

Introduction to Reversible Computing: Motivation, Progress, and Challenges

Michael P. Frank
FAMU-FSU College of Engineering
2525 Pottsdamer St., Rm. 341
Tallahassee, FL 32310
00-1*-850-410-6463
mpf@eng.fsu.edu

ABSTRACT

Reversible computing is motivated by the von Neumann-Landauer (VNL) principle, a theorem of modern physics telling us that ordinary irreversible logic operations (which destructively overwrite previous outputs) incur a fundamental minimum energy cost. Such operations typically dissipate roughly the logic signal energy, itself irreducible due to thermal noise. This fact threatens to end improvements in practical computer performance within the next few decades. However, computers based mainly on *reversible* logic operations can reuse a fraction of the signal energy that theoretically can approach arbitrarily near to 100% as the quality of the hardware is improved, reopening the door to arbitrarily high computer performance at a given level of power dissipation. In the 32 years since the theoretical possibility of this approach was first shown by Bennett, our understanding of how to design and engineer practical machines based on reversible logic has improved dramatically, but a number of significant research challenges remain, *e.g.*, (1) the development of fast and cheap switching devices with adiabatic energy coefficients well below those of transistors, (2) and of clocking systems that are themselves of very high reversible quality; and (3) the design of highly-optimized reversible logic circuits and algorithms. Finally, the field faces an uphill social battle in overcoming the enormous inertia of the established semiconductor industry, with its extreme resistance to revolutionary change. A more evolutionary strategy that aims to introduce reversible computing concepts only very gradually might well turn out to be more successful. This talk explains these basic issues, to set the stage for the rest of the workshop, which aims to address them in more detail.

Categories and Subject Descriptors

B.7.1 [Integrated Circuits]: Types and Design Styles – *Advanced technologies*. B.8.2 [Performance and Reliability]: Performance Analysis and Design Aids. C.1 [Processor Architectures]: Other Architecture Styles. F.2.3 [Analysis of Algorithms and Problem Complexity]: Tradeoffs between Complexity Measures

General Terms: Algorithms, Performance, Design, Economics, Reliability, Theory.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CF'05, May 4–6, 2005, Ischia, Italy.

Copyright 2005 ACM 1-59593-019-1/05/0005...\$5.00.

Keywords: Digital logic technologies, reversible computing, low-power computing, high-performance computing, limits of computing, power management, VLSI, field-effect devices, reversible logic, unconventional computing, computer architecture.

1. INTRODUCTION

1.1 The Impending Performance Crisis

To a great extent, the history of improvements in computing machinery can be summarized as a history of steady improvements in one key system-level figure of merit: namely, the achievable energy efficiency, defined as the number of useful information processing operations (including logic, storage, and communication operations) that can be performed per unit of available energy that is dissipated to the environment as heat. From ancient stone tablets to 19th-century mechanical calculators to the latest VLSI chips with virus-sized transistors, advances in practical computer performance have gone hand-in-hand with improvements in this key quantity. It is easy to see why by considering the following trivial equation:

$$R = \frac{N_{ops}}{t} = \frac{N_{ops}}{E_{diss}} \times \frac{E_{diss}}{t} = F_E \times P_{diss} \quad (1)$$

where R = performance, N_{ops} = number of useful operations performed during a job, t = total elapsed time to perform the job, E_{diss} = energy dissipated during the job, $F_E = N_{ops}/E_{diss}$ = energy efficiency, $P_{diss} = E_{diss}/t$ = average power dissipation during the job. The power dissipation that is tolerable in a given application context is always limited by some practical consideration, such as a requirement that a limited supply of available energy (such as in a battery) not be used up within a given time, or by the limited rate of heat removal in one's cooling system, or by a limited operating budget available for buying energy. Thus, improving system performance generally requires increasing the average energy efficiency F_E of useful operations.

In the last twenty years, since the field-effect transistor became the basis for most high-performance digital logic, energy efficiency for the lowest-level ops has been roughly given by $F_E \approx (1 \text{ op})/(\frac{1}{2}CV^2)$, where C is the typical capacitance of a node in a logic circuit, and V is the typical voltage swing between logic levels. This is because voltage-coded logic signals have an energy of $E_{sig} = \frac{1}{2}CV^2$, and this energy gets dissipated whenever the node voltage is changed by the usual irreversible FET-based mechanisms in modern CMOS technology. The improvements in the practical performance of digital systems over the last 20 years can

thus be attributed primarily to an exponential decline in C over this same period (in proportion to shrinking transistor lengths), together with an additional factor of $\sim 25\times$ coming from a reduction of the typical logic voltage V from 5V (TTL) to around 1V today.

Unfortunately, the “dirty little secret” of the semiconductor industry (actually, not much of a secret these days) is that they are all very quickly running out of good new ideas for further improving energy efficiency [11]. Logic voltages can’t go much below 1 V without compromising the ability of transistors to turn off effectively, which hurts energy efficiency, since it leads to large standby power consumption due to thermally-activated leakage of electrons across the transistor’s voltage barrier. Meanwhile, logic node capacitances can’t be decreased much further, because reduced transistor sizes also require lower voltages, in order to avoid problems with various additional forms of leakage and breakdown of devices. New transistor materials structures may alleviate these problems, but only to a limited extent.

No matter what, as soon as the signal energy $E_{sig} = \frac{1}{2}CV^2$ becomes small in comparison with the thermal energy $E_T = k_B T$, (where k_B is Boltzmann’s constant and T is the temperature), digital devices can no longer function reliably, due to problems with thermal noise [21]. For a reasonable level of reliability, the signal energy should actually be much larger than the thermal energy, $E_{sig} \gg E_T$. For example, a signal level of $E_{sig} \gtrsim 100 k_B T \approx 2.6$ eV (at room temperature) gives a decently low error probability of around $e^{-100} = 3.72 \times 10^{-44}$. (Incidentally, it may initially seem that simply operating at low temperatures would allow the energy dissipation for a given level of reliability to be smaller, but in reality, this does not help, because T effectively becomes the temperature of the environment when the added energy dissipation in even an ideal cooling system is taken into account [7].)

It is interesting to note that the energies of the smallest logic signals today are already only about $10^4 kT$ [20], which means there is only about a factor of 100 of further performance improvements remaining, before we begin to lose reliability. Error-correcting codes can be used, but they impose significant encoding overheads and so don’t end up helping energy efficiency; this is because it is the *total* energy of the encoded bit that is what really matters, for purposes of both dissipation and reliability. Furthermore, a firm lower limit on dissipation of $E_{diss} \geq k_B T \ln 2 \approx 18$ meV (in room-temperature environments) can be derived for conventional (irreversible) logic from basic thermodynamic considerations, even if reliability issues are ignored [33],[22].

A factor of 100 means only around 10 years remain of further performance improvements, given the historical performance doubling period of about 1.5 years. Thus, by about 2015, the performance of conventional computing will stop improving, at least at the device level—application-specific or reconfigurable hardware may potentially squeeze out another factor of 100, and new algorithms with lower asymptotic complexity may be found for many interesting problems. But for the majority of simple, general applications, where a poor algorithm is not the main bottleneck, and the architecture is a general-purpose processor, performance improvements will effectively halt at ~ 100 times today’s levels—for *ever*! This appears to be an imminent “end of the road” scenario for the history of computing technology, as far as performance on most applications is concerned. Or, is it?

1.2 Reversible Computing to the Rescue

Enter reversible computing [3]. In a phrase, the primary motivation for reversible computing lies in the fact that it provides the *only* way (that is, the only way that is logically consistent with the most firmly-established principles of fundamental physics) that performance on most applications within realistic power constraints might still continue increasing indefinitely, even after signal energies run up against the thermal noise floor of $\sim 100 k_B T$.

How is this possible? Let’s look back at eq. (1). We note that practical performance (given some power limit) is constrained by F_E , the number of operations that can be performed per unit energy *dissipated*. But, the $100 k_B T$ limit only directly refers to the energy *contained* in the logic signal, in the sense that this is the magnitude of the thermal fluctuation of that signal that would need to occur to spontaneously change a 0 to a 1 (or vice-versa). However, there is nothing fundamental about information processing that forces us to *dissipate* to heat all the energy that is contained in a signal, when that signal is processed in some way (storing, sending, or flipping a bit). Instead, we can potentially recover and retain (in an organized form that can be reused for subsequent operations) a fraction of the signal energy that approaches as close as we wish to 100% [3]. That is, there is *no* known positive lower bound on the fraction of signal energy that must be dissipated. Instead, as technology improves, potentially the fraction of the signal energy that gets dissipated with each operation can become ever closer to the limit of 0. Meanwhile, while still decreasing dissipation, the signal energy itself can even be made *larger* (at only a logarithmically increasing rate) in order to suppress the rate of dissipation due to errors resulting from thermal noise. All in all, the total number of computational operations that can be performed while only dissipating a fixed amount E_{diss} of energy to an outside environment at temperature T can (theoretically) approach infinity! Thus, it is possible that computer performance, within fixed power limits, can continue improving well beyond the limits of conventional technology.

What’s the catch? The catch is that in order for us to avoid all of the imminent energy dissipation limits, in particular the $k_B T \ln 2$ limit [22], it will be necessary for the *logical* design of the computer, as well as the hardware technology, to radically change [3]. That is, the computation must be *logically reversible*—every state of the machine (or, an increasingly large fraction of them) must have *only one* possible predecessor state that could be arrived at during the course of a computation. (Actually, the true requirement is that the number of predecessors must be, on average, the same as the number of successors; but in ordinary deterministic computations, the number of successor states is 1.)

That logical reversibility is required follows rather trivially from certain basic facts of fundamental physics [12]. All of our enormously successful fundamental physical theories (including Einstein’s general theory of relativity, quantum electrodynamics, and the standard model of particle physics) share the property that they can all be described as special cases of the more general concept of *Hamiltonian* dynamical systems; in these, the system’s dynamics arises from certain generic differential equations called the *Hamilton’s equations*, which are first-order in the time t [17]. The time differential dt in these equations can be taken as being either positive or negative, which means that the equations are deterministic looking either forwards or backwards in time. (In quantum mechanics, the behavior of quantum systems *appears* nondeterministic to us only because we can’t access the complete quantum state.) Thus, every physical state uniquely determines its

predecessor states. This is still true in quantum mechanics if by “state” we mean a quantum state vector or wavefunction, and in general relativity if by “state” we mean a spacelike hypersurface. Thus, we can say that *physics is reversible*. (This is fully as certain a statement as any we can make about physics; it is backed up by enormous, overwhelming mountains of evidence.)

Because of the reversibility of physics, whenever we have a situation where a single logical state of a machine could have been arrived at from either of two possible predecessors, either of which would also have comprised a valid logical state of the machine, it must be the case that the new state must have an increased number of possible representations in terms of *physical* states, namely, one for each representation of each possible predecessor state. For example, if the new state has 2 logical predecessors, each of which could be represented (in the context of a given machine implementation) by N distinct physical states, then the new logical state must have $2N$ possible distinct physical representations. Insofar as our model of the machine does not keep track of *which* one exactly of the possible physical representations is being used at any given moment, we can say that the number of possible representations (consistent with the instantaneous logical description) has doubled, or in other words, its logarithm has increased by an additive amount of $\log 2$. The logarithm of the number of possible states is also known as the *entropy*, and it can be measured in bits, giving the size of the optimal compressed encoding of the state. Thus, entropy has increased by an amount $\log 2 = 1$ bit. Because of the reversibility of physics, once our model has lost track of a bit’s worth of state information, we can never again regain it; that would require two physical states to merge into one, which is impossible, by the very mathematical form of Hamiltonian dynamics. This is, in fact, the proof of the 2nd law of thermodynamics, which says that entropy only increases, never decreases. It increases whenever our model of the system, over time, loses track of some part of the system’s state, or in other words, accumulates uncertainty regarding the actual state.

Now, when that bit’s worth of entropy gets expelled into the environment (as it must be eventually, if we don’t want the state of our computer to decay to a completely uncertain equilibrium state as entropy is accumulated), then if the environment is at temperature T , then energy $E_{diss} \geq T \cdot (1 \text{ bit})$ must also be expelled into the environment. This is true by the very thermodynamic definition of temperature, namely $T = \partial E / \partial S$, where S is entropy [30]. Now, 1 bit, or $\log 2$, is mathematically equal to $(\ln 2) \times (\log e)$, and $\log e$, considered as a unit of physical entropy, is the definition of Boltzmann’s constant k_B , so we get $E_{diss} \geq k_B T \ln 2$. The fact that one bit’s worth of lost logical information always leads to at least this amount of physical energy dissipation is called the *von Neumann-Landauer (VNL) principle*, after its discoverers [33],[22]. Thus, to avoid this limit, we must avoid losing track of logical information. That is, the information-processing operations within the machine must be *logically reversible*, meaning that they transform the state in an invertible way, and can be undone.

One might initially guess that this constraint of logical reversibility might be such a stringent one that general-purpose computation can not still be accomplished under its purview, but, in 1973, Charles Bennett of IBM research showed (with a simple Turing machine construction) that in fact, *any* desired computation can always be embedded within an equivalent reversible one, essentially by just keeping around all the information that would otherwise be discarded, and then “decomputing” it later, after it is

no longer needed [3]. However, Bennett’s construction only addressed the logical level, and left open the question of how, precisely, to build a practical physical mechanism for computation that would also be physically reversible, or very nearly so.

In the 32 years that have passed since Bennett’s paper, much progress has been made towards this goal; this will be reviewed in the next section. However, many engineering challenges still remain to be solved before reversible computing can become a practical basis for ultra energy-efficient, high-performance computing; these will be reviewed in section 3.

However, there are good physical reasons to expect that the remaining problems *can* be solved, given sufficient time, research, and good engineering. The underlying reason for this optimism is that, fundamentally, to say that energy has been *dissipated* (or that entropy has increased) is only to say that we have *lost control* of that energy, or that we have *lost track* of its detailed state, of exactly how it is distributed among the various degrees of freedom of our system. But, the entire centuries-long history of engineering and of physics is a largely a story of ever more precise manufacturing and control of physical artifacts, and ever more accurate characterizations of the fundamental dynamical laws that determine how those artifacts will behave. Thus, we can expect that this trend will continue, and that over time, we will be able to construct increasingly complex systems whose initial state is increasingly accurately specified (eventually in full atom-by-atom detail [8]), and whose built-in dynamical behavior is increasingly close to a pre-designed trajectory, with increasingly slow rates of drift away from that trajectory in unpredictable directions (increasingly low rates of entropy increase). At present, there is no known fundamental reason to think that such complex, precisely-specified, highly-predictable physical systems cannot include scalable, efficient, general-purpose programmable computers that generate only a negligibly small amount of new entropy with each logic operation that they perform.

2. REVIEW OF PROGRESS SINCE 1973

How is reversible computing to be physically implemented? In Bennett’s early papers on the subject, the only concrete physical constructions that were offered were so-called “Brownian motion machines,” in which the mean free path of the system’s trajectory was much shorter than the distance between neighboring computational states [4]. In absence of any energy input, the system progressed essentially via a random walk, taking an expected time of $\Theta(n^2)$ to advance n steps.

For the examples Bennett studied, even if the random walk is biased in the forwards direction by a small energy input, to achieve a linear rate of progress, the performance is still very low—Bennett [4] gave the biological example of DNA polymerization, which (under normal conditions, such as during cell division) proceeds at a rate on the order of only 1,000 nucleotides per second, with a dissipation of $\sim 40 k_B T$ per step.

In general, asymptotically reversible processes (including the DNA example) proceed forward at an adjustable speed, proportional to the energy dissipated per step. We can thus characterize the inefficiency of a given such process by the constant *energy coefficient* $c_E = E_{diss} / f_{op}$, where E_{diss} is the energy dissipated per operation, and f_{op} is the frequency of operations [12]. In Bennett’s original DNA process, the energy coefficient comes out to about $c_E = 1 \text{ eV/kHz}$. That is nice, but we’d prefer to run at GHz frequencies, or greater, with energy dissipation per op that is less than $k_B T$. So the question becomes, can we engineer reversible systems

that have lower energy coefficients than DNA? It turns out that we already have.

In the late 1970's, Ed Fredkin, Tommaso Toffoli, and other members of the Information Mechanics group at MIT envisioned the concept of *ballistic* computing, in which the state of the machine would proceed forwards along its trajectory under its own momentum, as it were, with only a small fraction of the signal energy being dissipated to heat upon each operation [15]. Their original physical picture was that of idealized, perfectly elastic billiard balls bouncing around on a frictionless pool table. Of course, this picture was just inspiration, and was not a serious implementation proposal. However, they also proposed a closely analogous, but more feasible electronic implementation, which involved charge packets bouncing around along inductive paths between capacitors [16]. Fredkin and Toffoli never got the opportunity to build an electronic implementation of their ideas, but other groups did.

In particular, the ideas already being tossed around at MIT were imported to Caltech with some help from the famous physicist Richard Feynman, who had been interacting with the IM group at the time [19], and teaching about reversible computing in his Lectures on Computation at Caltech [9]. Carver Mead (also at Caltech) had previously tried (but failed) to prove in his VLSI textbook [25] that reversible computing was impossible, which Feynman refuted by developing a full quantum model of a serial reversible computer, in a paper [10] that also helped to spawn the field of quantum computing. At about the same time, Charles Seitz and colleagues at Caltech described [29] a new technique for implementing reversible computing with MOSFETs which only required a small number of large inductors, which were brought off-chip; this key step made it easy to fabricate and test experimental reversible chip designs. In the early 1990's, these techniques were further refined by groups at ISI [2], Amherst [18], Xerox [26], and MIT [34], bringing them to the point where arbitrary, pipelined, sequential logic could be implemented in a fully-reversible fashion, limited only by the energy coefficients and leakage currents of the underlying transistors. There has been a small explosion of activity in these *adiabatic circuits* since then, with a wide variety of partially-adiabatic circuits being explored, and far too many references to list here.

What are the energy coefficients of these new technologies? In 1998 [12], I estimated the energy coefficient of a 0.5- μm transistor technology that we were using at the time as about 0.77 eV/kHz, already a little better than the biological DNA example. In a more recent, 0.18- μm technology with a newer circuit style [1], we estimated about $c_E = 3$ meV/kHz, a factor of $250\times$ less.

How much lower might energy coefficients become in the future? It is difficult to tell for certain, but a wide variety of post-transistor device technologies have been proposed (*e.g.*, [24],[27],[31]) that have energy coefficients ranging from 10^5 to 10^{12} times lower than present-day CMOS! This translates to logic circuits that could run at GHz to THz frequencies, with dissipation per op that is still less (in some cases orders of magnitude less) than the VNL bound of $k_B T \ln 2$ [22] that applies to all irreversible logic technologies. Some of these new device ideas have even been prototyped in laboratory experiments [28]. Thus, there appears to be enormous potential for ongoing improvement of reversible device technologies for many decades to come.

Meanwhile, on the system design front, our understanding of the various design issues and tradeoffs inherent to reversible computing has also improved substantially. In work that also had

its roots in the IM group at MIT, fully-reversible processor architectures [14] and instruction sets [32] have been designed and implemented in silicon. Meanwhile, on the algorithms side, a more spacetime-efficient embedding of general irreversible computations into reversible ones was introduced by Bennett in 1989 [5]; various authors have further explored the algorithmic space/time tradeoffs in reversible computing in the years since then [23],[6]. I myself carried out a variety of analyses (*e.g.*, [12],[13]) to determine how computer cost, performance, and power all interacted with each other in optimized reversible system designs, and to see how quickly the advantages of reversible machines would increase as various device-level technology parameters and application parameters were scaled. Generally, the advantages increase as sublinear polynomials—square roots, cube roots, *etc.*—of the key input parameters. Though they increase only slowly, these asymptotic advantages are real, and it is important to note that they still exist even despite the substantial overheads of the reversible paradigm. Therefore, we can expect that as technology improves, the advantage of the new reversible approach compared to the old irreversible one will gradually increase, until eventually the advantages could become so overwhelming that a *disruptive paradigm shift* may suddenly occur, in which the new paradigm rapidly overtakes the prevailing technology, when the advantages of switching to the new paradigm outweigh the costs of making the switch. It is difficult to forecast exactly when this transition will occur, although some projections based on present trends [13] suggest that the performance advantages of the reversible approach (which are negligible today) could be as much as $1,000\times$ by mid-century. A substantial switch to the new paradigm could be warranted even much sooner.

It is also possible that the shift could take place gradually, as reversible and adiabatic principles gradually penetrate more and more thoroughly throughout the system design. Processor designers today are already experimenting with adiabatic charging of large loads such as clock distribution networks and I/O buses. Once these subsystems are no longer the power dissipation bottleneck, then increasingly-adiabatic, increasingly thoroughly-reversible logic could become the next target for further power reductions. Meanwhile, several generations of new post-transistor devices could be introduced, each with a lower energy coefficient than the previous one, and each one may find a market in niche ultra-low-power applications, before being adopted for high-performance computing. Device costs may continue gradually decreasing, through improved manufacturing processes, allowing higher and higher degrees of logical reversibility to become cost-effective solutions for real high-performance products.

However, whether the shift to reversible computing takes place suddenly or only gradually, we can be confident in saying that such a shift *must* eventually occur, unless the entire world forever continues to turn a blind eye to the enormous potential value of the reversible approach. If even one nation or one large company manages to shake off their misplaced doubts and misconceptions about reversible computing long enough to turn serious resources (in the form of numerous very bright engineers, backed with substantial funding) towards solving the few key remaining engineering problems, I have little doubt that the problems can be hurdled, and that reversible computing can become a commercially viable reality, one that comes to entirely dominate the world of high-performance and low-power computing, and that leaves all non-reversible approaches in its dust.

Literally all of computer science and computer engineering will be affected by this transition—we will require new logic circuits, hardware description languages, processor architectures, programming languages, algorithms, and new frameworks for application design. Everything that is old news in computer science today will suddenly become fresh and new again, as we are forced to reconstruct the foundations of the entire field, from the bottom up, in a new style. It promises to be a most exciting adventure for everyone involved.

3. CHALLENGES TO BE FACED

In the previous sections, I have reviewed the underlying motivations for reversible computing, and the recent progress towards realizing it. Although many of the past doubts about the possibility and practicality of reversible computing have already been erased by various concrete developments that have taken place in the field over the course of the last 32 years, the R&D challenges that remain are nevertheless still significant ones. (If this were not the case, RC would likely already be on our desktops.) Here are what I think are some of the key, most important challenges that need to be addressed in coming years, in order to make significant progress in moving the field forward:

1. **Fast, cheap, low- c_E devices.** MOSFETs are fundamentally too resistive and leaky, and better switching devices are needed. We need new devices that have low manufacturing cost, high maximum frequency, low leakage rate and error probability, and a low adiabatic energy coefficient c_E , which recall is energy dissipated per op, per unit operating frequency. There are numerous nanoscale device concepts floating around (e.g., [27],[31]) that have been analyzed (on paper and in simulation) to have c_E that is many orders of magnitude better than CMOS, while retaining small size, high speed, low leakage, and high reliability. The most promising of these concepts need to be refined, prototyped, and empirically verified, and then inexpensive manufacturing techniques for them need to be developed.
2. **High- Q resonant power supplies.** All reversible computing technologies require some sort of resonant power/clock signal to drive and synchronize the adiabatic logic transitions throughout large design blocks. The quality factor of the resonator directly limits the advantages that can be gained from reversibility. Simple isolated systems (such as vibrating crystals in vacuum) are known to have quality factors of 10^{14} (and other well-isolated quantum systems are known to have even higher quality), which suggests there is a lot of room for improvement in this area. Certainly, we know of no firm limits on how good the Q factors can become. But the engineering of high- Q resonators is presently somewhat of a “black art;” designers typically find and eliminate dissipation mechanisms in a slow, iterative, empirical process. Better design methodologies for high- Q resonator design are needed. (This is an area where good progress would likely pay off in the near term for conventional irreversible technology as well, giving us low-power, energy-recovering clock distribution systems.)
3. **Avoiding back-action on the clock.** An important point to keep in mind is that if the instantaneous state of the clock signal becomes uncertain due to data-dependent interactions with the logic, this effectively means increased entropy in the clock signal, and effective dissipation which reduces the Q of the clock. (Theoretically, the clock must be a perfectly peri-

odic signal with zero bandwidth, in order to approach infinite Q .) The load in the logic must thus be carefully balanced so as to remain constant from one cycle to the next, or else to vary only in predictable ways that can be compensated for in the clock design. The implications of this constraint for the tradeoffs and overheads of reversible system design need to be more carefully studied. It would be reassuring to build a complete, self-contained model of logic+clock system that maintained a high overall Q while performing some non-trivial computation.

4. **Proof-of-concept prototypes.** In connection with the above, some physical prototype (even if it is initially too expensive to be practical) needs to be built that measurably dissipates substantially less than $kT \ln 2$ energy per logic operation (including in the clocking system) while performing a non-trivial computation, in order to finally silence the die-hard skeptics who still maintain today (though without any proof) that reversible computing must be impossible. Or, if it turns out that it really is impossible, for some as-yet-unknown reason, it is in the process of attempting to complete this prototype-building step that would finally give us the detailed empirical experience that might help us to understand *why* it must be so; this would then comprise a rather important new fundamental discovery about physics.
5. **Reversible design infrastructure.** Obviously, in order for reversible computing to become successful in practice, there must eventually be a large investment in the development of supporting design tools and application-specific hardware and software algorithms. This includes “reversibility-aware” versions of gate libraries, hardware description languages, ASIC libraries, processor and FPGA microarchitectures, instruction sets, and (eventually) even high-level languages, subroutine libraries, and reversible high-level algorithms for specific applications of interest. The lowest levels of this hierarchy of software tools will need to be modified first, with the higher levels becoming necessary to address only later, as the *degree of reversibility* (fraction of energy recovered per cycle) adds more 9’s (99.9% energy recovery, etc.)

4. CONCLUSION

The challenges that reversible computing faces are difficult, and it may take a concerted effort on the part of the semiconductor industry, the broader computing industry, and government to make the needed research progress occur quickly enough to prevent computer performance from stalling soon, perhaps for a noticeably extended period. However, despite my years of careful study of all the relevant issues, I don’t see any good reason yet to expect that the challenges listed above (and new ones that may yet arise) cannot be successfully tackled and overcome, through concerted engineering effort. Achieving reversible computing will clearly be a prerequisite in order for us to make significant further progress in computer performance. It is thus well worth trying to achieve, and the sooner we seriously get started on it, the sooner we may break free of the shackles of the present power dissipation crisis, and resume our past trend of steady progress in practical computer performance.

Thus, I implore the scientific and engineering community: Let us not give up the ghost at this early date. Let us face the situation as it stands, bravely tackle the remaining challenges of reversible computing, and see where this effort takes us. If we don’t make a serious and persistent effort fairly soon, and if pro-

gress dulls to permanent stagnation, we may never know the magnitude of the opportunities that we are missing by failing to act now. But if, instead, we give to reversible computing all of the best efforts that our brightest minds can reasonably muster today, then it just might take us farther than we ever would have imagined possible.

5. ACKNOWLEDGEMENTS

I would like to thank all of the Computing Frontiers organizers for inviting me to put together this special session, and for managing conference administrative matters. Sarah Frost of Notre Dame deserves a special thanks for helping to organize the special session, and, last but not least, I am grateful to all of the many authors participating in the special session for contributing a large number of excellent and very interesting papers.

6. REFERENCES

- [1] Anantharam, V., He, M., Natarajan, K., Xie, H., and Frank, M. P. Driving Fully-Adiabatic Logic Circuits Using Custom High-Q MEMS Resonators. In *Proc. Int. Conf. on Embedded Systems & Applications*, CSREA, 2004, 5-11.
- [2] Athas, W. C., Svensson, L. J., Koller, J. G., Tzartzanis, N. and Chou, E. Y.-C. Low-power digital systems based on adiabatic switching principles. In *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 2, 4, (Dec. 1994), 398-407.
- [3] Bennett, C. H. Logical reversibility of computation. *IBM J. Res. Dev.*, 17, 6 (1973), 525-532.
- [4] Bennett, C. H. The Thermodynamics of Computation—a Review. *Int. J. Theo. Phys.*, 21, 12 (1982), 905-940.
- [5] Bennett, C. H. Time/space trade-offs for reversible computation. *SIAM J. Comput.*, 18, 4 (1989), 766-776.
- [6] Buhrman, H., Tromp, J., Vitanyi, P. Time and space bounds for reversible simulation. *J. Phys. A: Mathematical and General*, 34 (2001), 6821-6830.
- [7] Carnot, S. *Reflexions sur la puissance motrice du feu sur les machines propres a developper cette puissance*. Bachelier, Paris, 1824. English translation in press: *Reflections on the Motive Power of Fire*, Peter Smith Publisher, 1992.
- [8] Drexler, K. E. *Nanosystems*, Wiley, 1992.
- [9] Feynman, R. P., *Feynman Lectures on Computation*, Perseus Books, 1996.
- [10] Feynman, R. P. Quantum mechanical computers. *Found. Phys.*, 16, 6 (1986), 507-531.
- [11] Frank, D. J. Power constrained CMOS scaling limits. *IBM J. Res. Dev.*, 46, 2/3 (2002), 235-244.
- [12] Frank, M. P. *Reversibility for Efficient Computing*, Ph.D. thesis, Massachusetts Institute of Technology, 1999.
- [13] Frank, M. P. Nanocomputer systems engineering. In *Nanotech 2003: Tech. Proc. of the 2003 Nanotechnology Conf. and Trade Show*, vol. 2, Computational Publications, 2003, 182-185.
- [14] Frank, M. P., Vieri, C., Ammer, M. J., Love, N., Margolus, N. H., and Knight, T. F., Jr. A scalable reversible computer in silicon. In *Unconventional Models of Computation*, Springer, 1998, 183-200.
- [15] Fredkin, E. F., Toffoli, T. Conservative logic. *Int. J. Theo. Phys.*, 21, 3/4 (1982), 219-253.
- [16] Fredkin, E. F., Toffoli, T. Design principles for achieving high-performance submicron digital technologies, proposal to DARPA, Nov. 1978. Reprinted in *Collision-Based Computing*, Fredkin, E. F. and Toffoli, T., eds., Springer-Verlag, 2001.
- [17] Goldstein, H., Poole, C. P., Safko, J. L., *Classical Mechanics*, 3rd ed., Addison Wesley, 2002.
- [18] Hall, J. S., An electroid switching model for reversible computer architectures. In *PhysComp '92: Proc. of the Wkshp. on Physics and Computation, Oct. 2-4, 1992, Dallas, Texas*, IEEE Computer Society Press, 1992, 237-247.
- [19] Hey, A. J. G., ed., *Feynman and Computation: Exploring the Limits of Computers*, Perseus Books, 1999.
- [20] International SEMATECH. *International Technology Roadmap for Semiconductors* (2004). <http://public.itrs.net>.
- [21] Kish, L. B. End of Moore's law: thermal (noise) death of integration in micro and nano electronics. *Phys. Lett. A*, 305, 3-4 (2 Dec. 2002), 144-149.
- [22] Landauer, R. Irreversibility and heat generation in the computing process. *IBM J. Res. Dev.*, 5 (1961), 183-191.
- [23] Lange, K.-J., McKenzie, P., and Tapp, A. Reversible space equals deterministic space. In *Proc. 12th Ann. IEEE Conf. on Computational Complexity (CCC '97)*, 1997, 45-50.
- [24] Likharev, K. K. Classical and quantum limitations on energy consumption in computation. *Int. J. Theo. Phys.*, 21, 3/4 (1982), 311-326.
- [25] Mead, C. and Conway, L., *Introduction to VLSI Systems*, Addison-Wesley, 1979.
- [26] Merkle, R. C. Reversible electronic logic using switches. *Nanotechnology*, 4 (1993), 21-40.
- [27] Merkle, R. C. and Drexler, K. E. Helical logic. *Nanotechnology*, 7, 4 (1996), 325-339.
- [28] Orlov, A. O., Kumamuru, R. K., Ramasubramaniam, R., Toth, G., Lent, C. S., Bernstein, G. H., Snider, G. L. Experimental demonstration of a latch in clocked quantum-dot cellular automata, *Appl. Phys. Lett.*, 78 (2001), 1625-1627.
- [29] Seitz, C. L., Frey, A. H., Mattisson, S., Rabin, S. D., Speck, D. A., van de Snepscheut, J. L. A. Hot-clock nMOS, in Fuchs, H., ed., *1985 Chapel Hill Conf. on Very Large Scale Integration*, Computer Science Press, 1985, 1-17.
- [30] Stowe, K. *Introduction to Statistical Mechanics and Thermodynamics*, Wiley, 1984.
- [31] Timler, J. and Lent, C. S. Maxwell's demon and quantum-dot cellular automata. *J. Appl. Phys.* 94 (2003), 1050-1060.
- [32] Vieri, C., *Reversible Computer Engineering and Architecture*, Ph.D. thesis, Massachusetts Institute of Technology, 1999.
- [33] von Neumann, J. *Theory of Self-Reproducing Automata*, University of Illinois Press, 1966. <http://www.walenz.org/vonNeumann/>.
- [34] Younis, S. G. and Knight, T. F., Jr. Asymptotically zero energy split-level charge recovery logic. In *Int. Wkshp. on Low Power Design*, 1994, 177-182.