Reversibility for Efficient Computing

Ph.D. Thesis Defense

Michael P. Frank
MIT EECS Dept./MIT AI Lab
`mpf@ai.mit.edu`

Monday, May 3, 1999

`http://www.ai.mit.edu/~mpf/rc/thesis/defense.html`

Thesis of doctoral work:

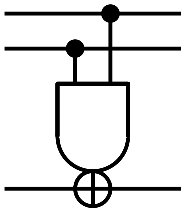
Reversible computing techniques (to be described) are:

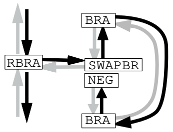
- Not *too* different from normal irreversible techniques.
- Straightforward to implement, at all levels.
- Definitely required for future computing efficiency.
- Beginning to become useful today.

It's time for this field to start getting more attention!

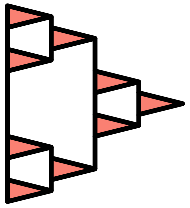
Overview of talk contents

- Basic principles.
- **Engineering & technology**:
  - Reversible logic circuits.
  - Programming reversible computers.
  - Near-term applications.

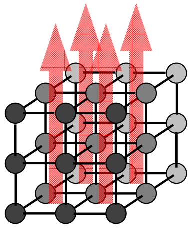




- **Theory & analysis**:
  - Traditional theory of reversible computing.
  - Physical computing theory, scaling.
  - Long-term applications.

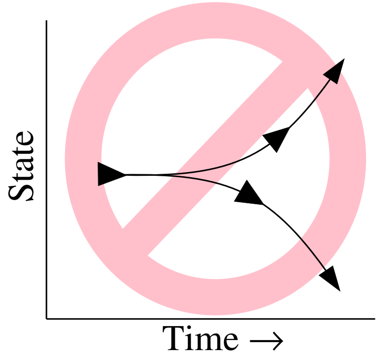


- Conclusions

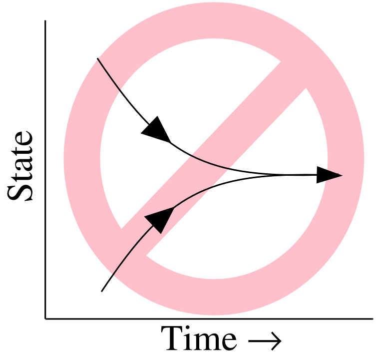


Forward and Reverse Determinism

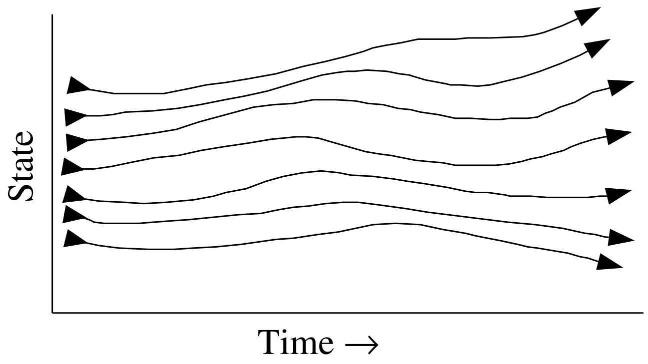
(Forward) Determinism:
No splits



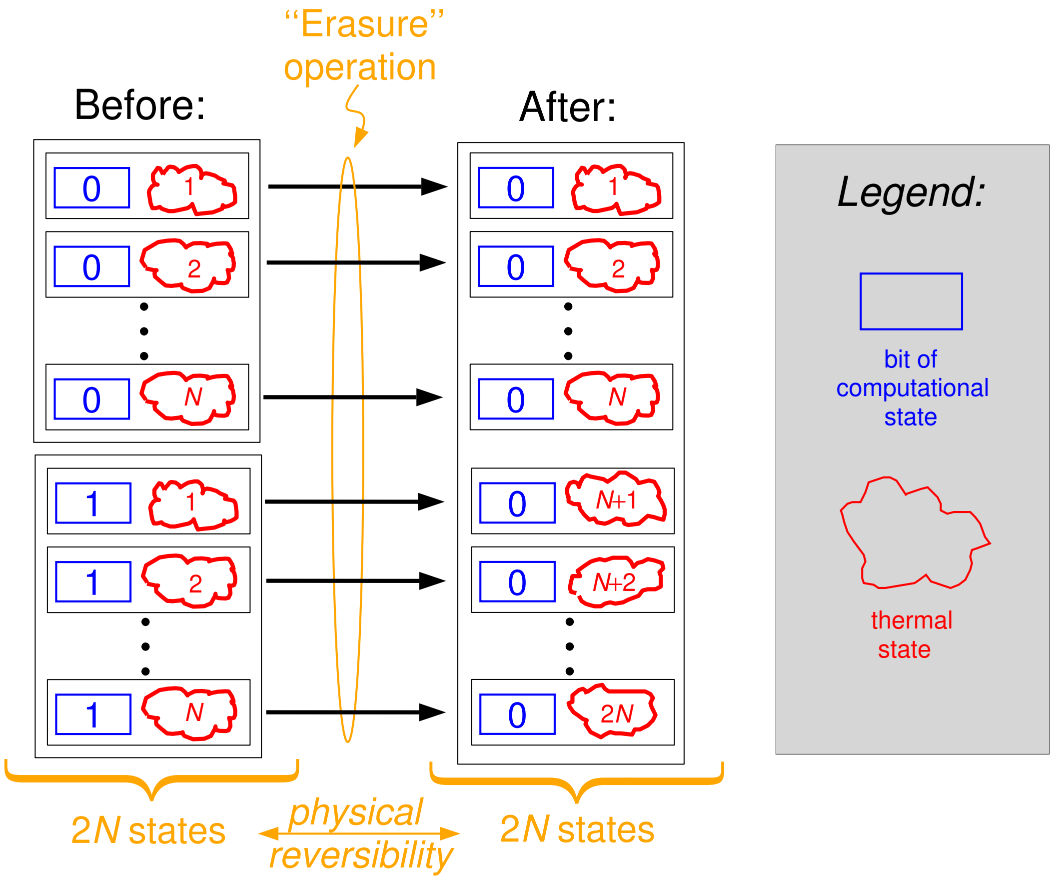
Reverse Determinism:
No merges



Forward and Reverse Determinism:



Information “erasure” in reversible physics

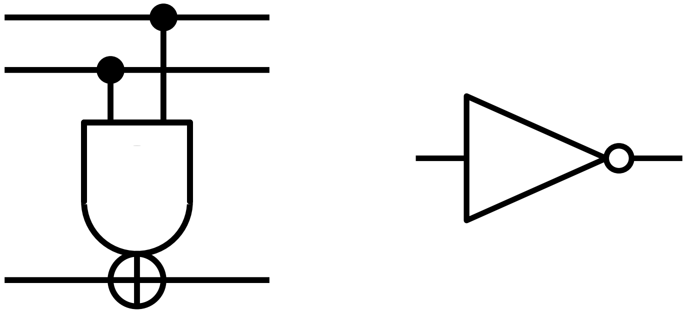


Before: N thermal states, entropy $S = k_{\mathrm{B}} \ln N$.

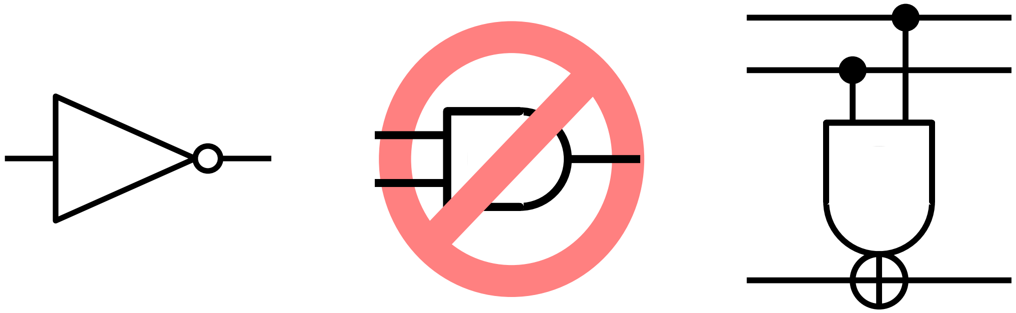
After: $2N$ thermal states, entropy $S = \ln 2N = k_{\mathrm{B}}(\ln 2 + \ln N)$.

$\Delta S = k_{\mathrm{B}}(\ln 2)$, $\Delta E = T\Delta S = k_{\mathrm{B}} T \ln 2$.

Part I: Reversible logic circuits

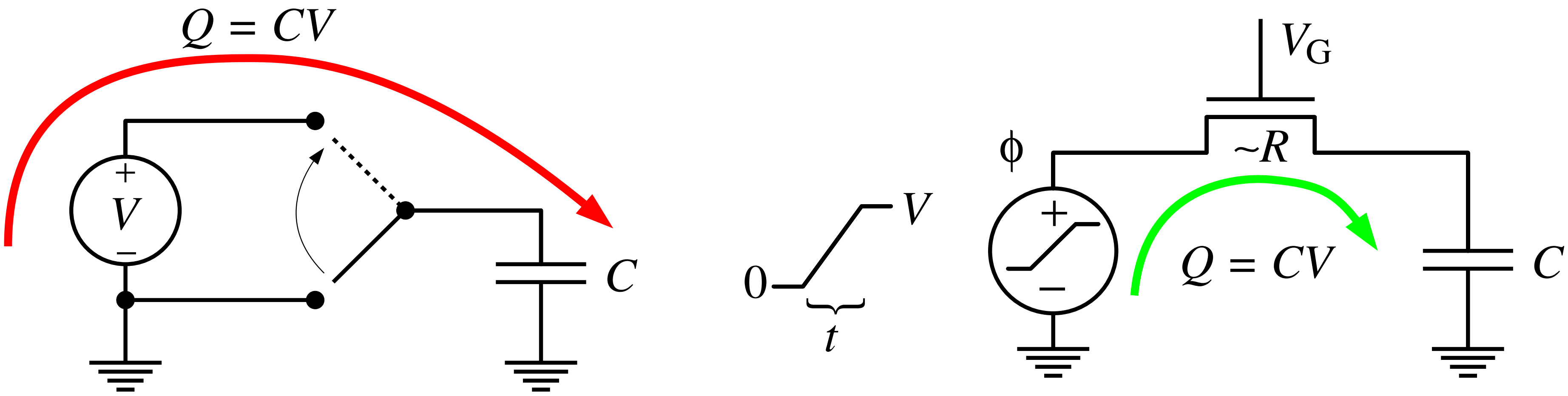


Fredkin-Toffoli reversible logic



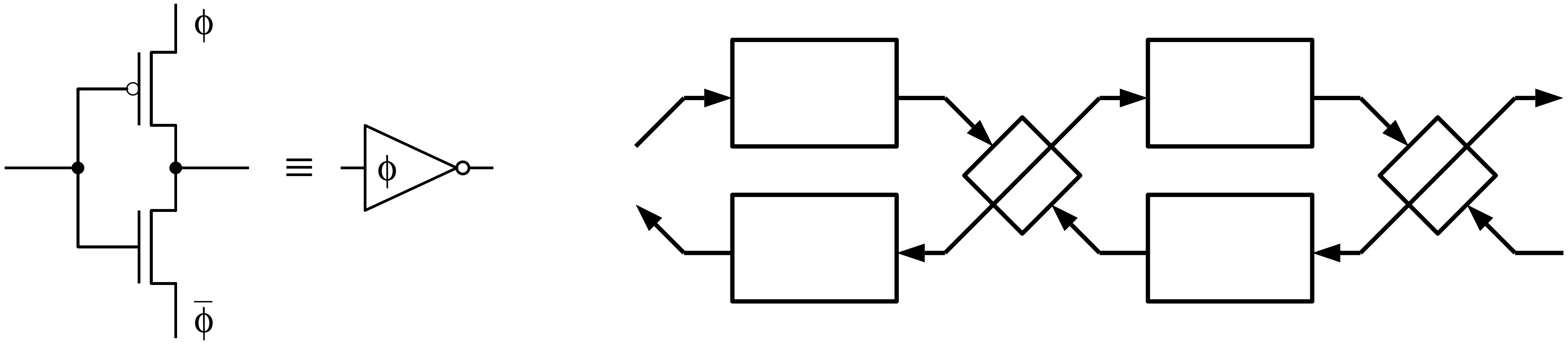
- Normal multi-input logic gates are not reversible.
  - Output insufficient to determine input.
- Toffoli's universal reversible 3-input, 3-output gate:
  - Inputs preserved, w. a delay.
  - Output reversibly toggled.
  - Subsumes AND/NAND/XOR/NOT.

Energy dissipation in irreversible vs. reversible electronics



- Dissipation in irreversible circuits at least $\frac{1}{2}CV^2$.
- Dissipation in reversible circuits is $CV^2(RC/t)$.
- For reliability of N ops/error despite thermal noise:
  - $\frac{1}{2}CV^2 \geq k_\mathrm{B}T \ln N$, in both.
- Decrease RC: less lossage at a given clock speed.

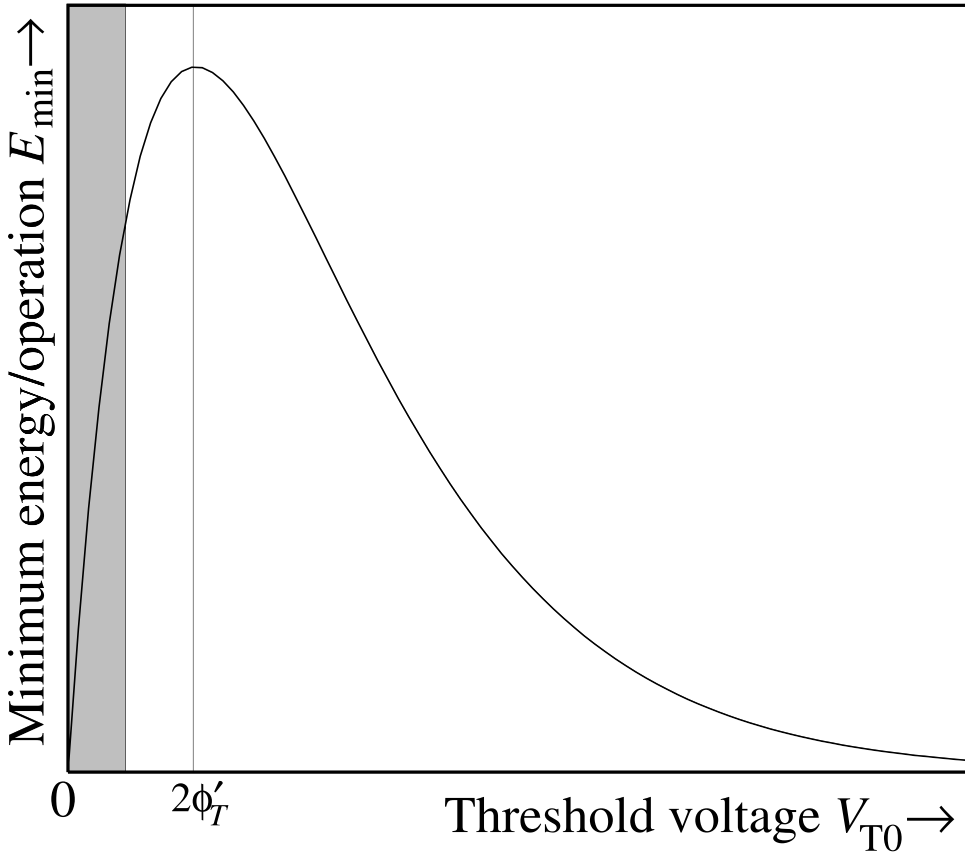
Younis-Knight SCRL reversible electronics



SCRL = *Split-level Charge Recovery Logic*:

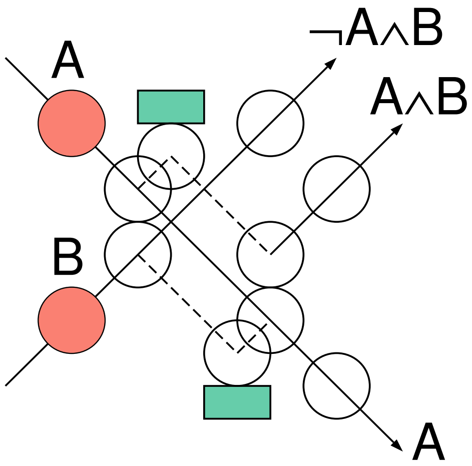
- Fully reversible sequential circuits.
- Ordinary semiconductor transistors.
- Resonant supply w. 3 levels.
- Never turn on trans. w. voltage across it.
- Enough info. in outputs to uncompute inputs.
- Compute in predefined phases.
- Me: voltage scaling analysis, bug fix.

Scaling of SCRL minimum energy with threshold voltage

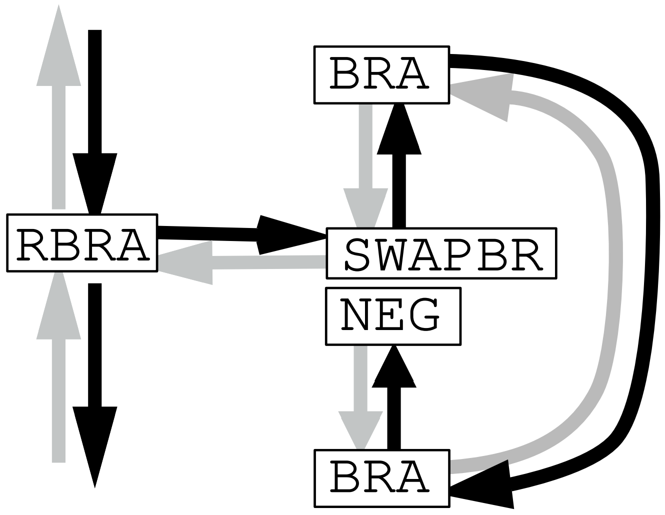


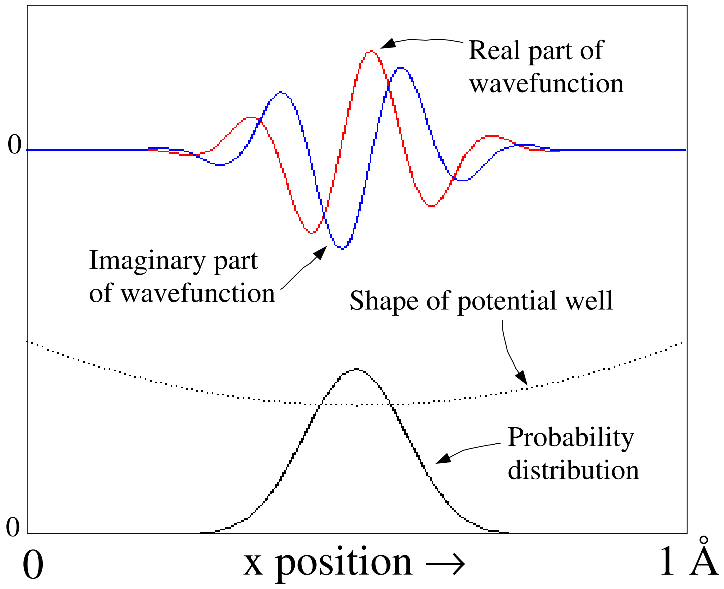
$$E_{\text{min}} \propto C_{\text{L}} V_{\text{T0}} e^{-\frac{1}{2} V_{\text{T0}}/\phi'_T}$$

FlatTop reversible mesh processor



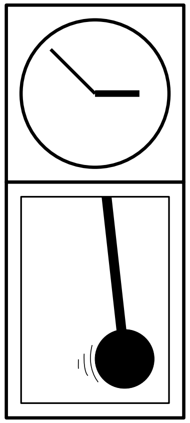
Part II: Programming reversible computers.





- PISA instruction set.
- R programming language.
- Reversible algorithms.

Pendulum reversible processor



- Carlin Vieri's Ph.D. project.
- Full RISC-style CPU.
- My contributions to PISA (Pend. instr. set arch.):
  - Asymptotic efficiency.
  - Guaranteed reversibility.
- Chips are being tested.

Example PISA-type instructions:

“Nonexpanding” arith./logic:

```
NEG  ra     (ra = -ra)
ADD  ra,rb  (ra += rb)
ADDI ra,imm (ra += [imm])
SUB  ra,rb  (ra -= rb)
XOR  ra,rb  (ra ^= rb)
RL   ra,imm (ra <=< imm)
```

“Expanding” arith./log.:

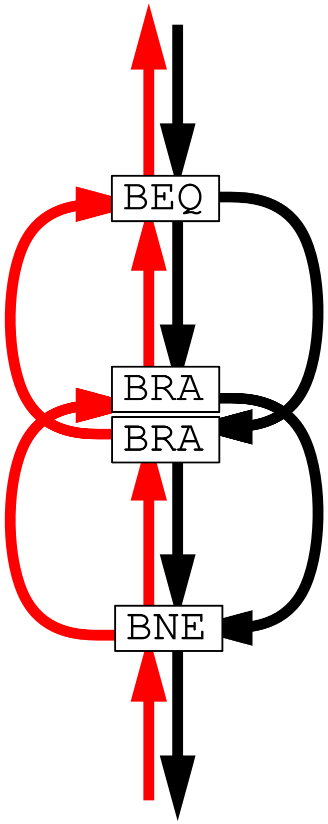
```
ANDX  ra,rb,rc  (ra ^= rb&rc)
ANDIX ra,rb,imm (ra ^= rb&[imm])
ORX   ra,rb,rc  (ra ^= rb|rc)
SLLX  ra,rb,imm (ra ^= rb<<imm)
SLTX  ra,rb,rc  (ra ^= (rb<rc)?1:0)
SRAX  ra,rb,imm (ra ^= rb>>imm)
```

Branch instructions:

```
BEQ    ra,rb,off (if ra=rb, BR+=off*dir)
BGEZ   ra,off    (if ra>=0, BR+=off*dir)
BGTZ   ra,off    (if ra>0, BR+=off*dir)
BRA    loff      (BR += loff*dir)
RBRA   loff      (dir=-dir, BR+=loff*dir)
SWAPBR ra        (ra <-> BR)
```

Some reversible control flow structures:

If/then/else



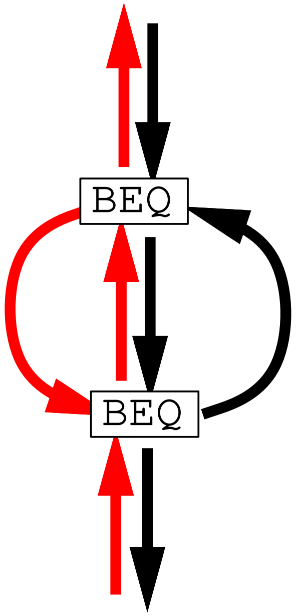
BEQ

BRA

BRA

BNE

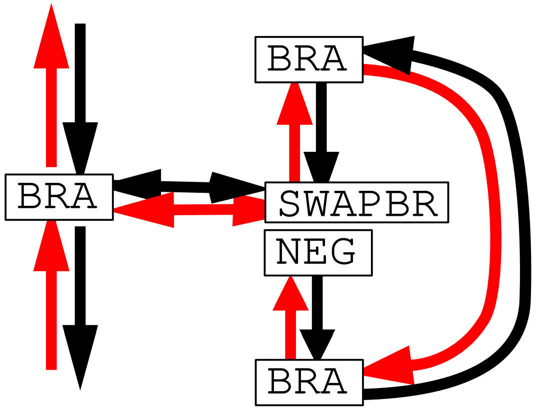
Simple loop



BEQ

BEQ

Subroutine call



BRA

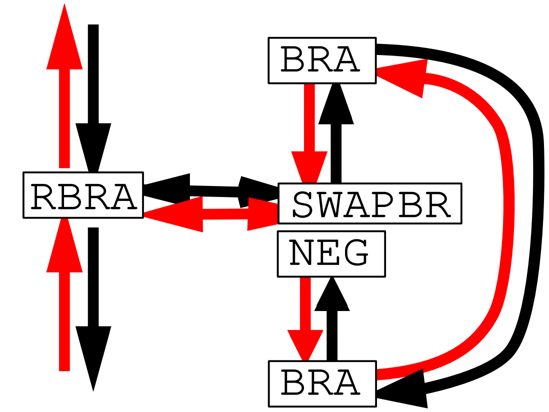
BRA

SWAPBR

NEG

BRA

Reverse subroutine call



BRA

RBRA

SWAPBR

NEG

BRA

The “R” reversible programming language

Features:

- Parenthesis-based syntax, for easy parsing.
- Simple C-like procedural semantics.
- Permits coding efficient reversible algorithms.
- The R language compiler:
  - Written in Common Lisp.
  - Targets PISA assembly code.

Example R language constructs

Control flow:

(if *condition* then
    statement ...)
(for *var* = *start* to *end*
    statement ...)
(defsub *subname* (arg_1 arg_2 ...)
    statement ...)
(call *subname* arg_1 arg_2 ...)
(rcall *subname* arg_1 arg_2 ...)

Data manipulation:

(let (*var* <- *val*)
    statement ...)
(*place* ++)
(*place* += *value*)
(*place* -= *value*)
(*place* ^= *value*)
⋮

Expressions:

(val_1 + val_2)
(val_1 - val_2)
(val_1 & val_2)
(* *address*)
(*array* _ *index*)
⋮

Data declaration:

(defword *name* *value*)
(defarray *name*
    $value_0$ $value_1$ ...)

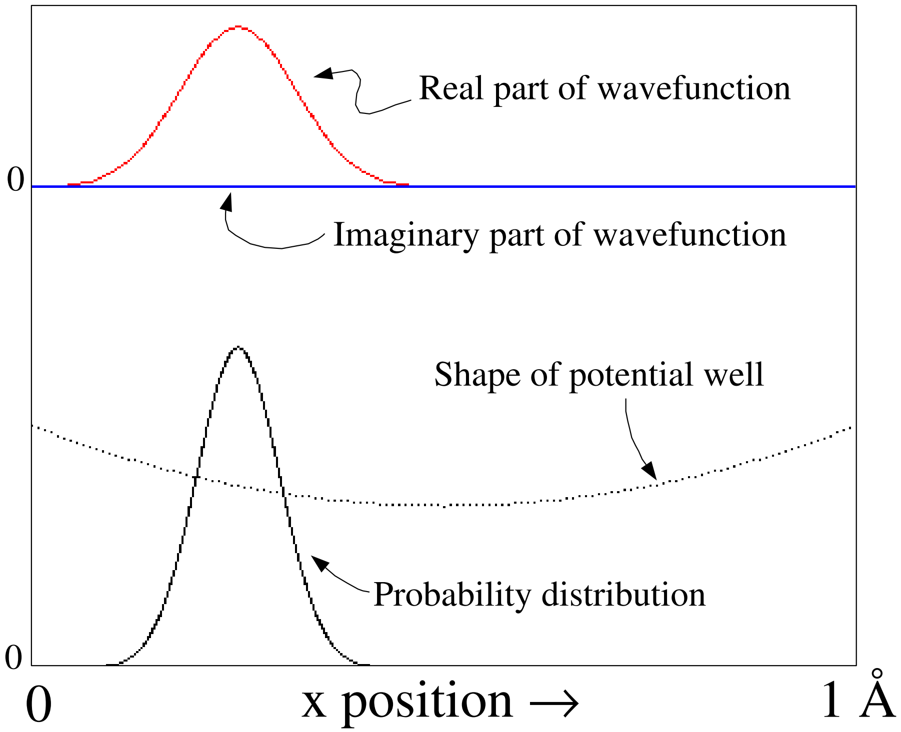
Output:

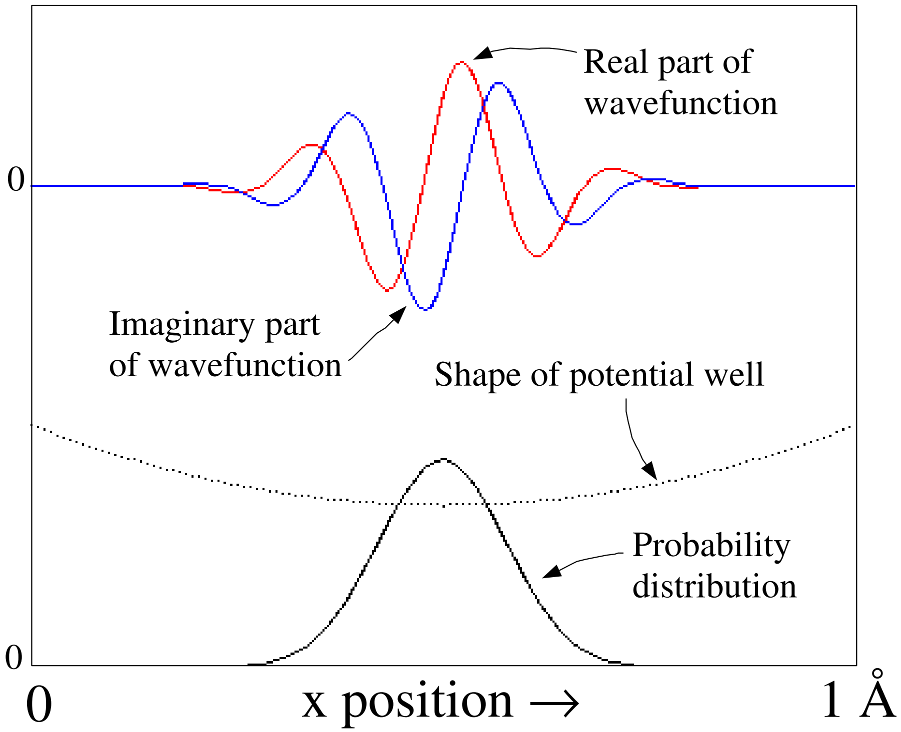
(printword *val*)
(println)

Reversible algorithms

- Sorting, arithmetic, depth-first & breadth-first search:
  - Straightforward, no space or time increase.
- Interesting case: All-pairs shortest path graph problem.
  - Floyd-Warshall: Time $\boldsymbol{\Theta}(n^3)$, space $\boldsymbol{\Theta}(n^2)$.
  - Direct reversible: Space $\boldsymbol{\Theta}(n^3)$.
  - Alt. reversible: Time $\boldsymbol{\Theta}(n^3 \log n)$, space $\boldsymbol{\Theta}(n^2 \log n)$.
  - Best reversible algs. may have different structure!
- Physical simulations (Margolus, D'Souza, …)
  - No overhead, good stability.
  - Reflect underlying reversibility of physics.
  - Example: Schrödinger wave simulation.

Reversible Schrödinger simulation



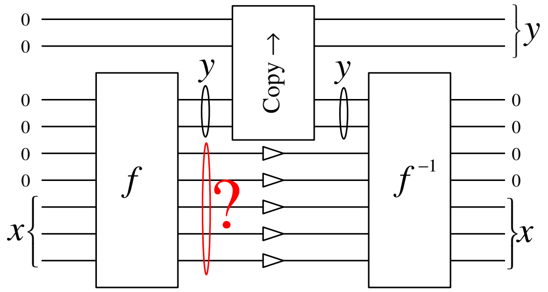


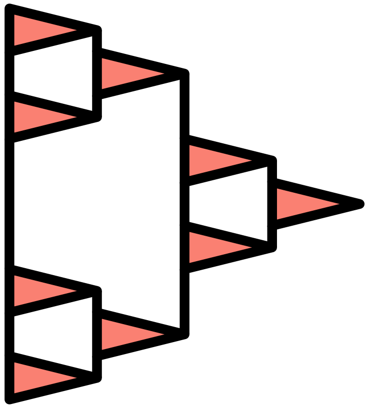
R source for Schrödinger program:

```
(defword epsilon 203667001) ; hbar*dt/m*dx^2 = .09 rad.  (dx=7.8125e-13m, dt=5e-22s)
(defarray alphas 458243442 456664951 ... [126 others]) ;Potential well.
(defarray psiR 2072809 3044772 ... [126 others]) ; Real part of wave.
(defarray psiI 0 0 ... [126 others]) ; Imaginary part of wave.
(defsub halfstep (dest src) ; Updates one wave based on the other.
  (let (e <- epsilon)
    (for i = 0 to 127
      (let (d <-> (dest _ i))
        (d += ((alphas _ i) */ (src _ i)))
        (d -= (e */ (src _ ((i + 1) & 127))))
        (d -= (e */ (src _ ((i - 1) & 127))))))))
(defsub printwave (wave) ; Print given wave to output.
  (for i = 0 to 127
    (printword (wave _ i)))
  (println))
(defmain schroed ;Main program.
  (for i = 1 to 1000 ;Time for electron to fall to well bottom.
    (call halfstep psiR psiI) (rcall halfstep psiI psiR) ; Update both components
    ;; Print both components.
    (call printwave psiR) (call printwave psiI)))
```

830 PISA instructions (395 data, 435 program).

Part III: Traditional reversible computing theory.





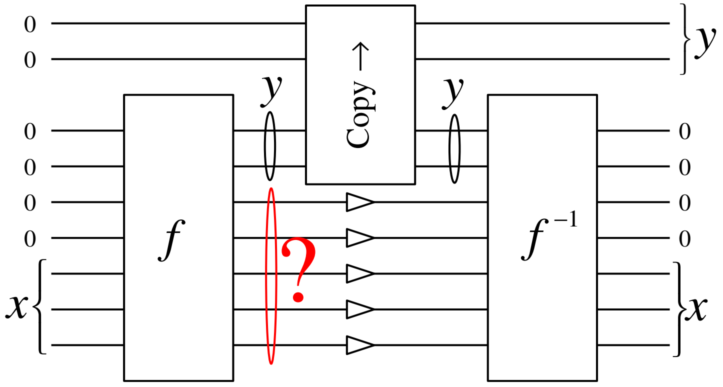
- Traditional modeling context.
- General irreversible → reversible transformations.
- Minimum overheads for reversibility.

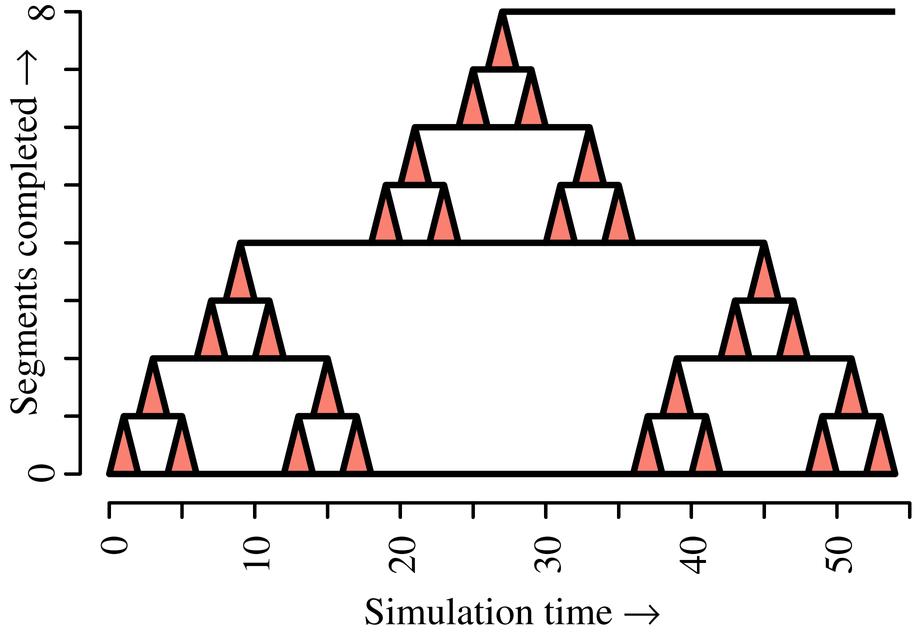
Traditional reversible computing theory—Caveats

Previous RC theory in traditional models of computation:

- Traditional focus on idealized space, time.
- Ignore energy cost & overhead of heat removal.
- Frank & Ammer: In that case, reversibility looks bad.
- Later: More *physical* models of computation.
- In more realistic models, reversible machines look good!

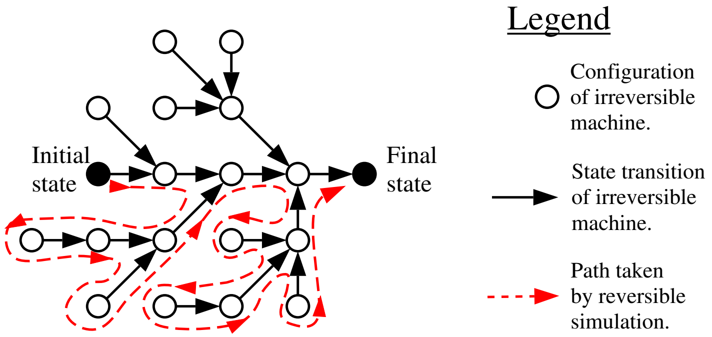
Bennett's reversible simulation techniques





- 1973: Compute w. full history, copy output, uncompute.
- 1989: Do in small segments, & recursively @ higher levels.
- Time $\mathcal{O}(\mathsf{T}^{1+\varepsilon})$, space $\mathcal{O}(\mathsf{S}\log\mathsf{T})$, w. constant $\varepsilon > 0$.
- Or: Time $\mathcal{O}(\mathsf{T})$, space $\mathcal{O}(\mathsf{S}\cdot\mathsf{T}^{\varepsilon})$.
- Levine & Sherman '90: Unrealistic to make ε *too* small.

More space-time efficient simulations?

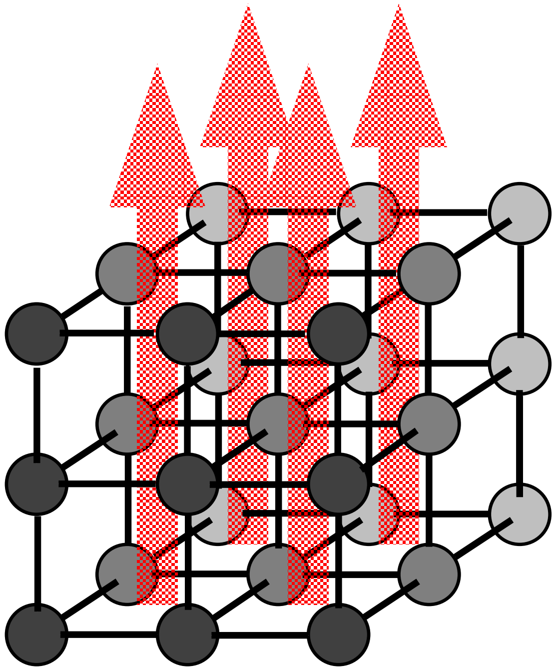


- Lange *et al.* '97: Space $\mathcal{O}(\mathsf{S})$, time $\mathcal{O}(2^{\mathsf{S}})$.
  - I helped them make result fully general.

- Open question: Possible to do space $\mathcal{O}(\mathsf{S})$, time $\mathcal{O}(\mathsf{T})$?
- Frank & Ammer '97: If so, would break if:
  - Add a (computable) reversible oracle.
  - Or, input is a reversible ROM.
  - Possibly also if 1-way functions exist.

- Remember: Assumes traditional context!

Part IV: Physical models & scaling

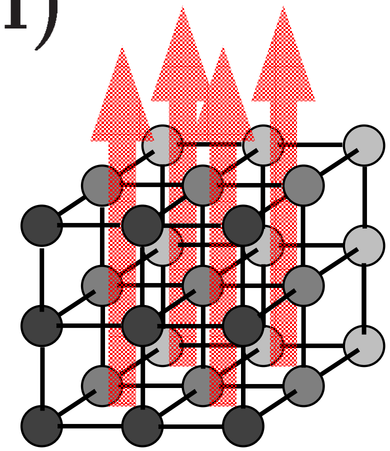


- Fundamental physical constraints on computation.
- More physically realistic models of computation.
- Scaling of physical reversible and irreversible machines.

Physical constraints on computation

Fundamental principle	**Constrained quantity**	**Symbol**	**Quantitative constraint**	**Impact on our model**
Quantum mechanics	Entropy density	ρ_{S}	$\lesssim$ 1–10 b/Å^3?	Finite state/ processor
	Entropy flux	F_{S}	$\leq \rho_{\mathrm{S}} v$	Finite info. flux
	Rate of state change	$\nu_{\perp}$	$\leq 4(E - E_0)/h$	Finite oper. frequency
Locality	Info. prop. velocity	v	$\leq c \approx 3 \times 10^8$ m/s	Mesh arch. (Vitányi ’88)
3-dimensionality of space	Connectivity		$\mathcal{O}(t^3)$	3-D mesh
Micro-reversibility, thermodynamics	Entropy change	ΔS	≥ 0 always, ≥ 1/erasure	Logical reversibility, entropy accounting
	Energy dissipation	ΔE	≥ 0 always, $\geq k_{\mathrm{B}} T \ln 2$/eras.	
Frictional effects	Entropy coefficient	k_{S}	> 0 b/Hz?	Time-prop. reversibility

My proposed model: The R3M (Reversible 3-D Mesh)



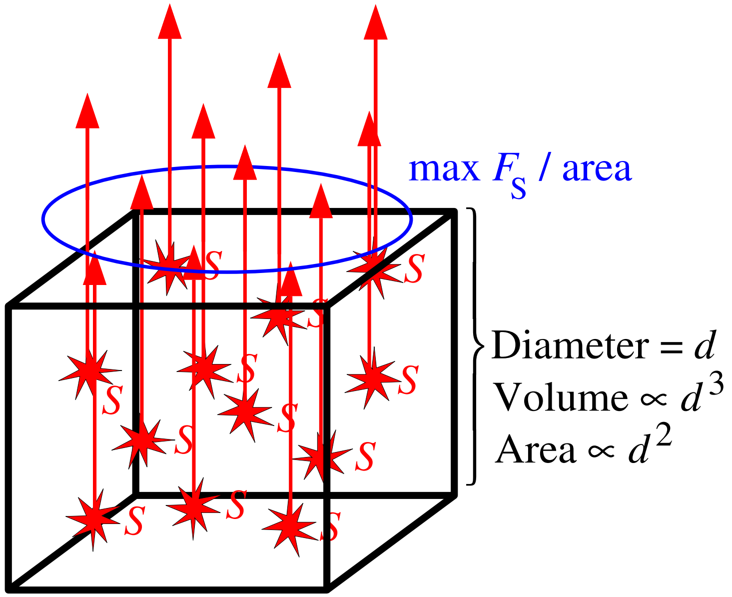
A locally-connected, 3-D mesh of finite *reversible* processors.

Assume only *time-proportionate* reversibility:

- Entropy/op proportional to speed, $S = k_{\mathrm{S}}/t$.
- We *know* how to do this with $k_{\mathrm{S}} > 0$ (*e.g.*, SCRL).
- It *still* scales better than irreversible models.
- No fundamental lower bound on k_{S} is known.

Conjecture: (*Tight Church's thesis*) The R3M is asymptotically optimal, within a constant factor.

Limit on irreversible processing rate



$$\mathcal{R}_{\mathrm{op}} \leq \frac{F_{\mathrm{S}} A}{S}$$

$\mathcal{R}_{\mathrm{op}}$ – total rate of ops per time

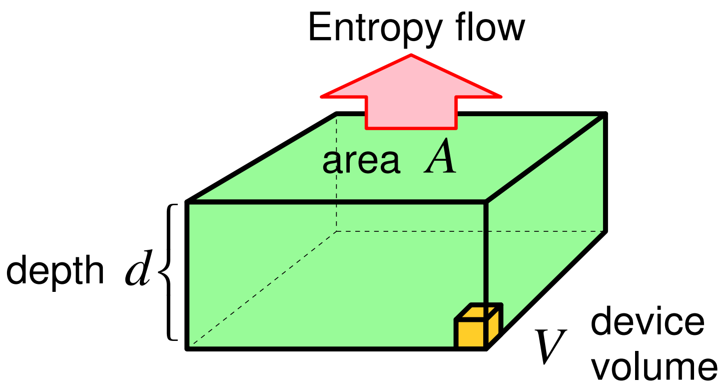
F_{S} – max. rate entropy flux per area

A – min. area of enclosing surface

S – min. entropy generated per op

- Note processing rate scales with surface area, not volume.
- The entropy S can never be less than 1 bit $(k_{\mathrm{B}} \ln 2)$ per bit irreversibly erased by an operation.
- We consider F_{S} to be a constant, assuming limits on temperature and energy density.

Per-area speed limit for reversible machines



$\mathcal{R}_{\mathrm{op}} \propto A\sqrt{d}$ with $t \propto \sqrt{d}$.

Beats best irrev. $\mathcal{R}_{\mathrm{op}} = A\frac{F_{\mathrm{S}}}{S}$

$$\text{when } d > \frac{F_{\mathrm{S}} k_{\mathrm{S}} V}{S^2}$$

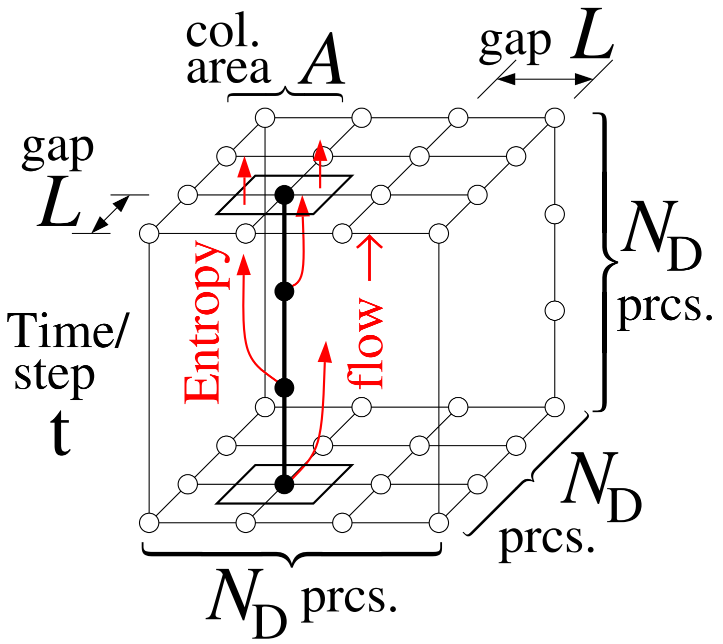
Present/near-future VLSI, w. ordinary cooling:

- $\gtrsim$ 200-layer circuits are faster if reversible.
- 1 m^3, 1 MW rev. machine: $\sim$ 24 PFLOPS (vs. 1 PFLOPS in supercond. irrev.).

Various nano-scale technologies, w. aggressive cooling:

- Rev. beats *best possible* irrev. if $d \gtrsim$ 1mm.

Speed per proc. in 3-D mesh computations



Irr.: $t \sim L \sim N_{\mathrm{D}}^{1/3}$, $A \sim N_{\mathrm{D}}^{2/3}$

$S/t \sim N_{\mathrm{D}}^{-1/3}$/prc., $\sim N_{\mathrm{D}}^{2/3}$/col.

Rev.: $t \sim L \sim N_{\mathrm{D}}^{1/4}$, $A \sim N_{\mathrm{D}}^{1/2}$

$S/t \sim N_{\mathrm{D}}^{-1/2}$/prc., $\sim N_{\mathrm{D}}^{1/2}$/col.

- Reversible advantage of $\mathcal{A}_{\mathrm{r}} = \mathbf{\Theta}(\sqrt[12]{N_{\mathrm{D}}})$
  - $10^{36}\times$ more processors to increase rev. advantage by 10×!
- $\mathcal{A}_{\mathrm{r}} = \mathbf{\Theta}(\sqrt[6]{N_{\mathrm{D}}})$ if communication distance is $\sqrt{N_{\mathrm{D}}}$.
  - Current VLSI: $\gtrsim$ \$25B machines benefit
- $\mathcal{A}_{\mathrm{r}} = \mathbf{\Theta}(\sqrt[3]{N_{\mathrm{D}}})$ if $k_{\mathrm{S}} = 0$.
- $\mathcal{A}_{\mathrm{r}} = \mathbf{\Theta}(N_{\mathrm{D}}^{\frac{1}{12}-\epsilon})$ to sim. any irrev. mesh.

Summary of major contributions:

- My work on circuits, ISAs, languages, algorithms:
  - Illustrates that reversibility is not hard to work with.
- My work on traditional reversible computing theory:
  - Helps establish overheads of full reversible computing.
- My work on scaling of physical machines:
  - Definitively establishes that substantial reversibility is best for high-performance computing in the long run.

Areas for future work

Computer science:

- More full-featured reversible programming languages.
- Good languages for programming reversible meshes.
- More serial and parallel reversible algorithms.

Computer engineering:

- Better reversible serial and mesh processor designs.
- Short-term applications of partial reversibility in energy-limited environments.

Electrical engineering:

- Lower-resistance switches for reversible ICs.
- Better resonant power supplies for reversible ICs.
- Adiabatic circuit styles with lower overheads.

Physics:

- Better characterize fundamental physical limits.
- Work on efficient nano-scale reversible devices.

Conclusion

Reversible computing *will* be a vital aspect of future computer science and computer engineering.

Let's proceed to develop the theory and technology more vigorously, and help facilitate that future!

Reversible computing:
It's here,
It's not *too* queer,[*]
Get used to it!

---

[*] It goes both ways.