FAMU-FSU College of Engineering

The Future of Computing

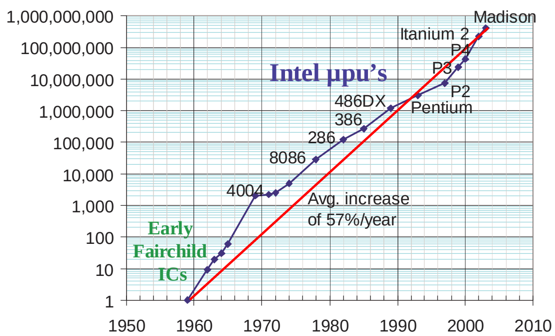
Dr. Michael P. Frank, Assistant Professor
Dept. of Electrical & Computer Eng.
FAMU-FSU College of Engineering

ECE Department Graduate Seminar
Thursday, September 2, 2004

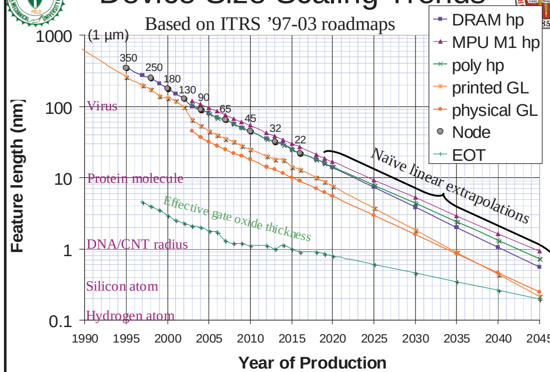
Abstract

- Throughout the 20th century, computer power has been improving at an exponentially increasing rate.
  - Some futurists have speculated about this trend continue indefinitely… perhaps towards infinity!?
- But, in the real world, it seems that no exponential trend can continue *forever*.
  - In fact, a variety of constraints from fundamental physics will prevent the present trend from continuing much longer…
    - Probably not much beyond roughly the next 1-3 decades.
- However, as technologists, we would like to keep computer power improving for as long as we can,
  - That is, to make computers as powerful as physics will allow.
    - The effort to do this reveals a number of deep connections between computing, and the laws of physics.
- In this talk, we survey some lessons that physics and the future of computing have to teach us about each other.

Moore's Law (Devices/IC)

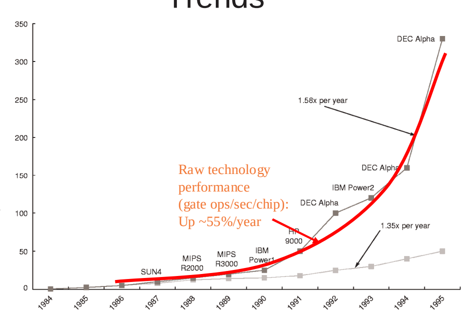


Device Size Scaling Trends

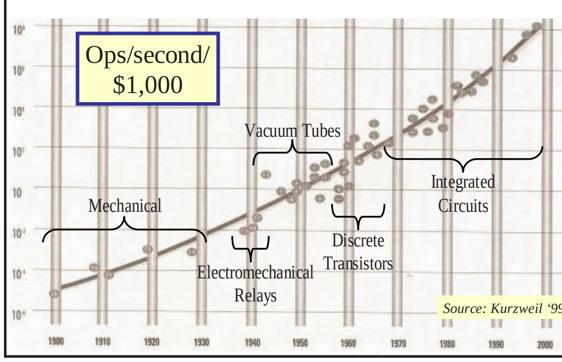


Microprocessor Performance Trends

Source: Hennessy & Patterson, *Computer Architecture: A Quantitative Approach.* Additional Performance analysis based on data from the ITRS 1999 roadmap.



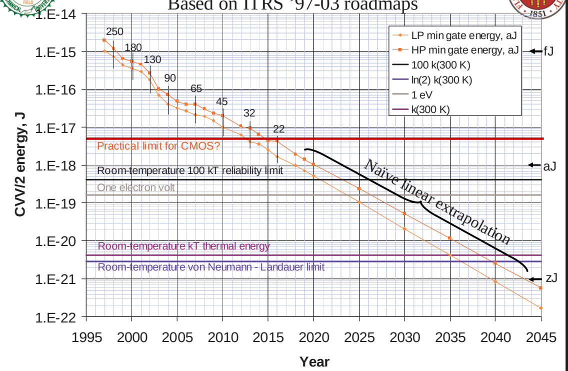
Super-Exponential Long-Term Trend



Importance of Energy

- In the real world, there is always *some* practical limit on a computer's tolerable level of power consumption…
  - Due to finite energy supplies (*e.g.*, in a battery)
  - Or, due to the difficulty and/or cost of cooling…
    - Cooling fan noise, liquid coolant hassles, fried laps, *etc.*
  - Or, due to the raw *cost* of power over time…
    - ($X/year of operating budget) ÷ (~$.10/kW-hr) =
      - at most so many W of power consumption is affordable
- And if power consumption is limited, the energy *dissipated* per logic gate operation directly limits raw (gate-level) computer performance!
  - Measured, say, in logic gate operations per unit time.
    - *Performance* (logic operations performed / time) = *Power consumption* (energy dissipated / time) × *Energy "efficiency"* (logic ops. / energy dissipated)

Trend of Min. Transistor Switching Energy



Important Energy Limits

- Near-term leakage-based limit for MOSFETs:
  - May be ~5 aJ, roughly 10× lower than today.
    - 10× faster, ~4-8 years left on the clock
- Reliability-based limit on bit energies:
  - Roughly 100 kT ≈ 400 zJ, ~100× below now.
    - 100× faster machines, ~8-15 years to go…
- Landauer limit on energy per bit erasure:
  - Roughly .7 kT ≈ 3 zJ, ~10,000× below today.
    - 10,000× faster machines, ~15-30 years left…
- No limit for reversible computing?
  - But other physical challenges come into play…

MOSFET Energy Limit

- A practical limit for transistors based on today's operating principles.
  - It's probably not an absolutely unavoidable, fundamental limit.
    - However, it is probably the biggest barrier to further transistor scaling today.
- The limit arises from the following chain of considerations:
  - We require reduced energy dissipation per logic operation.
  - → Want small $\frac{1}{2}CV^2$ logic node energy (normally dissipated when switching)
  - → Want small node capacitance C → small transistor size (also for speed)
  - → Need to lower switching voltage V, due to many factors:
    - Gate oxide breakdown, punch-through, also helps reduce CV^2.
  - → Reduced on-off ratio $R_{on/off} = I_{on}/I_{off} < e^{Vq/kT}$ (at room temperature)
    - Comes from Boltzmann (or Fermi-Dirac) distrib. of state occupancies near equil.
      - Independent of materials! (Carbon nanotubes, nanowires, molecules, *etc.*)
  - → Increased off-state current I_{off} and power $I_{off}V$, given high-performance I_{on}.
  - → Also, increased per-area leakage current due to gate oxide tunneling, *etc.*
  - → Previous two both *increase* total per-device power consumption floor
    - Adds to total energy dissipated per logic gate, per clock cycle
- Eventually, all the extra power dissipation from leakage overwhelms the power/performance reductions we gain from reducing CV^2!
  - Beyond this point, further transistor scaling hurts us, rather than helping.
    - Transistor scaling then halts, for all practical purposes!

Mitigating MOSFET Limits

- Reduce the portion of the $\frac{1}{2}CV^2$ node energy that gets *dissipated*
  - Reversible computing with adiabatic circuits does this
- Reduce parasitic capacitances that contribute to logic node's C
  - via silicon-on-insulator (SOI), low-κ field oxide materials, *etc.*
- Use high-κ gate dielectric materials →
  - Allows gate dielectrics to be thicker for a given capacitance/area
  - Reduces gate-oxide tunneling leakage current. Also:
  - Avoids gate oxide breakdown → allows higher V
    - → indirectly helps reduce off-state conduction.
- Use multi-gate structures (FinFET, surround-gate, *etc.*) to
  - reduce subthreshold slope $s = V/(\log R_{on/off})$ to approach theoretical optimum,
    - $s = T/q = (kT/q \ln 10)$/decade = 60 mV/decade
- Use multi-threshold devices & power-management architectures to turn off unused devices in inactive portions of the chip
  - The remaining leakage in the active logic is still a big problem, however…
- Lower operating temperature to increase Vq/kT and on-off ratio?
  - May lead to problems with carrier concentration, cooling costs, *etc.*
- Consider devices using non-field-effect based switching principles:
  - Y-branch, quantum-dot, spintronic, superconducting, (electro)mechanical, *etc.*

Reliability-Based Limit

- A limit on signal (bit) energy.
- Applies to any mechanism for *storing* a bit whose operation is based on the *latching* principle, namely:
  - We have some physical entity whose state (*e.g.* its location) encodes a bit.
    - *E.g.*, could be a packet of electrons, or a mechanical rod
  - If the bit is 1, the entity gets "pushed into" a state and held there by a potential energy difference (between there and not-there) of E.
    - The entity sits in there at thermal equilibrium with its environment.
  - A potential energy barrier is then raised in between the states, to "latch" the entity into place (if present).
    - A transistor is turned off, or a mechanical latching mechanism is locked down
- The Boltzmann distribution implies that $E > kT \ln N$, in order for the probability of incorrect storage to be less than $1/N$.
  - For electrons, we must use the Fermi-Dirac distribution instead…
    - But it gives virtually identical results for large N.
- When erasing a stored bit, typically we would dissipate the energy E.
  - However, this limit might be avoidable via special level-matching, quasi-adiabatic erasure mechanisms, *or* non-equilibrium bit storage mechanisms.

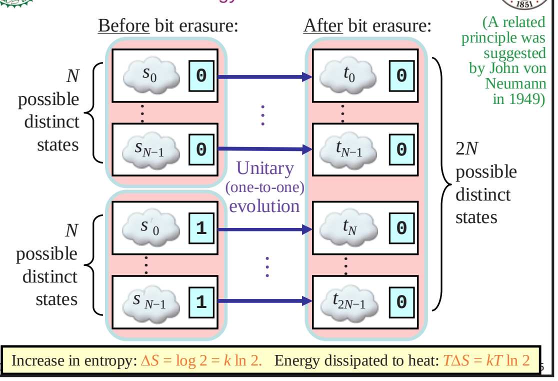
Numerical Example

- **Example:** Reliability factor of $N=10^{27}$ (*e.g.*, 1 error in a 10^9 gate processor running for ~3 years at 10 GHz)
  - The associated entropy is then: $\log 10^{27} = 27 \log 10 = 27\ k_B \ln 10 \approx 62\ k_B = 8.6\times10^{-22}$ J/K
  - Heat that must be output to a room-*T* (300 K) environment: k_B (300 K) $\ln 10^{27} = 2.6\times10^{-19}$ J (or 260 zJ, or 1.6 eV)
    - Sounds small, but…
  - If each gate dumped this energy @ a frequency of 10 GHz,
    - the total power dissipated by an entire 10^9-gate processor is 26 W.
    - Could have at most 4 such processors within a 100 W power budget!
  - Maximum performance: 4×10^{20} gate-cyles/sec.
    - or 4 PFLOPS, if processors require ~100,000 logic ops on average to carry out 1 standard (double-precision) floating-point op
      - a fairly typical figure for today's floating-point units
    - Typical COTS microprocessors today have ~100× additional overhead,
      - Leading to 40 TFLOPS max performance if using these same architectures
        - » A 40-TFLOP supercomputer (e.g. Red Storm) burns ~500 kW today
        - » Only 5,000× above the reliability-based limit!

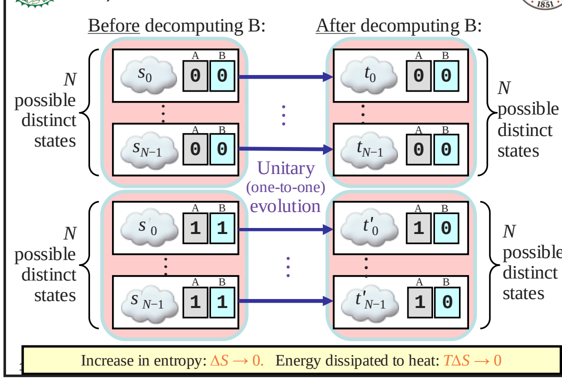
Von Neumann – Landauer (VNL) bound for bit erasure

- von Neumann-Landauer (VNL) bound for bit erasure:
  - "Oblivious" erasure/overwriting of a known logical bit moves the information it previously contained to the environment → It becomes entropy.
    - Leads to fundamental limit of $kT \ln 2$ for oblivious erasure.
  - Could *only* possibly be avoidable through *reversible computing*.
    - It "decomputes" unwanted bits, rather than obliviously erasing them!
      - Enables the signal energy to be mostly recycled, rather than dissipated.

Rolf Landauer's principle (IBM Research, 1961): The minimum energy cost of oblivious bit erasure



Non-oblivious "erasure" (by *decomputing* known bits) avoids the von Neumann–Landauer bound



Reversible Computing

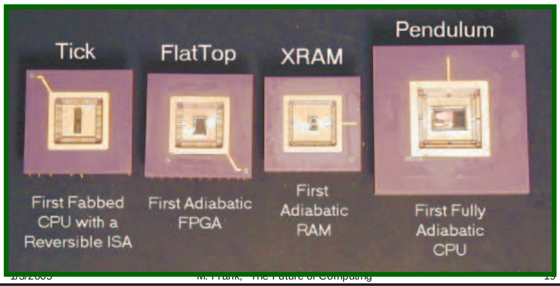
- A *reversible* digital logic operation is:
  - Any operation that performs an invertible (one-to-one) transformation of the device's local digital state space.
    - Or at least, of that subset of states that are actually used in a design.
- Landauer's principle only limits the energy dissipation of ordinary *irreversible* (many-to-one) logic operations.
  - Reversible logic operations can dissipate much less energy,
    - Since they can be implemented in a thermodynamically reversible way.
- In 1973, Charles Bennett (IBM Research) showed how any desired computation can in fact be performed using *only* reversible operations (with essentially no bit erasure).
  - This opened up the possibility of a vastly more energy-efficient alternative paradigm for digital computation.
- After 30 years of (sporadic) research, this idea is finally approaching the realm of practical implementability…
  - Making it happen is the goal of the RevComp project.

Adiabatic Circuits

- Reversible logic can be implemented today using fairly ordinary voltage-coded CMOS VLSI circuits.
  - With a few changes to the logic-gate/circuit architecture.
- We avoid dissipating most of the circuit node energy when switching, by transferring charges in a nearly *adiabatic* (literally, "without flow of heat") fashion.
  - *I.e.*, asymptotically thermodynamically reversible.
    - In the limit, as various low-level technology parameters are scaled.
- There are many designs for purported "adiabatic" circuits in the literature, but most of them contain fatal flaws and are not truly adiabatic.
  - Many past designers are unaware of (or accidentally failed to meet) all the requirements for true thermodynamic reversibility.

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.



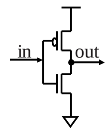
Conventional Logic is Irreversible

Even a simple NOT gate, as it's traditionally implemented!

- Here's what all of today's logic gates (including NOT) do continually, *i.e.*, every time their input changes:
  - They overwrite previous output with a function of their input.
  - Performs many-to-one transformation of local digital state!
  - ∴ required to dissipate $\gtrsim kT$ on avg., by Landauer principle
  - Incurs $\frac{1}{2}CV^2$ energy dissipation when the output changes.

Example:

Static CMOS Inverter:



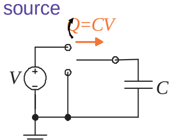
Inverter transition table:

Just before transition: in	out	*After transition:* in	out
0	0		
0	1	0	1
1	0	1	0
1	1		

Conventional *vs.* Adiabatic Charging

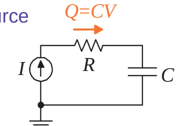
For charging a capacitive load C through a voltage swing V

- Conventional charging:
  - Constant voltage source



  - Energy dissipated: $E_{\text{diss}} = \frac{1}{2}CV^2$

- Ideal adiabatic charging:
  - Constant current source



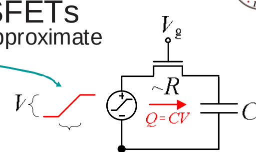
  - Energy dissipated:

$$E_{\text{diss}} = I^2Rt = \frac{Q^2R}{t} = CV^2\frac{RC}{t}$$

Note: Adiabatic beats conventional by advantage factor $A = t/2RC$.

Adiabatic Switching with MOSFETs

- Use a voltage ramp to approximate an ideal current source.
- Switch *conditionally*, if MOSFET gate voltage $V_g > V+V_T$ during ramp.



- Can discharge the load later using a similar ramp.
  - Either through the same path, or a different path.

$t \gg RC \Rightarrow E_{\text{diss}} \to CV^2\frac{RC}{t}$

$t \ll RC \Rightarrow E_{\text{diss}} \to \frac{1}{2}CV^2$

Exact formula: $E_{\text{diss}} = s\left[1+s\left(e^{-1/s}-1\right)\right]CV^2$ given *speed fraction* $s := RC/t$

Athas '96, Tzartzanis '98

Requirements for True Adiabatic Logic in Voltage-coded, FET-based circuits

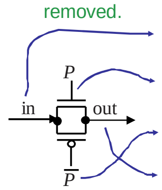
- Avoid passing current through diodes.
  - Crossing the "diode drop" leads to irreducible dissipation.
- Follow a "dry switching" discipline (in the relay lingo):
  - Never turn on a transistor when $V_{DS} \neq 0$.
  - Never turn off a transistor when $I_{DS} \neq 0$.

Important but often neglected!

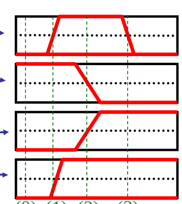
- Together these rules imply:
  - The logic design must be logically reversible
    - There is no way to erase information under these rules!
  - Transitions must be driven by a quasi-trapezoidal waveform
    - It must be generated resonantly, with high Q
- Of course, leakage power must also be kept manageable.
  - Because of this, the optimal design point will not necessarily use the smallest devices that can ever be manufactured!
    - Since the smallest devices may have insoluble problems with leakage.

A Simple Reversible CMOS Latch

- Uses a single standard CMOS *transmission gate* (T-gate).
- Sequence of operation:
  - (0) input level initially tied to latch 'contents' (output);
  - (1) input changes gradually → output follows closely;
  - (2) latch closes, charge is stored dynamically (node floats);
  - (3) afterwards, the input signal can be removed.



"Reversible latch"



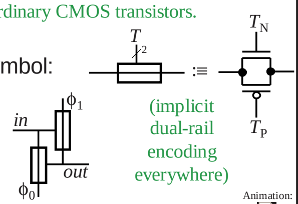
Before input: in	out	*Input arrived:* in	out	*Input removed:* in	out
0	0	0	0	0	0
		1	1	0	1

- Later, we can reversibly "unlatch" the data with an exactly time-reversed sequence of steps.

2LAL: 2-level Adiabatic Logic

A pipelined fully-adiabatic logic invented at UF (Spring 2000), implementable using ordinary CMOS transistors.

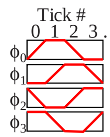
- Use simplified T-gate symbol:
- Basic buffer element:
  - cross-coupled T-gates:
    - need 8 transistors to buffer 1 dual-rail signal



(implicit dual-rail encoding everywhere)

Animation:

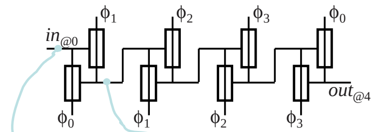
- Only 4 timing signals $\phi_{0\text{-}3}$ are needed. Only 4 ticks per cycle:
  - ϕ_i rises during ticks $t \equiv i \pmod 4$
  - ϕ_i falls during ticks $t \equiv i+2 \pmod 4$



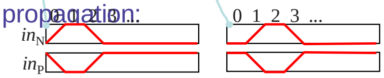
2LAL Shift Register Structure

Animation:

- 1-tick delay per logic stage:

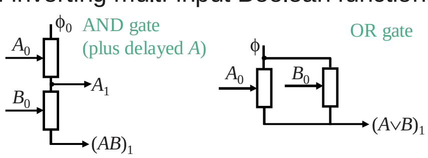


- Logic pulse timing and signal propagation:

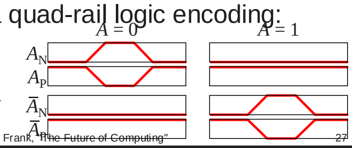


More Complex Logic Functions

- Non-inverting multi-input Boolean functions:



- One way to do inverting functions in pipelined logic is to use a quad-rail logic encoding:
  - To invert, just swap the rails!
    - Zero-transistor "inverters."

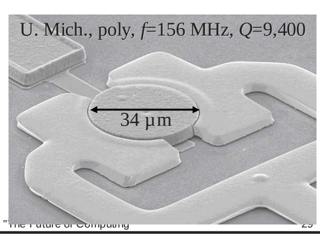


The Power Supply Problem

- In adiabatics, the factor of reduction in energy dissipated per switching event is limited to (at most) the Q factor of the clock/power supply.
  - $Q_{\text{overall}} = (Q_{\text{logic}}^{-1} + Q_{\text{supply}}^{-1})^{-1}$
- Electronic resonator designs typically have low Q factors, due to considerations such as:
  - Energy overhead of switching a clamping power MOSFET to limit the voltage swing of a sinusoidal LC oscillator.
  - Low coil count, substrate coupling in integrated inductors.
  - Unfavorable scaling of inductor Q with frequency.
- Our proposed solution:
  - Use electromechanical resonators instead!

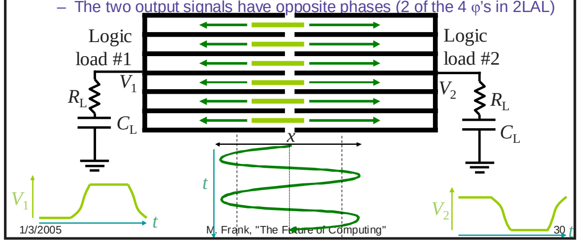
MEMS (& NEMS) Resonators

- State of the art of technology demonstrated in lab:
  - Frequencies up to the 100s of MHz, even GHz
  - Q's >10,000 in vacuum, several thousand even in air!
- An important emerging technology being explored for use in RF filters, *etc.*, in communications SoCs, *e.g.* for cellphones.



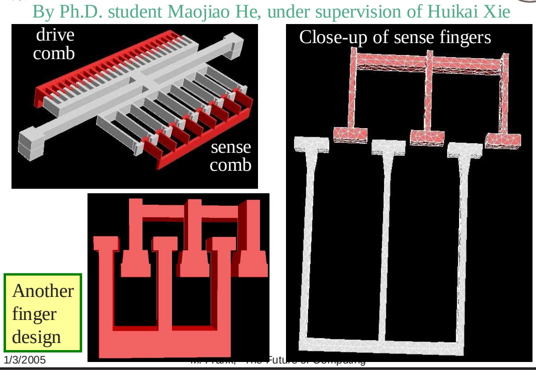
Original Concept

- Imagine a set of charged plates whose horizontal position oscillates between two sets of interdigitated fixed plates.
  - Structure forms a variable capacitor and voltage divider with the load.
- Capacitance changes substantially only when crossing border.
  - Produces nearly flat-topped (quasi-trapezoidal) output waveforms.
  - The two output signals have opposite phases (2 of the 4 φ's in 2LAL)



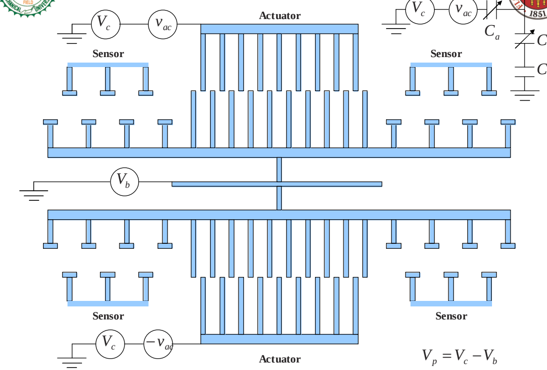
Early Resonator Designs

By Ph.D. student Maojiao He, under supervision of Huikai Xie



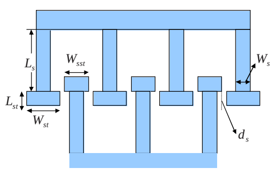
UF CONFIDENTIAL – PATENT PENDING

Resonator Schematic



UF CONFIDENTIAL – PATENT PENDING

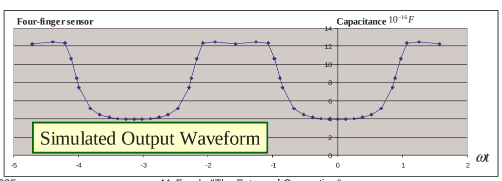
Sensor Design



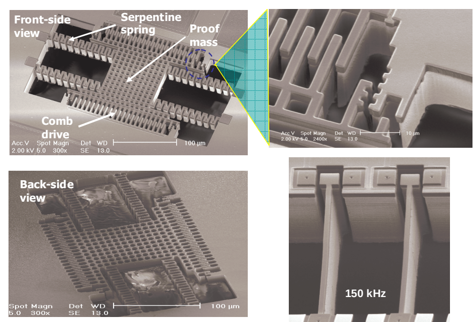
$d_s = d \quad W_s = \lambda$
$L_{st} = \lambda \quad X = 8L_{st}$
$W_{st} = W_s + 4d$
$W_{sst} = W_s + 2d$
$L_s >> d \; (L_s = 20d)$

$4C_{sf} \approx 8 \times 10^{-16}\ F$

(Early design w. thin fingers)



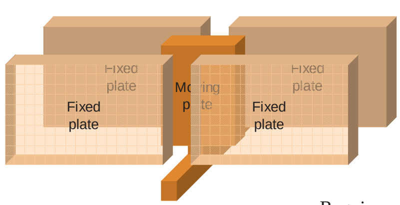
DRIE CMOS-MEMS Resonators



Resonators

UF CONFIDENTIAL – PATENT PENDING

New Comb Finger Shape V

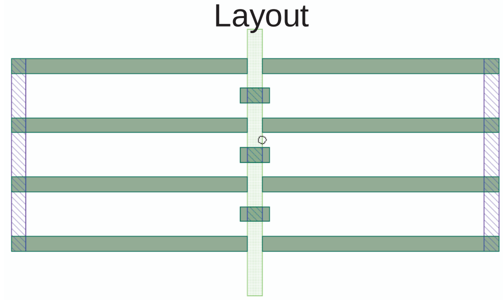


In this design, the plates are attached directly to a supprt arm which extends in the y direction instead of x. This arm can be the flexure, or it can be attached to a surrounding frame anchored to a flexure. Note that in the initial position, at all points, we only need etch from top and/or bottom, with no undercuts. Also, the flexure can be single-crystal Si.

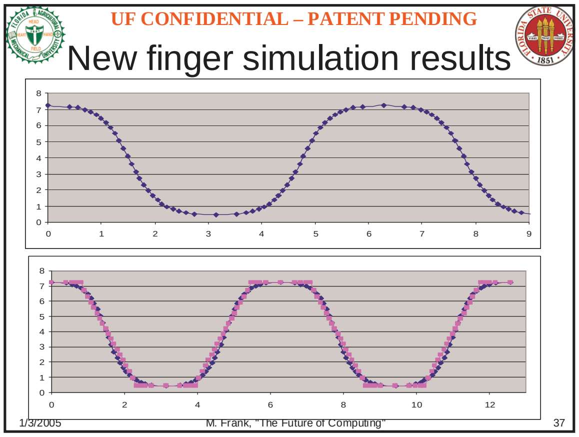
Requires accurate, variable-depth backside etch (not presently available).

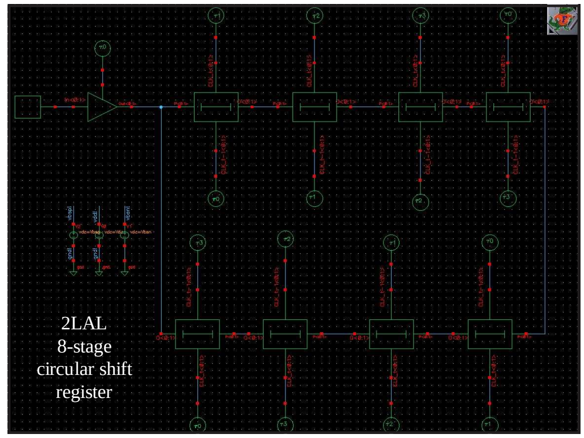
UF CONFIDENTIAL – PATENT PENDING

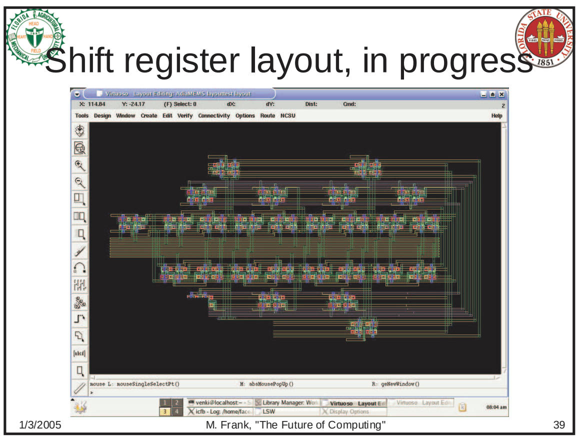
New finger: One Candidate Layout

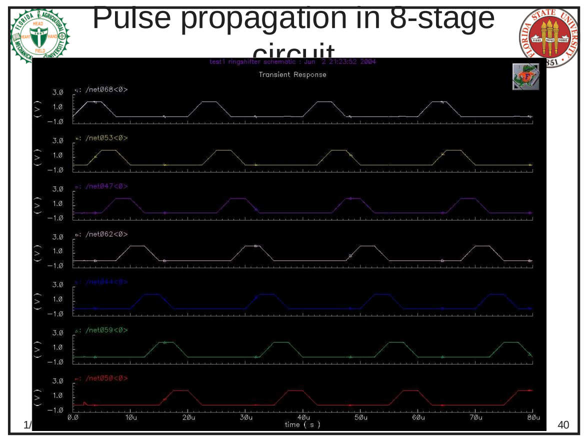


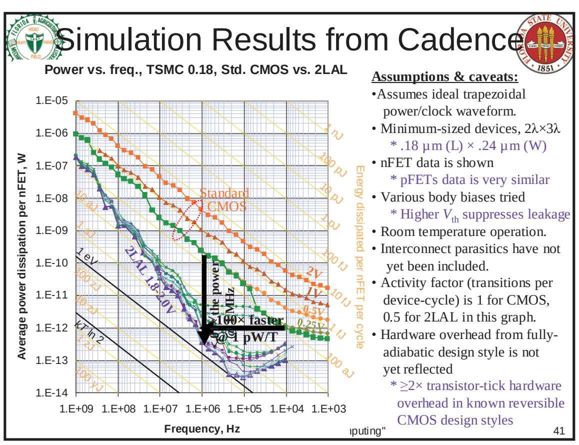
Coventor

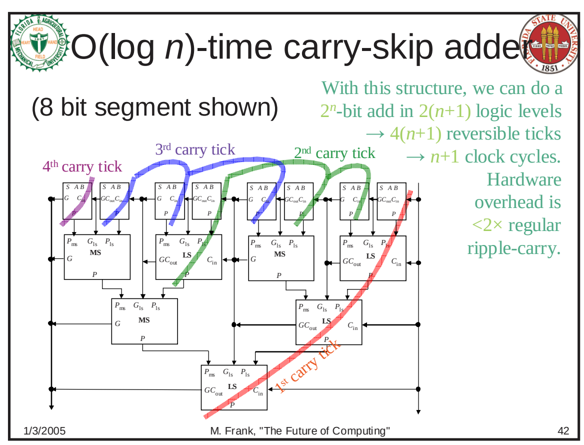

UF CONFIDENTIAL – PATENT PENDING
New finger simulation results
1/3/2005
M. Frank, "The Future of Computing"
37


2LAL
8-stage
circular shift
register

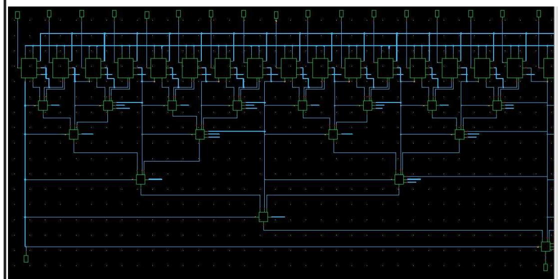

Shift register layout, in progress
1/3/2005
M. Frank, "The Future of Computing"
39


Pulse propagation in 8-stage circuit
Transient Response
40

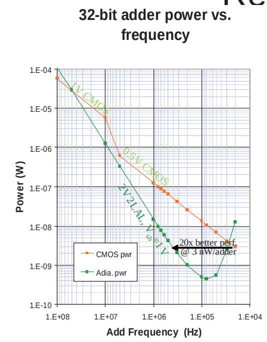

Simulation Results from Cadence
Power vs. freq., TSMC 0.18, Std. CMOS vs. 2LAL
Average power dissipation per nFET, W
Frequency, Hz
Energy dissipated per nFET per cycle
Standard CMOS
2LAL 1.8-2.0V
1 eV
kT ln 2
the power
100× faster
@ 1 pW/T
Assumptions & caveats:
•Assumes ideal trapezoidal power/clock waveform.
• Minimum-sized devices, 2λ×3λ
* .18 µm (L) × .24 µm (W)
• nFET data is shown
* pFETs data is very similar
• Various body biases tried
* Higher V_{th} suppresses leakage
• Room temperature operation.
• Interconnect parasitics have not yet been included.
• Activity factor (transitions per device-cycle) is 1 for CMOS, 0.5 for 2LAL in this graph.
• Hardware overhead from fully-adiabatic design style is not yet reflected
* ≥2× transistor-tick hardware overhead in known reversible CMOS design styles
41

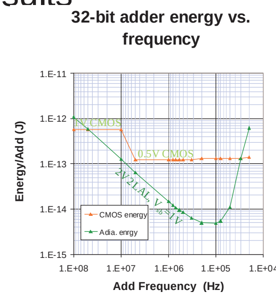

O(log n)-time carry-skip adder
(8 bit segment shown)
With this structure, we can do a 2^n-bit add in $2(n+1)$ logic levels
→ $4(n+1)$ reversible ticks
→ $n+1$ clock cycles.
Hardware overhead is <2× regular ripple-carry.
4th carry tick
3rd carry tick
2nd carry tick
1st carry tick
1/3/2005
M. Frank, "The Future of Computing"
42

Adder Schematic – High 16 Bits



32-bit Adder Simulation Results

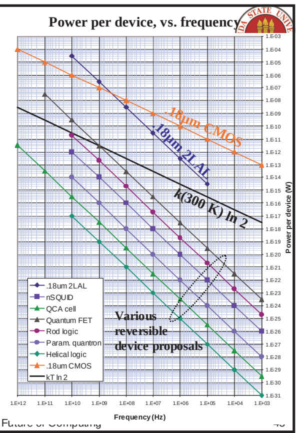




(All results normalized to a throughput level of 1 add/cycle)

Plenty of Room for Device Improvement

- Recall, irreversible device technology has at most ~3-4 orders of magnitude of power-performance improvements remaining.
  - And then, the firm $kT \ln 2$ limit is encountered.
- But, a wide variety of proposed reversible device technologies have been analyzed by physicists.
  - With theoretical power-performance up to 10-12 orders of magnitude better than today's CMOS!
    - Ultimate limits are unclear.

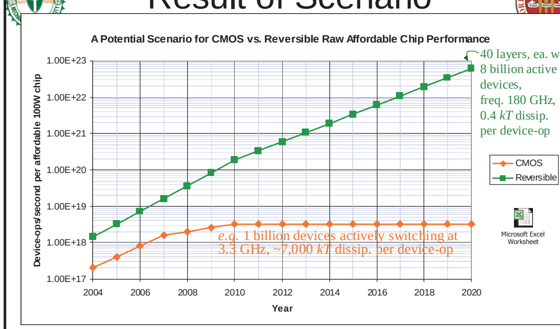


A Potential Scaling Scenario for Reversible Computing Technology

Make same assumptions as previously, except:

- Assume *energy coefficient* (energy diss. / freq.) of reversible technology continues declining at historical rate of 16× / 3 years, through 2020.
  - For adiabatic CMOS, $c_E = CV^2RC = C^2V^2R$.
    - This has been going as $\sim\ell^4$ under constant-field scaling.
  - But, requires new devices after CMOS scaling stops.
    - However, many candidates are waiting in the wings…
- Assume number of affordable *layers* of active circuitry per chip (or per package, *e.g.*, stacked dies) doubles every 3 years, through 2020.
  - Competitive pressures will tend to ensure this will happen, esp. if device-size scaling stops, as assumed.

Result of Scenario



Note that by 2020, there could be a factor of 20,000× difference in *raw* performance per 100W package. (*E.g.*, a 100× overhead factor from reversible design could be absorbed while still showing a 200× boost in performance!)

Quantum Computing

- An even more radical computing paradigm than reversible computing…
  - Not only reversible, but quantum-coherent!
- Harnesses some of the "weird power" of quantum mechanics to take "shortcuts" to solving certain problems.
  - Offers exponential speedups in some cases!
- Very difficult to physically implement…
  - Only 7-bit quantum computers have been built so far.
    - That's total bits of state, not bits per word of data!

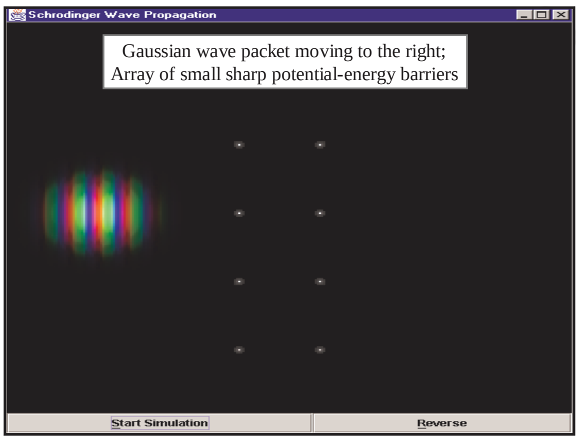
Quantum Mechanics Primer

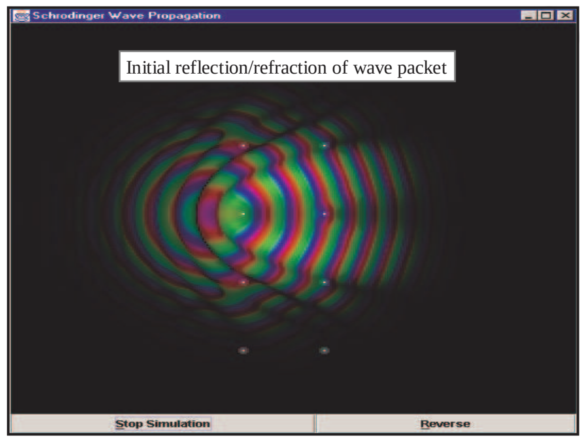
- If S is a maximal set of distinct states of a physical system,
  - Then the *quantum states* of that system are the functions $\Psi: S \to \mathbb{C}$ (complex-valued "amplitudes").
    - *I.e.*, vectors expressible as a list of $|S|$ complex numbers.
    - Vectors are normalized to a geometric length of 1.
    - $|\Psi(s)|^2$ is the *probability* of the *basis* state $s \in S$.
  - The Ψ are called *wavefunctions* or state vectors.
    - They are usually continuous, over topological spaces S.
    - Their time-evolution is continuous and obeys a differential equation which can be considered to be a wave equation.
- Wavefunctions Ψ evolve over time according to:

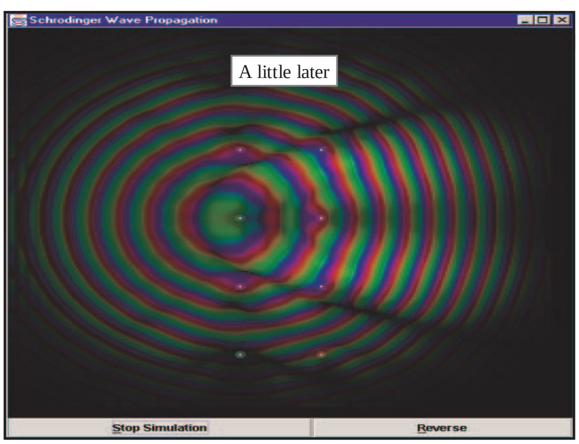
  $\Psi(t) = U(t)\Psi(0)$ with $U(t) = e^{iHt}$.
    - $U(t)$ is the unitary *time evolution operator*,
    - H is a hermitian operator - represents Hamiltonian energy

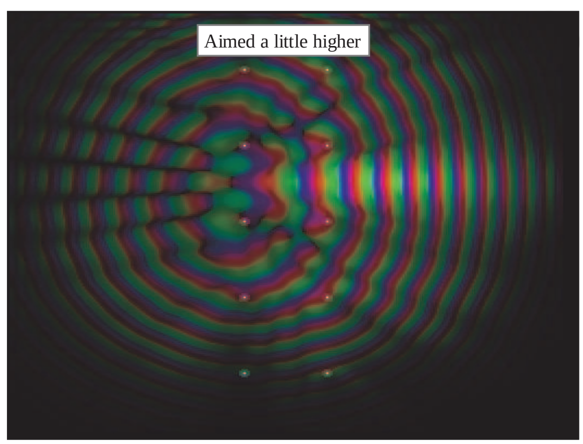
Some Features of QM

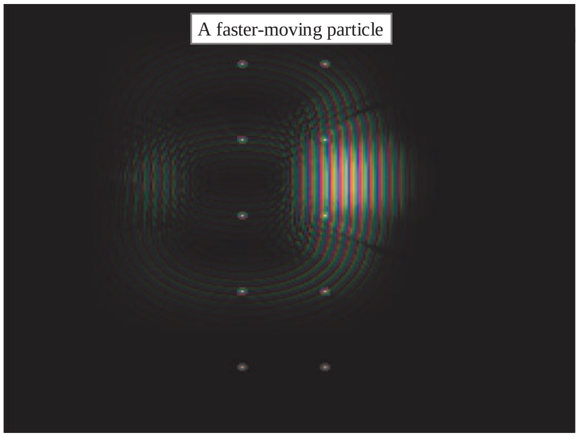
- Computing the precise behavior of a system generally requires considering its *entire* wavefunction Ψ.
  - Randomly sampling possible basis states is *not* sufficient!
- Many basis states may have nonzero values in the wavefunction simultaneously.
  - This leads to "Many Universes" picture of physics.
- But probability mass always flows locally in configuration space.
  - Local "peaks" in the wavefunction may split apart into smaller peaks, and later re-merge back together.
- When this happens, interference patterns may appear.
  - Specific basis states may end up more or less probable, depending on the relative phase of the incoming waves.











Quantum Computing

- In quantum computing, the basis states *S* are simply states of a digital computer…
  - Bit strings $b_0 b_1 \ldots b_{n-1}$ for an *n*-bit computer.
- The state of the quantum computer assigns an amplitude to each digital state.
  - Many different states may simultaneously have non-zero amplitudes!
- Logic is performed using unitary operators *U* applied to just 1 or 2 bits at a time.
  - This is sufficient to generate all unitary transformations! (2-bit gates are "universal")

Why Quantum Computing?

- It is exponentially more time-efficient than *any known* classical computing scheme at solving *certain* problems:
  - Factoring, discrete logarithms, related problems
  - Simulating quantum physical systems accurately
    - This application was the original motivation for quantum computing research first suggested by famous physicist Richard Feynman in the early 80's.!
- However, it's never really been proven that a fast classical algorithm for any of these problems is impossible…
  - If you want to win a sure-fire Nobel prize…
    - Find a *polynomial-time* algorithm for accurately simulating quantum computers on classical ones!
    - Or, prove rigorously that it can't be done!

Status of Quantum Computing

- Theoretical & experimental progress is being made, but slowly.
  - There are many areas where much progress is still needed.
- Physical implementations of very small (*e.g.*, 7-bit) quantum computers have been tested, and they work as predicted.
  - However, scaling them to large sizes is very difficult!
- There are no known *fundamental* theoretical barriers to large-scale quantum computing.
  - Guess: It may be a real technology in ~20 yrs. or so.

Gates without Superposition

- All classical input-consuming reversible gates can be represented as unitary transformations!
- *E.g.*, input-consuming NOT gate (like an inverter)

in	out
0	1
1	0

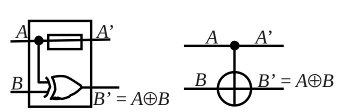
$$|\mathbf{0}\rangle :\equiv \begin{array}{c} \begin{pmatrix}1\\0\end{pmatrix} \end{array}\begin{array}{c}\mathbf{0}\\\mathbf{1}\end{array} \qquad |\mathbf{1}\rangle :\equiv \begin{pmatrix}0\\1\end{pmatrix}\begin{array}{c}\mathbf{0}\\\mathbf{1}\end{array}$$

$$N :\equiv \begin{array}{cc} \begin{array}{cc}\mathbf{0} & \mathbf{1}\end{array} & \\ \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix} & \begin{array}{c}\mathbf{0}\\\mathbf{1}\end{array}\end{array}$$

$$N|\mathbf{0}\rangle = |\mathbf{1}\rangle \qquad N|\mathbf{1}\rangle = |\mathbf{0}\rangle$$

Controlled-NOT

- A.k.a. CNOT (or input-consuming XOR)



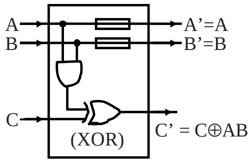
A	B	A'	B'
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

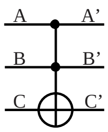
$$X :\equiv \begin{array}{cc} \begin{array}{cccc}\mathbf{00} & \mathbf{01} & \mathbf{10} & \mathbf{11}\end{array} & \\ \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix} & \begin{array}{c}\mathbf{00}\\\mathbf{01}\\\mathbf{10}\\\mathbf{11}\end{array}\end{array}$$

Example:

$$X\left|\overset{A\,B}{\mathbf{10}}\right\rangle = \left|\overset{A\,B}{\mathbf{11}}\right\rangle$$

Toffoli Gate (CCNOT)





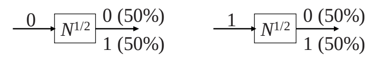
A B C	A' B' C'
0 0 0	0 0 0
0 0 1	0 0 1
0 1 0	0 1 0
0 1 1	0 1 1
1 0 0	1 0 0
1 0 1	1 0 1
1 1 0	1 1 1
1 1 1	1 1 0

$$X := \begin{array}{c} \begin{array}{cccccccc} \mathtt{000} & \mathtt{001} & \mathtt{010} & \mathtt{011} & \mathtt{100} & \mathtt{101} & \mathtt{110} & \mathtt{111} \end{array} \\ \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1&0 \end{bmatrix} \end{array} \begin{array}{c} \\ \mathtt{000}\\ \mathtt{001}\\ \mathtt{010}\\ \mathtt{011}\\ \mathtt{100}\\ \mathtt{101}\\ \mathtt{110}\\ \mathtt{111} \end{array}$$

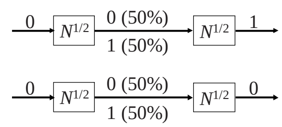
Now, what happens if the unitary matrix elements are *not* always 0 or 1?

The Square Root of NOT

- If you put in either basis state (0 or 1) you get a state that appears *random* if measured…



- But if you feed the output back into another $N^{1/2}$ *without measuring it*, you get the *inverse* of the original value!
- "How is that possible?"



NOT$^{1/2}$: Unitary implementation

$$\sqrt{N} := \begin{array}{c} \begin{array}{cc} \mathbf{0} & \mathbf{1} \end{array} \\ \begin{bmatrix} \frac{1+i}{2} & \frac{1-i}{2} \\ \frac{1-i}{2} & \frac{1+i}{2} \end{bmatrix} \end{array} \begin{array}{c} \\ \mathbf{0} \\ \mathbf{1} \end{array}$$

$$\left(\sqrt{N}\right)^2 = \begin{array}{c} \begin{array}{cc} \mathbf{0} & \mathbf{1} \end{array} \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ \mathbf{0} \\ \mathbf{1} \end{array} = N$$

$$\sqrt{N}|\mathbf{0}\rangle = \sqrt{N}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1+i}{2} \\ \frac{i-i}{2} \end{bmatrix} = \underbrace{\frac{1+i}{2}}_{\text{Prob. }1/2}|\mathbf{0}\rangle + \underbrace{\frac{1-i}{2}}_{\text{Prob. }1/2}|\mathbf{1}\rangle$$

The Hadamard Transform

- A randomizing "square root of identity" gate.
- Used frequently in quantum logic networks.

$$H := \begin{array}{c} \begin{array}{cc} \mathbf{0} & \mathbf{1} \end{array} \\ \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \end{array} \begin{array}{c} \\ \mathbf{0} \\ \mathbf{1} \end{array} \qquad H^2 = I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

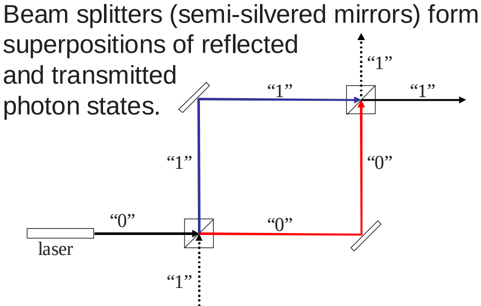
Another NOT$^{1/2}$

- This one negates the phase of the state if the input state was $|\mathbf{0}\rangle$.

$$(N')^{1/2} := \begin{array}{c} \begin{array}{cc} \mathbf{0} & \mathbf{1} \end{array} \\ \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \end{array} \begin{array}{c} \\ \mathbf{0} \\ \mathbf{1} \end{array} \qquad N' = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

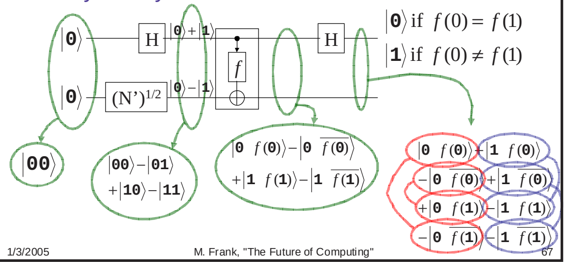
Optical Implementation of $N^{1/2}$

- Beam splitters (semi-silvered mirrors) form superpositions of reflected and transmitted photon states.



Deutsch's Problem

- Given a black-box function f:{0,1}→{0,1},
  - Determine whether $f(0)=f(1)$,
  - But you only have time to call f *once*!



Extended Deutsch's Problem

- Given black-box f:$\{0,1\}^n \to \{0,1\}$,
  - and a guarantee that f is either *constant* or *balanced* (1 on exactly ½ of inputs)
  - Answer the question, "Which of these is it?"
  - Minimize number of calls to f.
- Classical algorithm, worst-case:
  - Order 2^n time!
    - What if the first 2^{n-1} cases examined are all 0?
      - Function could *still* be either constant or balanced.
    - Case number $2^{n-1}+1$: if 0, constant; if 1, balanced.
- Quantum algorithm is exponentially faster!
  - (Deutsch & Jozsa, 1992.)

Unstructured Search

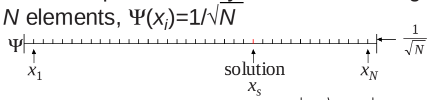
- Given a set S of N elements and a black-box function f:$S\to\{0,1\}$, find an element $x\in S$ such that $f(x)=1$, if one exists (or if not, say so).
  - Any NP problem can be cast as an unstructured search problem.
    - Not necessarily the optimal approach, however.
- Bounds on classical run-time:
  - $\Omega(N)$ expected queries in worst case (0 or 1 sol'ns):
    - Have to try $N/2$ elements on average before finding sol'n.
    - Have to try all N if there is no solution.
- If elements are length-ℓ bit strings,
  - Expected #trials is $\Omega(2^\ell)$ - exponential in ℓ. Bad!

Quantum Unstructured Search

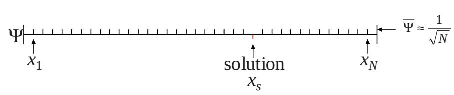
- Minimum time to solve unstructured search problem on a quantum computer is:
  - $\Omega(N^{1/2})$ queries = $(2^{\ell/2}) = (2^{1/2})^\ell$
    - Still exponential, but with a smaller base.
- The minimum # of queries can be achieved using *Grover's algorithm*.

Grover's algorithm:

- 1. Start w. amplitude evenly distributed among the N elements, $\Psi(x_i)=1/\sqrt{N}$



- 2. In each state x_i, compute $f(x_i)$: $|x_i\rangle \Rightarrow |x_i, f(x_i)\rangle$
- 3. Apply conditional phase shift of π if $f(x_i)=1$ (Negate sign of solution state.) Uncompute f.



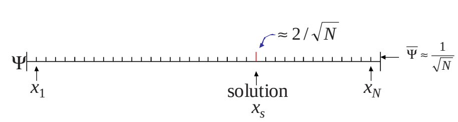
f=0 f=1

$$\begin{bmatrix} 1 & \\ & -1 \end{bmatrix}$$

Grover's algorithm, *cont.*

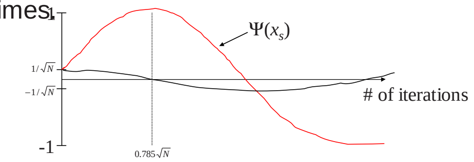
- 4. Invert all amplitudes with respect to the *average* amplitude:

$$\Psi' = \overline{\Psi} - (\Psi - \overline{\Psi}) = 2\overline{\Psi} - \Psi \qquad FRF = \begin{bmatrix} 2/N - 1 & 2/N & \cdots & 2/N \\ 2/N & 2/N - 1 & & 2/N \\ \vdots & & \ddots & \vdots \\ 2/N & 2/N & \cdots & 2/N - 1 \end{bmatrix}$$



Grover's algorithm, *cont.*

- 5. Go to step 2, and repeat 0.785 $N^{1/2}$ times.



Shor's Factoring Algorithm

- Solves the >2000-year-old problem:
  - Given a large number N, quickly find the prime factorization of N. (At least as old as Euclid!)
- No polynomial-time (as a function of n=lg N) classical algorithm for this problem is known.
  - The best known (as of 1993) was a *number field sieve* algorithm taking time $\mathbf{O}(\exp(n^{1/3}\log(n^{2/3})))$
  - However, there is also no proof that an (undis-covered) fast classical algorithm does *not* exist.
- Shor's quantum algorithm takes time $\mathbf{O}(n^2)$
  - No worse than multiplication of n-bit numbers!

Elements of Shor's Algorithm

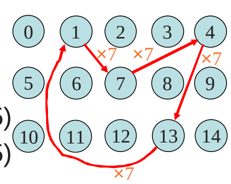
- Uses a standard reduction of factoring to another number-theory problem called the *discrete logarithm* problem.
- The discrete logarithm problem corresponds to finding the *period* of a certain periodic function defined over the integers.
- A general way to find the period of a function is to perform a *Fourier transform* on the function.
  - Shor showed how to generalize an earlier algorithm by Simon, to provide a Quantum Fourier Transform that is exponentially faster than classical ones.

Powers of numbers mod N

- Given natural numbers (non-negative integers) $N\geq 1$, $x<N$, and x, consider the sequence:
  x^0 mod N, x^1 mod N, x^2 mod N, …
  = 1, x, x^2 mod N, …
- If x and N are relatively prime, this sequence is guaranteed not to repeat until it gets back to 1.
- *Discrete logarithm* of y, base x, mod N:
  - The smallest natural number exponent k (if any) such that $x^k = y$ (mod N).
  - *I.e.*, the integer logarithm of y, base x, in modulo-N arithmetic. Example: dlog_7 13 (mod N) = ?

Discrete Log Example

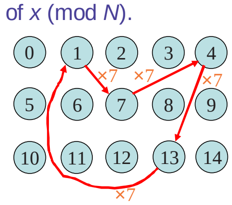
- N=15, x=7, y=13.
- x^2 = 49 = 4 (mod 15)
- x^3 = 4·7 = 28 = 13 (mod 15)
- x^4 = 13·7 = 91 = 1 (mod 15)



- So, dlog_7 13 = 3 (mod N),
  - Because 7^3 = 13 (mod N).

The *order* of x mod N

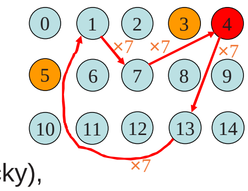
- **Problem:** Given N>0, and an $x<N$ that is relatively prime to N, what is the smallest value of k>0 such that x^k = 1 (mod N)?
  - This is called the *order* of x (mod N).
- From our previous example, the order of 7 mod N is…?



Order-finding permits Factoring

- A standard reduction of factoring N to finding orders mod N:
  - 1. Pick a random number $x < N$.
  - 2. If $\gcd(x,N)\neq 1$, return it (it's a factor).
  - 3. Compute the order of x (mod N).
    - Let $r := \min k>0: x^k \bmod N = 1$
  - 4. If $\gcd(x^{r/2}\pm 1, N) \neq 1$, return it (it's a factor).
  - 5. Repeat as needed.
- The expected number of repetitions of the loop needed to find a factor with probability > 0.5 is known to be only polynomial in the length of N.

Factoring Example

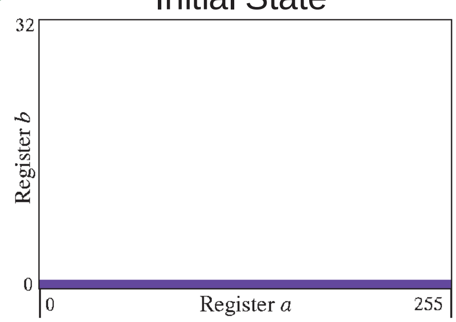


- For N=15, x=7…
- Order of x is r=4.
- $r/2 = 2$.
- $x^2 = 5$.
- In this case (we are lucky), *both* x^2+1 and x^2-1 are factors (3 and 5).
- Now, how do we compute orders efficiently?

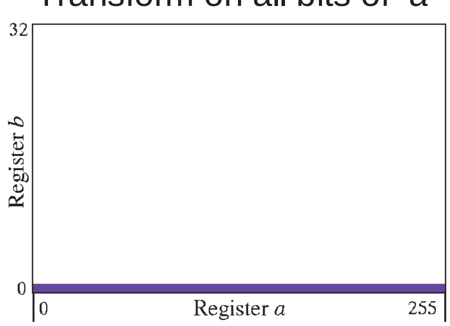
Quantum Order-Finding

- Uses 2 quantum registers (a,b)
  - $0 \leq a < q$, is the k (exponent) used in order-finding.
  - $0 \leq b < n$, is the y (x^k mod n) value
  - q is the smallest power of 2 greater than N^2.
- Algorithm:
  - 1. Initial quantum state is $|0,0\rangle$, *i.e.*, (a=0, b=0).
  - 2. Go to superposition of all possible values of a:

Initial State

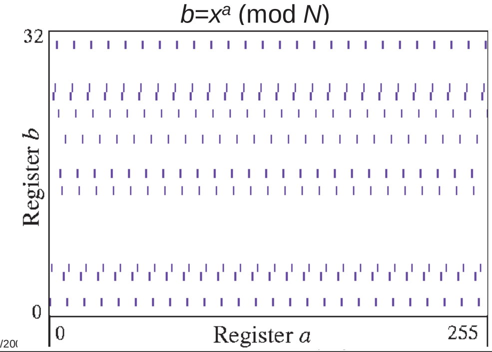


After Doing Hadamard Transform on all bits of *a*

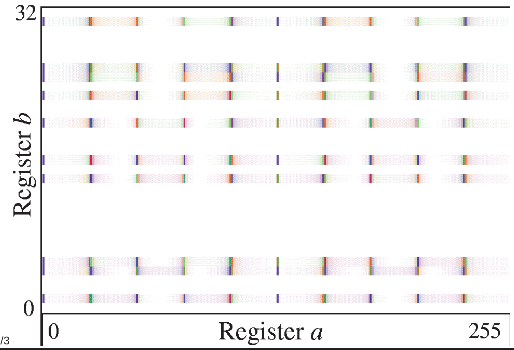


After modular exponentiation

$b=x^a \pmod N$



State After Fourier Transform



Physics as Computing

- Many physical quantities can be understood in computational terms:
  - Physical *entropy* is unknown/incompressible physical information.
  - Physical *energy* is the rate of physical quantum computation.
  - Physical *action* (energy×time) is an *amount* of computation.
    - *E.g.*, flipping a bit takes at least $h/4 = 90°$ of action.
  - Physical *temperature* is rate of computing per bit, or the "clock speed" of physical computation.
- These identities can be rigorously proven!

Physical Limits of Computing

- A computer implemented *in* a physical system can't be more computationally powerful then the underlying physical system is, itself!
  - This fact lets us derive technology-independent bounds on a computer's "power," for example, its:
    - Storage capacity
    - Total parallel processing rate
    - Serial processing rate
    - Information transmission "bandwidth"
  - Given the machine's physical characteristics, such as:
    - Physical size (diameter, volume, enclosing area)
    - Energy content
      - That is, actively-manipulated energy in its "moving parts"
    - Temperature
      - Generalized temperature of its computational degrees of freedom
    - Power consumption

Some Example Limits

- A $(10\text{ cm})^2$ tablet computer emitting 10W of power can never electromagnetically transmit/receive more than:
  - 2.2×10^{21} bps (2.2 Zb/s)
    - Independent of spectrum used, noise floor, *etc.*
  - Sounds big, but it's only 109 kb/s/nm^2!
    - Electromagnetic field not suitable for communicating between densely-packed nanoscale components at this power density…
- A digital device/signal with 1 eV of active energy can never transition between states faster than a rate of 484 THz.
  - Only ~100,000× faster than today's processors.
- "Moving parts" (*e.g.*, electrons) at a "generalized temperature" of only room temperature can't flip bits any faster than at 4.3 THz.
  - Only ~1,000× faster than today's processors.
- A computer consuming 100W of power in a room-T environment can't perform more than 3.48×10^{22} bit erasures/sec.
  - Only ~100,000× faster than today's processors.

The Ideal Digital Device?

- Has well-defined, well-separated physical states.
  - Suitable for representing bits.
- Active compute devices are *not* in an equilibrium state or quasi-static regime!
  - System evolves forward through configuration space under its own generalized momentum.
- Active particles in compute mechanism are very "hot" (generalized temp.)
  - They transition between subsequent distinct states very quickly
- Active particles are very well-isolated from surrounding structure/environment.
  - Energy is kept contained within the system, & recirculated with high efficiency.
- There are available "stationary bits" that remain stable in the long term
  - with low static power consumption – nonvolatile storage
- Fast communications available via high-speed "flying bits"
  - E.g., electronic or photonic pulses, signal energy confined to predetermined waveguides.
- There should be efficient interconversion between stationary & flying bits.
  - Signal energy nearly all recovered upon transmitting, or catching and storing, a flying bit
- Interactions should available that perform a universal set of classical ops
  - With as much gain as needed to replenish signal losses
- Should offers state transitions that are totally logically reversible
  - And that are implemented via high-*Q* ballistic, adiabatic physical transformations.
  - For avoiding the von Neumann - Landauer bound.
- Self-contained: No outside control signals need to be provided.
  - Time-independent Hamiltonian, (nearly) closed system apart from desired I/O, & power/cooling.
- For QC: Complete quantum gate set available, and state retains quantum phase coherence for many cycles.
  - Allows quantum error-correction techniques to be applied and quantum computing to occur.

What does the future hold?

- Prediction: Computers will keep getting faster & more powerful, for the next few years, at least…
- Then, one of two things will happen:
  - Computer performance will start to flatten out…
- OR:
  - Radical new devices and computing paradigms will begin to be introduced!
    - Such as reversible & quantum computing devices.
- Even then, things will probably still slow down before too many more decades go by!