Reversible Computing:
A Cross-Disciplinary Introduction

Invited talk presented Mar. 10th, 2014 to the Beyond Moore Research Challenge group at Sandia National Laboratories, Albuquerque, NM

Michael P. Frank
Associate in Engineering, Department of Electrical & Computer Engineering, FAMU-FSU College of Engineering
Outline of Talk

- Limits of traditional CMOS scaling
  - And of all irreversible digital technologies
- Brief history of reversible computing research
  - Review of some major developments
- An example reversible logic scheme: 2LAL
- RevComp’s alternate path for power scaling
  - Potential to sidestep limits of standard roadmap
  - Requires cooperation, with a new mindset, among:
    - Device physicists
    - Process engineers
    - Logic family designers / tool developers
    - Circuit/interconnect designers
    - Computer architects
Problems with continued CMOS power-performance scaling

- We have been aware these were coming for a long time!
  - E.g., *Proc. IEEE* paper by David Frank & colleagues, 1997
- Short-channel effects related to electron diffusion length
  - Low thresholds → hard to turn device off
- Even more fundamental limits on subthreshold slope of log(I)-log(V_{GS}) curve
  - ~60 mV/decade at best
  - Limits I_{on}/I_{off} for given logic swing V_{dd}

*Fig. 26.* Simulated threshold voltage versus channel length, comparing short-channel effect of double-gate FET’s (solid lines) with SOI MOSFET’s (dashed lines), where the threshold of the long-channel FET’s has been taken as zero. These values are extracted from drift diffusion simulations of the subthreshold regime of these FET’s. Inset: cross-sectional structure of a double-gated FET.
Boltzmann’s Distribution tells us that any system experiences energy fluctuations of order $\sim kT$

Thus, reliably suppressing undesired transitions requires energy barriers/differences of 10s of $kT$

- If you dissipate say $\sim 40 \ kT$ at any temperature $T$ when switching, implies $\sim 1 \ eV$ system dissip. into room-$T$ env.

Boltz. Dist. is derived by a totally general thermodynamic argument that is unaffected by the specific physical structure of the system in question

- Novel device physics cannot help (CNTs, optics, quantum)
- Fancy error correction codes cannot help
  - Total energy/reliability of encoded bit still subject to argument!
An Absolute Requirement for Ongoing Power-Performance Scaling

- Given that we must, therefore, dissipate $\sim 1 \text{ eV}$ for each (irreversible, terrestrial) bit operation,
  - We can only do at most $\sim 6 \times 10^{18}$ bit operations (6 Eops) per Joule of system energy dissipation
  - Convert to FLOPS/W with your favorite conversion

- If we hope to ever do significantly better than this, we absolutely **MUST** start learning how to:
  
  Avoid Dissipating the Entire Bit Energy When Manipulating Bits!

There literally is **no other choice**.
Brief History of Reversible Computing Research

- **Rolf Landauer, 1961**
  - Only irreversible (many-to-one) logical operations appear to require a fundamental minimum energy dissipation

- **Charles Bennett, 1973**
  - Irreversible operations are not required for universal computation

- **Fredkin & Toffoli, late ‘70s**
  - Construct reversible computations by composing reversible logic primitives
  - First proposal for an electronic reversible logic

- **Seitz et al., Kollar & Athas, etc., 1980s**
  - Earliest development of quasi-adiabatic circuits in CMOS

- **Younis & Knight & colleagues, 1990s**
  - First *fully* adiabatic sequential logic (early versions were still buggy)

  - Exploring computational complexity overheads of reversible computing

- **Frank & colleagues, 2000-2004**
  - Development & simulation of a simple *truly, fully* adiabatic logic scheme (2LAL)
One correct statement of the principle:
- On average, oblivious erasure of a known logical bit must increase total entropy by at least $k \ln 2$.
- Assuming 0 and 1 states arise equally often

Can be proven true via a trivial argument:
- Fundamental physics is believed to be reversible
  - Unitary quantum evolution is consistent with all data
- A many-to-one transformation of the logical state
  $\therefore$ implies a one-to-many transformation of the detailed physical state. $\Rightarrow$ Increased entropy
Bennett’s Insight

- **Landauer**: We can embed any irreversible (many-to-one) logical transformation into a larger reversible (one-to-one) transformation.
  - However, this in general generates extra unwanted “garbage” bits – what to do with them?

- **Bennett**: Just save the unwanted bits, reversibly copy the result, then undo the computation, de-computing the garbage.
  - Frees up space for reuse in later computation.

- Space-inefficient, but later work greatly improved on this.
Adiabatic Logic using FETs: Basic Principle

- Basic recipe to transition a logic level with negligible dissipation (“adiabatically”):
  1. Match levels between input node $I$ and storage node $S$ (with known data).
  2. Switch voltage on transistor gate $G$ to turn it on
     - Can be done using same method, in staggered fashion.
  3. Ramp voltage on input node $I$ gradually to new logic level over some time $t \gg RC$ ($R = \text{eff. on-resistance}$)
     - Dissipation $\sim CV^2RC/t$ can be as small as desired
     - Series/parallel combinations of devices can do logic
     - CMOS can be used for full-swing transitions
  4. Switch gate voltage on transistor to turn it off.
Is there a fundamental lower limit to energy dissipation of adiabatic charging?

- No!
- C.f., Boechler et al. (APL 97:103502, 2010) measured dissipation for charging a capacitor through a resistor adiabatically
  - Min. dissipation was much less than \((kT \ln 2)\), and was limited only by measurement uncertainty
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
- Only 4 timing signals $\phi_{0-3}$ are needed. Only 4 ticks per cycle:
  - $\phi_i$ rises during ticks $t \equiv i \pmod{4}$
  - $\phi_i$ falls during ticks $t \equiv i+2 \pmod{4}$

Animation:
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.”

\[ A = 0 \]
\[ \begin{array}{c}
  A_N \\
  A_P \\
  \bar{A}_N \\
  \bar{A}_P \\
\end{array} \]

\[ A = 1 \]
\[ \begin{array}{c}
  A_N \\
  A_P \\
  \bar{A}_N \\
  \bar{A}_P \\
\end{array} \]
Shift Register Simulation Results (Cadence/Spectre)

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

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

2LAL = Two-level adiabatic logic (invented at UF, ‘00)

M. Frank, RevComp
Cross-Disc. Intro for
$\Theta(\log n)$-time Recursive Adiabatic Wired-OR Carry-Skip Adder

(8 bit segment shown)

With this recursive structure, we can do a $2^n$-bit add in $2(n+1)$ logic levels.

Hardware overhead is $< 2 \times$ regular ripple-carry!
32-bit Adder Simulation Results
Further improvements may be attainable through more pipelining, carry-save adders, etc.

32-bit adder power vs. frequency

32-bit adder energy vs. frequency

(Results are normalized to a throughput level of 1 add/cycle)
What’s needed to create a viable path to pursue such ideas further?

- Requires large numbers of engineers, across multiple disciplines, to cooperate in pursuing this approach, while each adopting a very different new mindset:
  - Device physicists
  - Process engineers
  - Logic family designers / tool developers
  - Circuit/interconnect designers
  - Computer architects

- All these people need to learn that the optimal design points for reversible operation differ greatly from the optimal design points assuming irreversible operation.
The important figures of merit determining power-performance of digital technologies operated reversibly in terrestrial environments are things like:

- **For switching devices:** Minimize the *entropy coefficient* $c_S = Et/T$, where:
  - Device dissipates energy $E$ over an adiabatic transition time $t$ at operating temperature $T$.
  - Determines how system-level dissipation trades off against speed, while accounting for minimum cryogenic cooling overheads.

- **For storage technologies (static nodes, memory cells):** Minimize the *entropy rate* $r_S = (E/t)/T$, where:
  - Device dissipates energy $E$ per unit time $t$ at temp. $T$ just to reliably preserve a desired data value.
  - Determines limits on system energy efficiency due to power leakage from storage, again while accounting for minimum cooling overheads.
A New Perspective for Fabrication Process Engineers

- The best technology for maximizing system power-performance given the availability for reversible design is not the same as the best technology given traditional design practices!
  - E.g. leading-edge CMOS is highly suboptimal for reversible design b/c leakage currents are so high
    - It makes the wrong tradeoff if you don’t have to pay $CV^2$ costs!
  - Older-generation CMOS processes can actually achieve better overall energy efficiency when fully adiabatic design techniques are used! (Due to lower leakage)
    - System cost-performance is worse, but will improve over time
  - Looking forward, continuing to optimize processes for reversibility will lead to very different choices among the various potential post-CMOS device technologies.
The constraint of reversibility leads to very different abstractions being required at all levels in logic design. Designs must be rethought and re-optimized at every level.

- New logic families
- New standard cell libraries
- New hardware description languages
- New design tools
- New functional unit architectures
A New Perspective for Integrated Circuits & Systems Design

- New chip-level IC technologies
  - Integrated high-$L$ helical inductors would be useful
- New clocking and power-supply networks
  - Resonant delivery of trapezoidal clock-power signals
- New circuit-level constraints on logic layouts
  - Load balancing becomes even more important
- New interconnect design methodologies
  - Controlled-impedance signal paths become even more important
  - High-$Q$ transmission lines (integrated coax?)
A New Perspective for Computer Architecture

- Which functions best lend themselves to implementation with efficient reversible hardware algorithms?
  - Low computational complexity overheads from the use of reversibility

- Which reversible logic design technique is best for implementing a given functional unit?

- At what level in the system architecture should the reversibility constraint be broken?
  - As low-level device characteristics improve, we can expect the reversible/irreversible boundary to be pushed to higher and higher levels
Conclusion

- The digital technology industry is racing headlong towards a thermodynamic brick wall, due to irreversible logic practices
  - NO post-CMOS technology concept that ignores the requirement for reversibility can possibly get us very far beyond end-of-line CMOS
- But, I believe there’s a potential for the industry to transition over to a new track focused on development of reversible technologies
  - This approach has the potential to take us far beyond the limits of irreversible technology’s power-performance
- However, it will require a major shift in how things are done, which will be very disruptive to the industry
  - It requires rethinking design goals and constraints at every level from basic research in nanodevice physics up to microarchitecture (at least)
- I believe that massive investment in this approach needs to begin very soon if we want to avoid an extended flatline in power-performance delivered to end-user applications