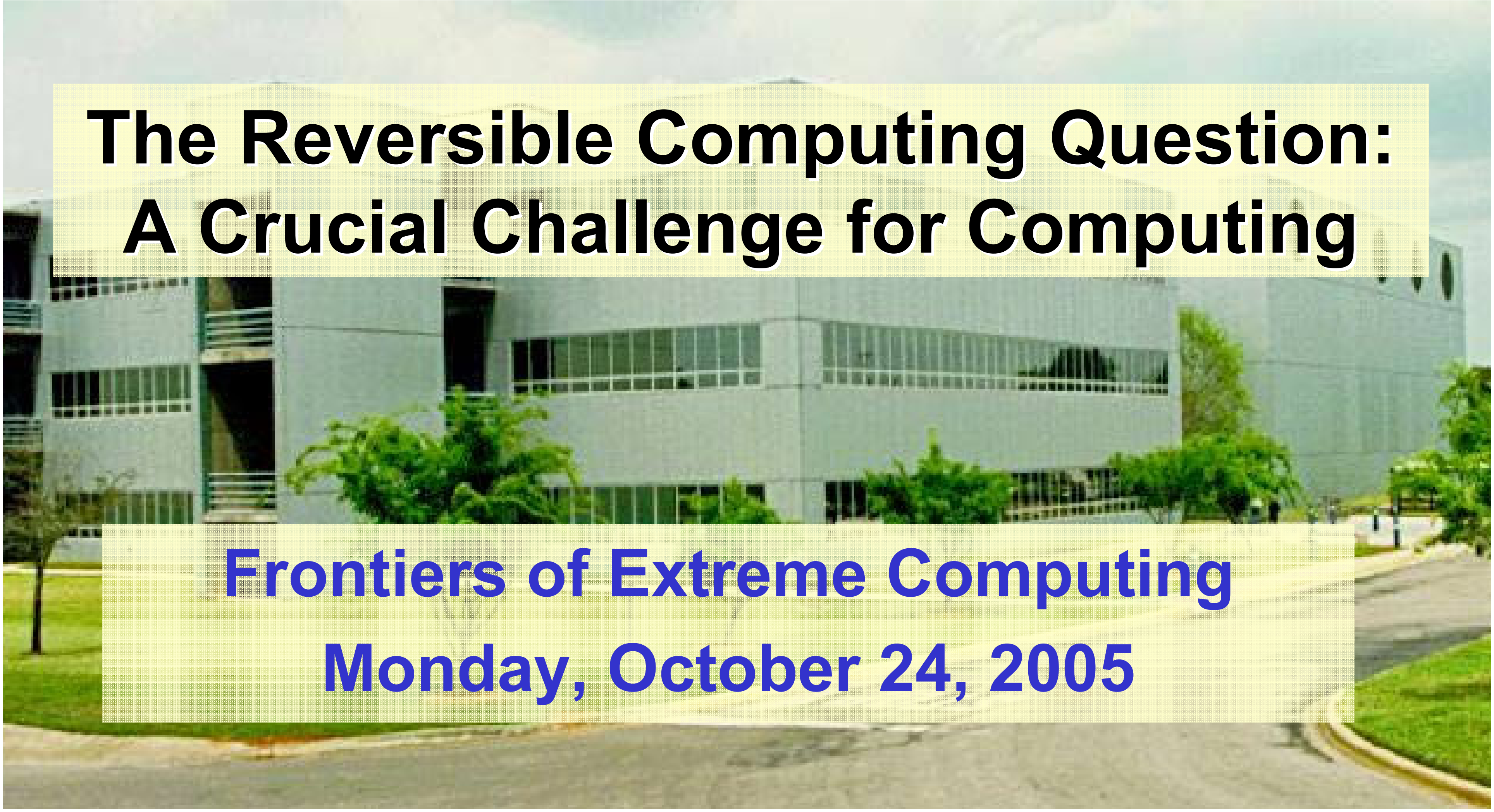
Florida Agricultural and Mechanical University

Michael P. Frank

FAMU-FSU College of Engineering

http://www.eng.fsu.edu/~mpf

Florida State University



The Reversible Computing Question: A Crucial Challenge for Computing

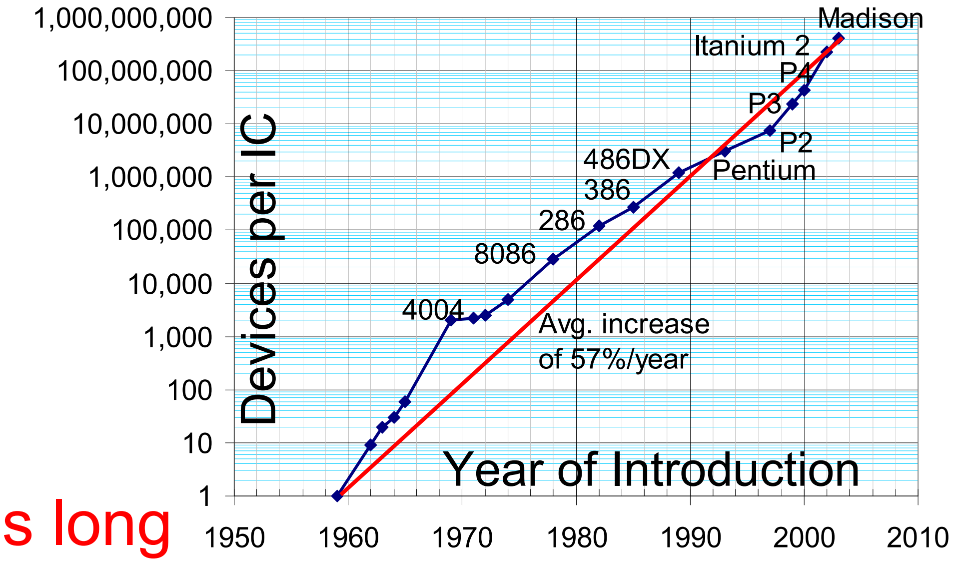
Frontiers of Extreme Computing

Monday, October 24, 2005

Outline of Talk

- Computational *energy efficiency* (η_{ec}) as the ultimate performance limiter in practical computer systems…
  - – Limits on the η_{ec} attainable in conventional machines
- Reversible computing (RC) as the *only* way out in the long term, after the next decade or two…
  - – Review of some basic concepts of reversible logic
- The "Reversible Computing Question:"
  - – Can we ever really build *competitive* RC machines?
- Why practical Reversible Computing is difficult…
  - – and why it might nevertheless be possible.
- A Call to Action!

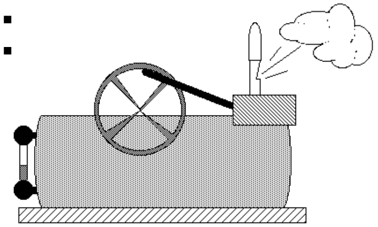
Moore’s Law and Performance



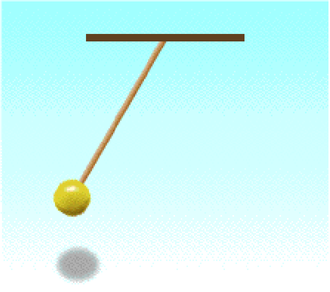
- Gordon Moore, 1975:
  - Devices per IC can be doubled every 18 months
    - Borne out by history, so far…
- Some associated trends:
  - Every 3 years: Devices ½ as long
  - Every 1.5 years: ~½ as much stored energy per bit!
    - This has enabled us to throw away bits (and their energies) 2× more frequently every 1.5 years, at reasonable power levels!
      - And thereby double processor performance 2× every 1.5 years!
- Increased *<u>energy efficiency</u>* of computation is a <u>prerequisite</u> for improved raw performance!
  - Given realistic fixed constraints on total power consumption.

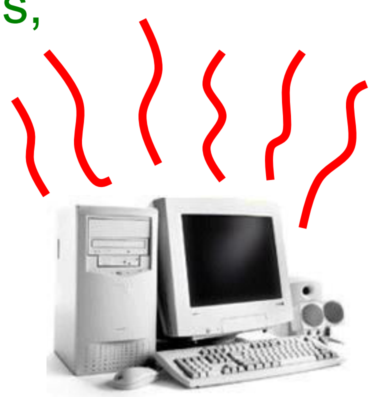
Efficiency in General, and Energy Efficiency

- The *efficiency* η of any process is: $\eta = P/C$
  - Where P = Amount of some valued product produced
  - and C = Amount of some costly resources consumed
- In *energy efficiency* η_e, the cost C measures energy.
- We can talk about the energy efficiency of:



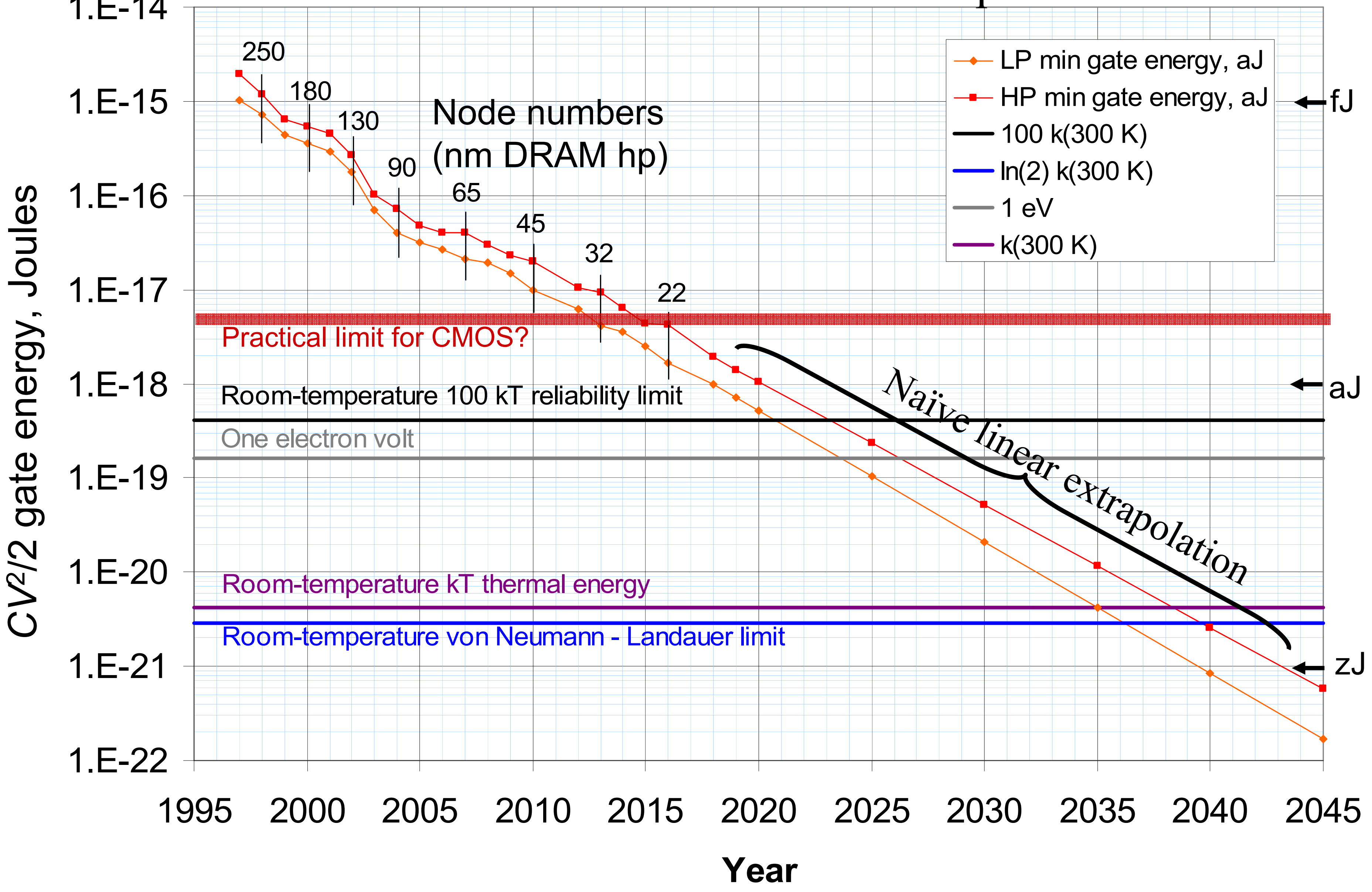
  - A *heat engine*: $\eta_{he} = W/Q$, where:
    - W = work energy output, Q = heat energy input
  - An *energy recovering process* : $\eta_{er} = E_{end}/E_{start}$, where:
    - E_{end} = available energy at end of process,
    - E_{start} = energy input at start of process
  - A *computer*: $\eta_{ec} = N_{ops}/E_{cons}$, where:
    - N_{ops} = # useful operations performed
    - E_{cons} = free-energy consumed





Trend of Minimum Transistor Switching Energy

Based on ITRS ’97-03 roadmaps



Some Lower Bounds on Energy Dissipation

- In today's 90 nm VLSI technology, for minimal operations (*e.g.*, conventional switching of a minimum-sized transistor):
  - $E_{\mathrm{diss,op}}$ is on the order of $1\ \mathrm{fJ}$ (femtojoule) ➔ $\eta_{\mathrm{ec}} \lesssim 10^{15}$ ops/sec/watt.
    - Will be a bit better in coming technologies (65 nm, maybe 45 nm)
- But, conventional digital technologies are subject to several lower bounds on their energy dissipation $E_{\mathrm{diss,op}}$ for digital transitions (logic / storage / communication operations),
  - And thus, corresponding upper bounds on their energy efficiency.
- Some of the known bounds include:
  - Leakage-based limit for high-performance field-effect transistors:
    - Maybe roughly $\sim 5\ \mathrm{aJ}$ (attojoules) ➔ $\eta_{\mathrm{ec}} \lesssim 2\times 10^{17}$ operations/sec./watt
  - Reliability-based limit for all non-energy-recovering technologies:
    - On the order of $1\ \mathrm{eV}$ (electron-volt) ➔ $\eta_{\mathrm{ec}} \lesssim 6\times 10^{18}$ ops./sec/watt
  - von Neumann-Landauer (VNL) bound for all irreversible technologies:
    - Exactly $kT \ln 2 \approx 18\ \mathrm{meV}$ (per bit erasure) ➔ $\eta_{\mathrm{ec}} \lesssim 3.5\times 10^{20}$ ops/sec/watt
      - For systems whose waste heat ultimately winds up in Earth's atmosphere,
        - » *i.e.*, at temperature $T \approx T_{\mathrm{room}} = 300\ \mathrm{K}$.

Reliability Bound on Logic Signal Energies

- Let E_{sig} denote the *logic signal energy*,
  - The energy *actively involved* (transferred, manipulated) in the process of storing, transmitting, or transforming a bit's worth of digital information.
    - But note that "involved" does not necessarily mean "dissipated!"
- As a result of fundamental thermodynamic considerations, it is required that $E_{\mathrm{sig}} \lesssim k_B T_{\mathrm{sig}} \ln r$ (with quantum corrections that are small for large r)
  - Where k_B is Boltzmann's constant, 1.38×10^{-12} J/K;
  - and T_{sig} is the temperature in the degrees of freedom carrying the signal;
  - and r is the *reliability factor*, *i.e.*, the improbability of error, $1/p_{\mathrm{err}}$.
- In non-energy-recovering logic technologies (totally dominant today)
  - Basically all of the signal energy is dissipated to heat on each operation.
    - And often additional energy (*e.g.*, short-circuit power) as well.
- In this case, minimum sustainable dissipation is $E_{\mathrm{diss,op}} \gtrsim k_B T_{\mathrm{env}} \ln r$,
  - Where T_{env} is now the temperature of the waste-heat reservoir (environment)
    - Averages around 300 K (room temperature) in Earth's atmosphere
- For a decent r of *e.g.* 2×10^{17}, this minimum is on the order $\sim 40\ kT \approx 1$ eV.
  - Therefore, if we want energy efficiency $\eta_{\mathrm{ec}} > \sim 1$ op/eV, we must *recover* some of the signal energy for later reuse.
    - Rather than dissipating it all to heat with each manipulation of the signal.

The von Neumann-Landauer (VNL) Principle

- First alluded to by John von Neumann in 1949.
  - Developed explicitly by Rolf Landauer of IBM in 1961.
- The principle is a rigorous theorem of physics!
  - It follows from the reversibility of fundamental dynamics.
- A correct statement of the principle is the following:
  - Any process that loses or *obliviously erases* 1 bit of known (correlated) information increases total entropy by at least

    $$\Delta S = 1\ \mathrm{bit} = k_\mathrm{B}\ln 2,$$

    and implies eventual system-level dissipation of at least

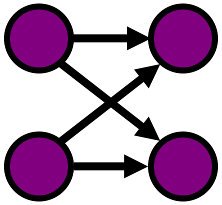
    $$E_\mathrm{diss} = \Delta S\cdot T_\mathrm{env} = k_\mathrm{B}T_\mathrm{env}\ln 2$$

    of free energy to the environment as waste heat.
    - where $k_\mathrm{B} = \mathrm{Log}\ e = 1.38\times10^{-23}\ \mathrm{J/K}$ is Boltzmann's constant
    - and T_env = temperature of the waste-heat reservoir (environment)
      - Not less than about room temperature ($300\ \mathrm{K}$) for earthbound computers. ➔ implies $E_\mathrm{diss} \geq 18\ \mathrm{meV}$.

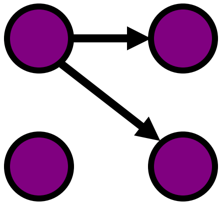
Types of Dynamical Systems

(We're using the physicist's, not the complexity theorist's meaning of "nondeterministic" below)

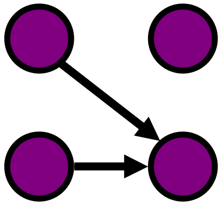
- Nondeterministic, irreversible



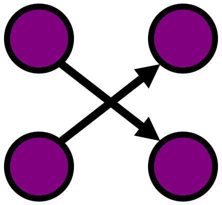
- Nondeterministic, reversible



- Deterministic, irreversible



- Deterministic, reversible



WE
ARE
HERE

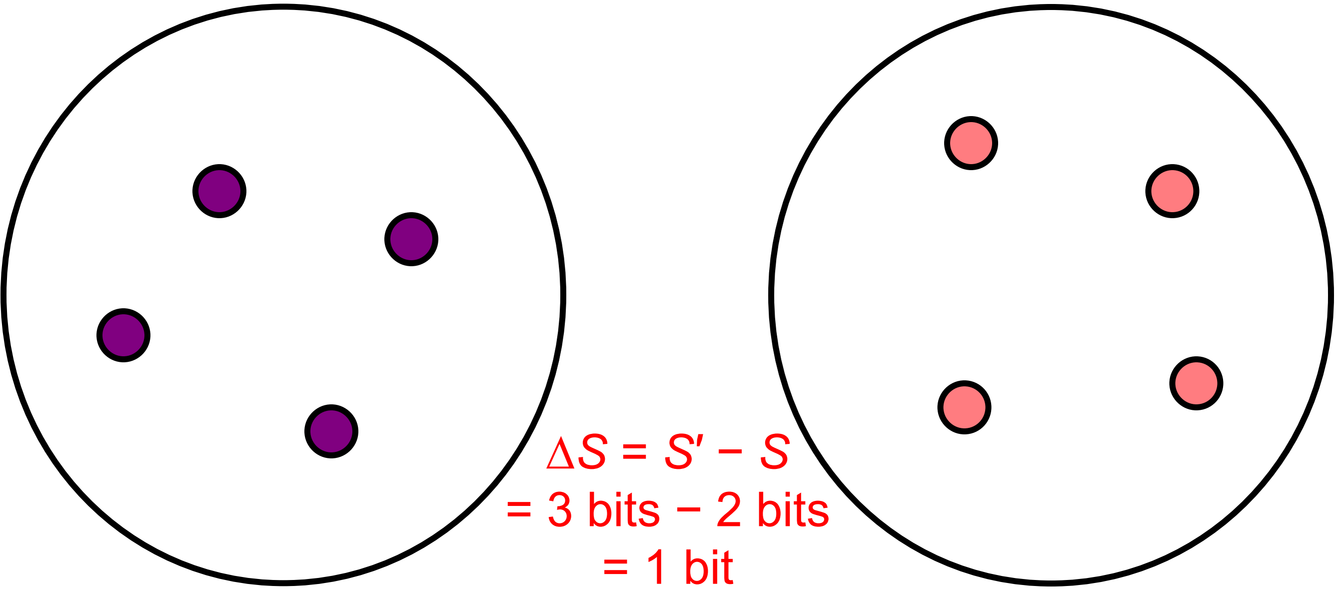
Physics is Reversible

- All the successful models of fundamental physics are expressible in the *Hamiltonian* formalism.
  - Including: Classical mechanics, quantum mechanics, special and general relativity, quantum field theories.
    - The latter two (GR & QFT) are backed up by enormous, overwhelming mountains of evidence confirming their predictions!
      - 11 decimal places of precision so far! And, no contradicting evidence.
- In Hamiltonian systems, the dynamical state $x(t)$ obeys a differential equation that's first-order in time,

  $\mathrm{d}x/\mathrm{d}t = g(x)$ (where g is some function)
  - This immediately implies determinism of the dynamics.
- And, since the time differential $\mathrm{d}t$ can be taken to be negative, the formalism also implies reversibility.
  - Thus, dynamical reversibility is one of the most firmly-established, inviolable facts of fundamental physics.

Illustration of VNL Principle

- Either digital state is initially encoded by any of *N* possible physical microstates
  - Illustrated as 4 in this simple example (the real number would usually be much larger)
  - Initial entropy S = log[#microstates] = log 4 = 2 bits.
- Reversibility of physics ensures "bit erasure" operation <u>can't possibly</u> merge two microstates, so it <u>must</u> double the possible microstates in the digital state!
  - Entropy S = log[#microstates] increases by log 2 = 1 bit = (log e)(ln 2) = k_B ln 2.
  - To prevent entropy from accumulating locally, it must be expelled into the environment.



Microstates representing logical "0"

Microstates representing logical "1"

Entropy $S' =$ log 8 = 3 bits

Entropy $S =$ log 4 = 2 bits

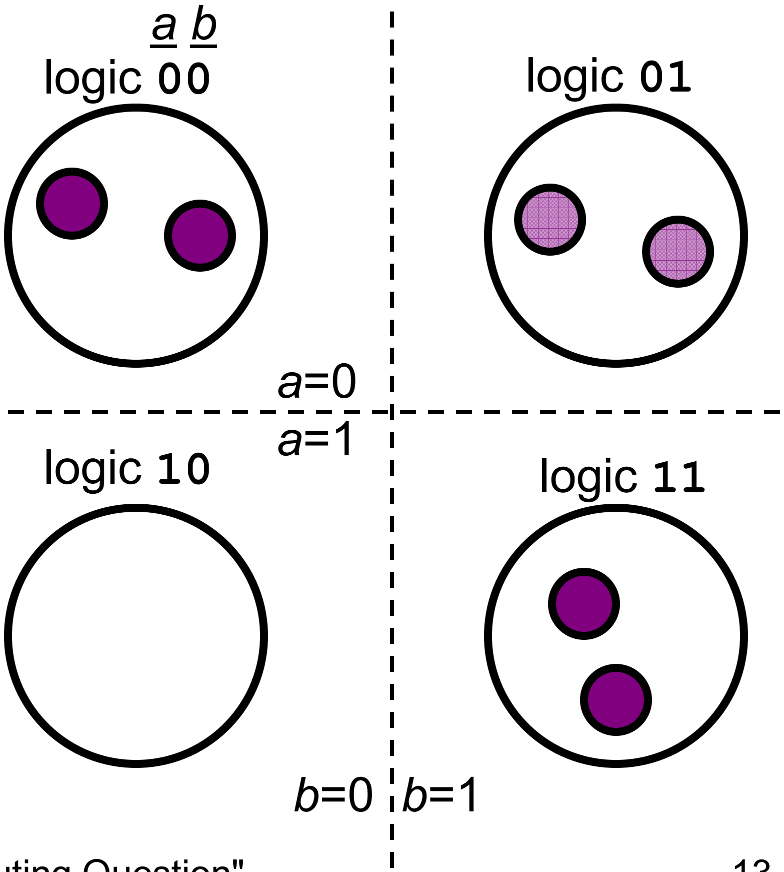
$\Delta S = S' - S$
= 3 bits − 2 bits
= 1 bit

Reversible Computing

- The basic idea is simply this:
  - *Don't discard information* when performing logic / storage / communication operations!
    - Instead, just reversibly (invertibly) transform it, in place!
- When reversible digital operations are implemented using well-designed energy-recovering circuitry,
  - This can result in local energy dissipation $E_{\mathrm{diss}} \ll E_{\mathrm{sig}}$,
    - this has already been empirically demonstrated by many groups.
  - and (in principle) total energy dissipation $E_{\mathrm{diss}} \ll kT \ln 2$.
    - This is easily shown in theory & simulations,
      - but we are not yet to the point of demonstrating such low levels of total dissipation empirically in a physical experiment.
    - Achieving this goal will require very careful design,
      - and verifying it requires very sensitive measurement equipment.

How Reversible Logic Avoids the von Neumann-Landauer Bound

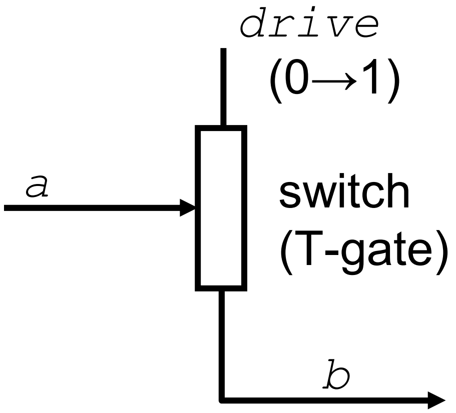
- We arrange our logical manipulations to never attempt to merge two distinct digital states,
  - but only to reversibly transform them from one state to another!
- *E.g.*, illustrated is a reversible operation "**`cCLR`**" (controlled clear)
  - Non-oblivious "erasure"
  - It and its inverse (**`cSET`**) enable arbitrary logic!

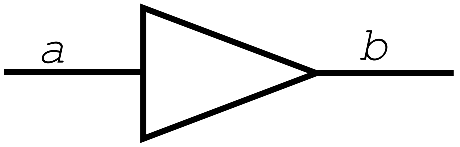


Notations for a Useful Primitive: Controlled-SET or `cSET`(a, b)

- **Function:** If $a=1$, then set $b:=1$.
  - *Conditionally* reversible, if the precondition $ab=0$ is met.
    - Note it's 1-to-1 on the subset of states used
      - Sufficient to avoid Landauer's principle!
- We can implement `cSET` in dual-rail CMOS with a pair of transmission gates
  - Each needs just 2 transistors,
    - plus one controlling "drive" signal
- This 2-bit semi-reversible operation with its inverse `cCLR` form a universal set for reversible (and irreversible) logic!
  - If we compose them in special ways.
    - And include latches for sequential logic.

a	**b**	**a′**	**b′**
0	0	0	0
0	1	0	1
1	0	1	1





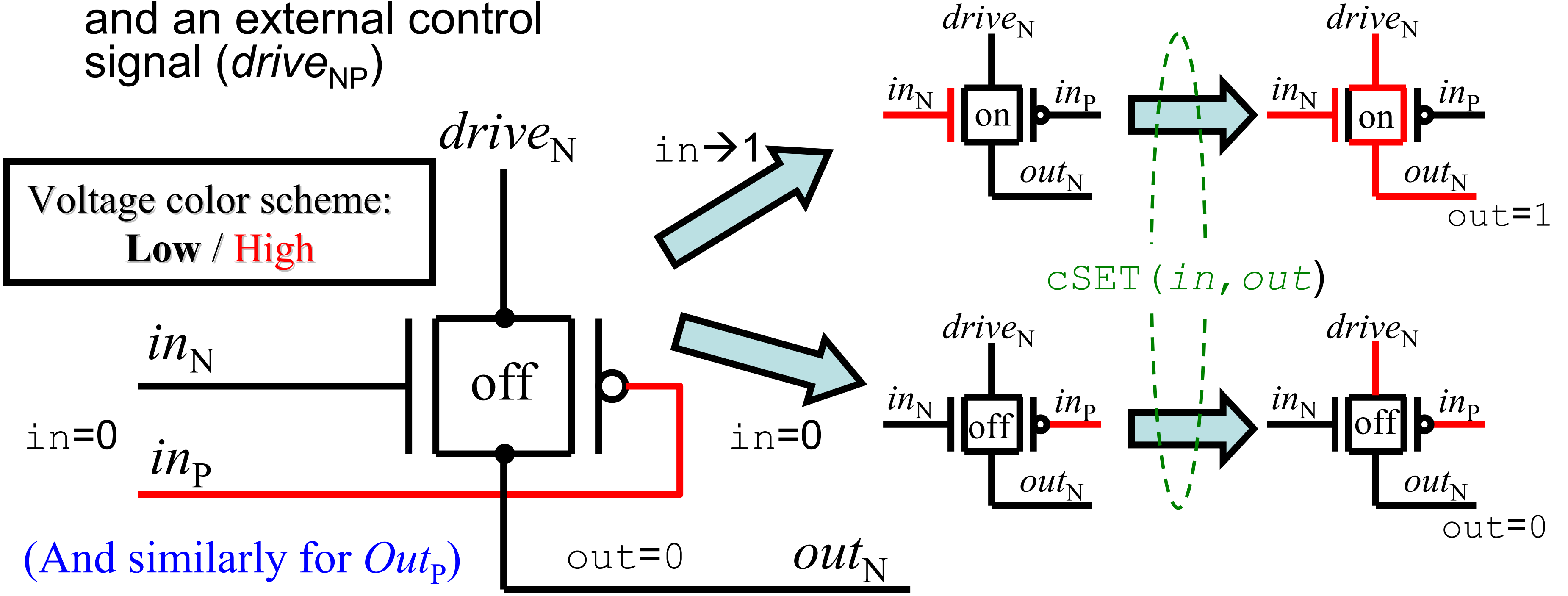
Example Implementation of a Reversible CMOS "`cSET/cCLR`" gate

- Formal semantics for a **controlled-SET** (**cSET**) operation:

```
cSET(in,out) ::=
  [~(in & out)]
  if in then out:0->1
  [~in | out]
```

Precondition: If *in*=1 we must have *out*=0 initially.
Action: If *in*=1, then take *out* from 0 to 1.
Postcondition: If *in*=1 then *out*=1 afterwards.

- The below implementation uses dual-rail signals, 2 T-gates, and an external control signal (*drive*$_{NP}$)

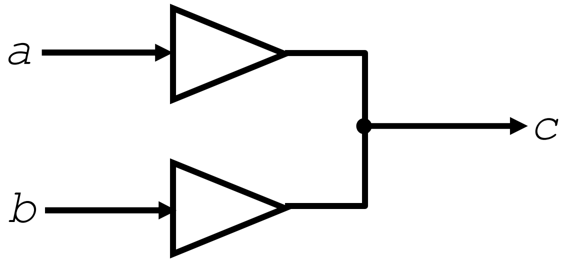


(And similarly for *Out*$_P$)

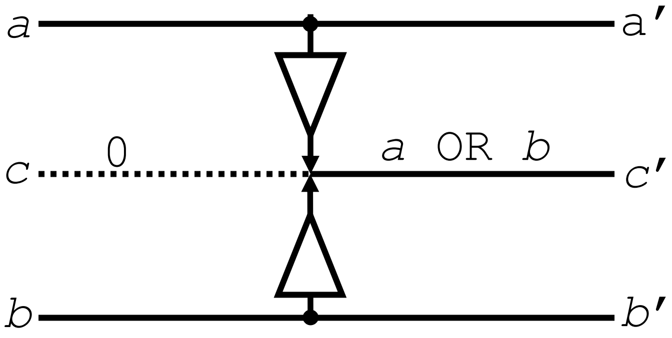
Reversible OR (`rOR`) from cSET

- **Semantics:** `rOR(a,b) ::= if a|b, c:=1.`
  - Set `c:=1`, on the condition that either `a` or `b` is `1`.
    - Reversible under precondition that initially `a|b → ~c`.
- Two parallel **`cSET`**s simultaneously driving a shared output bus implements the **`rOR`** operation!
  - This type of gate composition was not traditionally considered.
- Similarly one can do **`rAND`**, and reversible versions of all operations.
  - Logic synthesis with these is extremely straightforward…

Hardware diagram



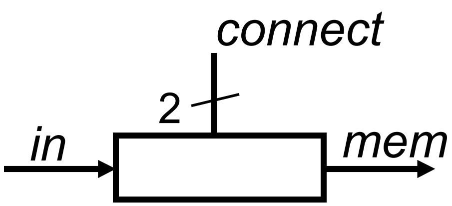
Spacetime diagram



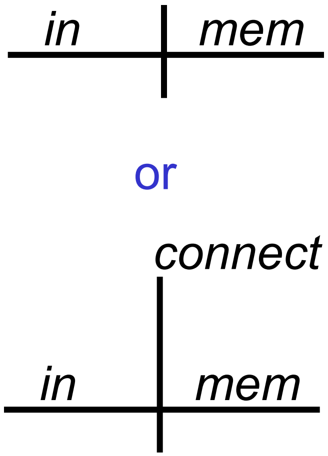
CMOS Gate Implementing `rLatch` / `rUnLatch`

- Symmetric Reversible Latch

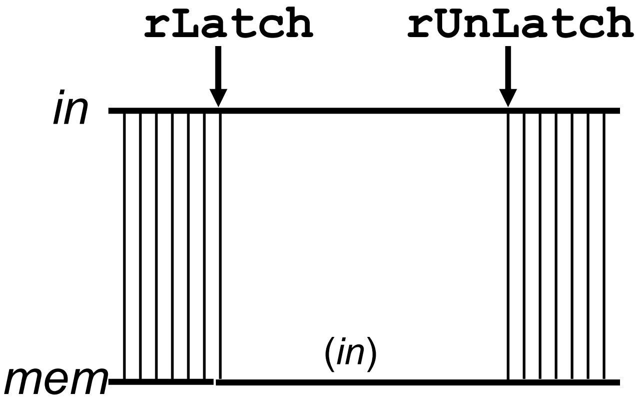
Implementation



Concise Icon



Spacetime Diagram

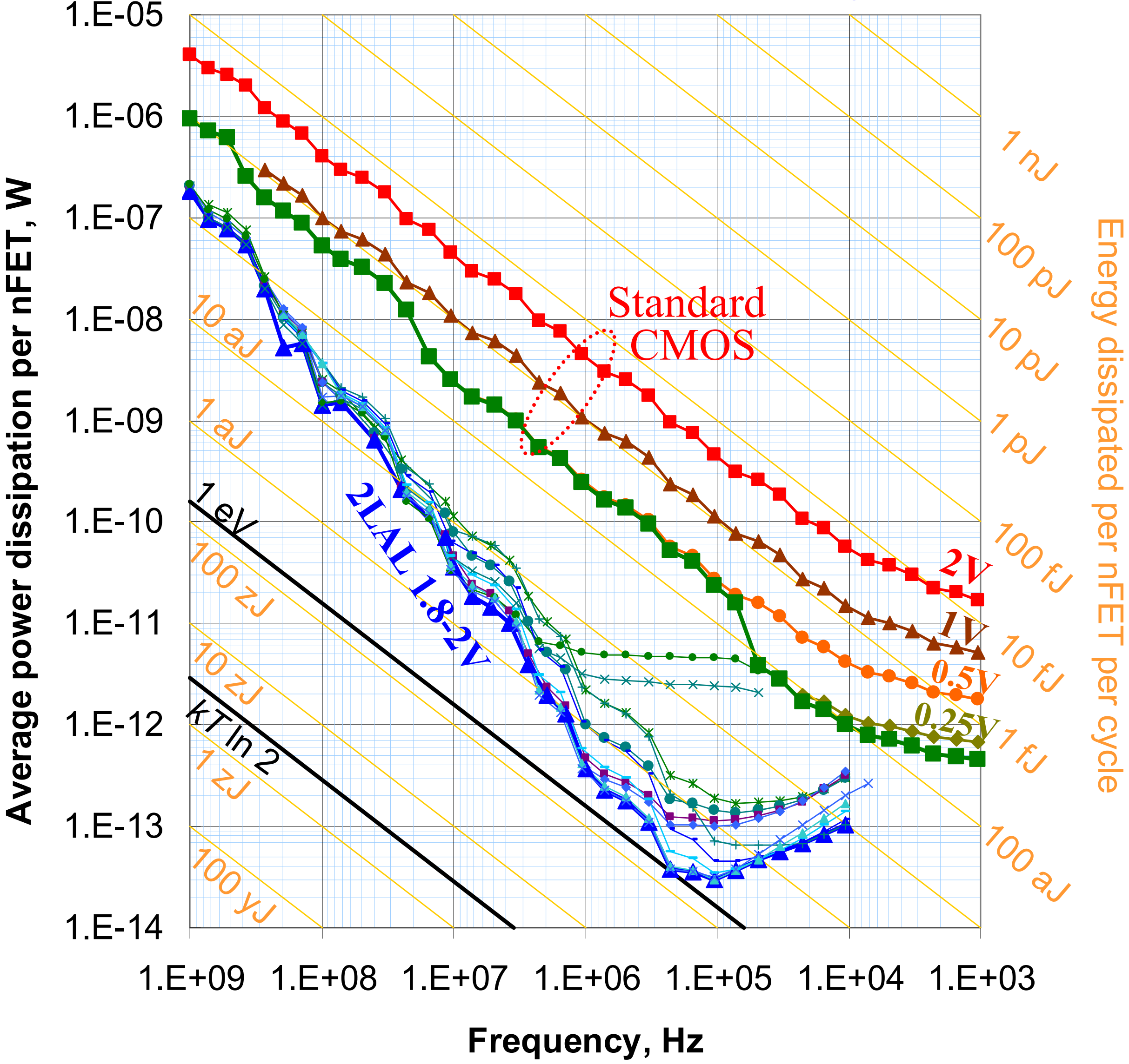


- The hardware is just a CMOS transmission gate again
  - This time controlled by a clock, with the data signal driving
- Concise, symmetric hardware icon – Just a short orthogonal line
- In spacetime diagram, thin strapping lines denote inter-node connection.

Cadence Simulation Results

Power vs. freq., TSMC 0.18, Std. CMOS vs. 2LAL

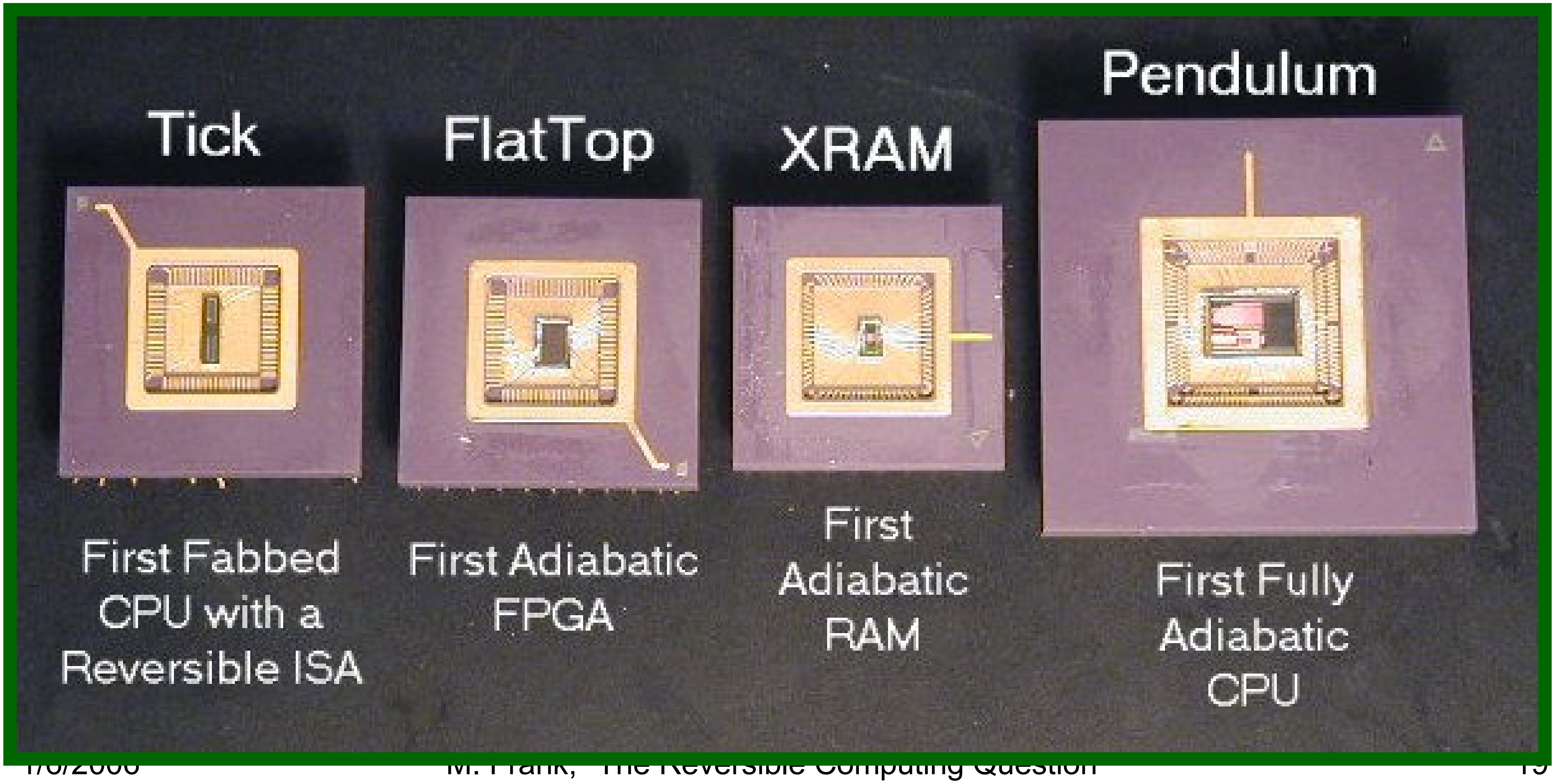
2LAL = Two-level adiabatic logic



- Graph shows power dissipation vs. frequency
  - in 8-stage shift register.
- At moderate frequencies (1 MHz),
  - Reversible uses < 1/100th the power of irreversible!
- At ultra-low power (1 pW/transistor)
  - Reversible is 100× faster than irreversible!
- Minimum energy dissipation < 1 eV!
  - 500× lower than best irreversible!
    - 500× higher computational energy efficiency!
- Energy transferred is still ~10 fJ (~100 keV)
  - So, energy recovery efficiency is 99.999%!
    - Not including losses in power supply, though

Reversible and/or Adiabatic VLSI Chips Designed @ MIT, 1996-1999

By Frank and other then-students in the MIT Reversible Computing group, under CS/AI lab members Tom Knight and Norm Margolus.



A Few Highlights Of Reversible Computing History

- Charles Bennett @ IBM, 1973-1989:
  - Reversible Turing machines & emulation algorithms
    - Can emulate irreversible machines on reversible architectures.
      - But, the emulation introduces some inefficiencies
  - Early chemical & Brownian-motion implementation concepts.
- Ed Fredkin and Tom Toffoli’s group @ MIT, late 1970’s/early 1980’s
  - Reversible logic gates and networks (space/time diagrams)
  - Ballistic mechanical and adiabatic circuit implementation proposals
- Paul Benioff, Richard Feynman, Norm Margolus, mid-1980s
  - Abstract quantum-mechanical models of “classical” reversible computers.
    - The field of quantum computing eventually emerged from this line of work
- Several groups @ Caltech, ISI, Amherst, Xerox, MIT, mid ‘80s-mid ’90s:
  - Concepts for & implementations of “adiabatic circuits” in VLSI technology
  - Small explosion of adiabatic circuit literature since then!
- Mid 1990s-today:
  - Better understanding of overheads, tradeoffs, asymptotic scaling
  - A few groups have begun development of post-CMOS implementations
    - Most notably, the Quantum-dot Cellular Automata group at Notre Dame

Reversibility and Reliability

- **A widespread claim:** “Future low-level digital devices will necessarily be highly unreliable.”
  - This comes from questionable lines of reasoning, such as:
    - Faster → more energy efficient ~~→~~ lower bit energies → high rate of bit errors from thermal noise
  - However, this scaling strategy doesn’t work, because:
    - High rate of thermal errors → high power dissipation from error correction → <u>less</u> energy efficient → ultimately <u>slower</u>!
- But in contrast, using reversible computing, in principle, we can achieve arbitrarily high energy efficiency <u>and</u> arbitrarily high reliability!
  - The key is to <u>keep bit energies reasonably high</u>!
    - Improve efficiency by <u>recovering</u> more and more of the bit energy…

Minimizing Energy Dissipation Due to Thermal Errors

- Let $p_{\mathrm{err}} = 1/r$ be the bit-error probability per operation.
  - Where r quantifies the "reliability level."
  - And $p_{\mathrm{ok}} = 1 - p_{\mathrm{err}}$ is the probability the bit is correct
- The minumum entropy increase ΔS per op due to error occurrence is given by the (binary) Shannon entropy of the bit-value after the operation:

$$H(p_{\mathrm{err}}) = p_{\mathrm{err}} \log p_{\mathrm{err}}^{-1} + p_{\mathrm{ok}} \log p_{\mathrm{ok}}^{-1}.$$

- For $r >> 1$ (*i.e.*, as $r \to \infty$), this increase approaches 0:

$$\Delta S = H(p_{\mathrm{err}}) \approx p_{\mathrm{err}} \log p_{\mathrm{err}}^{-1} = (\log r)/r \to 0$$

- Thus, the required energy dissipation per op also approaches 0:

$$E_{\mathrm{diss}} = T\Delta S \approx (kT \ln r)/r \to 0$$

- Could get the same result by assuming the signal energy $E_{\mathrm{sig}} = kT \ln r$ required for reliability level r is dissipated each time an error occurs:

$$E_{\mathrm{diss}} = p_{\mathrm{err}} E_{\mathrm{sig}} = p_{\mathrm{err}}(kT \ln r) = (kT \ln r)/r \to 0 \text{ as } r \to \infty.$$

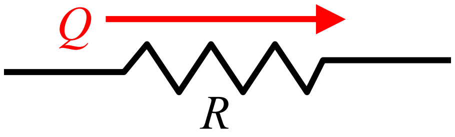
- Further, note that as $r \to \infty$, the required signal energy grows slowly…
  - Only logarithmically in the reliability, *i.e.*, $E_{\mathrm{sig}} = \Theta(\log r)$.

Some Device-Level Requirements for Reversible Computing

- A good reversible device technology should have:
  - Low manufacturing cost $¢_\mathrm{d}$ per device
    - Important for good overall (system-level) cost-efficiency
  - Low rate of static “standby” power dissipation P_sby due to energy leakage, thermally-induced errors, etc.
    - Required for energy-efficient storage especially (but also in logic)
  - Low *energy coefficient* $c_\mathrm{Et} = E_\mathrm{diss} \cdot t_\mathrm{tr}$ (energy dissipated per operation, times transition time) for adiabatic transitions.
    - Implies that we can achieve a high operating frequency (and thus good cost-performance) at a given level of energy efficiency.
  - High maximum available transition frequency f_max.
    - Especially important for those applications in which the latency of serial threads of computation dominates the total operating costs

Energy & Entropy Coefficients in Electronics

- For a transition involving the adiabatic transfer of an amount Q of charge along a path with resistance R:



  - The raw (local) energy coefficient is $c_{\mathrm{Et}} = E_{\mathrm{diss}}t = P_{\mathrm{diss}}t^2 = IVt^2 = I^2Rt^2 = Q^2R$.
    - Where V is the voltage drop along the path.
  - The entropy coefficient is $c_{\mathrm{St}} = Q^2R/T_{\mathrm{path}}$.
    - where T_{path} is the local thermodynamic temperature in the path.
  - The effective (global) energy coefficient is $c_{\mathrm{Et,eff}} = Q^2R(T_{\mathrm{env}}/T_{\mathrm{path}})$.
    - Note that we pay a penalty for low-T operation!

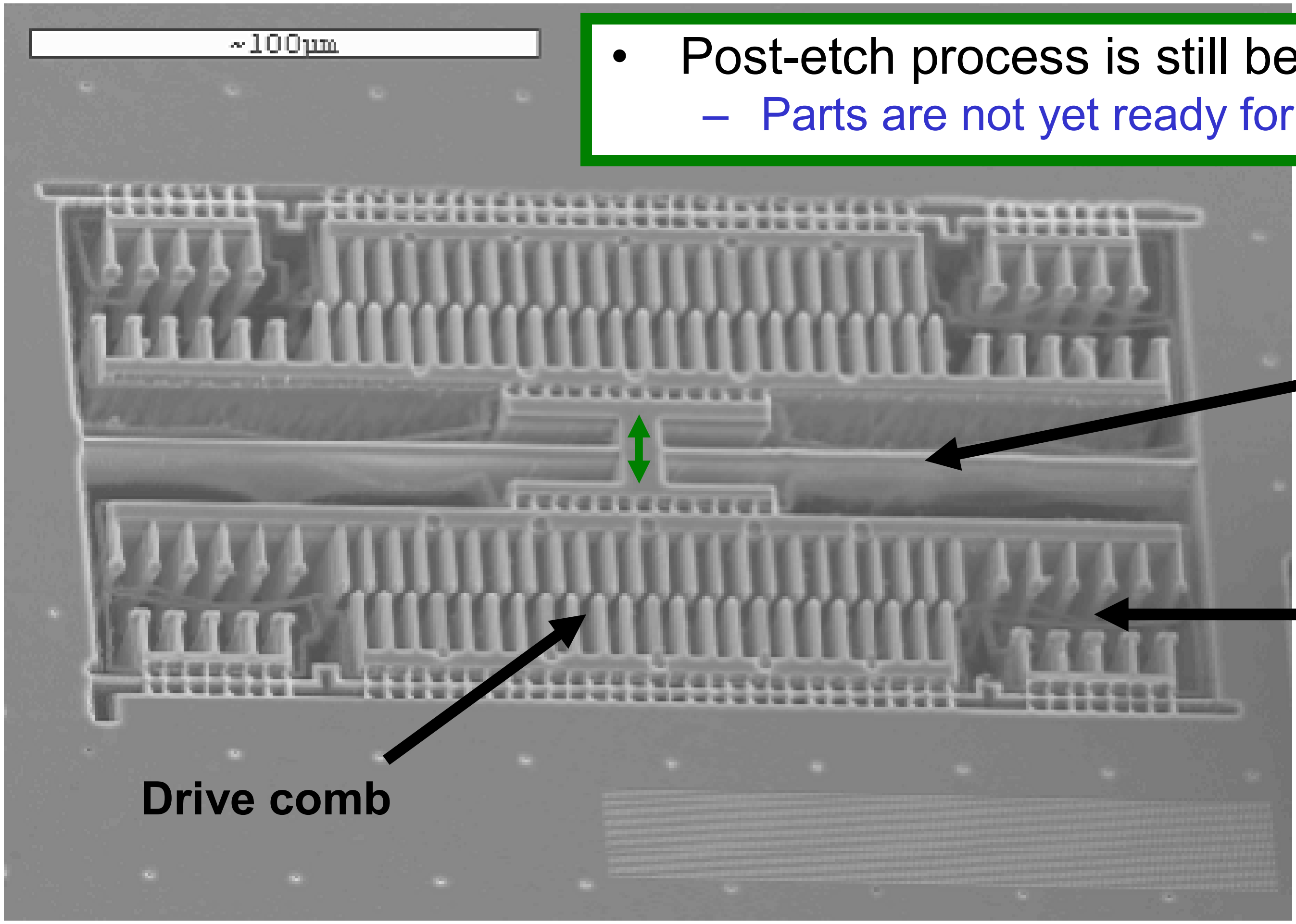
Requirements for Energy-Recovering Clock/Power Supplies

- All of the known reversible computing schemes invoke a periodic global signal that synchronizes and drives adiabatic transitions in the logic.
  - For good system-level energy efficiency, this signal must oscillate resonantly and near-ballistically, with a high effective quality factor.
- Several factors make the design of a resonant clock distributor that has satisfactorily high efficiency quite difficult:
  - Any uncompensated back-action of logic on resonator
  - In some resonators, Q factor may scale unfavorably with size
  - Excess stored energy in resonator may hurt effective quality factor
- There's no reason to think that it's <u>impossible</u> to do it…
  - But it is definitely a nontrivial hurdle, that we reversible computing researchers need to face up to, pretty urgently…
    - If we want to make reversible computing practical in time to avoid an extended period of stagnation in computer performance growth.

MEMS Quasi-Trapezoidal Resonator: 1st Fabbed Prototype

(Funding source: SRC CSR program)

- Post-etch process is still being fine-tuned.
  - Parts are not yet ready for testing…



(PATENT PENDING, UNIVERSITY OF FLORIDA)

General Reasons Why Practical Reversible Computing is Difficult

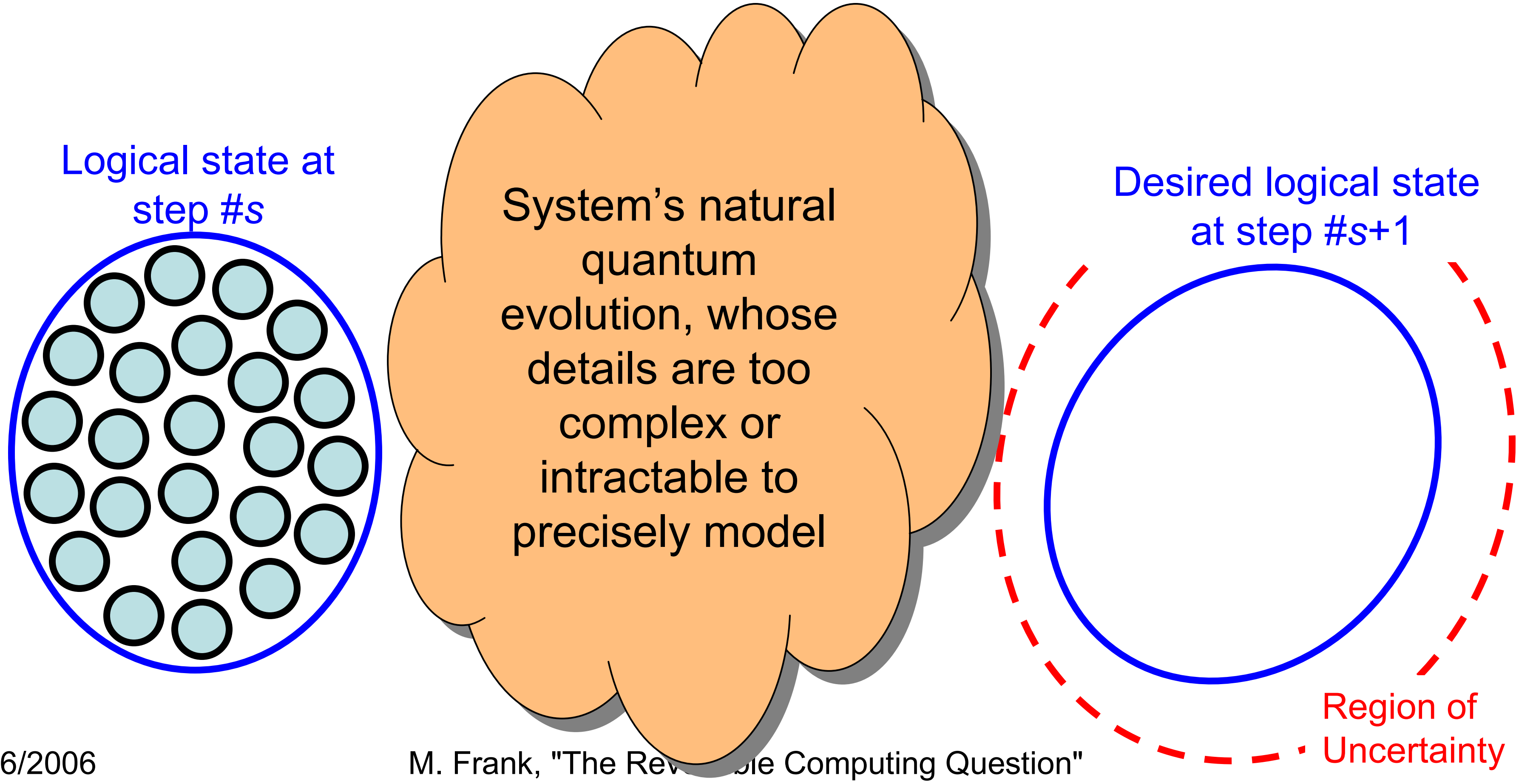
- Complex physical systems typically include *many* naturally occurring channels & mechanisms for energy dissipation.
  - Electromagnetic emission, phonon excitation, scattering, *etc.*
  - All must be delicately blocked to truly approach zero dissipation.
- We really must direct & keep track of where all (or nearly all) of the system's active energy is going at all times!
  - Accurately control/track the system's trajectory in configuration space.
  - Requires great care in design, & great precision in modeling.
- The physical architecture of the system is tightly constrained by the requirement for (near-) reversibility of the logic.
  - Gate-level synchrony, careful load balancing, elimination of unwanted reflections from impedance non-uniformities, *etc.*
  - Reversible logic, functional units, HW architectures & SW algorithms.
- Reversible logic itself introduces substantial (polynomial) space-time complexity overheads.
  - These bite a large chunk off of its energy-efficiency benefits.
  - This overhead appears to be inevitable in general-purpose apps.

Why Reversible Computing Might Still Be Possible, Eventually…

- Fundamentally, we know from quantum theory that physical systems intrinsically evolve with *no* inherent entropy increase.
  - A precisely characterized unitary evolution $\rho(t) = U(t)\rho(0)$ conserves the entropy $S(\rho)$ of any initial mixed state ρ.
- Thus, all "apparent" entropy increase ultimately arises from:
  - Imprecision in our knowledge of the fundamental physical laws (U).
  - Physical modeling techniques that (for practical reasons) explicitly neglect some of the information that we could infer about the state.
    - E.g., State vector projection, reduced density matrices, decoherence.
- To build systems with arbitrarily slow entropy increase, "just:"
  - Refine our knowledge of physical laws (values of constants, etc.) to ever more precision.
  - Develop ever more accurate, less approximate techniques for analytically/numerically modeling the time evolution of larger systems.
  - Learn how to design & construct increasingly complex systems whose engineered built-in dynamics is increasingly useful & powerful,
    - while still remaining feasible to model and track accurately.

One Big Reason for Optimism

- For a machine to have a high degree of *classical* reversibility *doesn't* appear to require that we maintain global phase coherence, or track the entire detailed evolution of all the quantum microstates…
  - It only requires that the rate of inflation of phase space volume is not too fast, and that most states end up *somewhere* in the desired region
    - Knowing which states go where within the desired region is not important



Logical state at step #*s*

System's natural quantum evolution, whose details are too complex or intractable to precisely model

Desired logical state at step #*s*+1

Region of Uncertainty

A Call to Action

- The world of computing is threatened by permanent performance-per-power stagnation in 1-2 decades…
  - We really should try hard to avoid this, if at all possible!
    - A wide variety of very important applications will be impacted.
- Many more of the nation's (and the world's) top physicists and computer scientists must be recruited,
  - to tackle the great "Reversible Computing Challenge."
- **Urgently needed:** A major new funding program; a "Manhattan Project" for energy-efficient computing!
  - **Mission:** Demonstrate computing beyond the von Neumann-Landauer limit in a practical, scalable machine!
    - Or, if it really can't be done for some reason, find a completely rock-solid proof from fundamental physics showing why.

Conclusions

- Practical reversible computing will become a necessity within our lifetimes,
  - if we want substantial progress in computing performance/power beyond the next 1-2 decades.
- Much progress in our understanding of RC has been made in the past three decades…
  - But much important work still remains to be done.
- I encourage my audience to help me urge the nation’s best thinkers to join the cause of finally answering the Reversible Computing Question, once and for all.