Common Mistakes in Adiabatic Logic Design and How to Avoid Them

Michael P. Frank[1]
CISE & ECE Departments
University of Florida

[1] Email: mpf@cise.ufl.edu

Abstract

Most so-called "adiabatic" digital logic circuit families reported in the low-power design literature are actually not truly adiabatic, in that they do not satisfy the general definition of adiabatic physical processes, as ones whose energy dissipation tends towards zero as their speed and/or parasitic interactions are decreased. Yet, the need for truly adiabatic design can be proven to be a key requirement for cost-efficient digital design for the majority of general-purpose computing applications in the long run, as technology advances and power dissipation becomes an increasingly stringent limiting factor on system performance. Although they may remain useful for some specialized applications, all of these only semi-adiabatic logic styles (as well as all non-adiabatic logics) are doomed to eventual irrelevance to the bulk of the computing market, most likely within only a few decades. It therefore behooves us to begin emphasizing today how to design truly adiabatic circuits.

In this paper, I describe the most common departures from true adiabaticity in the logic designs that have been published to date, and discuss how these problems can be avoided in the future. The most common problems are: (1) use of diodes, (2) turning off transistors when there is nonzero current across them, (3) failure of the design style to accommodate arbitrarily much logical reversibility, which can be proven to be required to approach truly adiabatic operation, and (4) failure to accommodate the asymptotically most cost-efficient possible circuit algorithms, in terms of both hardware-time and energy.

I also summarize the key characteristics of a new "most general" truly adiabatic CMOS logic family that avoids all of these problems. In my group at UF, we are beginning to create an associated hardware description language and design tools that will enable complex, hierarchical adiabatic circuits to be easily composed (by hand and/or by automatic generation from irreversible designs) and automatically analyzed to locate and minimize any departures from fully adiabatic operation.

1. Introduction

In applied physics, an *adiabatic* process is defined as any process that is asymptotically isentropic (thermodynamically reversible), that is, whose total entropy generated tends towards zero in some appropriate limit (typically, of low speed and/or improved isolation of the system). As the most famous example, asymptotically reversible heat engines were first described by Carnot in 1825 [1], and were shown by him to provide the maximum possible thermodynamic efficiency. Part of the cycle of Carnot's engines involved processes with no heat flow, and this lack was the original and literal meaning of the term "adiabatic." But today, we would call the *entire* Carnot cycle adiabatic, in the more general applied-physics sense of the term, which has departed from the literal meaning.

Of course, no real physical process can be arranged to be absolutely *perfectly* isentropic (with entropy generated being *exactly* zero) since there will always be some nonzero base rate of unwanted dissipative interactions with the environment (*e.g.*, quantum tunneling, cosmic rays, asteroid impact). However, in practice, if the goal is to minimize the energy dissipation of some process, much can be done to bring the expected dissipation of the process as close to zero as is possible, within the constraints of the available technology. I use the term *adiabatics* to refer to the general engineering study of ways to minimize the entropy generation of real physical processes. The field of *adiabatic circuits* applies the general concepts of adiabatics to the design of low-power electronic circuits in particular, consisting primarily today of digital MOSFET-based switching circuits.

Some history. To my knowledge, the explicit use of the term *adiabatic* in connection with the design of nearly reversible low-power switching circuits was first made by Koller and Athas of ISI, at the 1992 Workshop on Physics and Computation in Dallas [2]; this event can be considered the formal birth of *adiabatic circuits*, as a well-defined discipline named by these two words. However, the same general circuit design concepts were also studied in the late 1970's and early 1980's by Ed Fredkin and Tomasso Toffoli at MIT [3], and by Carver Mead [4], Richard Feynman [5], and Charles Seitz and colleagues at Caltech [6]. Even earlier was work on

similar techniques by Boyd Watkins of Philco-Ford (a then-subsidiary of Ford Motor) published in JSSC in 1967 [7], though Watkins did not explicitly mention the connection between his specific circuits, and the more general phenomenon of reversible, adiabatic processes.

2. The Need for True Adiabaticity

Why is adiabatics important? First, simple economic arguments show that over the long run, as manufacturing process efficiency improves, and the cost of raw hardware resources (*e.g.*, gate-cycles) decreases, the cost of energy dissipated must eventually become the dominant part of the total cost of any computation. Even today, energy transport systems (power supplies, packaging, fans, enclosures, air-conditioning systems) comprise a significant fraction of the manufacturing and installation cost in many computing applications.

However, an even more dominant consideration is that the practical limits on cooling-system capacity (in terms of the total Watts of power that may be dissipated harmlessly in a system of given size) imply that practical hardware efficiency (*e.g.* useful bit-ops per gate-second) in any limited-size system is itself immediately impacted by the energy efficiency of the system's components. As we rapidly approach (at current rates, by the 2030's [8]) the fundamental limits to the energy efficiency of traditional irreversible technology, this effect will become even more of a concern. Moreover, the cooling problem for a given logic technology is not one that can be solved by mere engineering cleverness in one's cooling system design, as there exist absolutely fundamental and unavoidable quantum-mechanical limits on the rate at which entropy can be exported from a system of given size by a coolant flow of given power [9].

Still, engineering cleverness in the *logic*, via truly adiabatic design, *can* enable us to avoid the energy efficiency limits suffered by traditional irreversible technology, allowing us to continue improving hardware efficiency for cooling-limited applications by many orders of magnitude beyond the limits that would apply if non-adiabatic or even non-*truly* adiabatic techniques (such as most "adiabatic" techniques in the literature) were used.

The existence of adiabatic processes is an everyday fact, exemplified by the ballistic motion of a projectile in a near-vacuum environment (*e.g.* orbiting satellites). An adiabatic, ballistic process can carry out a computation, as illustrated by a simple mechanical model of adiabatic computation by Fredkin [10]. Fredkin's model was criticized by some for being unstable [11], but a little creative thought—which I will leave here as an exercise for the reader—shows that the instabilities can be easily fixed, while preserving adiabaticity, via some additional constraining mechanisms.

The *degree of adiabaticity* of any process can be defined as equal to its quality factor Q, in the sense used in electrical and mechanical engineering, *i.e.*, the ratio between the amount of free energy involved in carrying out the process, and the amount of this energy that gets dissipated to heat. Interestingly, this quantity turns out also to be the same thing as the *quantum quality factor q* given by the ratio of operation times to decoherence times in a quantum computer [12,13]. This is because the energy of any system can be interpreted as carrying out a quantum computation which updates the system's state at a certain rate of operation [14] while each quantum decoherence event effectively transforms 1 bit's worth of the quantum information in the system's state into entropy, and therefore transforms the associated energy into heat.

So, in computers, high adiabaticity implies high isolation of the system's computational state from parasitic, decoherent interactions with the environment. In ordinary voltage-coded electronic logic, such interactions include: (1) interference from outside EM sources, (2) thermally-activated leakage of electrons over potential-energy barriers, (3) quantum tunneling of electrons through narrow barriers (roughly Fermi wavelength or shorter), (4) scattering of ballistic electrons by lattice imperfections in wire/channel materials, which causes Ohmic resistance, and (5) low Q of intentionally inductive circuit components (*e.g.* in RF filters). Finally, high adiabaticity implies a low relative frequency of operations that *intentionally* transform physical coding-state information into entropy, to erase it, *e.g.*, when a circuit node is tied to a reference voltage at a different level.

Most adiabatic circuit designs today have focused on avoiding only the last mechanism of dissipation mentioned, because this one is relatively easy to avoid solely through changes in circuit design. In contrast, the other dissipation mechanisms typically require non-circuit-level solutions such as (1) electromagnetic shielding, (2) high threshold devices and/or low-temperature devices, (3) thicker, high-κ gate dielectrics, (4) low-temperature current-pulse coded superconducting circuits [15] or ballistic MOSFETs [16], (5) high-Q MEMS/NEMS electromechanical resonators [17].

We should emphasize that *both* general areas must be addressed in the long run: that is, not only the *intentional* sources of dissipation (*e.g.*, $\frac{1}{2}CV^2$ switching energy of irreversible transitions), which can be prevented through adiabatic circuit design methodologies, but also the parasitic sources of dissipation, which must be addressed through engineering device physics and package-level shielding/cooling. Both intentional and parasitic dissipation must eventually be addressed to meet the fundamental long-term requirement for maximally energy-efficient computation. In this paper, which is addressed to a circuit-design audience, I focus on what can be done at the circuit level, but this is not to imply that the other areas are not also critical ones for long term research. The efficiency benefits that be gained by working at the circuit level alone are limited (a simple application of the Generalized Amdahl's Law [18]), but we can foresee that in the

long run, further improvements can and will be made in all of these areas, so that an unlimited degree of adiabaticity in the circuit design will be beneficial.

Finally, many researchers complain that they don't see the point in adiabatic design, thinking that its overheads necessarily outweigh its benefits. This may be true for many specific low-power applications in the current technology generation and economic context, with the still-improving energy-efficiency of traditional approaches to low power, like voltage-scaling. But this is a very narrow, short-term view. A simple point of fact is that these intuitions are not borne out by a proper long-term theoretical analysis of the situation that takes all factors into account [19]. In the long run, for most applications, energy dissipation overwhelms all other concerns. This is especially so for the majority of applications which require either a compact enclosure footprint, or some degree of tightly-coupled parallelism with minimized communication delays, and therefore suffer a practical limitation on the convex-hull surface area available for cooling, so that energy efficiency ends up directly impacting not only the cost of energy itself but also the attainable hardware efficiency, and thus the effective hardware cost per unit of performance.

In the remainder of this paper, I will take for granted that the capability for arbitrarily high adiabaticity will be an essential element of our logic design methodology if it is to retain long-term relevance. In the next section, I will outline the primary mistakes, in light of this requirement, that have bedeviled most of the adiabatic circuit approaches that have been proposed to date.

3. Common Mistakes to Avoid

3.1. Don't Use Diodes

The first and simplest rule of true adiabatic design is: *never use diodes*. At the very least, one should always recognize that whenever one includes a diode as a necessary functional element in part of one's circuit (in contrast to, for example, junction diodes that are used only for device isolation or ESD protection), then that part of the design has no long-term viability and will eventually have to be replaced, as the requirements for energy efficiency become ever more stringent. The reason is that diodes, in their role as a one-way valve for current, are fundamentally thermodynamically irreversible, and cannot operate without a certain irreducible entropy generation. For example, ordinary semiconductor diodes have a built-in voltage drop across them, and this "diode drop" results in an irreversible energy dissipation of QV for an amount of charge Q carried through it. No matter what the device structure or mechanism, a dissipationless diode is equivalent to a "Maxwell's demon" for electrons, which is thermodynamically impossible (see, *e.g.*, the introduction to [20]); it is equivalent to a perpetual-motion machine, and it would violate the fundamental laws of Hamiltonian dynamics that are incorporated in all of modern physics, through quantum mechanics.

Many of the early adiabatic circuit designs, from Watkins on, used diodes in the charge return path. To the extent that the diode drop is less than logic voltage swings, so that the diode losses are much less than non-adiabatic $\frac{1}{2}CV^2$ losses, this approach may still be useful in the short run, but it must eventually be abandoned when we need still greater energy efficiency.

3.2. Don't Disobey Transistor Rules

Although diodes are fundamentally non-adiabatic, fortunately, transistors, despite being non-ideal switches, remain acceptable for adiabatic operation, so long as two basic rules are followed:

(1) Never turn on a transistor when there is a significant (non-negligible) voltage difference between its source and drain terminals.

(2) Never turn *off* a transistor when there is significant *current* flowing through its channel.

The first rule is fairly obvious, because, for example, when a dynamic node of capacitance C is directly connected to a static reference signal of voltage different from it by V, we all know there is an unavoidable dissipation of $\frac{1}{2}CV^2$ in the node's transition to its new level. Even in the best case, where both nodes are isolated and both of capacitance C, the dissipation as they converge to their average level is still $\frac{1}{4}CV^2$. (In the worst case, when the two nodes are connected to differing voltage sources, turning on the transistor results in a continuous power dissipation thereafter.) Nearly all adiabatic logic styles obey this rule, at least approximately—in light of noise considerations, leakage, *etc.*, it will in general be impossible to ensure that voltages *exactly* match before the transistor is turned on. But we should try to get as close as possible to a match.

The second rule is less obvious, and most of the purportedly adiabatic logic styles in fact fail to follow it. Why does it necessarily cause dissipation to shut off a flow of current by turning off a transistor that it is flowing through? The reason comes from the fact that real transistors are not perfect switches, which go instantaneously from perfect-on to perfect-off the moment the gate-to-source voltage crosses some threshold (however slowly). In fact, such ideal switches can be shown thermodynamically impossible, because they can be used to build lossless diodes [21].

No, as we all know from looking at I-V curves, real transistors turn off only gradually. This is especially so when the gate voltage itself is changing only gradually over time, which is the case when the gate voltage is being controlled adiabatically, as will be the case for most of the transistors in any mostly-adiabatic digital circuit. Because of this, during part of any on/off transition, the transistor will have an intermediate level of effective resistance—not the ~10 kΩ of a minimum sized saturated MOSFET, nor the many gigaohms or more of a low-

leakage, turned-off device, but some intermediate level, perhaps in the megaohms, at which the voltage drop across the device increases substantially, but the resistance is not yet so high as to bring the $P = V^2/R$ power dissipation back down towards zero. This can lead to a significant non-adiabatic dissipation that does not scale down very much as the overall frequency is decreased.

To validate this expectation, I wrote a simple numerical model of energy dissipation in a typical MOSFET transistor in a current process, using standard subthreshold conduction models, when the transistor is being turned off adiabatically while a dynamic node is being charged through it [22]. The dissipation was ~3000 kT even when all logic transitions were taking place so slowly that there was $<kT$ dissipation in transitions through fully turned-on transistors.

It is easy to fail to notice this effect, but it is important not to do so. For example, the logically reversible "adiabatic" logic style of de Vos [23] involves some transistors being turned off while they are simultaneously being used to turn off other transistors. This is bad, because it is (unavoidably) not truly adiabatic. So, there will be significant irreducible dissipation as a result. Essentially, de Vos's work assumes that CMOS transistors behave like ideal switches, and this practice was rightly criticized by Schlaffer and Nossek [21] on the basis that ideal switches are actually thermodynamically impossible. However, Schlaffer and Nossek went too far in their conclusions, and assumed that just because one particular reversible adiabatic logic style (de Vos's) was fatally flawed, this must therefore be true in general of all reversible logic. In fact this is fallacious; there are no flaws in the circuit-level adiabaticity of, for example, the logically reversible SCRL circuit style of Younis and Knight, if repaired as I described in my Ph.D. thesis [22]. I have designed a number of other truly adiabatic logic styles, including one summarized in section 4 below.

Why was a repair needed in SCRL? Essentially, because in its original version [24,25], it too unwittingly broke the same transistor rule (2). SCRL's designers knew better than to *intentionally* schedule a transistor's turn-off transition concurrently with a transfer of charge through the device. However, what was neglected was that a transistor might turn itself off if the voltage on the source terminal becomes too close to the gate voltage. This actually occurs in SCRL in the series networks in NAND pull-downs (or NOR pull-ups) when the inner FET is on while the outer one remains off. As the internal node between the two devices, which is the source node of the inner FET, passes the $V_G - V_T$ cutoff level, the transistor gradually turns itself off, thereby violating the rule (2), since the device is being turned off while there is still a current passing through it.

Fortunately, in the case of SCRL, it is easy to modify the circuit style in a way that avoids this phenomenon [22], by placing a third transistor of complementary type in parallel with the inner FET, with its gate tied to the gate of the outer FET. This additional FET will remain strongly on during the entire output transition in the case described, and so the current charging the source node will be shunted away from the FET which is turning off, keeping the voltage drop across it small, and preventing the unwanted non-adiabatic dissipation as the source node is being charged. More generally, use of dual-rail logic can enable the designer to provide full-swing charging pathways everywhere and thereby prevent the rule from being violated in the pull-up/pull-down networks of inverting Boolean gates of any desired complexity.

However, this case illustrates that the rule (2) can be easily broken by accident, even when one is aware of it. Therefore, we wished to bring this rule to the design community's attention, and emphasize that adiabatic circuits must be carefully checked to ensure that the rule is never broken, through either gate-activated or source-activated switch-off of transistors.

An implication of the rule is that while a transistor is being turned off, source and drain voltages must either be held constant, or they must be varying along identical trajectories, and with both nodes being driven through charging pathways that include some routes that do not pass through any transistors that are undergoing on/off transitions.

3.3. Use Mostly-Reversible Logic

Many supposedly adiabatic logic styles are not logically reversible, either explicitly or implicitly, and many authors do not even understand the reasons why logical reversibility is absolutely necessary for physical reversibility. However, the relationship can be easily proven directly from the most fundamental and well-established laws of modern physics.

The field of fundamental-particle physics has convincingly demonstrated, through myriads of precise experiments, that all the observable phenomena of our world obey the Standard Model of particle physics, specifically, the Yang-Mills quantum field theory, to a very high degree of accuracy. No departures from the theory have been found in the decades since it was established. It is the *de facto* standard model of fundamental physics today.

As a quantum theory, the model includes at its very core the postulate that the quantum state of any closed system (such as, the entire universe) evolves over time according to a Hamiltonian dynamics that is embodied in the Schrödinger wave equation. In any Hamiltonian dynamical system, the state variables (in quantum theory, the wavefunction amplitudes) evolve over time according to a differential equation that is first-order in time, and this mathematically implies the reversibility (bijectivity) of the time-evolution. No departures from this micro-reversibility have ever been observed; apparently irreversible phenomena such as "wavefunction collapse" are explained away in the pure quantum theory as expected emergent phenomena that are predicted by an entirely reversible underlying theory [26]. And, macro-scale irreversibility reflects simply the modeler's inability to keep exact track of the reversible evolution of a microstate interacting with an unknown environment.

The reversibility of consensus quantum physics is incontrovertible, and the connection between adiabatics and reversible logic follows immediately from it. I first described this connection in my Ph.D. thesis [22], in a figure reproduced below. Since time-evolution in quantum mechanics proceeds via a unitary, invertible time-evolution, two initially completely distinguishable states can never evolve to become the same or not completely distinguishable (at the micro-level) at some later time. Say a mechanism purports to be able to operate on a bit which can take either of two distinct values, and transform that bit to have a single value unconditionally. This operation cannot actually reduce the number of distinct physical states. Since the states are no longer distinguished by the logical content of the bit, they *must* become distinguished by other physical variables, for example by the thermal vibrational state of atoms in the nearby environment. The total *physical* information content of the system is *unchanged*. If the information about the given bit's state was known (correlated in a known way with other known information) before the erasure operation, but becomes unknown afterwards, then we have had a transformation of known information to entropy, by the very definition of entropy: unknown information. Therefore, total entropy has increased. The amount of increase is the logarithm of the factor by which the size of the possible state space has expanded, in this case by 2. The logarithm base e (in this case ln 2) gives the entropy in units of Boltzmann's constant k, in this case k ln 2. Since the entropy, once produced, cannot be destroyed (by the second law of thermodynamics), it can only be coped with by exporting it into the external environment, into some thermal reservoir at temperature T. By the very definition of temperature, $T = (\partial E/\partial S)_V$, meaning that (for constant-volume systems), an increase in entropy of ∂S requires an increase in energy (in the form of heat) of ∂E, in this case kT ln 2.

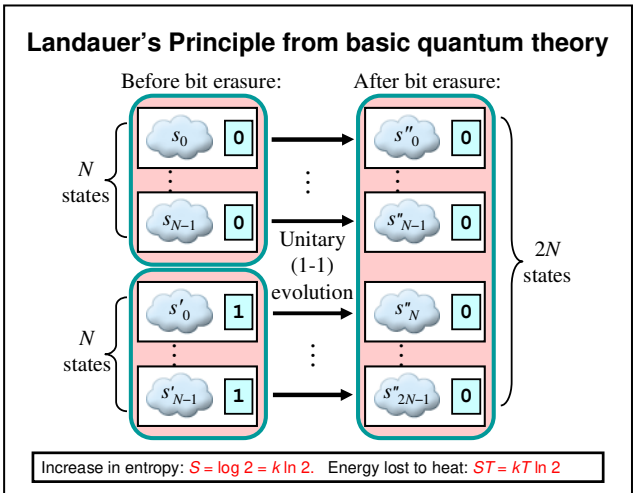

Figure 1. Rolf Landauer's principle that bit erasure causes energy dissipation can be proven trivially from basic quantum theory. Before a logical bit is erased, it has 2 possible distinct states, and suppose there are N possible distinguishable states of the rest of the system of which it is a part. Therefore there are $2N$ states for the system as a whole. The microreversible, bijective nature of unitary quantum time-evolution guarantees that however the bit "erasure" operation is performed, after the erasure there must remain $2N$ distinct states. If, before the erasure, the state of the bit is known, that part of the information in the system is by definition known information, and therefore is not entropy. If, after the erasure, we retain no knowledge of which half of the state space we are in, this information is then unknown, and is by definition entropy. Therefore, entropy has increased by $S = 1$ bit, which is equal to k ln 2, and this entropy requires an energy expenditure of ST to store in a thermal reservoir at temperature T, by the very definition of temperature: $T = \partial E/\partial S$.

Even worse, suppose we encode our logical bits by voltage levels. The difference between 1 and 0 voltage levels is *physically* encoded by a change in the Fermionic occupation numbers (1 or 0) of some large number (today) of electron states in the circuit node. Each of these states contains exactly 1 bit of physical information, which, because it is correlated with the logical bit value, is all known information before the erasure. *All of these known physical bits become entropy when the logical bit is erased*, unless special measures are taken to uncompute some of the bits based on the redundant values of the other ones (*e.g.*, Younis describes one such method in §4.8 of his thesis [25]). But even in the limit of the most aggressive possible partially-adiabatic mechanism, at least *one* bit of physical information *must* become entropy if the logical state is lost. Therefore, *complete physical reversibility requires complete logical reversibility* for manipulation of known bits. (However, if some "logical" bits are already unknown, for example, if they are produced by measuring bits of entropy in the environment, then these logical bits are *already* entropy, and a non-logically-reversible mixture of those bits with other unknown bits need not create any new entropy ([20], p. 22]).)

So although a circuit technique can be semi-adiabatic without being logically reversible, it cannot be *truly* adiabatic, in the sense of scaling to ever-lower levels of dissipation with further engineering refinements, or as

speed is decreased. Therefore, non-logically-reversible "adiabatic" circuit techniques are, at best, a temporary stopgap measure; as requirements for low power become ever more-stringent, an increasing degree of logical reversibility in the circuit style is required.

So, any claims that logically irreversible erasure of known information can be performed truly adiabatically (or with less than kT ln 2 energy dissipation per bit erased) can be summarily discarded without needing further consideration (unless backed up by rigorous, replicable experiment) since such claims directly contradict the entire, extremely thoroughly-tested logical framework underlying all of modern physics.

3.4. Don't Over-Constrain the Design

For designing large-scale circuits approaching true adiabaticity (which therefore must also approach total logical reversibility), a number of different logic schemes have been proposed. However, many of the existing schemes are unsatisfactory in the sense that the costs (spacetime, energy) to perform some computations in those schemes is asymptotically greater, by unboundedly-large factors, than in alternative schemes.

One example has to do with adiabatic logic families that don't permit reversibility across multiple stages of sequential, pipelined logic, for example Hall's early technique [27]. Fixing this particular problem was the goal of the SCRL project of Younis & Knight [24,25]. They succeeding in providing a pipelined, reversible, fully-adiabatic logic (with only a minor bug that is easily fixed as I described above in section 3.2). But, it turns out that SCRL itself is still over-constrained, for the simple reason that it forces every logic node to undergo a transition on every clock cycle. Even memories which are just statically storing data, if implemented in SCRL, perform active logic transitions for each stored bit on each cycle [28]. Because of this, SCRL is actually asymptotically *less* cost-efficient than could be achieved in an alternative adiabatic scheme that allowed stored bits to remain quiescent. I have developed such a scheme (essentially, it is just a straightforward generalization of SCRL) and will be publishing the details of it in future papers. In the next section I summarize its key characteristics.

4. GCAL: General CMOS Adiabatic Logic

GCAL is a CMOS-based adiabatic logic style that has been developed at UF that has the following features:

- Enables designing circuits having asymptotically optimal cost-efficiency, for any combination of time, space, spacetime, and energy costs.
- Supports circuits having unboundedly high reversibility.
- Supports both two-level and three-level truly adiabatic CMOS logic gates using as few transistors as possible.
- Requires only 4 (for two-level logic only) or 12 (for two- and three-level logic) distinct externally supplied power/clock signals to drive all circuits.
- Supports both fully-pipelined and non-fully-pipelined (retractile) logic styles.
- Allows on-chip generation of multiple independent arbitrary patterns of adiabatic timing signals.
- Supports quiescent (not constantly switching) but still fully-adiabatic dynamic and static latches, registers, and RAM cells.

To support adiabatic design in this logic style, we are developing the following design tools:

- An hierarchical hardware description language for describing partially- to fully-adiabatic circuits, together with the constraints on their I/O signal timing required to achieve the intended degree of reversibility.
- An adiabaticity checker that can automatically verify whether a complete design satisfies all timing constraints, optimize the design within the constraints, and pinpoint any unanticipated departures from true adiabaticity.
- An adiabatic logic synthesis tool that can automatically generate adiabatic circuits of any desired degree of reversibility given HDL-level descriptions of ordinary irreversible logic (including legacy designs), based on Bennett's 1989 reversiblization algorithm.

Unfortunately, we cannot at this time provide all the technical details of GCAL effort, as UF will be applying for a patent on many of the novel ideas contained in the GCAL approach. Future papers will release the details after provisional patents are secured.

However, when this effort is completed, asymptotically adiabatic logic design to any desired degree of adiabaticity (up to the device technology's limit) will become much more accessible to the average hardware designer. For the first time, designers will be able to straightforwardly create logic designs that can scale to the maximum cost-efficiency physically possible in a given device technology. This capability will become essential for digital design over the next century, in order to maximize the range of digital applications while remaining maximally competitive within the major application areas driving digital technology today, such as general-purpose computing, graphics computing, and digital signal processing.

For example, at the 2003 Nanotechnology Conference and Trade show [19], I reported the results of a recent numerical analysis (detailed in [29]) showing that under reasonable technology assumptions, adiabatics (with a high degree of reversible logic) will be capable of achieving 1,000-100,000 times the cost-efficiency of irreversible logic in desktop- or laptop-scale computing systems by around the 2050's.

5. Conclusion

Most adiabatic logic design families proposed to date have been relatively short-sighted, in view of the requirements for cost-efficient computing in the long run (meaning, after a few decades), which will require closely approaching the real physical limits of computing, in particular the need for near-total physical (and therefore logical) reversibility of the computing mechanism. These requirements preclude the use of diodes (except increas-

ingly sparsely) in adiabatic logic circuit designs. Attention must be paid to the current-carrying state of transistors (or other current switches) at times when the device is being switched *off*, as well as the voltage state when the devices are switched on. And, the logic family must be sufficiently flexible that it doesn't preclude expressing the most asymptotically efficient possible hardware algorithms. Meeting all of these requirements will become an absolute economic necessity in the coming century; this prediction is no less certain than is the conjunction of the bedrock of modern physics, and the juggernaut march of continued technological progress, since we can confidently prove it using nothing apart from these hypotheses. However, none of the adiabatic logic families that have been proposed so far by other researchers (as far as I've seen) actually manage to meet all of these requirements.

In the Reversible and Quantum Computing group at the University of Florida, in addition to identifying the flaws mentioned above in the existing adiabatic logic schemes, and characterizing the requirements for a satisfactory alternative, we have designed a new adiabatic logic family that completely meets all of these requirements, and moreover, we are in the process of developing an extensive related suite of HDL-based design tools, which will bring this logic family and the revolutionary new design capabilities associated with it well within the reach of the average logic designer. We expect that these developments will help to make truly-adiabatic reversible computing the rule rather than the exception, over the course of the coming century.

1 Sadi Carnot, *Reflections on the Motive Power of Heat*, Bachelier, Paris, 1825, http://www.history.rochester.edu/steam/carnot/1943/.

2 J. G. Koller and W. C. Athas, "Adiabatic switching, low energy computing, and the physics of storing and erasing information," in *PhysComp '92: Proceedings of the Workshop on Physics and Computation, October 2-4, 1992, Dallas Texas*, IEEE Computer Society Press, 1992, pp. 267-270.

3 Edward F. Fredkin and Tommasso Toffoli, "Design principles for achieving high-performance submicron digital technologies," proposal to DARPA, Nov. 1978.

4 Carver Mead and Lynn Conway, "Physics of Computational Systems," chapter 9 of *Introduction to VLSI Systems*, Addison-Wesley, 1980, pp. 333-371.

5 Richard P. Feynman, *The Feynman Lectures on Computation*, Perseus Books, 1996 (posthumous collection of transcripts of lectures from a course taught at Caltech from 1983-1986).

6 Charles L. Seitz, Alexander H. Frey, Sven Mattisson, Steve D. Rabin, Don A. Speck, and Jan L. A. van de Snepscheut, "Hot-clock nMOS," in Henry Fuchs, ed., *1985 Chapel Hill Conference on Very Large Scale Integration*, pp. 1-17, Computer Science Press, 1985.

7 Boyd G. Watkins, "A low-power multiphase circuit technique," *IEEE Journal of Solid-State Circuits*, pp. 213-220, Dec. 1967.

8 Michael P. Frank, "Physical Limits of Computing", *Computing in Science and Engineering*, 4(3):16-25, May/June 2002, http://www.cise.ufl.edu/research/revcomp/physlim/plpaper.html.

9 Warren D. Smith, "Fundamental physical limits on computation," Technical Report, NECI, May 1995, http://www.neci.nj.com/homepages/wds/fundphys.ps.

10 E. F. Fredkin and T. Toffoli, "Conservative logic," *International Journal of Theoretical Physics*, 21(3/4):219-253, 1982.

11 W. H. Zurek, "Reversibility and stability of information processing systems," *Physical Review Letters* **53**(4):391-394, July 23, 1984.

12 Michael A. Nielsen and Isaac L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.

13 Michael P. Frank, "Scaling of Energy Efficiency with Decoherence Rate in Closed, Self-Timed Reversible Computing", UF Reversible Computing Project Memo #M18, Nov. 2002, http://www.cise.ufl.edu/research/revcomp/memos/Memo18-Timing.doc.

14 Norman H. Margolus and Lev B. Levitin, "The maximum speed of dynamical evolution," *Physica D* **120**(1/2), 1998, http://arXiv.org/abs/quant-ph/?9710043.

15 See, for example, the web pages and publications liked from http://pavel.physics.sunysb.edu/RSFQ/.

16 Y. Naveh and K. Likharev, "Modeling of 10-nm-scale ballistic MOSFET's", IEEE Electron. Device Lett., vol. 21, pp. 242-244, May 2000, http://rsfq1.physics.sunysb.edu/~likharev/nano/FET_EDL.ps.

17 *E.g.*, see publications by Michael Roukes' group at CalTech, http://www.its.caltech.edu/%7Enano/publicat.html.

18 Michael P. Frank, *Principles of Computer Architecture*, Fall 2002, lectures #4-5, slide #56, http://www.cise.ufl.edu/class/cda5155fa02/lecs/Lec04-05.ppt.

19 Michael P. Frank, "Nanocomputer Systems Engineering," *NanoTech 2003: Technical Proceedings of the 2003 Nanotechnology Conference and Trade Show*, **2**:182-185, held Feb. 23-27, 2003, San Francisco, CA, http://www.cise.ufl.edu/research/revcomp/theory/NanoTech2003/Frank-NanoTech-2003.doc, .ps.

20 Harvey S. Leff and Andrew F. Rex, eds., *Maxwell's Demon 2: Entropy, Classical and Quantum Information, Computing*, Institute of Physics Publishing, 2003.

21 A. Schlaffer and J. A. Nossek, "Is there a connection between adiabatic switching and reversible computing?", Institute for Network Theory and Circuit Design, Munich University of Technology, http://citeseer.nj.nec.com/schlaffer97is.html.

22 Michael P. Frank, *Reversibility for Efficient Computing*, manuscript based on Ph.D. thesis, Dec. 1999, http://www.cise.ufl.edu/~mpf/manuscript, §7.6.4, pp. 197-199.

23 Alexis de Vos, "A 12-transistor c-MOS building-block for reversible computers," *Int. J. Electronics*, 1995, **79**(2):171-182.

24 S. G. Younis and T. F. Knight, Jr., "Asymptotically zero energy split-level charge recovery logic," in *International Workshop on Low Power Design*, pp. 177-182, 1994, http://www.cise.ufl.edu/~mpf/scrl94.pdf.

25 S. G. Younis, *Asymptotically Zero Energy Computing Using Split-Level Charge Recovery Logic*, Ph.D. thesis, MIT EECS Dept., 1994, http://www.cise.ufl.edu/~mpf/younis-phd.ps.

26 Wojciech H. Zurek, "Decoherence, einselection, and the quantum origins of the classical," preprint http://arxiv.org/abs/quant-ph/0105127, July, 2002.

27 J. Storrs Hall, "An electroid switching model for reversible computer architectures," in *PhysComp '92* (*ibid.* [2]), pp. 237-247.

28 Carlin Vieri, M. Josephine Ammer, Amory Wakefield, Lars "Johnny" Svensson, William Athas, and Tom Knight, "Designing Reversible Memory," *Unconventional Models of Computation*, Springer, 1998, pp. 386-405.

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