Nanocomputer Systems Engineering

Laying the Key Methodological Foundations for the Design of 21st-Century Computer Technology

Michael P. Frank
University of Florida
College of Engineering
Departments of CISE and ECE
`mpf@cise.ufl.edu`

NanoEngineering World Forum
International Engineering Consortium
Marlborough, Massachusetts
June 23-25, 2003

Abstract

- What is Nanocomputer Systems Engineering?
  - Interdisciplinary engineering of computers w. nanoscale parts.
  - Recognizes tight interplay between physics and computing.
- Physical Computing Theory
  - Models of computing based on fundamental physics.
  - Powerful, accurate, and technology-independent.
  - Key capabilities include *reversible* and *quantum* computing.
- Technology Scaling and Systems Analysis
  - Compared cost-efficiency of reversible vs. irreversible technologies.
  - Reversible computing may win by factors of ≥1,000× by mid-century.
  - We outline how this projection was obtained.
- **Conclusion:** More attention should be paid to the design of reversible, ballistic device mechanisms.
  - Low leakage, high Q factor will both be critically important in bit-device engineering for nanocomputers.

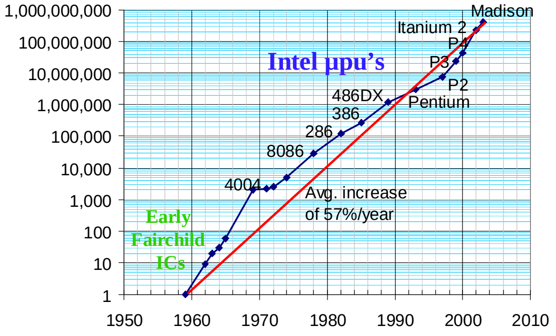
Organization of Talk

1. Moore's Law vs. Fundamental Physics
2. Methodological Principles of NCSE
3. Physical Computing Theory
4. Reversible Computing
5. Cost-Efficiency Analysis of RC
6. Conclusions

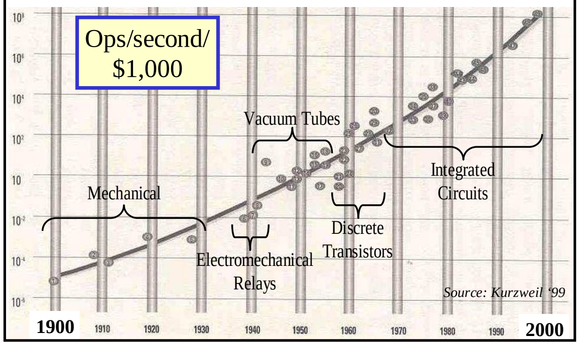
Organization of Talk

1. Moore's Law vs. Nanoscale Limits
2. Methodological Principles of NCSE
3. Physical Computing Theory
4. Reversible Computing
5. Cost-Efficiency Analysis of RC
6. Conclusions

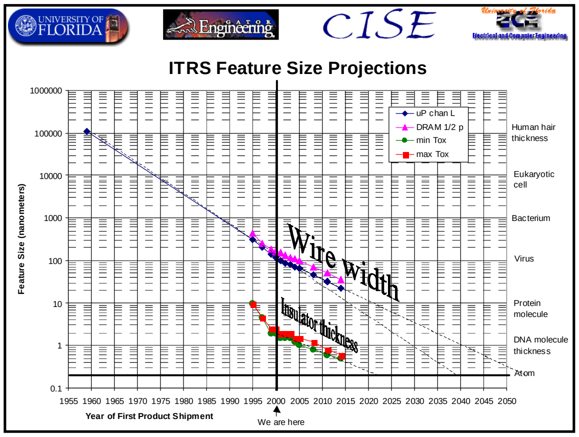
Moore's Law – Devices per IC

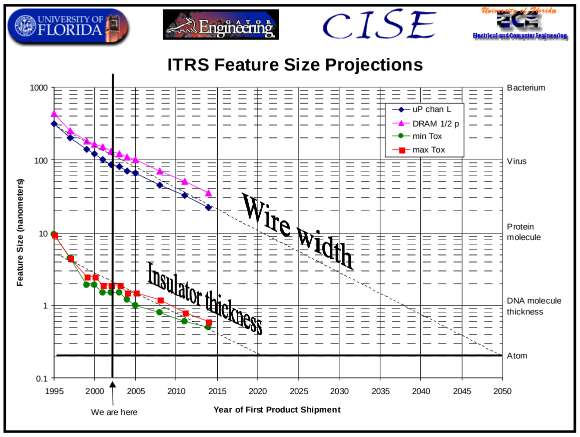


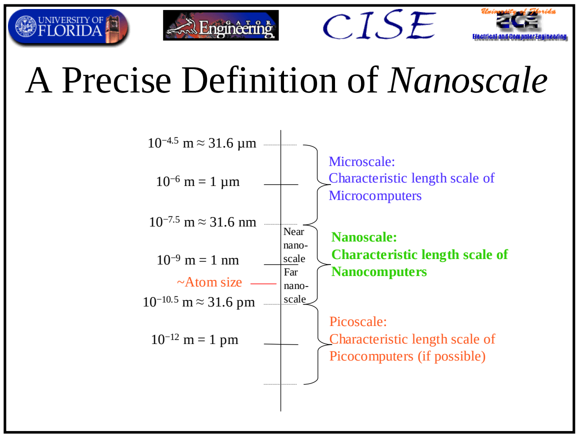
Super-Exponential Long-Term Trend



Source: Kurzweil '99


ITRS Feature Size Projections


ITRS Feature Size Projections


A Precise Definition of *Nanoscale*

$10^{-4.5}$ m ≈ 31.6 μm

10^{-6} m = 1 μm

$10^{-7.5}$ m ≈ 31.6 nm

10^{-9} m = 1 nm

~Atom size

$10^{-10.5}$ m ≈ 31.6 pm

10^{-12} m = 1 pm

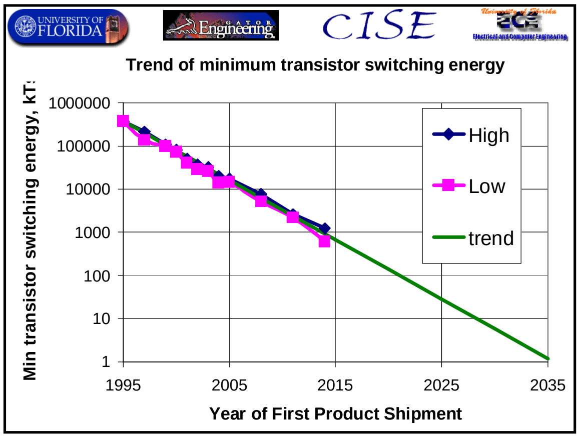
Near nano-scale

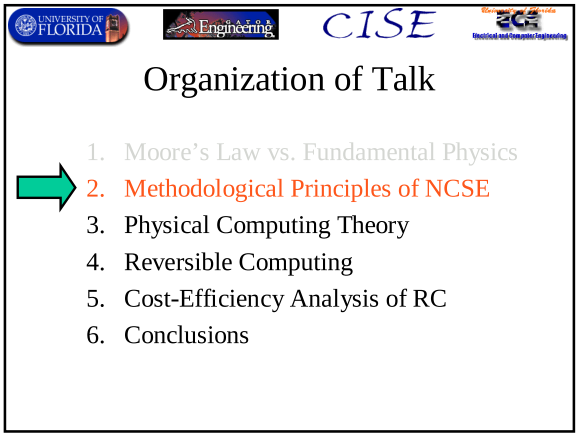
Far nano-scale

Microscale:
Characteristic length scale of Microcomputers

**Nanoscale:
Characteristic length scale of Nanocomputers**

Picoscale:
Characteristic length scale of Picocomputers (if possible)


Trend of minimum transistor switching energy


Organization of Talk

1. Moore’s Law vs. Fundamental Physics
2. Methodological Principles of NCSE
3. Physical Computing Theory
4. Reversible Computing
5. Cost-Efficiency Analysis of RC
6. Conclusions


Key Principles of NCSE

- Design for Generalized Cost-Efficiency
- Physics-Based Modeling
- Technology-Independent Models
- Multi-Domain Modeling
- Hierarchical Modeling
- Global System Design Optimization

Cost-Efficiency:
The Key Figure of Merit

- **Claim:** *All* practical engineering design-optimization can arguably be ultimately reduced to maximization of a generalized, system-level *cost-efficiency* characteristic.
  - Given an appropriate model of cost "$".
- Definition of the Cost-Efficiency $\%_\$$ of a process:
  - $\%_\$ \overset{\text{def}}{=} \$_{\text{min}}/\$_{\text{actual}}$
- Maximize $\%_\$$ by minimizing $\$_{\text{actual}}$
  - Note: This is valid even when $\$_{\text{min}}$ is unknown

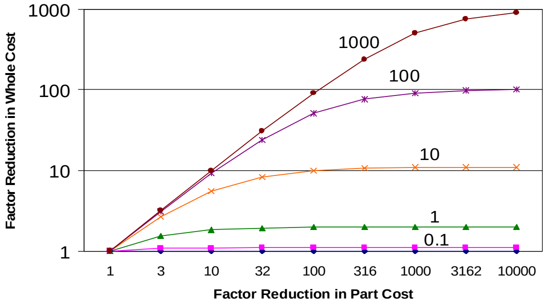
Important Cost Categories in Computing

- Hardware-Proportional Costs:
  - Initial Manufacturing Cost
- Time-Proportional Costs:
  - Inconvenience to User Waiting for Result
- (Hardware×Time)-Proportional Costs:
  - Amortized Manufacturing Cost
  - Maintenance & Operation Costs
  - Opportunity Costs
- Energy-Proportional Costs:
  - Adiabatic Losses
  - Non-adiabatic Losses From Bit Erasure
  - Note: These may both vary *independently* of (HW×Time)!

Focus of most traditional theory about computational "complexity."

These costs ***must*** be included also in practical theoretical models of nanocomputing!

The Generalized Amdahl's Law of Diminishing Returns



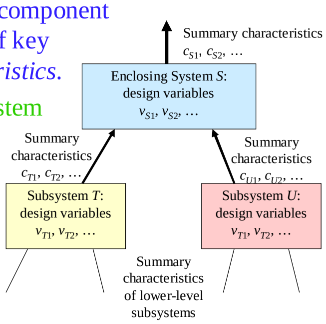
Computer Modeling Areas

1. Logic Devices
2. Technology Scaling
3. Interconnections
4. Synchronization
5. Processor Architecture
6. Capacity Scaling
7. Energy Transfer
8. Programming
9. Error Handling
10. Performance
11. Cost

Any Optimal, Physically Realistic Model of Computing Must Accurately Address *All* these Areas!

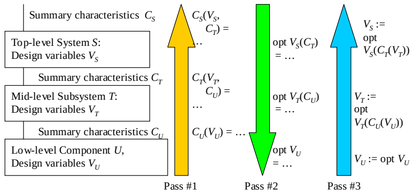
Hierarchical System Design

- Abstract from sub-component designs to values of key *summary characteristics.*
- Separates super-system design from sub-system design.
- Facilitates global optimization of system across all levels of design.



Three-Pass System Optimization

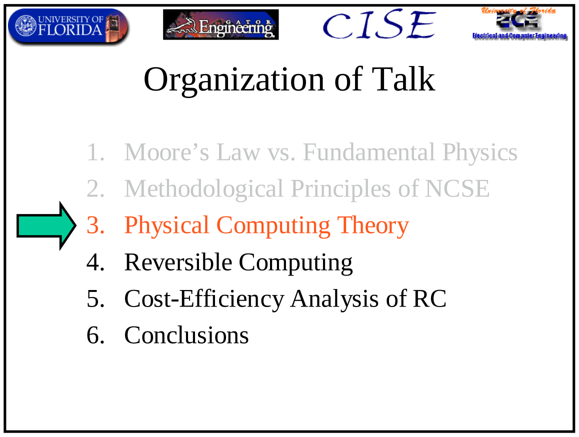
- A general methodology for the interdisciplinary optimization of the design of complex systems.



1) Express system performance characteristics as functions of component design variables.

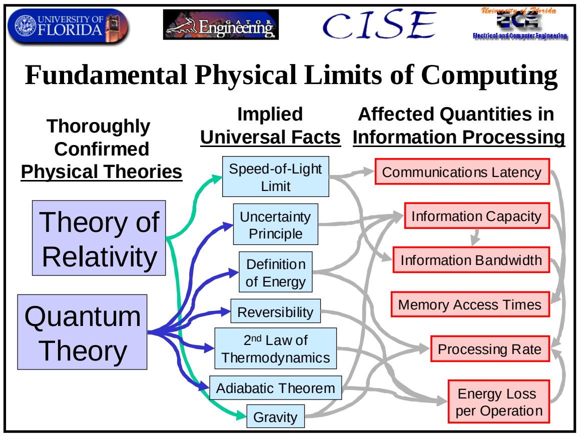
2) Compose optimization procedures.

3) Select optimized values of design parameters.



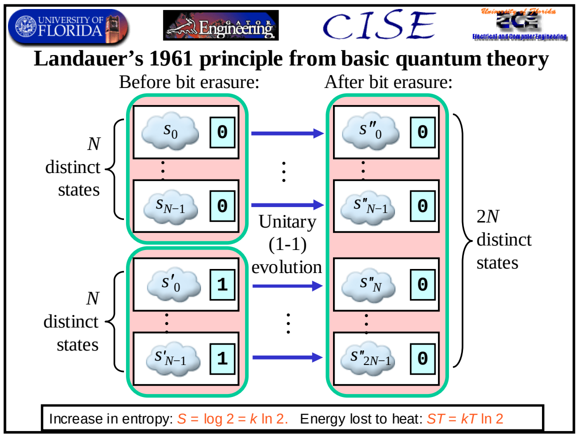
Organization of Talk

1. Moore's Law vs. Fundamental Physics
2. Methodological Principles of NCSE
3. Physical Computing Theory
4. Reversible Computing
5. Cost-Efficiency Analysis of RC
6. Conclusions



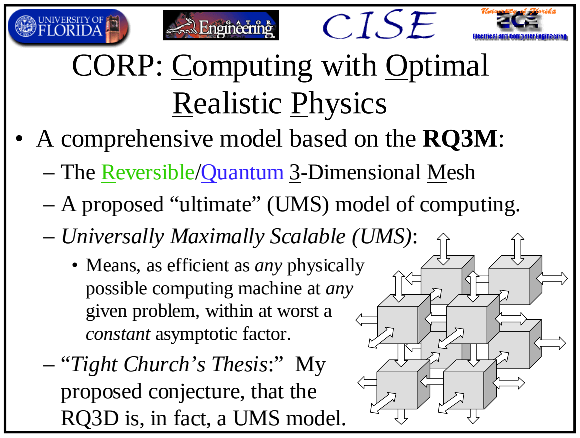
Fundamental Physical Limits of Computing

Thoroughly Confirmed Physical Theories	Implied Universal Facts	Affected Quantities in Information Processing
Theory of Relativity	Speed-of-Light Limit	Communications Latency
Quantum Theory	Uncertainty Principle	Information Capacity
	Definition of Energy	Information Bandwidth
	Reversibility	Memory Access Times
	2nd Law of Thermodynamics	Processing Rate
	Adiabatic Theorem	Energy Loss per Operation
	Gravity	



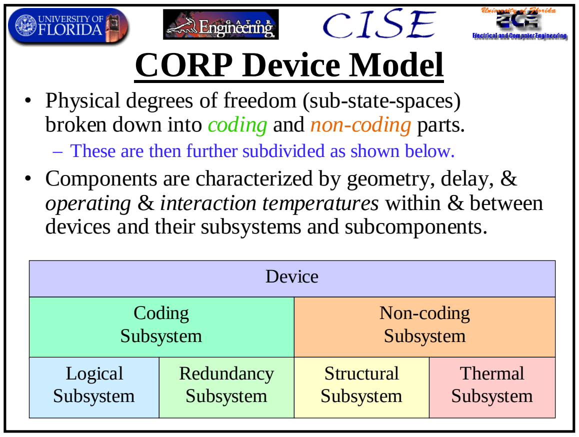
Landauer's 1961 principle from basic quantum theory

Increase in entropy: $S = \log 2 = k \ln 2$. Energy lost to heat: $ST = kT \ln 2$



CORP: Computing with Optimal Realistic Physics

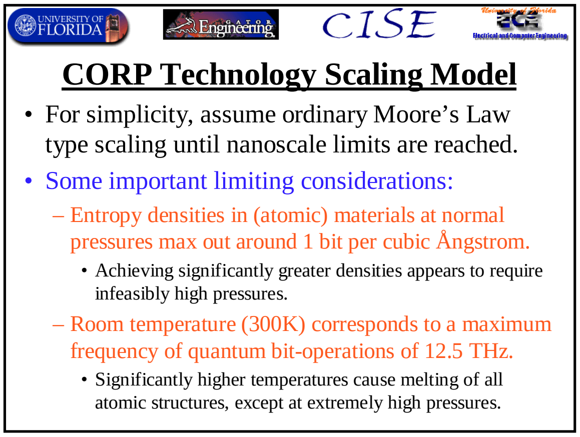
- A comprehensive model based on the **RQ3M**:
  - The Reversible/Quantum 3-Dimensional Mesh
  - A proposed "ultimate" (UMS) model of computing.
  - *Universally Maximally Scalable (UMS)*:
    - Means, as efficient as *any* physically possible computing machine at *any* given problem, within at worst a *constant* asymptotic factor.
  - "*Tight Church's Thesis*:" My proposed conjecture, that the RQ3D is, in fact, a UMS model.



CORP Device Model

- Physical degrees of freedom (sub-state-spaces) broken down into *coding* and *non-coding* parts.
  - These are then further subdivided as shown below.
- Components are characterized by geometry, delay, & *operating* & *interaction temperatures* within & between devices and their subsystems and subcomponents.

Device			
Coding Subsystem		Non-coding Subsystem	
Logical Subsystem	Redundancy Subsystem	Structural Subsystem	Thermal Subsystem



CORP Technology Scaling Model

- For simplicity, assume ordinary Moore's Law type scaling until nanoscale limits are reached.
- Some important limiting considerations:
  - Entropy densities in (atomic) materials at normal pressures max out around 1 bit per cubic Ångstrom.
    - Achieving significantly greater densities appears to require infeasibly high pressures.
  - Room temperature (300K) corresponds to a maximum frequency of quantum bit-operations of 12.5 THz.
    - Significantly higher temperatures cause melting of all atomic structures, except at extremely high pressures.

CORP Capacity Scaling Model

- Multiprocessing model
- Mesh-type (locally connected) interconnect structure
- Thermal pathways explicitly represented!
- Scaling in 3D up to thermal limits
- Device frequencies can be scaled down as number of devices increases, for maximum energy efficiency and cost-efficiency

---

Other Aspects of CORP Modeling

- Interconnect & Timing Models
  - Interconnects and oscillators can be treated as just special cases of devices.
  - Generalized mesh-style interconnect network.
- Architectural Model (Logic gates up to Processors)
  - Architectural design tools & methodologies should not preclude efficient reversible & quantum hardware designs!
- Programming Model
  - Should support standard programming paradigms.
  - But, should ***also*** permit expressing efficient reversible & quantum algorithms, in cases where these are beneficial.

Organization of Talk

1. Moore's Law vs. Fundamental Physics
2. Methodological Principles of NCSE
3. Physical Computing Theory
4. Reversible Computing
5. Cost-Efficiency Analysis of RC
6. Conclusions

Terminology / Requirements

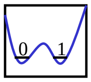
Property of Computing Mechanism	Approximate Meaning	Required for Quantum Computing?	Required for Reversible Computing?
(Treated As) Unitary	System's full invertible quantum evolution, w. all phase information, is modeled & tracked	Yes, device & system evolution must be modeled as ~unitary, within threshold	No, only reversible evolution of classical state variables need be tracked
Coherent	Pure quantum states don't decohere (for us) into statistical mixtures	Yes, must maintain full global coherence, locally within threshold	No, only maintain stability of local pointer states+transitions
Adiabatic	No entropy flow in/out of computational subsystem	Yes, must be above a certain threshold	Yes, as high as possible
Isentropic / Thermodynamically Reversible	No new entropy generated by mechanism	Yes, must be above a certain threshold	Yes, as high as possible
Time-Independent Hamiltonian, Self-Controlled	Closed system, evolves autonomously w/o external control	No, transitions can be externally timed & controlled	Yes, if we care about energy dissipation in the driving system
Ballistic	System evolves w. net forward momentum	No, transitions can be externally driven	Yes, if we care about performance

---

Some Claims Against Reversible Computing	Eventual Resolution of Claim
John von Neumann, 1949 – Offhandedly remarks during a lecture that computing requires $kT \ln 2$ dissipation per "elementary act of decision" (bit-operation).	No proof provided. Twelve years later, Rolf Landauer of IBM tries valiantly to prove it, but succeeds only for logically irreversible operations.
Rolf Landauer, 1961 – Proposes that the logically irreversible operations which necessarily cause dissipation are unavoidable.	Landauer's argument for unavoidability of logically irreversible operations was conclusively refuted by Bennett's 1973 paper.
Bennett's 1973 construction is criticized for using too much memory.	Bennett devises a more space-efficient version of the algorithm in 1989.
Bennett's models criticized by various parties for depending on random Brownian motion, and not making steady forward progress.	Fredkin and Toffoli at MIT, 1980, provide ballistic "billiard ball" model of reversible computing that makes steady progress.
Various parties note that Fredkin's original classical-mechanical billiard-ball model is chaotically unstable.	Zurek, 1984, shows that quantum models can avoid the chaotic instabilities. (Though there are workable classical ways to fix the problem also.)
Various parties propose that classical reversible logic principles won't work at the nanoscale, for unspecified or vaguely-stated reasons.	Drexler, 1980's, designs various mechanical nanoscale reversible logics and carefully analyzes their energy dissipation.
Carver Mead, CalTech, 1980 – Attempts to show that the kT bound is unavoidable in electronic devices, via a collection of counter-examples.	No general proof provided. Later he asked Feynman about the issue; in 1985 Feynman provided a quantum-mechanical model of reversible computing.
Various parties point out that Feynman's model only supports serial computation.	Margolus at MIT, 1990, demonstrates a parallel quantum model of reversible computing—but only with 1 dimension of parallelism.
People question whether the various theoretical models can be validated with a working electronic implementation.	Seitz and colleagues at CalTech, 1985, demonstrate working energy recovery circuits using adiabatic switching principles.
Seitz, 1985—Has some working circuits, unsure if arbitrary logic is possible.	Koller & Athas, Hall, and Merkle (1992) separately devise general reversible combinational logics.
Koller & Athas, 1992 – Conjecture reversible *sequential* feedback logic impossible.	Younis & Knight @MIT do reversible sequential, pipelineable circuits in 1993-94.
Some computer architects wonder whether the constraint of reversible logic leads to unreasonable design convolutions.	Vieri, Frank and coworkers at MIT, 1995-99, refute these qualms by demonstrating straightforward designs for fully-reversible, scalable gate arrays, microprocessors, and instruction sets.
Some computer science theorists suggest that the algorithmic overheads of reversible computing might outweigh their practical benefits.	Frank, 1997-2003, publishes a variety of rigorous theoretical analysis refuting these claims for the most general classes of applications.
Various parties point out that high-quality power supplies for adiabatic circuits seem difficult to build electronically.	Frank, 2000, suggests microscale/nanoscale electro-mechanical resonators for high-quality energy recovery with desired waveform shape and frequency.
Frank, 2002—Briefly wonders if synchronization of parallel reversible computation in 3 dimensions (not covered by Margolus) might not be possible.	Later that year, Frank devises a simple mechanical model showing that parallel reversible systems can indeed be synchronized locally in 3 dimensions.

Bistable Potential-Energy Wells

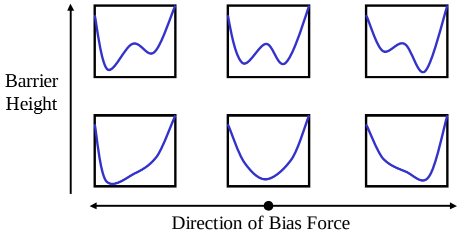
- Consider any system having an adjustable, bistable potential energy surface (PES) in its configuration space.
- The two stable states form a natural *bit*.
  - One state represents 0, the other 1.
- Consider now the P.E. well having two adjustable parameters:
  - (1) Height of the potential energy barrier relative to the well bottom
  - (2) Relative height of the left and right states in the well (bias)



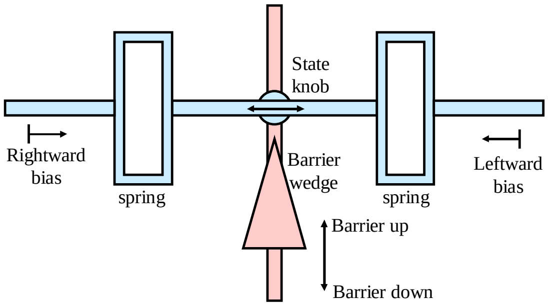
(Landauer '61)

Possible Parameter Settings

- We will distinguish six qualitatively different settings of the well parameters, as follows…

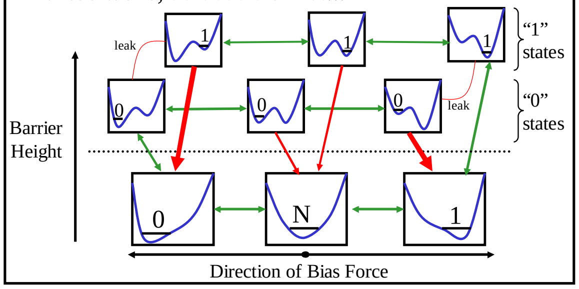


One Mechanical Implementation



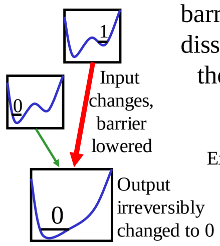
Possible Adiabatic Transitions

- Catalog of all the possible transitions in these wells, adiabatic & not… (Ignoring superposition states.)



Ordinary Irreversible Logics

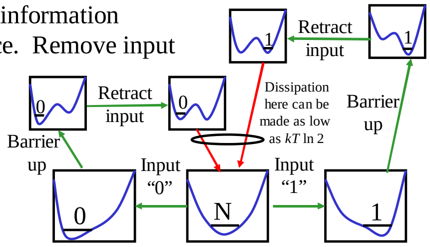
- Principle of operation: Lower a barrier, or not, based on input. Series/parallel combinations of barriers do logic. Major dissipation in at least one of the possible transitions.
  - Amplifies input signals.



Example: Ordinary CMOS logics

Ordinary Irreversible Memory

- Lower a barrier, dissipating stored information. Apply an input bias. Raise the barrier to latch the new information into place. Remove input bias.



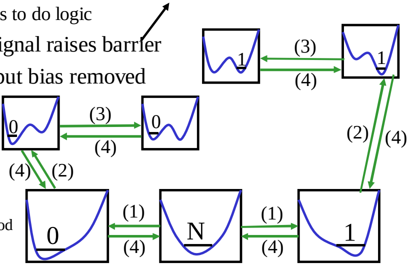
Example: DRAM

Input-Bias Clocked-Barrier Logic

- Cycle of operation:
  - (1) Data input applies bias
    - Add forces to do logic
  - (2) Clock signal raises barrier
  - (3) Data input bias removed

Can amplify/restore input signal in the barrier-raising step.

Can reset latch reversibly (4) given copy of contents.

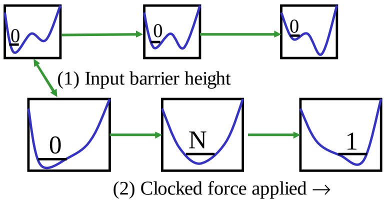


Examples: Adiabatic QDCA, SCRL latch, Rod logic latch, PQ logic, Buckled logic

Input-Barrier, Clocked-Bias Retractile

- Cycle of operation:
  - – Inputs raise or lower barriers
    - Do logic w. series/parallel barriers
  - – Clock applies bias force which changes state, or not

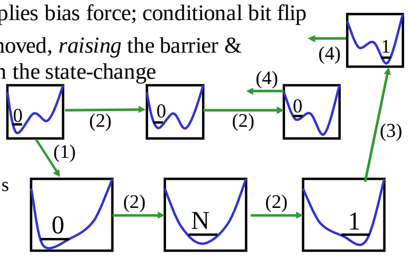
- Barrier signal amplified.
- Must reset output prior to input.
- Combinational logic only!



Examples:
Hall's logic,
SCRL gates,
Rod logic interlocks

Input-Barrier, Clocked-Bias Latching

- Cycle of operation:
  1. Input *conditionally lowers* barrier
     - Do logic w. series/parallel barriers
  2. Clock applies bias force; conditional bit flip
  3. Input removed, *raising* the barrier & locking in the state-change
  4. Clock bias can retract



Examples: Mike's 4-cycle adiabatic CMOS logic

Full Classical-Mechanical Model

The following components are sufficient for a complete, scalable, parallel, pipelinable, linear-time, stable, classical reversible computing system:

(a) Ballistically rotating flywheel driving linear motion.

(b) Scalable mesh to synchronize local flywheel phases in 3-D.

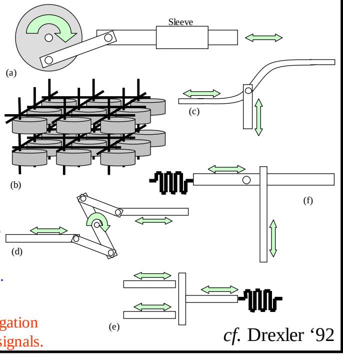
(c) Sinusoidal to flat-topped waveform shape converter.

(d) Non-amplifying signal inverter (NOT gate).

(e) Non-amplifying OR/AND gate.

(f) Signal amplifier/latch.

Primary drawback: Slow propagation speed of mechanical (phonon) signals.

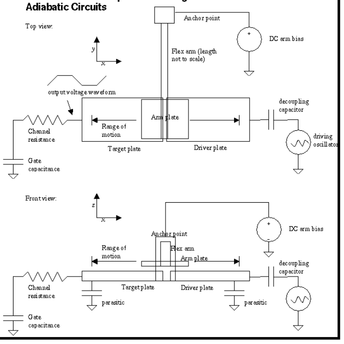


cf. Drexler '92

A MEMS Supply Concept

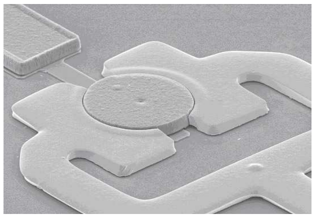
- Energy stored mechanically.
- Variable coupling strength → custom wave shape.
- Can reduce losses through balancing, filtering.

Sketch of MEMS Trapezoidal Voltage-Waveform Resonator for Adiabatic Circuits



MEMS/NEMS Resonators

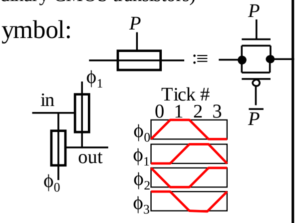
- State of the art technologies demonstrated in lab:
  - – Frequencies up into the microwave (>1 GHz) regime
  - – Q's >10,000 in vacuum, several thousand even in air!
- Are rapidly becoming the technology of choice for commercial RF filters, *etc.*, in embedded communications SoCs (Systems-on-a-Chip), *e.g.* for cellphones.



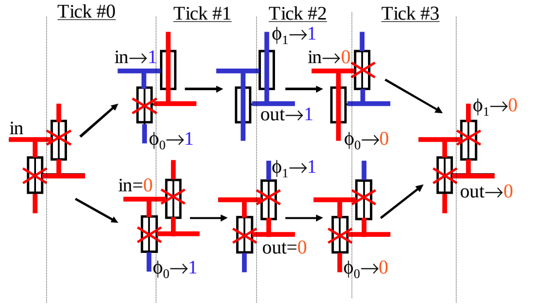
2LAL: 2-level Adiabatic Logic

(Implementable using ordinary CMOS transistors)

- Use simplified T-gate symbol:
- Basic buffer element:
  - – cross-coupled T-gates
- Only 4 timing signals, 4 ticks per cycle:
  - – ϕ_i rises during tick i
  - – ϕ_i falls during tick i+2 mod 4

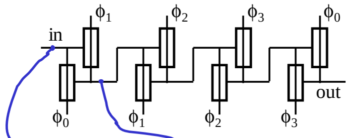


2LAL Cycle of Operation

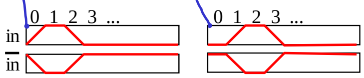


2LAL Shift Register Structure

- 1-tick delay per logic stage:

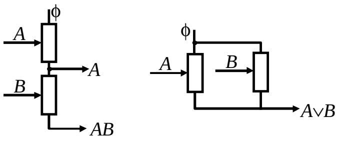


- Logic pulse timing & propagation:

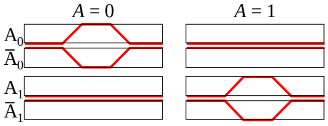


More complex logic functions

- Non-inverting Boolean functions:

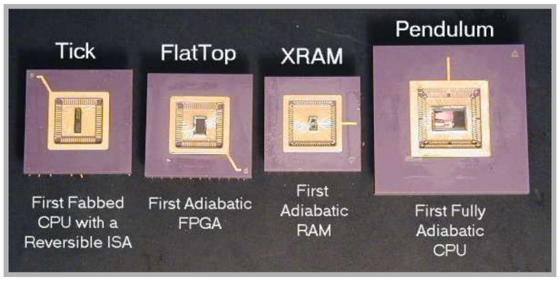


- For inverting functions, must use quad-rail logic encoding:
  - To invert, just swap the rails!
    - Zero-transistor "inverters."

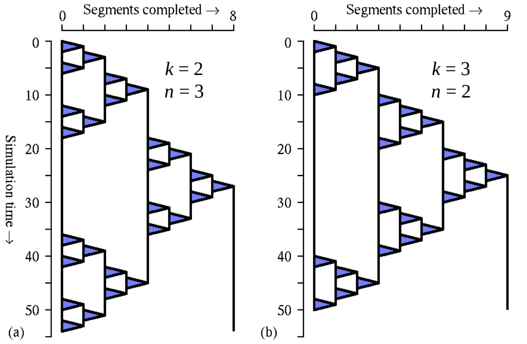


Reversible / Adiabatic Chips Designed @ MIT, 1996-1999

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



Reversible Emulation - Ben89



Organization of Talk

1. Moore's Law vs. Fundamental Physics
2. Methodological Principles of NCSE
3. Physical Computing Theory
4. Reversible Computing
5. Cost-Efficiency Analysis of RC
6. Conclusions

A Showcase Application of Our NCSE Methodology

- An important research question to be answered:
  - As nanocomputing technology advances, will reversible computing ever become very cost-effective, and if so, when?
- We applied our methodology as follows:
  - Made Realistic Model (Obeying Constraints)
  - Optimized Cost-Efficiency in the Model
  - Swept Model Parameters over Future Years

Important Factors Included in Our Model

- Entropic cost of irreversibility
- Algorithmic overheads of reversible logic
- Adiabatic speed vs. energy-usage tradeoff
- Optimized degree of reversibility
- Limited quality factors of real devices
- Communications latencies in parallel algorithms
- Realistic heat flux constraints

Technology-Independent Model of Nanoscale Logic Devices

I_{d} – Bits of internal logical state information per nano-device

S_{iop} – Entropy generated per irreversible nano-device operation

t_{ic} – Time per device cycle (irreversible case)

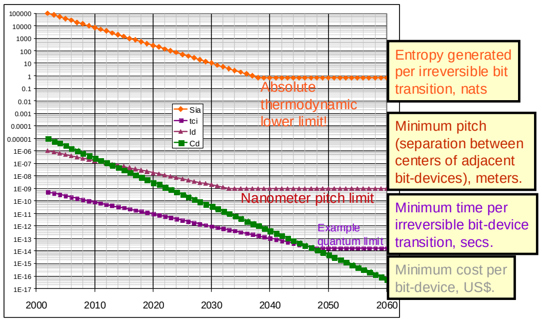
$S_{\mathrm{d,t}}$ – Entropy generated per device per unit time (standby rate, from leakage/decay)

$S_{\mathrm{rop,f}}$ – Entropy generated per reversible op per unit frequency

ℓ_{d} – Length (pitch) between neighboring nanodevices

$S_{\mathrm{A,t}}$ – Entropy flux per unit area per unit time

Technological Trend Assumptions



Entropy generated per irreversible bit transition, nats

Minimum pitch (separation between centers of adjacent bit-devices), meters.

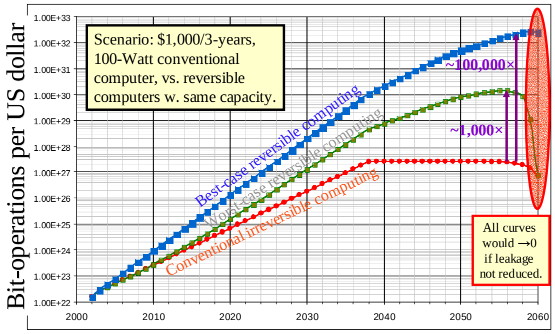
Minimum time per irreversible bit-device transition, secs.

Minimum cost per bit-device, US$.

Fixed Technology Assumptions

- Total cost of manufacture: US$1,000.00
  - User will pay this for a high-performance desktop CPU.
- Expected lifetime of hardware: 3 years
  - After which machine is obsolete and mostly depreciated.
- Total power limit: 100 Watts
  - Much greater than this and it would burn up your lap!
- Power flux limit: 100 Watts per square centimeter
  - Approximate limit of air-cooling capabilities
- Standby entropy generation rate: 1,000 nat/s/device
  - Arbitrarily chosen, but achievable in today's technology

Cost-Efficiency Benefits



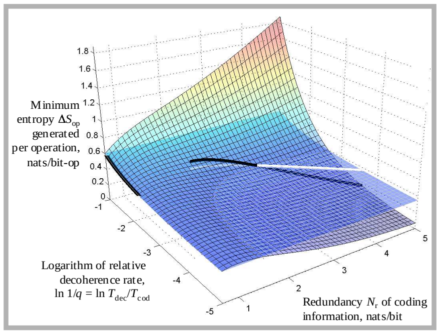
Scenario: $1,000/3-years, 100-Watt conventional computer, vs. reversible computers w. same capacity.

All curves would →0 if leakage not reduced.

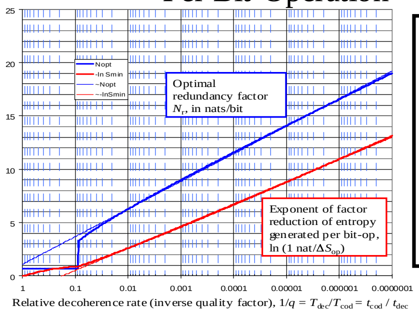
More Recent Work

Optimizing device size to
minimize entropy generation

Minimizing Entropy Generation in Field-Effect Nano-devices



Lower Limit to Entropy Generation Per Bit-Operation



Scaling with device's quantum "quality" factor q.

- The optimal redundancy factor scales as: $1.1248(\ln q)$
- The minimum entropy generation scales as: $q^{-0.9039}$

Conclusions

- We are developing an integrated and principled methodological foundation for analysis in the new field of **NanoComputer Systems Engineering (NCSE)**.
  - Techniques like our Physical Computing Theory are needed in order to properly address important and difficult questions.
    - *E.g.*, the realistic cost-efficiency of reversible computing.
- Results from our analytical models to date indicate that **Reversible Computing** offers ***extreme*** potential cost-efficiency advantages for future nanocomputing.
  - Even when taking its overheads into account!
- Thus, nanocomputing device engineers ***must*** focus harder on the requirements for efficient reversible operation:
  - *E.g.*, Low per-device leakage rates, high resonant Q factors.