

Exceptional service in the national interest



Generalizations of the Reversible Computing Paradigm

Michael P. Frank
Center for Computing Research
Sandia National Laboratories

Presented at the workshop on
Thermodynamics and Computation: Towards a New Synthesis,
Santa Fe Institute, Aug. 16-18, 2017

Approved for Unclassified Unlimited Release
SAND2017-8442 C



Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Abstract



In this talk, I survey recent work exploring various ways in which traditional concepts of classical reversible computing can be usefully generalized without requiring full-blown quantum computing. First, I show that the traditional Landauer-Fredkin-Toffoli model of *unconditionally* logically-reversible operations is not, in fact, the most general concept of classical logical reversibility that is sufficient to avoid entropy ejection from the computational state; more generally, we can consider a logically reversible *computation* composed of what I call *conditioned* reversible operations, whose preconditions for reversibility are satisfied with certainty by design. This model facilitates simpler designs for reversible hardware, and is well-suited to modeling adiabatic circuits. Next, I describe an *asynchronous* model of ballistic reversible computing, which reduces clocking overhead compared to staged adiabatic approaches, while avoiding the chaotic instabilities of synchronous ballistic models. Finally, I discuss a concept of *chaotic* reversible computing, another approach for reducing clocking in which a deep combinational network can be updated in a single step via an adiabatic transformation of a chaotically-fluctuating dynamical system.

Talk Outline

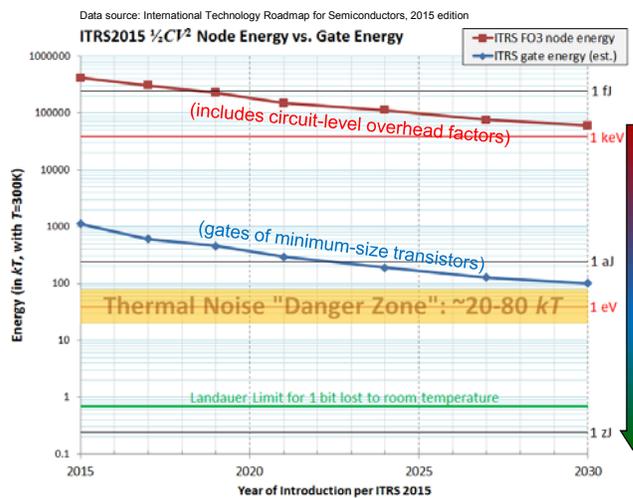


- Motivation:
 - The end of scaling, and implications for supercomputing
 - Need for practical classical reversible computing
- Landauer's Principle from fundamental physics
 - A simple, rigorous proof of a general formulation
- **Examples of reversible computing hardware**
- Generalized Reversible Computing Theory
 - Conditional logical reversibility
 - Foundation for designing fully adiabatic computing mechanisms
- Asynchronous Ballistic Reversible Computing
 - General model, proof of universality
 - Possible superconducting implementations
- Chaotic Reversible Computing
 - Sketch of another approach for reducing clocking overhead
- Conclusion

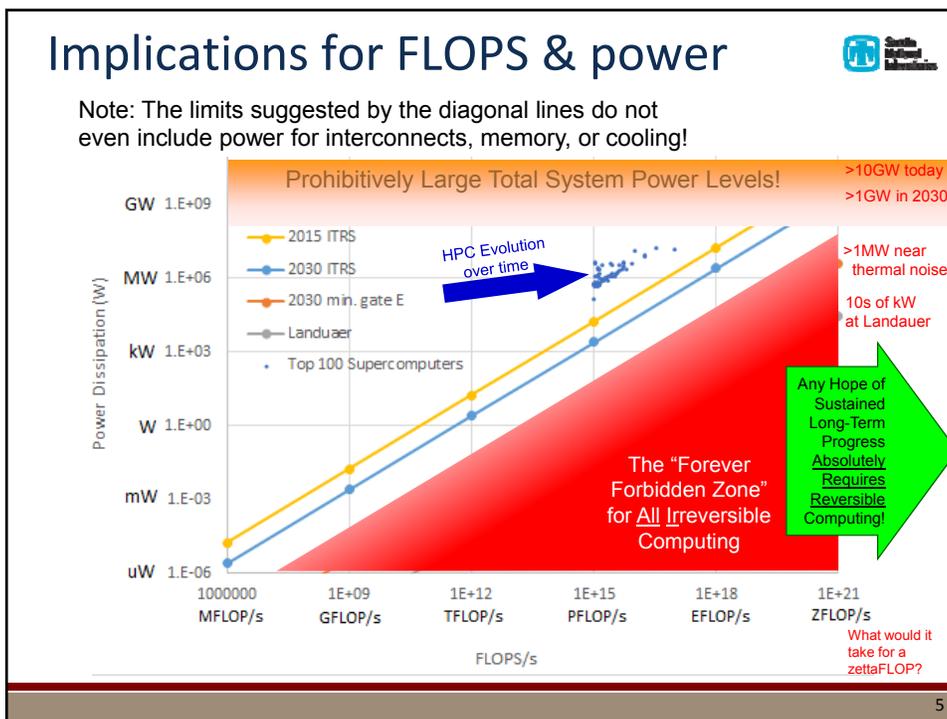
Semiconductor Roadmap is Ending...



- Thermal noise on gates of minimum-width segments of FET gates leads to channel PES fluctuations when $E_g \lesssim 1-2 \text{ eV}$
 - Increases leakage, impairs practical device performance
 - Thus, ITRS has minimum gate energy asymptoting to $\sim 2 \text{ eV}$
- Also, real logic circuits incur many further overhead factors:
 - Transistor width $10-20 \times$ min.
 - Parasitic (junction, etc.) transistor capacitances ($\sim 2 \times$)
 - Multiple (~ 2) transistors fed by each input to a given logic gate
 - Fan-out to a few (~ 3) logic gates
 - Parasitic wire capacitance ($\sim 2 \times$)
- Due to all these overheads, the energy of each bit in real logic circuits is many times larger than the min.-width gate energy
 - $375-600 \times$ (!) larger in ITRS'15
 - \therefore Practical bit energy for irreversible logic asymptotes to $\sim 1 \text{ keV!}$
- Practical, real-world logic circuit designs can't just magically cross this $\sim 500 \times$ architectural gap!
 - \therefore Thermodynamic limits imply much larger practical limits!
 - The end is near!



Only reversible computing can take us from $\sim 1 \text{ keV}$ at the end of the CMOS roadmap, all the way down to $\ll kT$.



Thermodynamics and Information

- Physical **entropy** quantifies *uncertainty about the detailed microstate of a system*.
 - First postulated by Boltzmann (H-theorem)
 - Integral to modern physics (Von Neumann entropy)
 - Depends on modeler's state of knowledge (Jaynes)
- The **reversibility (injectivity)** of microphysics underlies the Second Law of Thermodynamics.
 - Entropy of a closed system cannot decrease!
 - Conserved by unitary quantum time-evolution.
 - Entropy can *increase* if we have any uncertainty about the dynamics, or do not track it in detail
- At the most fundamental level, physical information *cannot be destroyed*.
 - Only **reversibly** transformed, and/or transferred between different subsystems...

$S[p] = E_p[\log p^{-1}]$

$S_I = 1.03 k$
 $S_F = 1.03 k$

Bijjective microphysics → No "true" entropy change

$S_I = 0.69 k$
 $S_F = 0 k$

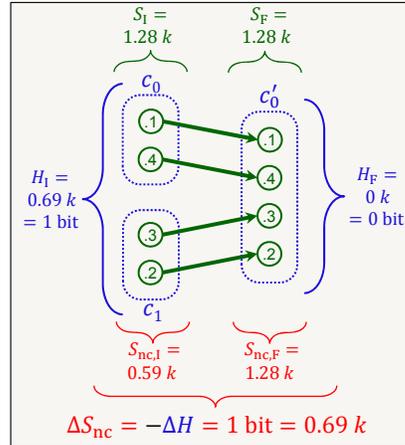
Irreversible microphysics → Entropy would decrease (Second Law of Thermo. would be violated)

$S_I = 1.03 k$
 $S_F = 1.29 k$

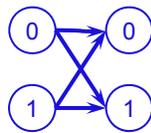
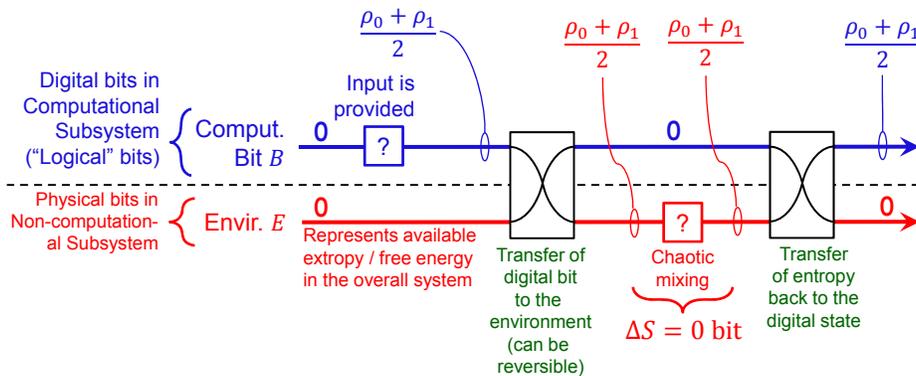
True dynamics uncertain (or not tracked in detail) → Entropy increases

Proving Landauer's Limit

- Follows directly from thermodynamics!
- A *computational state* c_j is just an equivalence class of physical states ϕ_i
 - On the left we see two computational states c_0, c_1 , each with probability 0.5
- The computational subsystem has an induced information entropy $H(c)$.
 - Here, it is $H(c) = \log 2 = 1 \text{ bit} = k \ln 2$.
- Thus, the non-computational subsystem (everything else) has expected entropy
 - $S_{nc} = S(\phi|c) = S(\phi) - H(c) = S - H$
 - The conditional entropy of the physical state ϕ , given the computational state c .
- Thus, if the computational entropy *decreases* (note here $\Delta H = -1 \text{ bit}$),
 - The *non-computational* entropy *must* increase by $\Delta S_{nc} = -\Delta H$ (here, $k \ln 2$).
- So, to lose a computational bit, we *must* eventually dissipate energy $\Delta E_{diss} = kT \ln 2$ as heat to some environment at some temperature T .

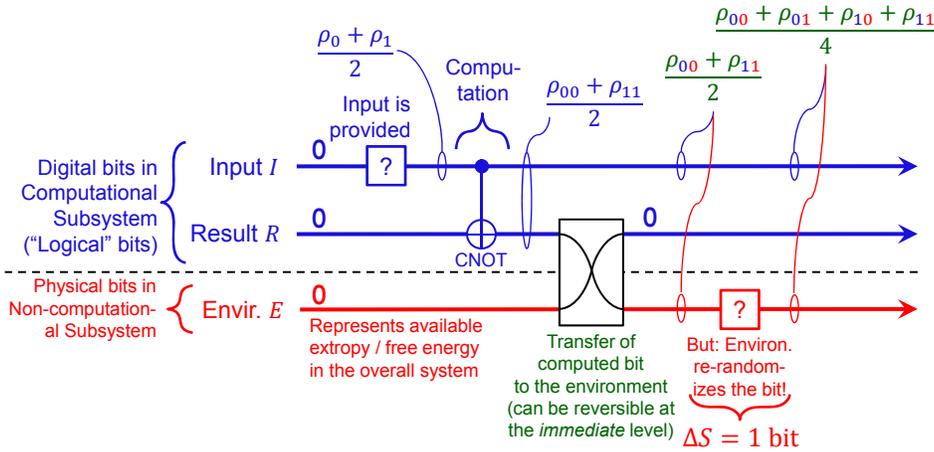


Thermodynamically Reversible Erasure of an Uncorrelated Bit



- Overall map including mixing is non-injective
- No autocorrelation between initial & final state
- Not "logically reversible" in traditional sense

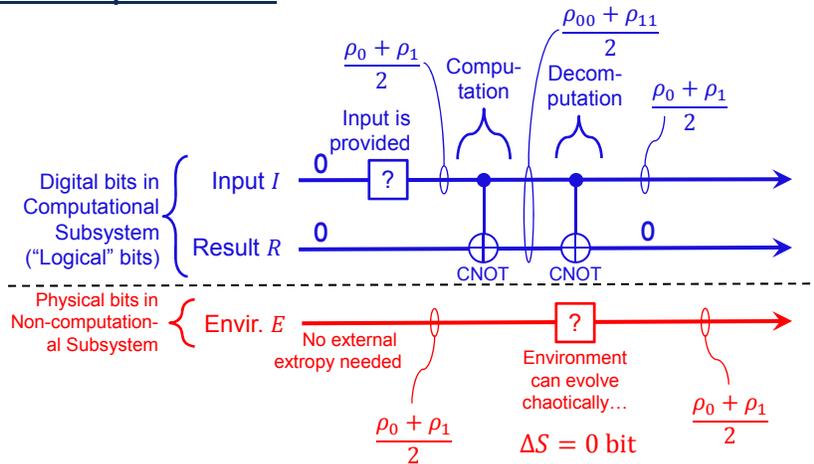
Logically Irreversible, Oblivious Erasure of a Correlated Bit



Moving a computed, correlated bit to an (*unpredictable!*) thermal environment necessarily, inevitably loses its correlations, and thus increases entropy!

9

Logically Reversible, Non-oblivious Decomputation of a Correlated Bit



Decomputing correlated bits, instead of ejecting them to the thermal environment, avoids losing correlations & increasing entropy!

10

Reversible Computing with Adiabatic Circuits



Some early history (pre-2000):

- Fredkin and Toffoli, 1978 (DOI:10.1007/978-1-4471-0129-1_2)
- Seitz *et al.*, 1985 (CaltechCSTR:1985.5177-tr-85)
- Several papers at PhysComp '92:
 - Koller and Athas (DOI:10.1109/PHYCMP.1992.615554)
 - Hall; Merkle (DOIs:10.1109/PHYCMP.1992.615549; 10.1109/PHYCMP.1992.615546)
 - General-purpose reversible methods, but for combinational logic only
- Younis & Knight, 1993 (<http://dl.acm.org/citation.cfm?id=163468>)
 - First fully-reversible, fully-adiabatic sequential circuit technique (CRL)
- Younis & Knight, 1994
 - Simplified 3-level adiabatic CMOS design family (SCRL) – Buggy though.
- Subsequent work at MIT by myself and other students, 1995-99
 - Several reversible and/or adiabatic demonstration chips

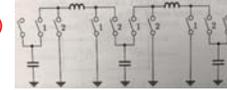
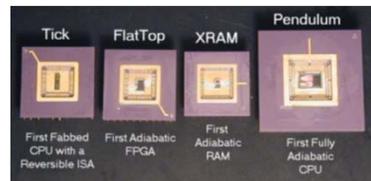


Figure reproduced with permission

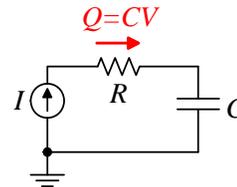
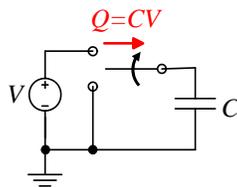


Conventional vs. Adiabatic Charging



For charging a capacitive load C through a voltage swing V

- **Conventional charging:**
 - Constant *voltage* source
- **Ideal adiabatic charging:**
 - Constant *current* source



- Energy dissipated:

$$E_{\text{diss}}^{\text{conv}} = \frac{1}{2} CV^2$$

- Energy dissipated:

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

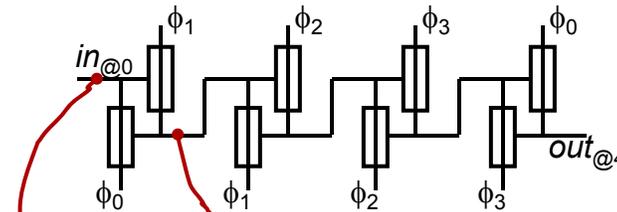
Note: Adiabatic charging beats the energy efficiency of conventional by advantage factor:

$$A = \frac{E_{\text{diss}}^{\text{conv}}}{E_{\text{diss}}^{\text{adia}}} = \frac{1}{2} \frac{t}{RC}$$

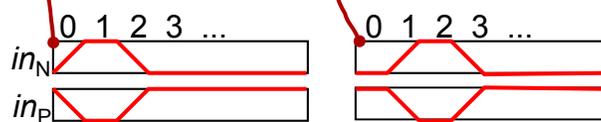
2LAL Shift Register Structure



- 1-tick delay per logic stage:



- Logic pulse timing and signal propagation:

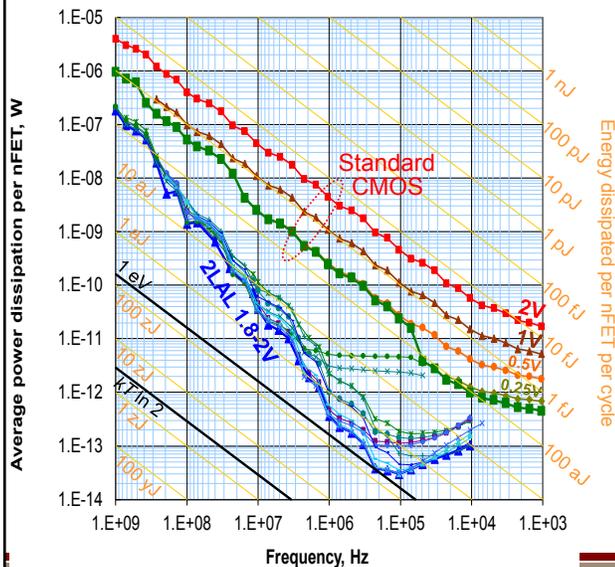


9/13/2017

Simulation Results (Cadence/Spectre)



Power vs. freq., TSMC 0.18, Std. CMOS vs. 2LAL
 2LAL = Two-level adiabatic logic (invented at UF, '00)



- Graph shows per-FET power dissipation vs. frequency
 - in an 8-stage shift register.
- At moderate freqs. (1 MHz),
 - Reversible uses $< 1/100^{\text{th}}$ the power of irreversible!
- At ultra-low power levels (1 pW/transistor)
 - Reversible is $100 \times$ faster than irreversible!
- Minimum energy dissipation per nFET is **< 1 electron volt!**
 - $500 \times$ lower dissipation than best irreversible CMOS!
 - $500 \times$ higher computational energy efficiency!
- Energy transferred per nFET per cycle is still on the order of 10 fJ (100 keV)
 - So, energy recovery efficiency is on the order of 99.999%!
 - Quality factor $Q = 100,000!$
 - Note this does not include any of the parasitic losses associated with power supply and clock distribution yet, though

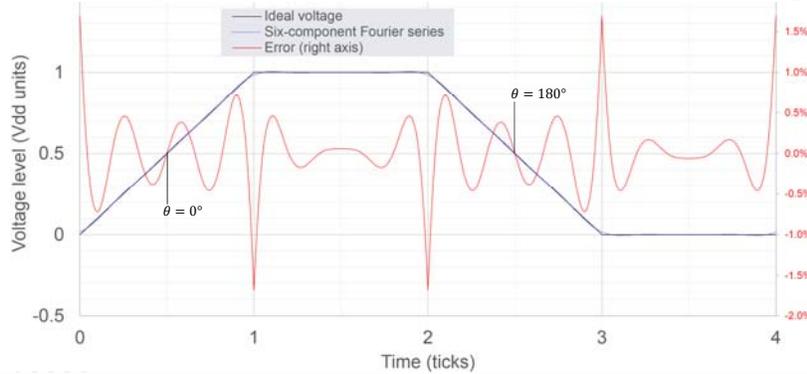
15

Spectrum of Trapezoidal Wave



- Relative to mid-level crossing, waveform is an odd function
 - Spectrum includes only odd harmonics $f, 3f, 5f, \dots$
- Six-component Fourier series expansion is shown below
 - Maximum offset with $11f$ frequency cutoff is $< 1.7\%$ of V_{dd}

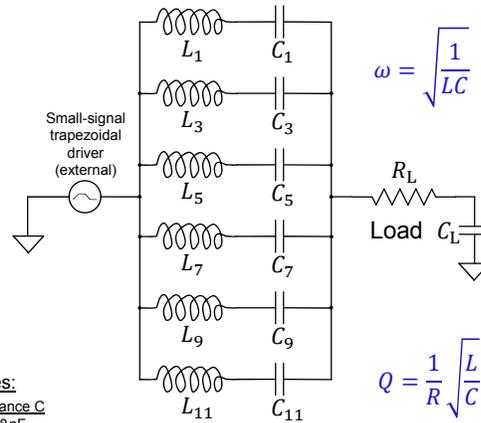
$$v_{f6}(t) = V_{dd} \left[\frac{1}{2} + \frac{4\sqrt{2}}{\pi^2} \left(\sin \theta + \frac{\sin 3\theta}{3^2} - \frac{\sin 5\theta}{5^2} - \frac{\sin 7\theta}{7^2} + \frac{\sin 9\theta}{9^2} + \frac{\sin 11\theta}{11^2} \right) \right]$$



Ladder Resonator Structure



- Can build trapezoidal resonator w. a ladder circuit made of parallel passive bandpass filters, each a sinusoidal LC resonator
 - Each "rung" of ladder passes a different odd multiple of the fundamental clock frequency f
 - Adjust L/C ratio to obtain a target Q value on each path, given parasitic R, C values
- Excite the circuit with a driving signal containing the right distribution of frequency component amplitudes
 - Each frequency component gets amplified by the Q value of its corresponding rung
 - If all rungs are designed to the same target Q , we can just use a trapezoidal driver
- For high Q , clock period must be long compared to the total parasitic RC ...
 - Max. possible $Q_n = \frac{1}{2\pi} \cdot \frac{t_{period,n}}{(RC)_{parasitic}}$



Ladder Resonator for Odd Harmonics

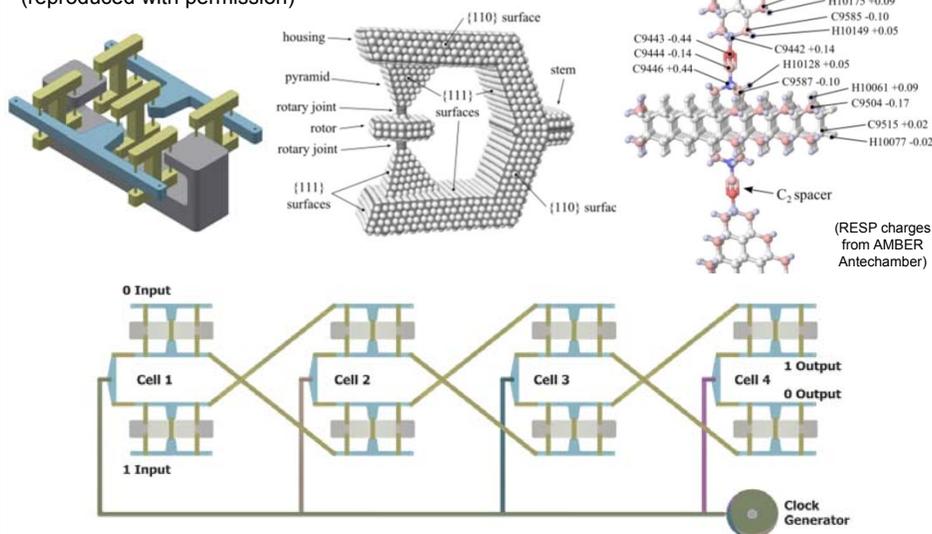
(for $V_{dd} \approx 1.75 \text{ V} \downarrow$)

harmonic mode (n)	frequency f	amplitude Va	inductance L	capacitance C
1	230 kHz	1000.00 mV	691.98 nH	691.98 nF
3	690 kHz	111.11 mV	230.66 nH	230.66 nF
5	1150 kHz	-40.00 mV	138.40 nH	138.40 nF
7	1610 kHz	-20.41 mV	98.85 nH	98.85 nF
9	2070 kHz	12.35 mV	76.89 nH	76.89 nF
11	2530 kHz	8.26 mV	62.91 nH	62.91 nF

Example values:

Nanomechanical Rotary Logic

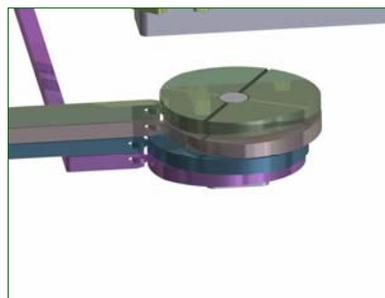
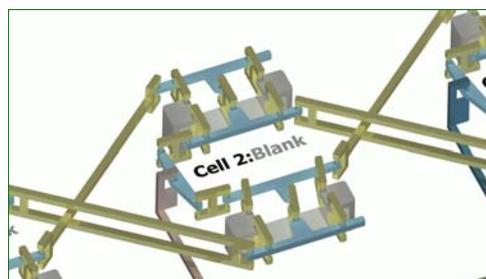
Merkle et al., IMM Report 46 and Hogg et al., arxiv:1701.08202
(reproduced with permission)



18

Rotary Logic Lock Operation

- Videos animate schematic geometry of a pair of locks in a shift register
- Molecular Dynamics modeling/simulation tools used for analysis include:
 - LAMMPS, GROMACS, AMBER Antechamber
- Simulated dissipation:
 - $\sim 4 \times 10^{-26}$ J/cycle at 100 MHz
 - 74,000 \times below the Landauer limit for irreversible ops!
- Speeds up into GHz range should also be achievable



19

Reinventing Reversible Computing Theory

(Frank, "Foundations of Generalized Reversible Computing," RC17.)

- Our very first task, in developing the theory of reversible computing, can be characterized as answering this question:
 - What are the necessary and sufficient conditions that must be met, *at the logical level*, in order for a computational process to avoid ejecting entropy from the computational state? (I.e., for $\Delta H \geq 0$.)
 - Or more generally, to approach 0 entropy ejection, $\Delta H \rightarrow^- 0$.
- Landauer attempted to answer this question in his 1961 definition of **logical reversibility**, and the entire traditional theory of classical reversible computing has been based on this definition...
 - Only problem: **Landauer's definition is wrong!**
 - In the sense that, in fact, it is provably a *sufficient*, but **not** a *necessary* logical-level condition to avoid entropy ejection from a computation.
 - Let's see why...

... RESTORE TO ONE is an example of a logical truth function which we shall call *irreversible*. We shall call a device *logically irreversible* if the output of a device does not uniquely define the inputs.

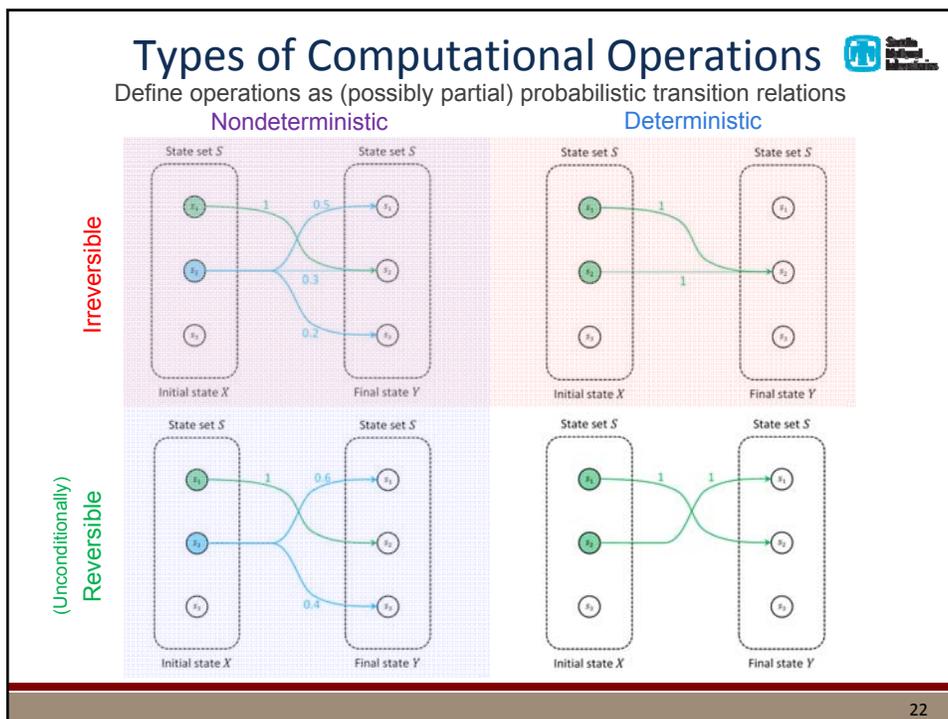
... Now assume that the computer is *logically reversible*. Then the machine cycle maps the 2^N possible initial states of the machine onto the same space of 2^N states, rather than just a subspace thereof.

20

Devices, Operations, Computations

- We'll distinguish several different concepts:
 - **Device** – Can perform one (or more) types of *operations*.
 - A given device has some associated local state information
 - Includes states of I/O terminals, internal states of device
 - **Operation** – a (*computational*) *operation* is a map O from *initial (computational) states* to *final states* (locally)
 - The terms “input” and “output” are ambiguous – avoid!
 - We can also consider partial maps (undefined = don't-care)
 - More generally, the operation O could even be a stochastic map...
 - A probabilistic transition rule $r_{ij} = \Pr[c_F = c_{Fj} \mid c_I = c_{Ii}]$.
 - » However, that case is not our main focus at present
 - **Computation** – for us, a *computation* is a computational operation performed within a specific operating context
 - Context specifies/constrains the initial state probabilities
 - These are essential for a meaningful thermodynamic analysis!

21



Avoiding Entropy Ejection

- Considering what's required of an operation if it is to be non-entropy-ejecting in all definable operating contexts leads to the traditional definition of logical reversibility for *operations*:

 - Theorem:** A total deterministic operation O is not even *potentially* entropy-ejecting if and only if O is a bijective map (permutation) of the full space C consisting of all of the device's describable computational states.

 - Definition:** We call such an O an (unconditionally) *logically reversible operation*.
- But, considering requirements for an operation to be non-entropy ejecting in a *specific* operating context P_1 gives a *different* logical reversibility concept suited for (contextualized) *computations*:

 - Theorem:** A total deterministic operation O is not *specifically* entropy-ejecting in a *given* operating context P_1 if and only if O is injective over at least (i.e., when restricting its domain to) the *active subset*

$$A = \{c_i \mid p_i \equiv P_1(c_i) > 0\} \subseteq C$$

of initial computational states that are assigned nonzero probability by P_1 .

 - Definition:** Here, we can call $e = (O, P_1)$ a *logically reversible computation*.
- Given that $e = (O, P_1)$ is logically reversible, we can also say that:

 - O is a conditionally (logically) reversible operation (this is always true),
 - O_A is a conditioned (logically) reversible operation with assumed set A ,
 - A is an adequate precondition for the (logical) reversibility of O .

23

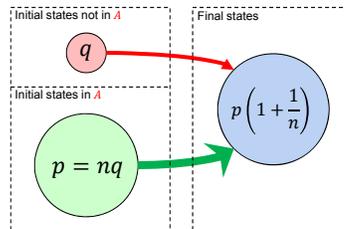
Almost-Logically-Reversible Computations



- You might object, “But real probabilities are almost never *exactly* 0.”
 - But, if they’re just *close* to 0, that’s good enough to be *almost* fully reversible.
- Theorem:** For any operation that is conditionally reversible under a given adequate precondition A , if we consider any progression of operation contexts in which the probability that A is not satisfied (i.e., that $c_i \notin A$) approaches 0, the entropy ejected by the computation due to Landauer’s principle also falls to 0 accordingly.
 - Lemma:** For a state with any probability q not in A that merges with some state in A that has a larger probability $p = nq$ (where $n > 1$), the contribution Δs of this state merger to the total entropy ΔS ejected from the computation approaches the following expression as the probability ratio n increases (i.e., as the probability q falls, relative to p), to first order in n :

$$\Delta s \rightarrow \frac{p}{n} (1 + \ln n) k_B$$

- And this value itself approaches 0, almost in proportion to $q = p/n$ as that value falls.

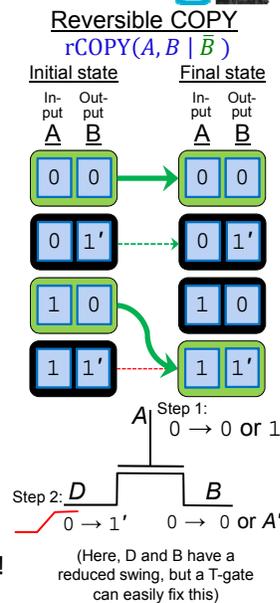


The concepts of Generalized Reversible Computing are essential for understanding Adiabatic Circuits!



Switching is, inherently, only conditionally reversible!

- E.g.: Even a *single* MOSFET can implement a certain (conditioned) reversible COPY operation...
 - Operation sequence is as follows:
 - Driving node D is initially statically held at 0, input A also 0.
 - Input A is externally supplied ($D \& B$ connected iff A is high)
 - Externally transition driver D from 0 to (weak) logic high $1'$
 - Voltage level on node B follows D iff A is strong logic high (1)
 - Note: Given a (loose) assumed precondition of \bar{B} ,
 - i.e., if all initial states with $B = 1$ have prob. 0,
 - this indeed performs a reversible COPY operation, $rCOPY(A, B | \bar{B})$.
 - Note: The output in this case is not full-swing,
 - In this diagram, primes (') denote reduced-voltage logic high signals
 - A notation precisely describing this operation’s semantics is:
 - $[A\bar{B}]$ if $B = 0$ then $B := A$ (else, leave state unchanged)
- Note: Traditional reversible computing theory based on unconditionally reversible operations is insufficient to model the logical/physical reversibility of this operation!



Conditionally-Reversible Logic in Adiabatic Circuits



- This simple CMOS structure can be used to do/undo latched reversible rOR operations

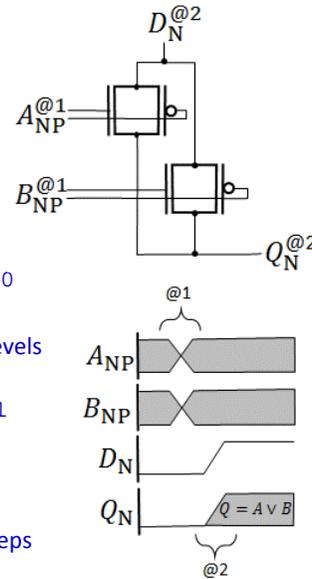
- Example of 2LAL logic family (Frank '00)
 - Based on CMOS transmission gates
 - Implicit dual-rail complementary signals (PN pairs) in this notation

- Computation sequence:

- Precondition: Output signal Q initially at logic 0
- Driving signal D is also initially logic 0
- At time 1 (@1), inputs A, B transition to new levels
 - Connecting D to Q if and only if A or B is logic 1
- At time 2 (@2), driver D transitions from 0 to 1
 - Q follows it to 1 if and only if A or B is logic 1
 - Now Q is the logical OR of inputs A, B

- Reversible things that we can do afterwards:

- Restore A, B to 0 (latching Q), or, undo above steps



Asynchronous Ballistic Reversible Computing



- Some problems with all of the existing *adiabatic* schemes for reversible computing:

- In general, numerous power/clock signals are needed to drive adiabatic logic transitions
- Distributing these signals adds substantial complexity overheads and parasitic power losses

- Ballistic logic schemes can eliminate the clocks!

- Devices simply operate whenever data pulses arrive
- The operation energy is carried by the pulse itself
 - Most of the energy is preserved in outgoing pulses
 - Signal restoration can be carried out incrementally

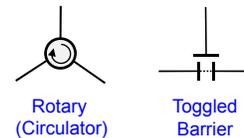
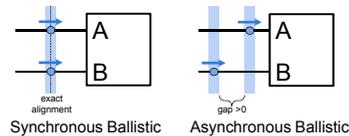
- But, *synchronous* ballistic logic has some issues:

- Unrealistically precise timing alignment required
- Chaotic amplification of timing uncertainties when signals interact

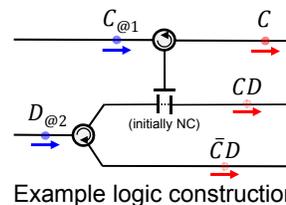
- Benefits of *asynchronous* ballistic logic:

- Much looser timing constraints
- Linear instead of exponential increase in timing uncertainty per logic stage
- Potentially simpler device designs

- A new effort at Sandia to try implementing ABRC in superconducting circuits



Example ABR device functions

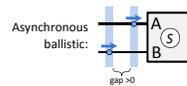
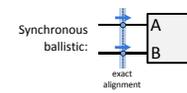
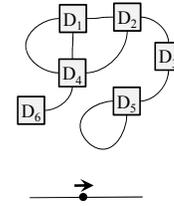


Example logic construction

ABRC Model: Starting Requirements



1. Universality – For reversible and embedded irreversible
2. Network model – Devices, bidirectional terminals, links
3. Localized signals
 - a. Spatial confinement – Along 1-D signal paths (wires)
 - b. Temporal localization – Pulse width specified as bounded
4. Ballistic propagation – at sufficiently large scales
5. Digital interpretation – m distinct signal types
6. Asynchrony – exact arrival times not important
7. Determinism – future depends non-randomly on past
8. Reversibility – using our generalized definition
9. Quiescence – devices unchanged between pulses



28

ABRC Model: Derived Requirements



These follow from the starting requirements:

10. Non-overlap of arriving pulses – Needed for determinism
11. Non-overlap of departing pulses – Needed for reversibility
12. One-to-one correspondence between incoming and outgoing pulses – Necessary to carry away pulse energy/timing information
13. Statefulness – To do logic, devices must have a stable internal state.
14. The possible ABRC device behaviors are exactly characterized by (isomorphic to) a restricted set of Mealy machines:
 - I/O symbol alphabet consists of $N = n \cdot m$ compound signal characters:

$$\Sigma = \{c_i^j\} = \left\{ \begin{pmatrix} t_j \\ T_i \end{pmatrix} \right\}$$
 - where $T_i \in \{T_1, T_2, \dots, T_n\}$ is any of n I/O terminals, each multiplicity m ,
 - and $t_i \in \{t_1, t_2, \dots, t_m\}$ is any of the m signal types.
 - Can easily generalize this to cases where not all terminals have the same arity
 - Transition function $f: \Sigma \times S \rightarrow S \times \Sigma$ is conditionally reversible
 - Machine implements an injective transformation of the subset of input strings for which its preconditions for reversibility are met at each step

29

ABRC Primitives

- Here, we enumerate some simple unary ABRC primitives:

- One-terminal unary primitives:

- Pulse Reflector (PR)

- Two-terminal unary primitives:

- The one-state, two-terminal primitives:

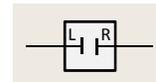
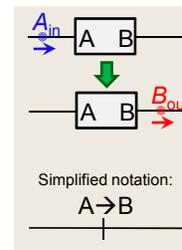
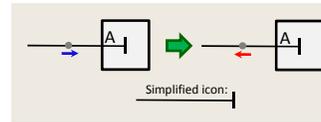
- Wire (W) a.k.a. signal renamer

» Functionally identical to a section of wire

- Barrier (B)

» Two pulse reflectors back-to-back

- (Continued on next slide...)



30

ABRC Primitives, cont.

- Unary primitives, cont.

- Two-terminal unary primitives, cont.

- Two-state, two-terminal unary primitives:

- We can categorize them using these symmetry groups:

» T – Time-reversal symmetry

» D – Data-terminal reversal symmetry

» TS – Time/state reversal symmetry

- All nontrivial 2-state, 2-terminal unary devices can then be classified as follows:

» Devices with both T and D symmetries

- ❖ Flipping Diode (FD)

» Devices with both D and TS symmetries

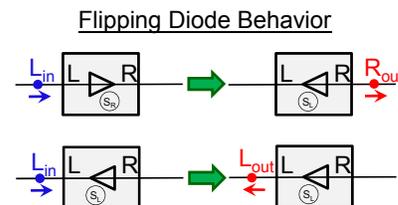
- ❖ Anti-Flipping Diode (AFD)

- ❖ Toggling Barrier (TB)

» Devices with none of these symmetries

- ❖ Directional Flipping Diode (DFD)

- ❖ Flipping Comparator (FC)



31

ABRC Primitives, cont.



Unary primitives, cont.

Three-terminal unary primitives:

One-state, three-terminal primitives:

- Rotary (R)

Rotary (CW)

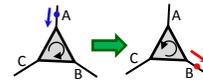


Two-state, three-terminal primitives:

- Some important symmetries:

- » D3 – All 3 data terminals treated symmetrically
- » D2 – A specific 2 of the 3 data terminals are interchangeable with each other

Flipping Rotary Behavior



- Some interesting cases:

- » Devices with both T and D3 symmetry:

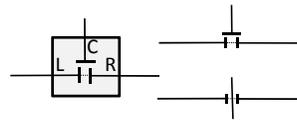
❖ Flipping Rotary (FR)

- » Devices with T and D2, but not D3 symmetry:

❖ Controlled Flipping Diode (CFD)

❖ **Toggleing Controlled Barrier (TCB)**

Toggleing Controlled Barrier



Universality Construction

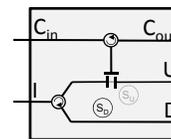
(slide 1 of 2)

- **Theorem:** {R, TCB} comprise a universal set of primitives for reversible (and embedded irreversible) computing

Constructive proof proceeds as follows:

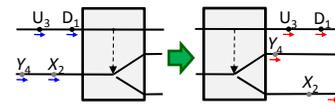
1. Using two rotaries and a toggleing controlled barrier,
 - We can structure a toggleing version of the reversible "switch gate" studied by Feynman and others
 - » We can then also build up a non-toggleing version of it...
2. A toggleing switch gate can be used as an asynchronous pulse (de)multiplexer
 - Requires supply of control pulses
3. A toggleing switch gate plus a demux can make a pulse duplicator
 - Produces incidental output ("garbage")

Toggleing Switch Gate:



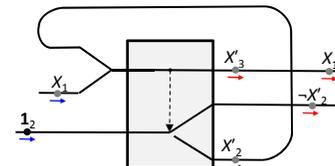
Block symbol

Asynchronous (De)Mux:



Simplified icon: (control & state implicit)

Pulse Duplicator:

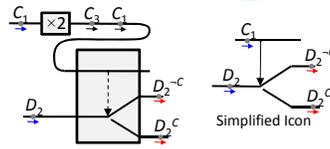


Simplified icon: X1, X'1, X'2, -X'2

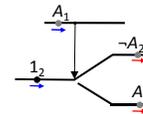
Universality Construction (slide 2 of 2)

- Universality theorem, cont.
 - Constructive proof, cont.
 4. With a pulse duplicator plus a toggling switch gate, we can build a non-toggling switch gate
 - Previously this gate was shown by Feynman and others to be universal!
 5. E.g., the (non-toggling) switch gate can be used to build a single-rail to dual-rail converter...
 - This can also be considered as a NOT gate that also produces an extra (garbage) copy of its input
 6. ...and the switch gate can also be used to produce a reversible AND function
 - Also produces \overline{AB} as a garbage output
 7. Standard logic constructions can then be used to build up arbitrary functions from NOT and AND
 8. Standard techniques like Lecerf reversal and the Bennett trick can be applied to decompute all garbage,
 - while leaving just the desired result and a copy of the input.
 - The above construction is sufficient for proving universality...
 - But, considered as a logic synthesis method, it clearly has some practical drawbacks...
 - This construction requires a great many control signals
 - Open research problem:
 - Find much simpler constructions for general functions
 - Using primitives other than {R, TCB} could be helpful

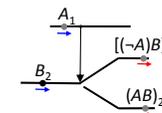
Non-toggling Switch Gate:



Single-rail to Dual-rail Converter:
(Includes NOT function)



Asynchronous reversible AND gate:



Physical realizations of ABRC?

- Of course, to be useful, this model needs to be realized in a specific physical implementation technology that actually provides near-thermodynamically-reversible operation.
 - Need some kind of soliton-like, near-ballistically-propagating pulse,
 - or some sort of particle or quasiparticle.
 - Need some physical state variable that can stably maintain at least binary state
 - for the toggling devices
 - Need a means of physically interacting the pulses with the states
 - in ways that can reliably, and almost physically-reversibly, implement at least a universal subset of the 2- and 3-terminal primitive devices.
- One intriguing possible candidate implementation technology is to use superconducting circuits...
 - SFQ (single flux quantum, or fluxon) pulses on appropriately constructed superconducting transmission lines can carry info. with relatively low dispersion and high propagation velocity (approx. 1/3 c)
 - Fluxons are naturally quantized by the SQUID-like circuits that produce them, and are naturally polarized (carry 1 bit's worth of +/- polarization state information per pulse)
 - Need to select suitable ABRC primitives operating on arity-2 signals
 - Fluxons trapped in loops (SQUID-like structures) can hold data quiescently
 - Generally, loops hold integer numbers of fluxons in some range: ..., -2, -1, 0, +1, +2, ...
 - How exactly to implement the reversible interactions?
 - A 3-year, internally-funded project is just starting at Sandia to investigate this...

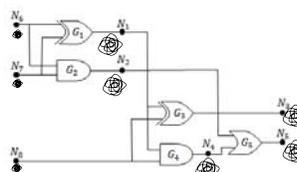
Chaotic Logic – Brief Summary



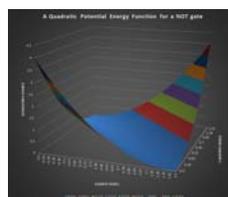
- Shannon teaches us that reliable *communication* is still possible with signal energies below the noise floor
 - Why not also reliable computation?
- Chaotic Network Model of logic:
 - Nodes are dynamic variables
 - Gates are Hamiltonian interaction terms
 - Node values chaotically fluctuate around a long-term average that encodes the result of the computation
- A simulator for this model was built...
 - cs.sandia.gov → Software → Dynamic
 - Page also links to a paper & a full talk

$$C = B \log_2 \left(1 + \frac{S}{N} \right)$$

Channel capacity theorem



Full Adder dynamical network



Logic gates implemented by potential energy surfaces

Frank & DeBenedictis '16, "A Novel Operational Paradigm for Thermodynamically Reversible Logic: Adiabatic Transformation of Chaotic Nonlinear Dynamical Circuits"

36

Example Interaction Functions

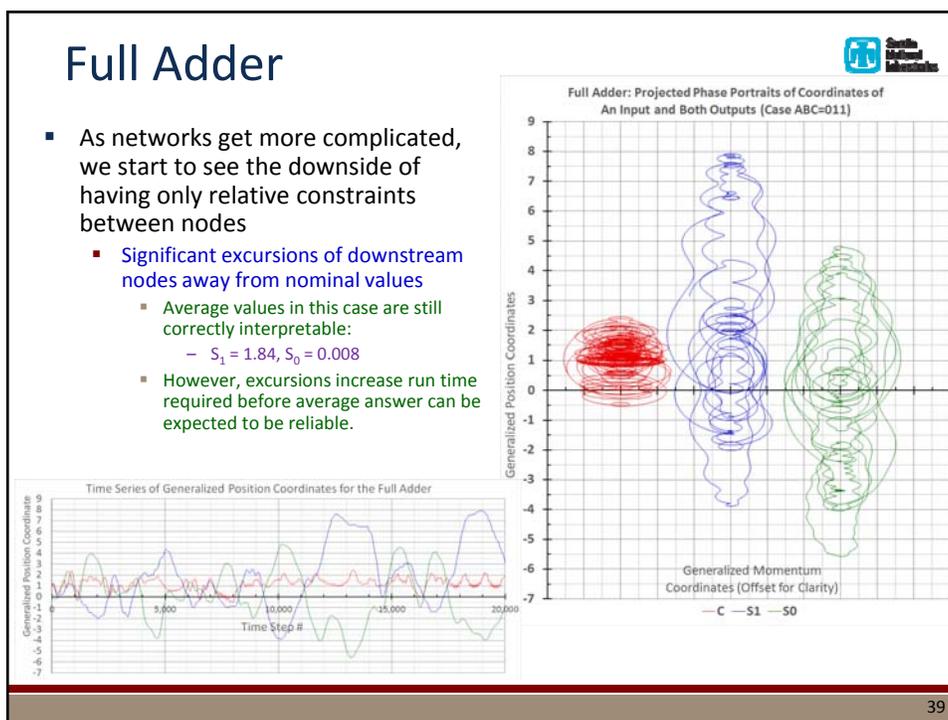
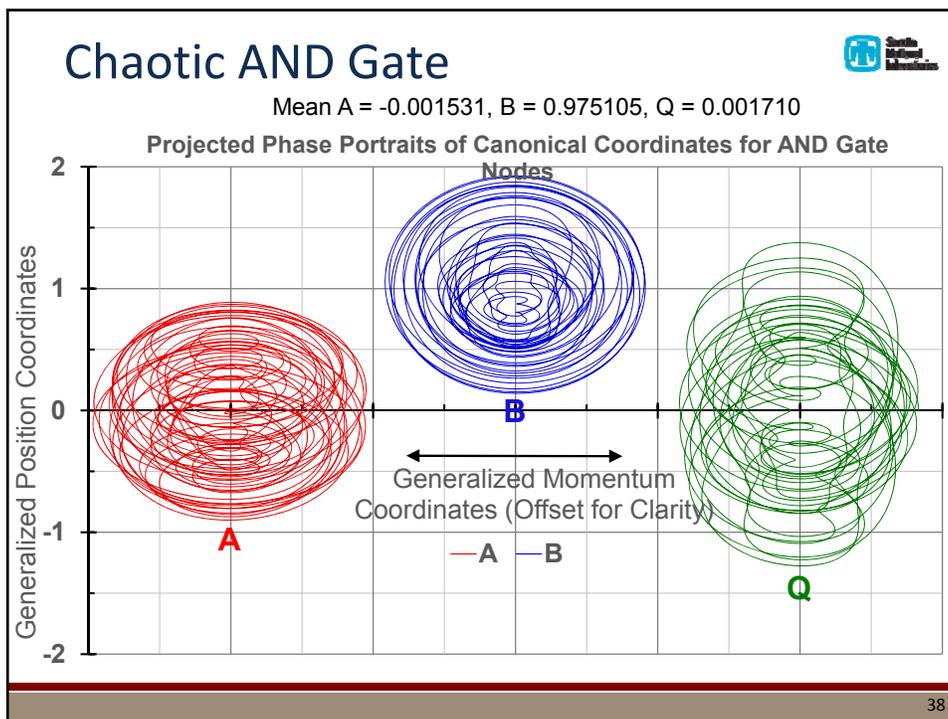


- Here are some simple quadratic interaction functions that assign minimum energy to correct logical outputs. Let x_i be the (generalized q) coordinate of the output node, and let x_j, x_k be the coordinates of input nodes:

	<u>Interaction Function</u>	<u>Minimized When</u>
▪ NOT gate: (coupling input x_j to output x_i)	$E_i = \frac{1}{2} e_{\text{NOT}} (x_i + x_j - 1)^2$	$x_i = 1 - x_j$
▪ AND gate: (coupling inputs x_j, x_k to output x_i)	$E_i = \frac{1}{2} e_{\text{AND}} (x_i - x_j x_k)^2$	$x_i = x_j x_k$
▪ OR gate (inclusive):	$E_i = \frac{1}{2} e_{\text{OR}} (x_i - x_j - x_k + x_j x_k)^2$	$x_i = x_j + x_k - x_j x_k$
▪ XOR gate (exclusive):	$E_i = \frac{1}{2} e_{\text{XOR}} (x_i - x_j - x_k + 2x_j x_k)^2$	$x_i = x_j + x_k - 2x_j x_k$

- The e_{GATETYPE} constants parameterize the energy scale of the gate interactions.
 - At *incorrect* logic values, the gate's potential energy above its minimum, or "stress" energy, will be $\frac{1}{2}$ this constant. → Rough equiv. of "signal energy" in this system.
 - In the experiments shown here, the energy scales e of all gates were set equal to kT .

37



Conclusion



- Reversible computing will be required to get beyond energy efficiency limits that threaten to slow industry's progress fairly soon...
 - However, more effective/efficient implementation techniques are still needed.
- In this talk, we summarized three novel variations of traditional models of reversible computing, which potentially can help lead to simpler, faster, more cost-efficient designs:
 - **Generalized Reversible Computing** theory clarifies that considering context-dependence when deriving the logical-level requirements for reversibility opens up a more general class of simpler, *conditionally*-reversible primitives.
 - Model still needs to be extended to handle the case of stochastic operations...
 - **Asynchronous Ballistic Reversible Computing** is a new class of reversible circuit models that are potentially capable of higher speeds and reduced clocking overheads compared to the (non-ballistic) adiabatic approaches.
 - An effort to demonstrate this new concept in SFQ JJ circuits has been funded.
 - **Chaotic Logic** is a very preliminary concept that also reduces clocking needs, and can potentially also utilize signal energies below the thermal noise floor (!)
- These new developments illustrate that the space of possibilities for reversible computer engineering is still just beginning to be explored...