Reversible Logic for Supercomputing

Erik P. DeBenedictis
Sandia National Laboratories
P. O. Box 5800 MS 1110
Albuquerque, NM USA 87185-1110
+1 (505) 284-4017
epdeben@sandia.gov

ABSTRACT

This paper is about making reversible logic a reality for supercomputing. Reversible logic offers a way to exceed certain basic limits on the performance of computers, yet a powerful case will have to be made to justify its substantial development expense. This paper explores the limits of current, irreversible logic for supercomputers, thus forming a threshold above which reversible logic is the only solution. Problems above this threshold are discussed, with the science and mitigation of global warming being discussed in detail. To further develop the idea of using reversible logic in supercomputing, a design for a 1 Zettaflops supercomputer as required for addressing global climate warming is presented. However, to create such a design requires deviations from the mainstream of both the software for climate simulation and research directions of reversible logic. These deviations provide direction on how to make reversible logic practical.

Categories and Subject Descriptors

C.1.3 [**Processor Architectures**]: Other architecture styles – *cellular architecture.* C.4 [**Performance of Systems**]: Modeling techniques. I.6.3 [**Simulation and Modeling**]: Applications. J.2 [**Physical Sciences and Engineering**]: Earth and atmospheric sciences – *climate change.*

General Terms: Algorithms, Performance, Design.

Keywords

Reversible logic, quantum dot cellular automata, climate change global warming, applications modeling, supercomputing, computer architecture.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. Public release as SAND2004-6603C.

CF'05, May 4–6, 2005, Ischia, Italy.

1. INTRODUCTION

Supercomputing is a key driver of science, defense, and other priorities of national and worldwide interest – but the future of this asset may not be assured. "In the last decade, the power of computation – our ability to model and simulate experiments that we have not conducted in a laboratory – has become so great that it must now be considered a third pillar, along with theory and experiment, in the triad of tools used for scientific discovery" [9]. To assure the third pillar remains available, the supercomputing community has periodic efforts to study the future of applications and their ability to run on hardware likely to be available [18].

While concentrating on incremental advances, these studies do cover the demands of the most ambitious applications [14, 16, 17]. Yet where an application needs more computer power than can be provided by Moore's Law, authors tend to say so and move on to the next topic without providing a solution.

This might lead the reader to wonder about the security of Moore's Law – because continued progress in science, technology, and society seems to be depending on it. The most authoritative source is the Semiconductor Industries Association (SIA) and their International Technology Roadmap for Semiconductors (ITRS) [13]. This is a yearly study of the future of the semiconductor industry for up to a dozen years into the future. This study covers hundreds of constituent technologies that collectively contribute to continuing progress. The ITRS has committees of experts that meet yearly to assess the suitability of all these technologies, creating color-coded tables indicating areas and years where technological breakthroughs will be needed to assure continued progress. There is cause for alarm. Recent versions of this report have lots of red markings indicating that current technology is inadequate and breakthroughs are needed. A careful review indicates that the semiconductor industry's experts are unsure how CMOS scaling can continue even a dozen years into the future.

Thus, the security of the "third pillar" and all it implies must involve something other than CMOS transistors – and the ITRS has an "emerging devices" section, which should cover this. The ITRS includes an overview of promising post-transistor devices, which appears in Table I (except that I added the black shaded row). However, a breakdown of the table by device is disappointing: the emerging devices that might compete with transistors are in gray shaded cells of Table I. All the devices are either slower (white text) or larger (black text).

Table I: Annotated ITRS Emerging Devices Information [13 ERD 2003 p. 42]

Technology	T_{min} sec	T_{max} sec	CD_{min} m	CD_{max} m	Energy J/op	Cost min $/gate	Cost max $/gate
Si CMOS	3E-11	1E-6	8E-9	5E-6	4E-18	1E-11	3E-3
RSFQ	1E-12	5E-11	3E-7	1E-6	2E-18	1E-3	1E-2
Molecular	1E-8	1E-3	1E-9	5E-9	1E-20	1E-12	1E-10
Plastic	1E-4	1E-3	1E-4	1E-3	4E-18	1E-7	1E-6
Optical (digital, all optical)	1E-16	1E-12	2E-7	2E-6	1E-12	1E-3	1E-2
NEMS (conservative)	1E-7	1E-3	1E-8	1E-7	1E-21	1E-8	1E-5
Biologically Inspired	1E-13	1E-4	6E-6	5E-5	3E-25	5E-4	3E-1
Quantum	1E-16	1E-15	1E-8	1E-7	1E-21	1E3	1E5
QDCA [21, 22]	1E-13	1E-11	1E-9	1E-8	1E-24	1E-12	

Reversible logic may represent a way to beat the physical limit to current progress, known as Landauer's limit. The scientific community has believed for decades (the key paper was in 1961 [15]) that reversible logic offers a way to cut the key power consumption attribute of logic substantially, thus increasing the throughput of power-limited chips. There is no known bound on the amount of computing possible per Joule, although nobody has actually created and measured devices that beat Landauer's limit. The black shaded cells in Table I represent molecular quantum dot cellular automata [21, 22] operating in "reversible logic" mode, which the reader will see are smaller, faster, and lower power than transistors. The ITRS authors did not use these figures even though they cited the paper [21], suggesting that industry has not factored the potential of reversible logic into its plans.

Government and industry acknowledge a need to consider reversible logic, but have not done so to date [2]. Representatives of the Semiconductor Industry and the US National Nanotechnology Initiative (NNI) held a workshop in 2003 and included in the official conclusions that a "basic understanding and demonstration of system configuration with reversible logic to reduce power dissipation" was an appropriate goal for these organizations.

This paper seeks to address the advantage possible from reversible logic for supercomputing, in part by exploring basic system configurations.

2. LIMITS OF IRREVERSIBLE LOGIC

The audience at this workshop is certainly familiar with Landauer's limit on the minimum amount of energy per logical operation [15]. However, for the purposes of this paper we need to derive the maximum number of FLOPS available to solve a problem of a given importance.

The amount of money available for a supercomputer is essentially independent of time, but depends on the importance of the problem being solved. A department in a university has always been able to purchase a computer for about US$1M for science or technology research by a few professors. Big Science and defense define the top entries on the Top500 [23] list with computers costing around US$100M. While not a supercomputer, the US space station represents major international science and technology priority where the primary component costs over US$10B. In broad terms, I suggest that University research, Big Science, and major international programs can consider components costing US$1M, US$100M, and US$10B.

However, the Laudauer Limit is based on energy and the paragraph above is based on US$. A supercomputer budget will go to capital cost, maintenance, electric bill, etc. The money spent on electricity will buy Joules, of which only a fraction will go to the logic. These factors are much to numerous to calculate from first principles, so I suggest using the US$ to energy ratio from a recent supercomputer and assume that this ratio will apply more or less unchanged in the future.

To that end, I have personally just finished working on the 40 Teraflops ASCI Red Storm supercomputer. This US$100M supercomputer draws about 2 MW electric power from the wall, putting about 750 KW into the microprocessor chips (the remainder being dissipated for power converters, interprocessor communications logic, I/O system, fans, and so forth – but not air conditioning in this analysis). My conversion factor is to say that a US$100M supercomputer project puts 750 KW into computational logic. Thus, University research warrants 7.5 KW power to logic, Big Science 750 KW, and a major international priority 75 MW.

Figure 1 illustrates the limits of a US$100M supercomputer based on irreversible logic, both from theoretical and industrial planning standpoints This analysis is similar to one in an existing publication [8], but accounts for reversible logic.

Let us discuss the Landauer's limit for a US$100M supercomputer, which is illustrated in by reading figure 1 from the top down to the black box. The US$100M supercomputer will have 750 KW entering its active components. Landauer's limit for irreversible logic with this much power will be 2×10^{26} logic operations per second.

However, we rate supercomputers by FLOPS. There are about 20,000 logic operations in a 64 bit floating-point operation, taking a typical mix of adds and multiplies. While this yields a limit of 10 Zettaflops, this limit leaves does not account for noise margin, device tolerances, or other losses. Let us further derate the limit by 4× to account for these inefficiencies, though I cannot cite any source of authority. The resulting number of 2.5 Zettaflops would be the limit of performance of a system that were solidly packed floating point units, such as a radar signal processor.

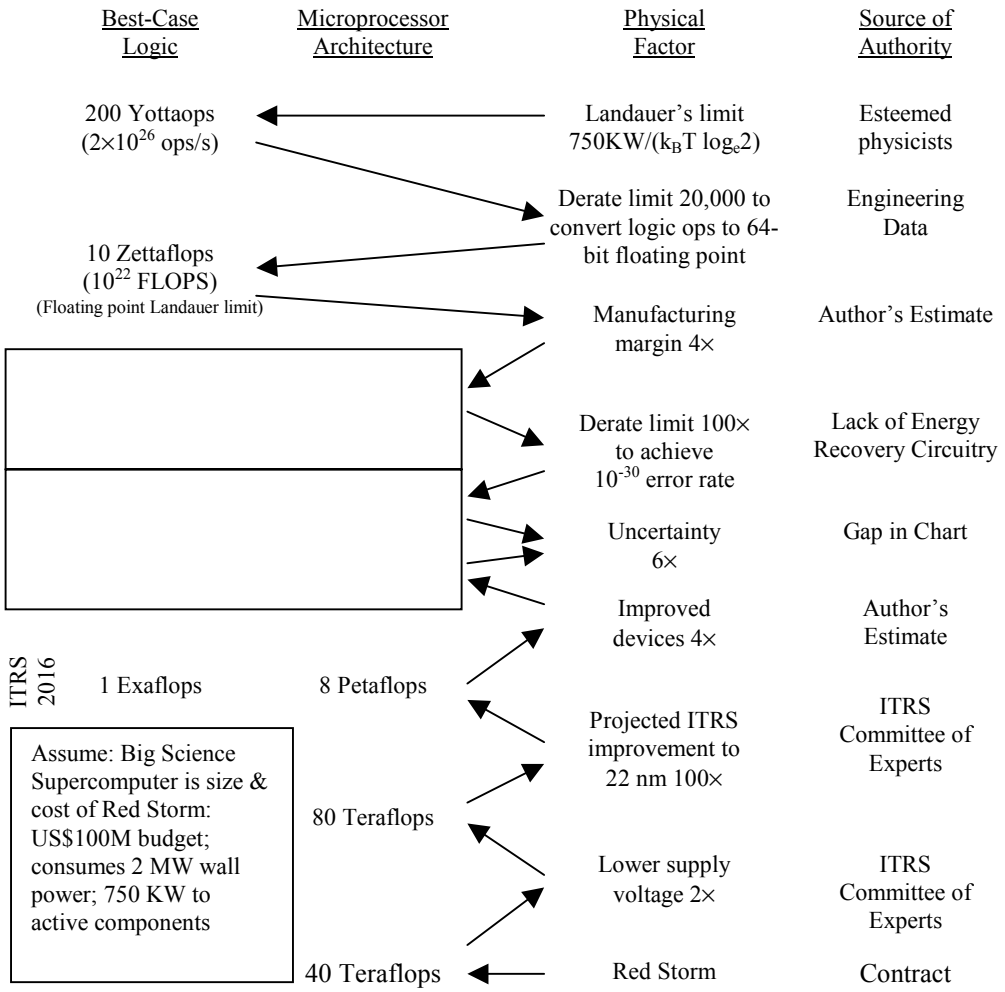


Figure 1. Performance limits on a Big Science irreversible logic supercomputer. This chart derives the upper bounds on performance by derating the physical limits while simultaneously building up possible performance from known supercomputers and industry plans. The gray region shows limits based on industry plans, including a small region of uncertainty related to how close practical technology can come to the physical limits. Programmers seeking the limits of software should consider the column labeled "microprocessor architecture" whereas hardware designers can aspire to the limits in the "best-case logic" column by embedding the needed behavior in hardware. "Advanced architectures" such as Processor-In-Memory [6, 11, 18, 20, 24] or Blue Gene [1] tend to cross the 125:1 gap, although their degree of success varies. The upper or lower numbers in the gray region will be determined by engineering considerations in device design. The black region represents Landauer's theoretically significant limit, but is not necessarily of practical importance. To the author's knowledge, there is no research on irreversible logic devices or circuits that would exceed the thermal noise limit and thus be governed by the black region (although reversible logic can exceed all limits in this chart).

However, we expect to program supercomputers with high-level languages, like Fortran or C++. To execute these languages requires a microprocessor with instruction decoding, program counters, cache, etc. Microprocessors have been very successful devices because of their flexibility, yet their flexibility comes at the expense of a lot of logic gates consuming power without contributing to what we count (FLOPS). In fact, modern microprocessors burn only about 1% of their power in the floating-point units (this factor is represented in figure 1 as 125:1). I cannot say that the a future supercomputer should be designed with flexible (but power inefficient) microprocessors or efficient (but inflexible) special or new architecture, so I've drawn figure 1 with columns corresponding to both choices. This yields the limit for a Big Science supercomputer at 20 Exaflops for microprocessor-based solutions to 2.5 Zettaflops for custom architectures.

However, Moore's Law and industrial developments are not headed down a technology track that will ever approach Landauer's limit. CMOS circuits are implemented in such a way that all the bit energy is turned into heat when not needed, instead of being largely recovered as needed to approach Landauer's limit. At low energies, this type of circuitry is subject to spontaneous errors determined by thermal noise. To avoid errors of this type will require increasing signal energy by about 100 fold, from $k_BT \log_e 2 \approx .7\ k_BT$ to about 70 k_BT. Figure 1 shows this derating of 100× to yield the top set of numbers in gray shaded "industrial limit" region, 200 Petaflops for microprocessor-based solutions to 25 Exaflops for custom architectures.

For confirmation, let us see how close industry believes it can come to the figures just obtained. The Sandia Red Storm system (which defines the power levels used for the top-down analysis), delivers a peak of 40 Teraflops. According to the ITRS, a 200× increase in performance will be possible by the year 2016. However, the ITRS does not claim that Moore's Law ends in 2016. Let us assume that there is at least an additional 4× improvement possible between 2016 and the end of the evolution of transistors. Reading upwards in figure 1, the Red Storm performance will increase to 32 Petaflops for microprocessor-based solutions and 4 Exaflops for a custom architecture. This leaves a gap of about 6×, acknowledging our inability to predict the future with confidence.

3. PROBLEMS NOT SOLVABLE WITH IRREVERSIBLE LOGIC

Investments in reversible logic will be justified if there are really important problems with resource requirements above the numbers listed in the shaded boxes of figure 1. Many supercomputer simulations are scalable, being able to solve progressively larger problems with progressively more compute resource (FLOPS). The importance of solving the larger problems varies.

Geophysicists would like to have a 3D model of the solid portion of the Earth, showing the positions of faults, major rock formations, and so forth. This would improve our understanding of Earthquakes and could result in public policy changes that would reduce damage and loss of life due to buildings falling down in Earthquakes. Estimates suggest a 1 Exaflops supercomputer could construct such a 3D model by "inverse" simulation of seismic data already collected [Jeroen Tromp 2004, private discussion].

While it is easy to find problems where the compute requirements exceed the smallest numbers in a shaded region of figure 1 (200 Petaflops), I don't believe they will be sufficient for justifying reversible logic. A critic could counter-argue that CMOS's maturity makes further investment less risky and a better deal given that solutions are possible both ways.

Bigger problems would be less prone to counter argument, and they exist. Projections of simulations of the non-solid portions of the Earth (ocean, surface, and atmosphere) reach 1 Zettaflops. In the context of global warming, simulating the Earth's climate is of compelling importance [16].

The sections below will outline specific algorithms for climate and weather predictions, but there is a broader point applicable to reversible logic. For many years, simulation has been the major use of supercomputers. The algorithms have almost universally calculated the time evolution of a region in space based on physical laws – justified on an argument that is almost intuitive but was articulated very effectively by Feynman [10]. These algorithms represent the state of space at some time t as a vector x_t, and have some modeling function $\mathcal{M}$ that calculates $x_{t+\Delta t} = \mathcal{M}(x_t)$. It seems that irreversible logic is sufficient to calculate $\mathcal{M}$ for all systems of interest, and hence irreversible logic ought to be sufficient for all supercomputing applications. However, there are problems whose solution requires "more" than simulation. Geophysicists today simulate the flow of seismic waves through their best guess of the Earth's structure, but their need is to deduce the structure of the Earth from analysis of these waves; climate modelers propose to simulate the Earth for 10,000 years, not that they intend to live 10,000 years to observe the result but just to find the fixed point of $\mathcal{M}$. My point is that the reason for reversible logic to exist may not be to make today's supercomputing problems run faster, but to enable an emerging set of problems that go beyond the capacity of present-day supercomputers.

4. MODELING THE EARTH'S CLIMATE

The global climate study [16] calculates a series of "compute factors" that apply to current climate modeling codes in order to make them sufficiently accurate and versatile to fulfill the objectives of the US Climate Change Science Program [3]. Simulation science has a rigorous basis [10] provided that all the necessary physical processes are simulated to sufficient accuracy. While today's climate codes show correlation with current climate trends, they were never intended to model the necessary range of physical processes nor have sufficient resolution. The study cited [16] collected information on the range of physical processes that determine the Earth's climate and the accuracy necessary for each. By scaling up current codes by designated amounts, the overall computing resources for a proper climate code can be predicted. Table II illustrates the factors from this paper, yielding a sustained throughput of 1 Zettaflops (actually [16] specifies a range of 10^3-10^5 for the first compute factor which would strictly speaking yield a range of 100 Exaflops – 10 Zettaflops; I use the midpoint.)

To understand the needed balance of FLOPS to other resources (memory, communications bandwidth and latency, etc.),

Table II: Compute Factors for Global Climate Initiative (from [16])

Issue	*Motivation*	*Compute Factor*	*Peak Performance*
Spatial Resolution	Provide regional details	10^3-10^5 (10^4)	100 Teraflops
Model completeness	Add "new" science	10^2	10 Petaflops
New parameterizations	Upgrade to "better" science	10^2	1 Exaflops
Run length	Long-term implications	10^2	100 Exaflops
Ensembles, scenarios	Range of model variability	10	1 Zettaflops

we must know the nature of climate codes. We will describe the CCSM (Community Climate System Model) in this paper, yet this application quite representative of many evolving supercomputing applications. CCSM is an interdisciplinary group of scientists organized into nine groups, eight of which are writing computer models for some physical process. The processes include atmosphere, land, ocean, polar climate, biogeochemistry, paleoclimate, climate variability, and climate change, and for all practical purposes can be considered as writing eight computer codes. The ninth group is called "software engineering," and is creating a framework that can execute all nine codes in a single, time based simulation.

These codes all model physical processes by subdividing space and simulating sequentially in time, yet the codes differ in mesh dimensionality (2D, spherical, 3D), mesh size, and time step. The code also uses sub grid parameterization, a form of the emerging area of multi-scale simulation. For example, the study predicts that the atmosphere will need to use multi-layer spherical grids of 15 km horizontal resolution, 300-600 layers, and time steps of 10 minutes to achieve accurate atmospheric modeling. The framework's job is to execute the various models on different sections of the supercomputer such that simulated time passes at similar rates. The framework translates data between models by interpolating mesh boundaries.

The inner loop for today's models comprises evaluations of an equation of state. These models are comprised of undistinguished code based to some extent on heuristics. The equation of state for the atmosphere involves models for cloud, rain, snow, hail, etc. formation; for land it involves the growth and decay of various types of trees.

Equations of state like the one above consume FLOPS in today's runs on par with communications-intensive dynamical solvers, although this will change. Today's simulations are dominated by atmospheric simulation. An atmospheric simulation time step comprises evaluating the equation of state for the atmosphere followed by a whole-Earth solution of atmospheric dynamics. Today's atmospheric dynamics includes non-local communications patterns as a part of FFT evaluations, although future models are likely to rely only on local communications [11]. However, one notes from Table II that the complexity of the equation of state is projected to rise by 10,000× (two factors of 100× in the second and third compute factors). This 10,000× increase in FLOPS count without any increase in communications load will tend to reduce the required "bytes/FLOPS" communications balance by 10,000×.

Thus, we can paint a picture of a future computer required for climate simulations: The supercomputer will need a sustained performance of 1 Zettaflops. The supercomputer will be comprised of some large number of computing nodes connected by some form of interconnect fabric, illustrated in figure 2 as a hatched square (it is premature to speculate on interconnection topology or the number of "nodes," so figure 2 should be interpreted as just representing resources by equal area on the printed page). In executing the climate simulation, it would be quite effective to divide the computer into about 4 unequally sized sections to run the physical models given above such that the simulated time would run at the same rate on each section (today's approximate ratios are illustrated in figure 2). Each

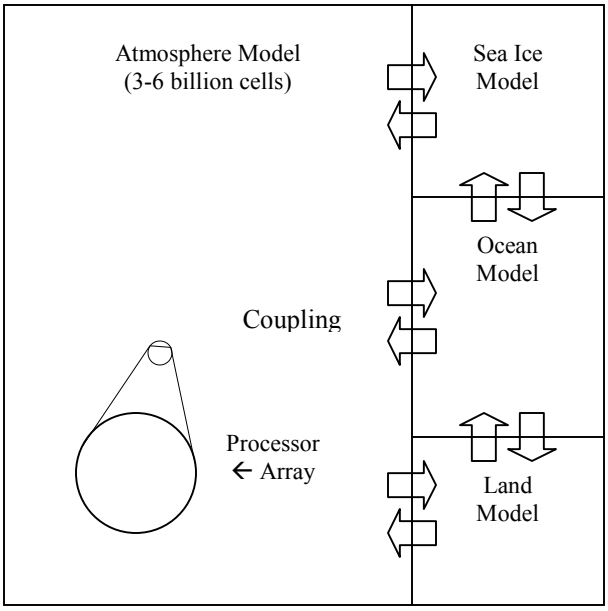


Figure 2: Overall design of parallel supercomputer for climate modeling

physical model will be similar to today's CCSM modules (i. e. Fortran and C++ code) but with 10,000× as many floating operations per evaluation of the equation of state. Let us guess that the 10,000× increase in floating operations comes from 20× as many lines of code running on 20× as much data in loops that are 20× longer. Each time step will involve solving dynamics equations within each section of the machine in accordance with the 10^3 to 10^5 increases in resolution, with boundary values moved between sections at various multiples of the simulated time step.

5. A PLAUSIBLE SUPERCOMPUTER DESIGN

I would like to see if reversible logic could be the technology increment that makes global climate simulation feasible within a US$100M budget and for simulations with the running time required by the scientists. I am strongly motivated to seek solutions involving a parallel von Neumann architecture, as scientists are familiar with this architecture and the CCSM code is already written for it.

However, I found that by following the individual paths outlined by the mainstreams of the applications and "physics of computation" communities, I ended up with a supercomputer that did not meet requirements (in other words, a supercomputer that might result from mainstream "physics of computation" research would not run the global climate simulation as currently coded fast enough to be useful). Yet, it was clear that deviations from the mainstream could reduce the gap. I succeeded in closing the gap by making the tradeoffs in figure 3, writing an optimizer/expert system program to track the details.

My optimizer program made tradeoffs between innovations in several areas (hardware, software) in accordance with my judgment. Should a reversible logic supercomputer be constructed in the future, the practitioners would certainly renegotiate these

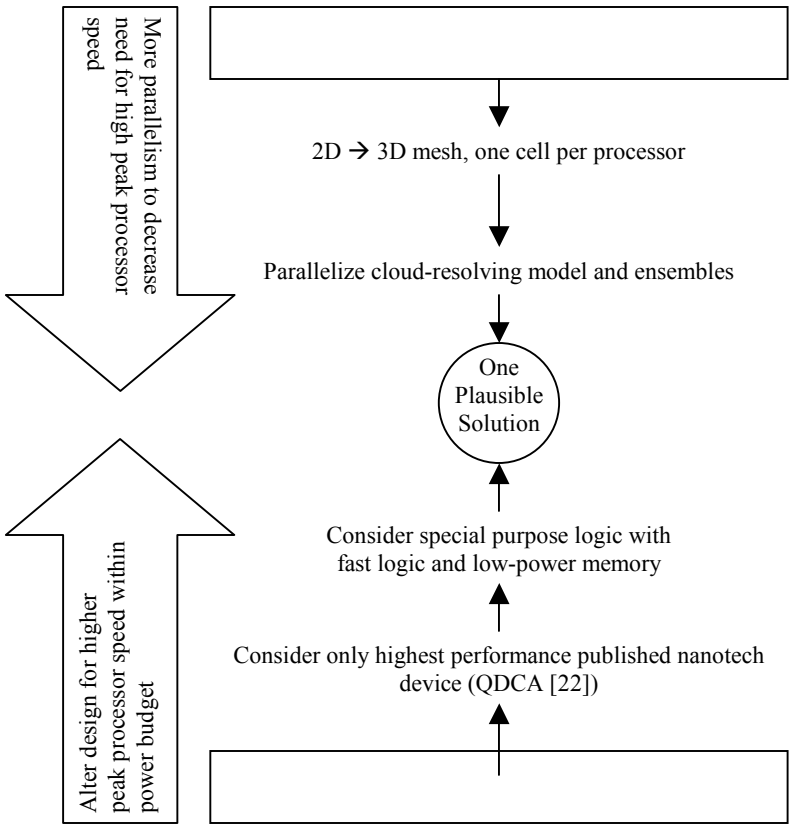


Figure 3: Exploiting degrees of freedom in order to find a plausible solution.

tradeoffs. Thus, the specific solution that follows may not be meaningful, although I hope the process of making tradeoffs to meet collective requirements may be instructive.

The specific solution is probably not meaningful because it was based on a series of arbitrary tradeoffs, but that is not the point. The process shows how software and hardware tradeoffs interact in order to reach a solution. The practitioners of the various fields can negotiate more acceptable tradeoffs in the future.

6. DEFINING THE EXAMPLE PROBLEM SPECIFICALLY

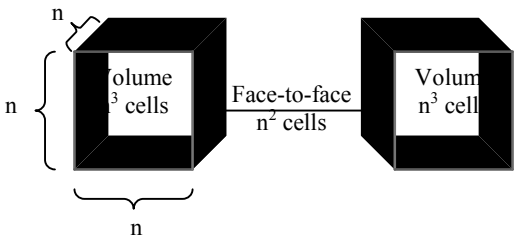
Table II and figure 3 together show how to change the current CCSM into a complete simulation of the global climate, with Table III describing the model after the changes have been made. However, Table III and the remainder of this paper will consider only the atmospheric modeling part of climate simulation (due to space limitations). I have also generalized Table II [16] into a continuous geometric scaling by replacing the compute factors of 10, 100, 1000 with s, s^2, and s^3 for scale factor $1 \le s \le 10$.

7. APPLICATIONS MODELING AND DESIGN OPTIMIZATION

Through a process called applications modeling, supercomputer buyers know how to predict the performance of a supercomputer on algorithms like those described in the preceding pages. This involves creating a mathematical formula expressing the running time of an algorithm in terms of the design and performance parameters of the supercomputer. By substituting the parameters of equivalent reversible logic components, we should be able to predict running time for a supercomputer based on some different technology as well. By using the running time as part of an objective function, an optimizer/expert system can pick the best design.

Table III: Model of Future Climate Modeling Application

	Today	**To Solve Problem**	**Compute Factor; s**
Grid	150 km × 150 km × 30 layers = 660K cells	15 km × 15 km × 300 layers = 660M cells	1000×; s^3
Time step interval × number of time steps = simulation time	100 minutes × 500K = 100 years	10 minutes × 500M = 10,000 years	1000×; s^3
Basic atmospheric physics per cell update; memory bytes/cell	12.5K floating point operations; 1.6 Kbytes	12.5K floating point operations; 1.6 KBytes	
Add atmospheric chemistry for 100 tracers (chemicals)	N/A	12.5K × 100 → 1.25M floating point operations per cell update; 1.6 → 3.2 KBytes	100×; s^2
Add 10×10 cloud resolving sub grid	N/A	1.25M × 100 → 125M floating point operations; 320 Kbytes	100×; s^2
Floating point operations per simulation × scenarios; bytes of memory	4.4 P × 1 = 4.4 P; 1.6 Gbytes	43 Y × 10 = 430 Y; 83 PBytes	10×; s



T_{step} = Wall clock time per simulated time step
n = atmospheric cells per processor
C_{bytes} = number of bytes communicated per surface cell
T_{byte} = communications time per byte
F_{grind} = number floating operations to evaluate each of n cells
F_{inner} = average FLOPS rate in inner loop (from simulation)

$$T_{step} = 6\, n^2\, C_{bytes}\, T_{byte} + n^3\, F_{grind}/F_{inner}$$

Figure 4: Simple example of applications modeling.

Table IV: Technology Parameters, with QDCA Exemplary Values

Parameter	Description	Exemplary Value [22]
T	Temperature in Kelvins during operation.	60°K
Tau	Speed of operation; gate delay for logic or access time for memory.	8×10^{-14} s
Lambda	Size of a logic gate or memory bit when included in a circuit, thus including both the gate and an average amount of wiring as would correspond to hand or automatic layout. Specified as linear dimension to be squared or cubed depending on the packaging.	5×10^{-7} m
Erase	Energy to erase a bit.	100 k_BT
E	Energy to compute a logic function, or energy to access a bit for a memory.	1.2×10^{-24} J
$F_{Powerdown}$	A dynamic range factor $F_{Powerdown}<1$ permitting an engineer choose to reduce power consumption of the device to as little as $F_{Powerdown}$ in exchange for reducing speed to $F_{Powerdown}^{½}$. It doesn't matter if this factor is due to the "adiabatic theorem" or voltage scaling.	100
SPwr	Static power dissipation.	2×10^{-10} W
SigProp	Velocity of signal propagation for a logic family constructed of this device, as fraction of the speed of light.	.002 c

Table V: Architectural Parameters

Parameter	Description	Exemplary Values
N_{float}, Tau_{float}, $E_{float,}$ $F_{reversible}$	Number of logic gates in a 64-bit floating point unit, number of gate delays, and number of units of units of energy (E) in a typical floating point operation. Flag to indicate reversibility principles used.	25,000 gates 200 prop. delays; 20,000 E; TRUE
N_{core}, N_{FLOPS}, $E_{core,}$ $F_{reversible}$	Number of gates in a microprocessor core, number of FLOPS/clock, and number of E units per clock. Flag to indicate reversibility principles used.	250 N_{float}; 2 FLOPs/clock; 125 E_{float}; TRUE

Table VI: Packaging Parameters

Parameter	Description	Exemplary Value
Coolant-Capacity	Performance of coolant in watts per square meter for full system cooling. This measurement assumes pipes moving the coolant at the maximum feasible velocity are piping the coolant in and out of the vicinity of the computer. The capacity will be in watts per sum of inlet and outlet cross section.	47KW/m^2 (air cooling)
CooledPackage-Edge	For density calculation, chips with associated cooling apparatus are assumed to occupy a cube with this edge dimension.	50 mm
MaxPower	This is the maximum power that can be dissipated by a chip.	150 W
Pins	This is the maximum number of conductors per chip, each operating at logic speed.	1500 pins
CostPerChip	US$ per chip, with all overheads included such that the system cost is the number of chips times this cost.	US$9000

Based on human analysis of an algorithm or application, an analytical formula is developed for the running time, such as $T_{run} = f(s, P_{design})$, where T_{run} is the running time per simulated time step, f is a manually derived function, s is the scale of the problem (the scaling factor per Tables II and III), and P_{design} represents various design parameters (e. g. memory size). Figure 4 illustrates the function f for a straightforward finite difference equation where each processor has an n×n×n array of cells. The running time formula, includes a component based on the surface area of the sub region in each processor and another component based on the number of cells. In both cases, the constants T_{byte} and F_{inner} represent technology-based performance parameters. To be accurate, actual formulas should include load balance and global synchronization [4, 5, 6].

8. TECHNOLOGY AND ARCHITECTURE PARAMETERS

To extend applications modeling to other technologies (in combination with reversible logic as appropriate) requires parameters for the technologies involved (as there may be different technologies for logic, memory, and interconnect). Table IV lists these technology parameters as I use them. Table V lists other parameters associated with technology-independent logic design. Table VI covers packaging parameters, which are used to calculate system cost (the dimensions of the overall package also affect communications latency). I have put values rightmost column of Tables IV-VI representative of quantum dot cellular automata [21, 22], which will be used, in later sections.

I have added a factor of 2 overhead for reversible logic based on ideas I've heard from others but which do not appear to be published.

1. The Notre Dame QDCA group proposes to use a clocking scheme where a wave passes over a QDCA combinational logic circuit in a forward direction and then retreats. This clocking makes combinational logic nets reversible and surprisingly introduces no overhead – although it would prevent pipelining.

2. It seems well known that reversible architectures need not be fully reversible. If reversible gates dissipate 1/K times the energy of a irreversible gate, letting o(1/K) gates operate irreversibly will not substantially increase overall system power. An instruction in a standard microarchitecture does thousands of gate operations per instruction yet only updates a register (or memory) and increments a program counter. Thus, a computer system where the microarchitecture was reversible only to the granularity of a single instruction would not need much overhead beyond what would be required to make the combinational logic reversible and would be close to optimal on power.

I regret that there seem to be no references on the above matters.

9. AN OPTIMIZER/EXPERT SYSTEM FOR FINDING ARCHITECTURES

I use an optimizer/expert system to convert the properties of unspecified designs discussed in the preceding pages into design trends. The structure of the program is illustrated in figure 5 in general terms. In essence, the program searches the design space by decreasing values of the scaling parameter s used in Tables II and III. For each value of the scaling parameter, the program creates a candidate design of a supercomputer and evaluates its acceptability. The first acceptable design is printed out and the search finishes. The printout will therefore be the most highly scaled problem that can be solved with the specified technology. Scaling is via all of the factors in Table II (resolution, complexity of physics model, run time length, and ensembles).

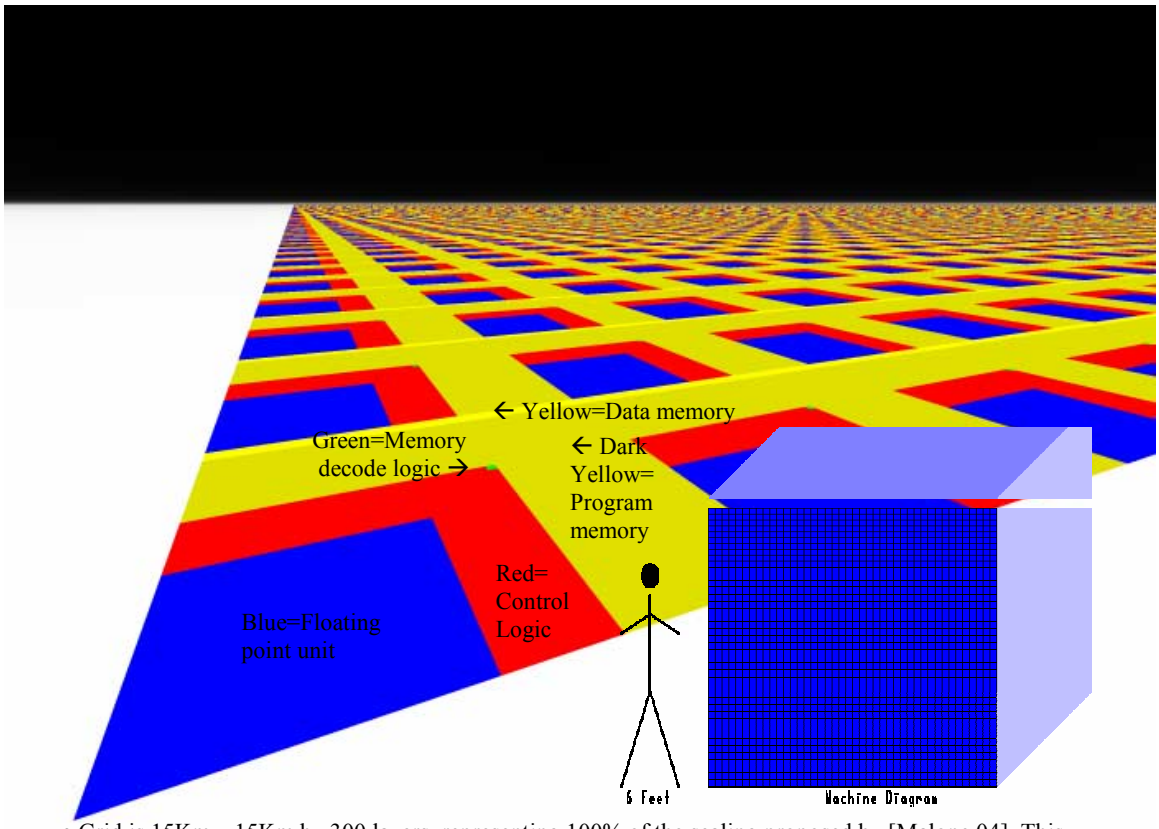
The optimizer/expert system also has an operating mode for comparing time trends, graphing the results over various years into the future. The ITRS roadmap gives projected CMOS device parameters and line widths possible with future lithographic technology, which have been coded into the optimizer/expert system. For each future year, the software consults the ITRS for CMOS parameters and lithographic line width. The software implements the idea that no matter how small the proposed device could be, it can never be manufactured smaller than the lithographic line width at the year of manufacture. This means the device dimension is updated to:

max(device size from technology table, lithographic line width in year of manufacture)

The utility of these time trends has to be considered on a case-by-case basis. For short-term CMOS predictions, the semiconductor industry is committed to manufacturing devices as reported. However, there are no plans to manufacture many interesting nanotech at any line width on any schedule

```
for (scaling factor = 10…1 by small increments)
    for (number of cells per processor iterating over valid range)
        for (power level = 1…F_Powerdown (adiabatic principle))
            for (number of processors per chip = 1, 2 … n/2, n) {
                if (chip would be >100% full) continue;
                if (chip overheats) continue;
                if (yearly cost > budget) continue;
                if (modeled running time > target) continue;
                print (s, FLOPS, and design)
                break;
            }
```

Figure 5: Schematic of Optimizer/Expert System



• Grid is 15Km x 15Km by 300 layers, representing 100% of the scaling proposed by [Malone 04]. This results in 1.5Kx1.5Kx300 (x10x10 cloud subgrid)=67.8G cells.

• Simulated timestep is 10 min x 526M steps yields 10K years simulated time.

• Performing 10 scenarios as 1 at a time in 10 sequential groups.

Basic atmospheric dynamics operation requires 12.5K floating-point operations. Scaling increases this to 1.25M for atmospheric subgrid and 12.5M for atmospheric chemistry. However, cloud subgrid parallelism is enabled. The 15Km x 15Km atmosphere cells have been each divided into 100 cloud subgrid cells, with one processor assigned to each. This increases the number of cells and cuts the per-cell floating point operation count to 1.25M.

• Supercomputer is 77.6K chips, each with 437K nodes of 2 cells of 6.17K floats; solves 67.8G=4.08Kx4.08Kx4.08K cell problem.

• System dissipates 742KW from the faces of a cube 2.29m on a side, for a power density of 47KW/m2.

Power: 742KW active components; 2.97MW refrigeration; 7.42MW wall power; 14.8MW from power • company.

• Yearly cost is $26M.

5M, including $19.5M power, $1.28K facility rental, and 30% of the $23.3M capital cost of the machine per year.

• Compute power is 325EFLOPS, completing an iteration in 260µs and a simulation in 1.06s.

• Timestep takes 260µs: (40µs compute + 185µs synchronize waste time + 35.3µs AllReduce) | 4.96µs communications.

Figure 6: Example of resource allocation for reversible-logic chips for global-warming calculation. Diagram is a 1 cm chip viewed in perspective for a corner. Area and number of "nodes" correspond to optimized output from "expert system." Cube and human stick figure show sizes given specified packaging technologies.

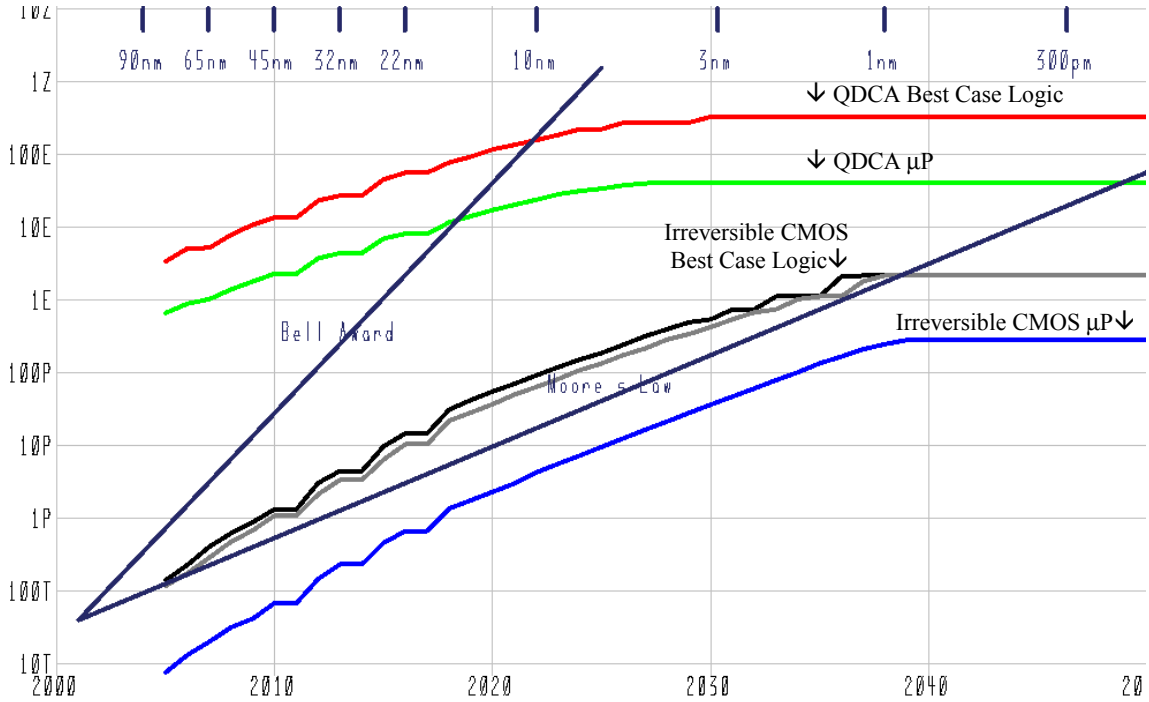


Figure 7: Time Trend of various technologies for performing global climate modeling per [15]. Red=Quantum Dot Cellular Automata [22] with reversible logic and special purpose (non-µP architecture); Green=Quantum Dot Cellular Automata [22] with reversible logic and µP parameters; Black=irreversible CMOS with special purpose (non µP) architecture, Blue= irreversible CMOS µP.

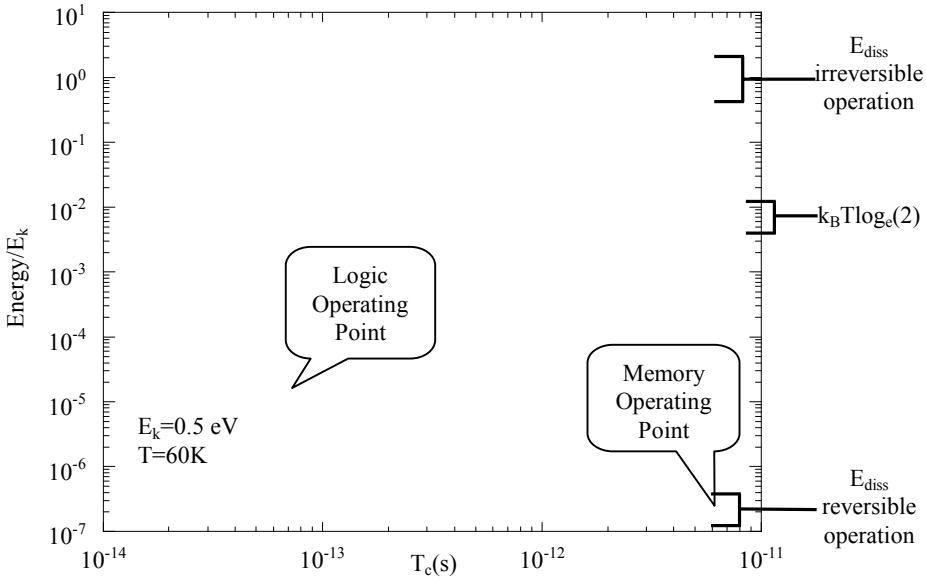


Figure 8: Molecular Quantum Dot Cellular Automata speed-energy curve for irreversible and reversible operation (courtesy of C. Lent) with operating points used in this paper labeled.

My objective in constructing the optimizer/expert system was to find which emerging nanotech devices show promise for solving the very largest known problems. While the expert system approach shows promise, I found it surprisingly difficult to find the necessary technology parameters for many devices. CMOS is well characterized, but does not seem to have sufficient performance. Quantum dot cellular automata [22] are the only device that appears well enough characterized to evaluate, and where the evaluation suggests sufficient performance.

Figure 6 shows the output of the optimizer/expert system for molecular quantum dot cellular automata applied to the global warming problem. To have sufficient volume to be cooled, the package would need to be the size of the cube shown with a human stick figure for scale. The colored shape shown in perspective is a 11.8 mm × 11.8 mm chip. Repetitions of the colored regions on the surface of the chip show the 437K nodes. The blue, red, green, and yellow (lightening shades of gray in B&W rendition) regions illustrate floating-point units, processor control logic, memory decoder, and memory functions. The proportional size of the regions relates to the size on the surface of the chip given device sizes and layout densities from the technology tables.

Figure 7 shows a time trend for various nanotech devices.

Figure 8 additionally illustrates the projected speed-power curve for QDCA similar to [22] with the operating points illustrated. The optimization program determined that the logic operating point correspond to the fastest allowable speed; I defined the memory operating point to be $1/10^{th}$ the speed of the logic.

10. CONCLUSIONS

Reversible logic offers performance advantages over conventional technology, but faces a long and expensive technology development path. A promising way to secure funding for the required technology development would be to join forces with a high-priority problem that can be solved by computers but exceeds the limits of conventional technology.

I've outlined in this paper the limits of conventional supercomputer technology – which for Big Science range from 32 Petaflops to 25 Exaflops depending on various factors. University research and major international efforts will be ~100× smaller or larger. A supercomputing need that exceeds these levels would be a good candidate for reversible logic.

I have outlined classes of applications that exceed these limits. This class often includes problems bigger than simulations, such as inverse problems, imaging problems, adjoint problems, large-scale data analysis, data assimilation, etc. The example developed most thoroughly in this paper is the problem of understanding and mitigating changes in the Earth's climate.

I conclude a supercomputer based on reversible logic has some chance of addressing the example climate problem, but only if the direction of research into both reversible logic and climate modeling are altered somewhat. The problem lies with the tradeoff between parallelism and the combination of speed and power.

- Software developers tend to code to only as much parallelism as their current supercomputers require; yet future supercomputers must have more parallelism and the developers will have to accommodate.
- Reversible logic advocates tend toward slow devices both because of the "adiabatic principle" and because nanotech devices are hard to manufacture. However, reversible logic designers will need to work towards fast and low power devices.

This paper considered only supercomputing: it may also be possible to justify reversible logic on the basis of lower capacity, low power computers for mobile devices (such as cell phones and robots); I did not study this problem.

In my view, this analysis reveals:

- Algorithms need more parallelism.
- Algorithms need to be developed to the uneven landscape of emerging nanotech devices operating reversibly.
- Nanotech devices need to be better characterized.
- Nanotech devices need to be fast and low power.
- Reversible microprocessor-like architectures for executing Fortran and C++ are needed.
- Synchronization and load balance issues were investigated but not reported in this paper due to length limitations.
- Molecular quantum dot cellular automata [22] have no equivalent of a wire, using instead a string of buffer gates strung end-to-end with a propagation speed of about 10^{-4} of the speed of light. This paper accounts for this wire speed for long distance communications, but it will also impact microacrhitecture. I ignored this effect, but Sarah Frost is doing research in this area.

11. ACKNOWLEDGMENTS

This paper is based on, but is a substantial extension to, material presented at the Workshop on Extreme Supercomputing held in Santa Fe, New Mexico on October 12, 2004 [7] and acknowledgement is made to participants of that workshop.

12. REFERENCES

[1] IBM Blue Gene/L system, http://www.research.ibm.com/bluegene/.

[2] Cavin, R. K. and Zhirnov, V. V. Silicon nanoelectronics and beyond: reflections from a semiconductor industry-government workshop, *Journal of Nanoparticle Research* 6: 137-147, 2004.

[3] Climate Change Science Program (CCSP), Strategic Plan for the U.S. Climate Change Science Program, Washington, DC, July 2003, available at http://www.climatescience.gov.

[4] Christopher, T. W. Radiation Transport Algorithms on Trans-Petaflops Supercomputers of Different Architectures *Sandia National Laboratories Technical report SAND2003-2814*, August 2003.

[5] Christopher, T. W. Pressures the Radiation Transport Problem Places on Future PIM-Based Supercomputer Designs, *Workshop on Software for Processor-In-Memory Based Parallel Systems held in conjunction with the Second Annual IEEE/ACM International Symposium on Code Generation and Optimization*, March 21, 2004.

[6] Davis, K., Hoisie, A., Johnson, G., Kerbyson, D. Lang, M., Pakin, S., Petrini, F. Blue Gene: A Performance and Scalability Report at the 512-Processor Milestone, *Los Alamos National Laboratories unlimited release technical report LA-UR- 04-1114*.

[7] DeBenedictis, E. *Workshop on Extreme Supercomputing, held in Santa Fe, New Mexico, October 12, 2004*, presentations available at http://www.zettaflops.org.

[8] DeBenedictis, E. Will Moore's Law be Sufficient?, in *proceedings of Supercomputing 2004*.

[9] US Department of Energy, The Challenge and Promise of Scientific Computing, *Report from US Department of Energy*.

[10] Feynman, R. P. Simulating Physics with Computers, *International Journal of Theoretical Physics*, Vol. 21. Nos. 6/7, 1982.

[11] Fournier, A., Taylor, M. A. and Tribbia, J. The Spectral Element Atmosphere Model (SEAM): High-resolution parallel computation and localized resolution of regional dynamics," *Monthly Weather Review*, 132 (2004) 726-748.

[12] Kozyrakis, C., Patterson, D., Overcoming the Limitations of Conventional Vector Processors, *30th International Symposium on Computer Architecture*, San Diego, CA, June 2003.

[13] International Technology Roadmap for Semiconductors, http://public.itrs.net, 2002 update.

[14] Jardin, S.C., Plasma Science Contribution to the SCaLeS Report, *Princeton Plasma Physics Laboratory, PPPL-3879 UC-70*, available on Internet.

[15] Landauer, R., Irreversibility and heat generation in the computing process, *IBM J. Res. Dev.* 5, 183-191, 1961.

[16] Malone, R. C., Drake, J. B., Jones, P. W., Rotman, D. A. High-End Computing in Climate Modeling, *Los Alamos National Laboratory technical report LA-UR-04-7128*.

[17] NASA Goddard Space Flight Center, *Advanced Weather Prediction Technologies: NASA's Contribution to the Operational Agencies*, available on Internet.

[18] Workshop on the Science Case for Large-scale Simulation (SCaLeS), June 24-25, proceedings on Internet http://www.pnl.gov/scales/.

[19] Sterling T. L. and Zima, H.P. Gilgamesh: A Multithreaded Processor-In-Memory Architecture for Petaflops Computing, *Proceedings of Supercomputing02*, Baltimore, Maryland, November 18-22, 2002.

[20] Sunaga, T., Kogge, P. M. et. al. A Processor In Memory Chip for Massively Parallel Embedded Applications, *IEEE J. of Solid State Circuits*, Oct. 1996, pp. 1556-1559.

[21] Timler, J. and Lent, C.S. Power gain and dissipation in quantum-dot cellular automata, *Journal of Applied Physics* 91, 823-831 (2003).

[22] Timler, J. and Lent, C. S. Maxwell's demon and quantum-dot cellular automata, *Journal of Applied Physics* 94, 1050-1060 (2003).

[23] The list of the top 500 largest supercomputers. Maintained online at http:// www.top500.org.

[24] Upchurch, E., Sterling, T., and Brockman, J. Analysis and Modeling of Advanced PIM Architecture Design Tradeoffs, in *Proceedings of the 6th International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems* (IWIA03), p66-77, 2003.