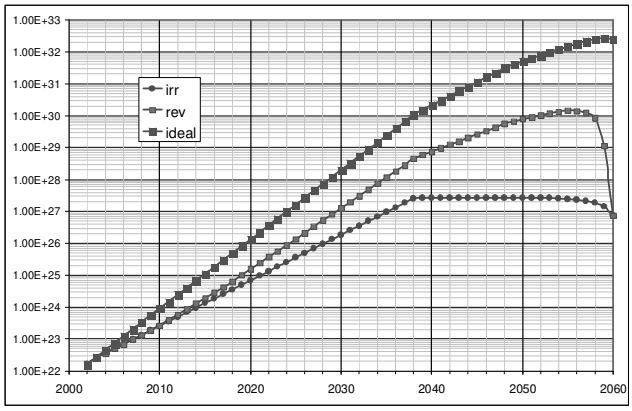

Figure 2. Cost-efficiency, in effective logic operations per dollar, for irreversible (lower line), general reversible (middle line), and best-case reversible (upper line) computations, as a function of year, for the $1,000, 100W desktop scenario described in section 5.

Readers that question our technology scaling assumptions can easily change the parameter sweep in our optimization program, which is available online at `http://www.cise.ufl.edu/research/revcomp/theory/three-d/optimizer.cpp`, re-run the program, and see how different assumptions would affect the results.

6 RESULTS AND CONCLUSION

The results of our analysis are plotted in figure 2, which shows bit-ops per dollar as a function of year. All curves dip towards zero after 2060 due to an artifact of the model—namely, that we are not modeling any technological reductions in per-device standby power, and so eventually, $1,000 worth of devices dissipates more than 100W even when performing no operations at all.

Note that reversible machines begin to become more cost-efficient than irreversible machines for general-purpose computations around 2010. By 2030, they are about 10× as cost-efficient; by 2045, 100×; and by 2055, about 1,000× improved. For special-purpose reversible computations not requiring Bennett's algorithm (or if a perfect general-purposes reversiblization algorithm were discovered), the benefits mount even faster, reaching 10× by 2016, 1,000× by 2041, and 100,000× by 2058.

We conclude that an integrated methodology such as we are proposing is essential for an accurate analysis of future nanocomputing, and that the possibility of reversible computing promises great enough cost-efficiency benefits to warrant further study.

1 M. Frank, *Reversibility for Efficient Computing*, Ph.D. thesis, MIT, 1999. http://www.cise.ufl.edu/~mpf/manuscript.

2 Nielsen and Chuang, *Quantum Computation and Quantum Information*, Cambridge U. Press, 2000.

3 M. Frank and T. Knight, "Ultimate Theoretical Models of Nanocomputers," *Nanotechnology* **9**(3):162-176, 1998; from 5th Foresight Conf. on Molecular Nanotechnology, Palo Alto, CA, Nov. 1997. http://www.cise.ufl.edu/~mpf/Nano97/abstract.html.

4 M. Frank, T. Knight, and N. Margolus, "Reversibility in Optimally Scalable Computer Architectures," in Calude, Casti, Dinnen, eds., *Unconventional Models of Computation*, Springer, 1998, pp. 183-200; from 1st Int'l. Conf. on Unconventional Models of Computation (UMC'98), U. Auckland, Jan. 5-9, 1998. http://www.cise.ufl.edu/~mpf/rc/scaling_paper/scaling.html.

5 M. Frank, "Cost-Efficiency of Adiabatic Computing with Leakage and Algorithmic Overheads," UF Reversible Computing Project, Research Memo #M15, Mar. 2002, http://www.cise.ufl.edu/research/revcomp/memos/Memo15-newalg.doc.

6 M. Frank, "Realistic Cost-Efficiency Advantages for Reversible Computing in Coming Decades," UF Reversible Computing Project, Research Memo #M16, Oct. 2002, http://www.cise.ufl.edu/research/revcomp/memos/Memo16-three-d.doc.

7 W. Smith, "Fundamental physical limits on computation," NECI TR, May 1995. http://www.neci.nj.nec.com/homepages/wds/fundphys.ps.

8 S. Lloyd, "Ultimate physical limits to computation," *Nature*, **406**:1047-1054, 2000.

9 A. Einstein, *Relativity: The Special and the General Theory*, Crown Publishers Inc., 1961.

10 N. Margolus and L. Levitin, "The maximum speed of dynamical evolution," *Physica D*, **120**:188-195, 1998. http://arxiv.org/abs/quant-ph/9710043.

11 M. Frank, "Physical Limits of Computing," *Computing in Science and Engineering*, 2002, no. 3, pp. 16-26. http://www.cise.ufl.edu/research/revcomp/physlim/plpaper.html.

12 R. Landauer, "Irreversibility and heat generation in the computing process," *IBM J. of Research and Development*, **5**:183-191, 1961. http://www.research.ibm.com/journal/rd/441/landauerii.pdf.

13 Wald, *General Relativity*, U. of Chicago Press, 1984.

14 F. Preparata and G. Bilardi, "Horizons of Parallel Computation," *25th Anniversary of INRIA 1992*, pp. 155-174. http://www.cs.brown.edu/publications/techreports/reports/CS-93-20.html.

15 P. Vitányi, "Locality, communication and interconnect length in multicomputers," *SIAM J. Computing*, **17**:659-672, 1988. http://www.cwi.nl/~paulv/papers/multicomp.ps.

16 C. Bennett, "Logical reversibility of computation," *IBM J. Res. Devel.*, **17**(6):525-532, 1973.

17 Semiconductor Industry Association, "International Technology Roadmap for Semiconductors," 2001 ed, http://public.itrs.net/Files/2001ITRS/Home.htm.

18 G. Moore, "Progress in digital integrated electronics," *Technical Digest 1975 International Electron Devices Meeting*, IEEE, 1975, pp. 11-13.