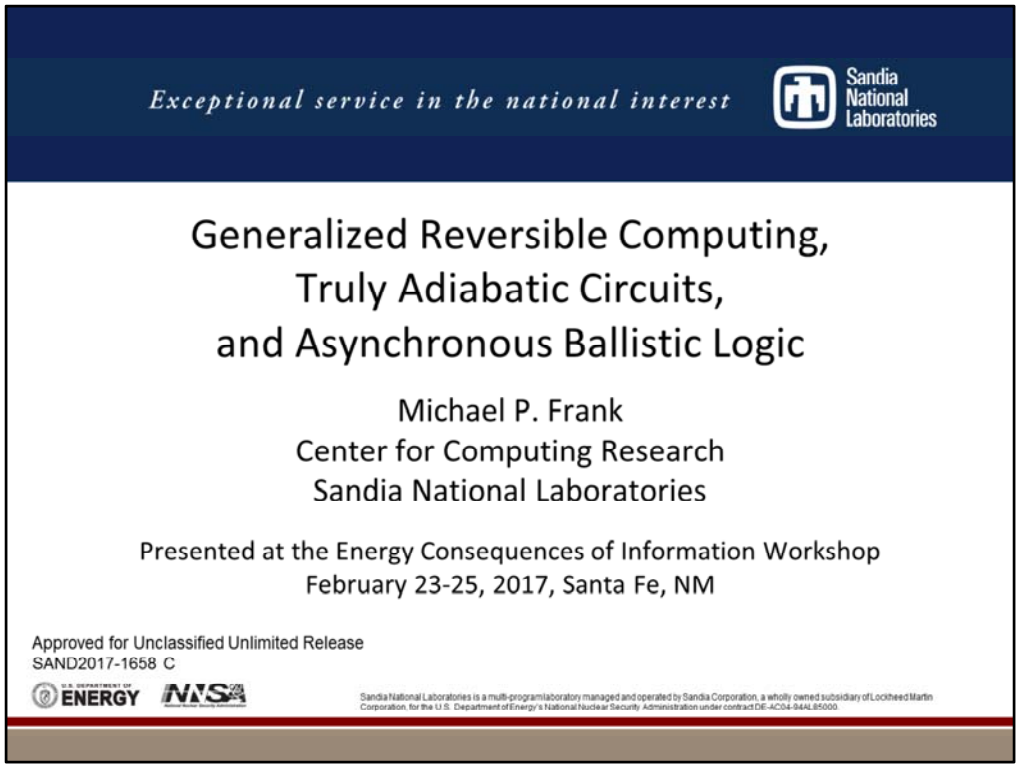


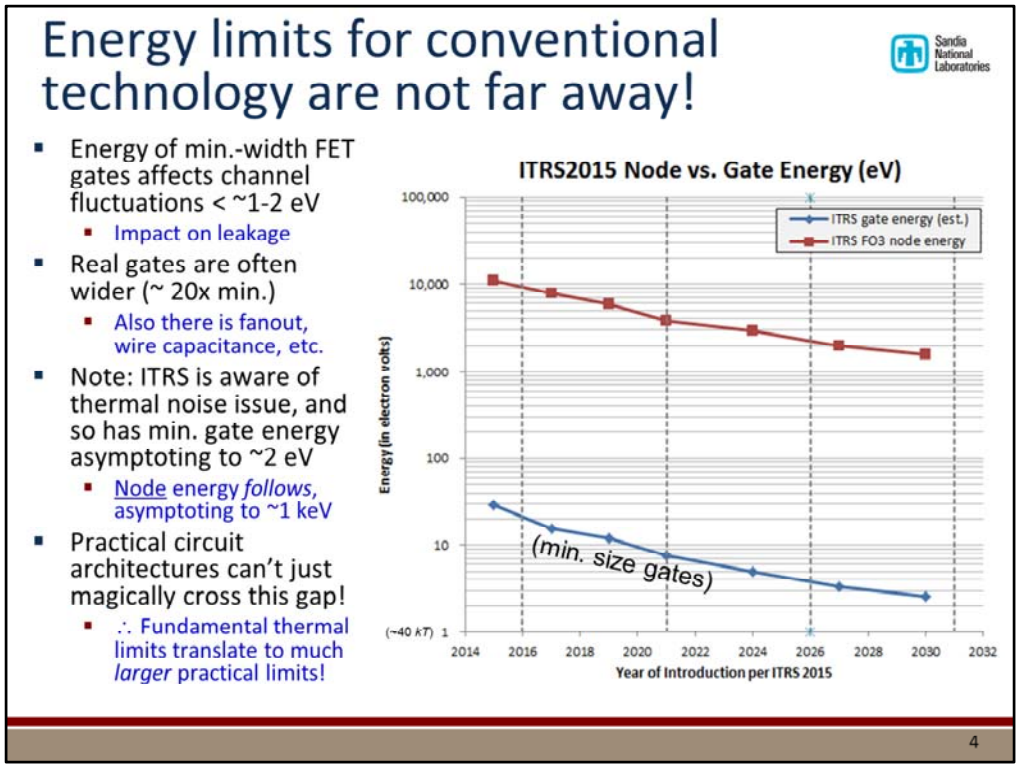
Title: **Generalized Reversible Computing, Truly Adiabatic Circuits, and Asynchronous Ballistic Logic**

Abstract: The concept of doing *reversible computing* in order to avoid information erasure, and the Landauer minimum energy dissipation associated with it, has been around for a long time. Unfortunately, the traditional theoretical models of reversible computing do not translate very easily into hardware. But, it turns out that those traditional models are more restrictive, and more complicated, than is necessary. The proper theoretical understanding of Landauer's principle leads directly to a more general model of what I call *conditionally* reversible operations, which can be mapped directly to simple hardware. The possible adiabatic physical transformations that can occur in circuits can be shown to correspond exactly to abstract conditionally-reversible computational operations in this generalized model. Furthermore, the generalized model, unlike the more restricted traditional one, can also be used as the foundation for novel *asynchronous*, ballistic models of reversible logic, which offer the prospect of being implementable with greatly reduced clocking overheads compared to all of the known synchronous implementations of reversible logic.

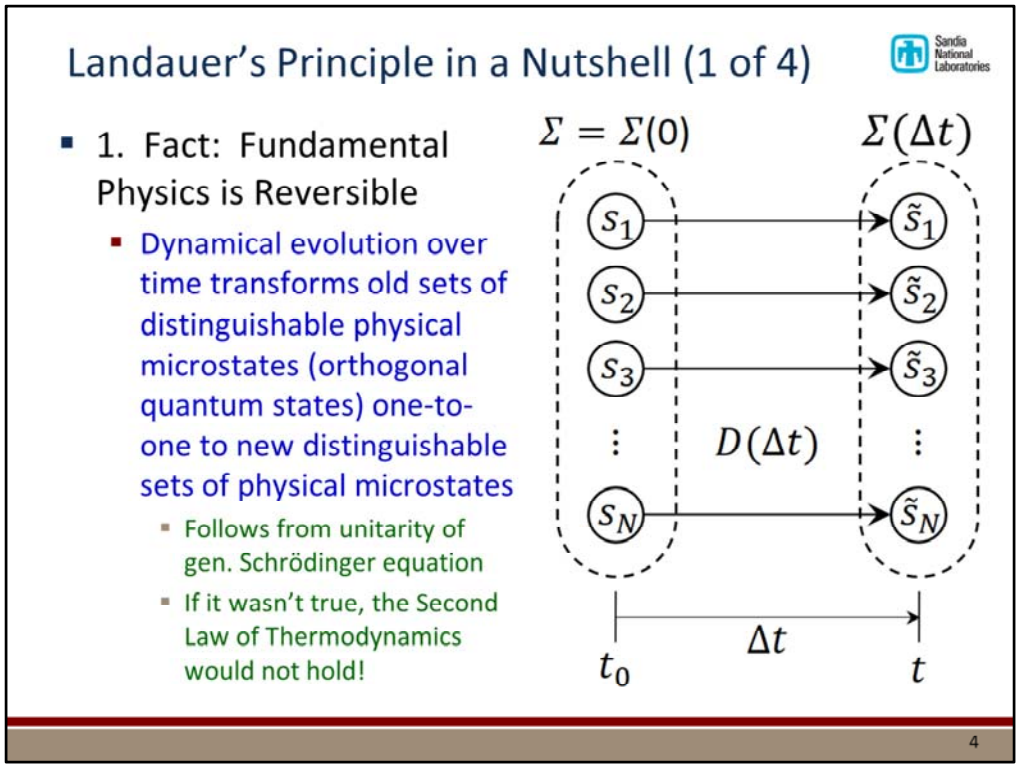
Revised talk outline

- Fresh prologue for this audience:
  - Leveling off of signal energies
  - A simple and rigorous proof of Landauer's Principle
  - What is reversible computing (RC)?
  - A concept for *fast* room-temperature reversible computing
- The original talk:
  - Motivation – Fix problems of traditional RC theory
  - Generalized Reversible Computing (GRC) theory
  - Correspondence between GRC and fully-adiabatic circuits
  - Asynchronous Ballistic Reversible Computing
  - Conclusion

The first part of the talk was added after the first day of the workshop to give a better introduction to the field to this audience.



An important point to be aware of is that reversible computing doesn't just save "the last kT", it can save the entire energy between where we are (10s of thousands of kTs) and 0.



Over any given time interval Δt, the operation of the physical dynamics $D(\Delta t)$, which, in quantum models, is a unitary transformation, maps old states one-to-one onto new states (an injective map).

Every viable quantum theory of fundamental physics (in particular, all quantum field theories, including the Standard Model of particle physics) have this property. It's believed that whenever we have a working theory of quantum gravity, it will exhibit this property as well.

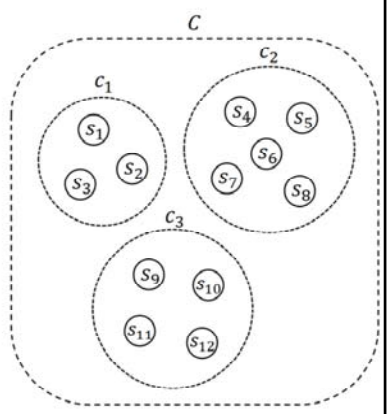
If multiple distinct states could transform to the same resulting state over some time interval, then the 2nd Law of Thermodynamics would be false, because the entropy of that pair of states (if they both have nonzero probability) would necessarily be reduced when they are merged (because $p \log p^{-1}$ is subadditive), and so entropy could just vanish. Because merging of states is not possible, entropy cannot decrease.

It's also believed that fundamental quantum dynamics is deterministic, and therefore that "true" entropy cannot increase, either; however, subjectively, we perceive entropy as increasing, because in our models of the world, we discard information about the state that in principle we could know, if we knew the exact laws of physics and tracked the evolution exactly.

The bijectivity of fundamental physics is one of the facts of physics that we are more certain of than anything else! (Hawking eventually conceded that even black holes are unitary.)

Landauer's Principle in a Nutshell (2 of 4)

- 2. A *computational state* is just an equivalence class of distinct physical microstates that we interpret alike for computational purposes.
  - *E.g.* any state of a circuit node in which its average voltage V is in some range, $V_{1\mathrm{L}} < V < V_{1\mathrm{H}}$, may represent a logic "1"
    - But, there are many detailed physical microstates consistent with this!
      - E.g., at nonzero temperature, many electron states near Fermi level may or may not be occupied

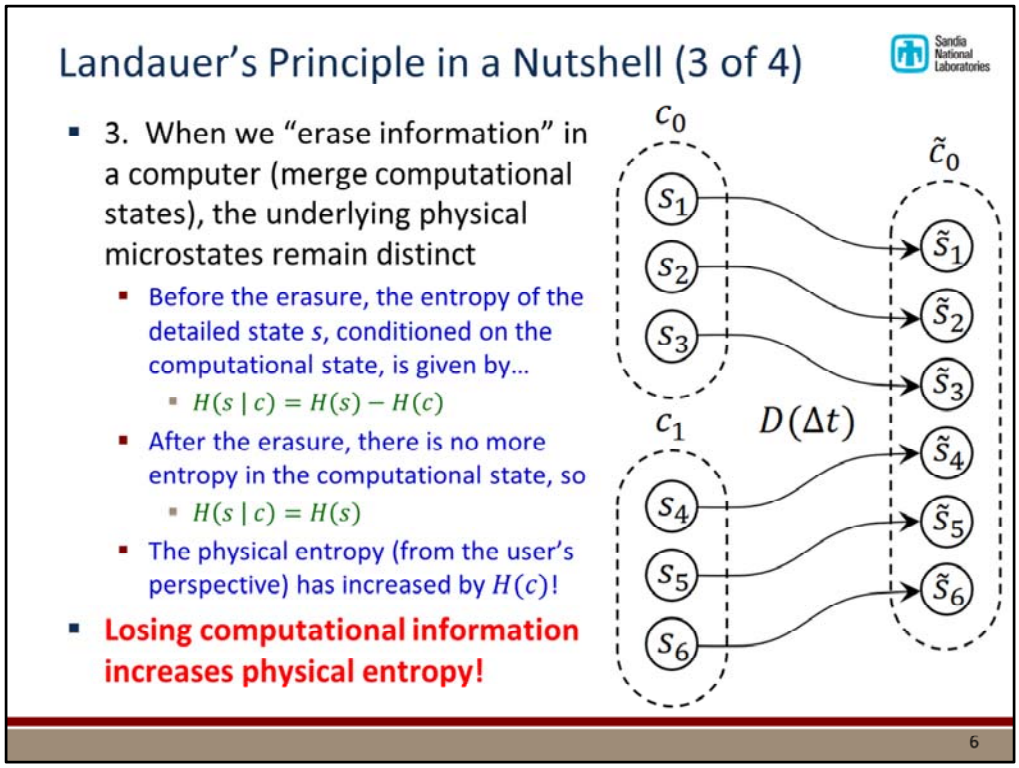


Example of a computational state space C consisting of 3 distinct computational states c_1, c_2, c_3, each defined as a set of equivalent physical states.

5

Even at zero temperature, for an extended circuit node, there would still be many distinguishable states (different Fermi levels) between $V_{1\mathrm{L}}$ and $V_{1\mathrm{H}}$.

And, even for quantum computers, wherein we may use a single 2-state quantum system (e.g. an electron spin) to represent a bit, there are still many physical states per computational state when we include the thermal state of other parts of the machine and its environment.



This is the conceptual core of Landauer's principle.

Before the computational states are merged, the information about which computational state we are in may, in principle, be known information – not true entropy – if, for example, it is information that was computed from other known information. However, if we then run those two state sets through an erasure mechanism that does not have this a priori knowledge – *i.e.*, a mechanism that is supposed to change the bit to a 0 regardless of whether it was a 0 or a 1 originally – all of the entropy that existed in the computational state, from the device's perspective, will simply get transferred to the non-computational state; thus, the entropy in the non-computational state is increased.

Another way of saying this is that the amount of physical entropy *conditioned* on knowledge of the computational state, $H(s \mid c) = H(s) - H(c)$, which is the user's view of the physical entropy, is increased, because the information entropy $H(c)$ has gone from some nonzero value to 0.

Another way of describing this process is that some known bit of physical information (whether the computational state was c_0 or c_1) got pushed out into the environment, becoming part of the non-computational state, where that bit was subsequently randomized, and became entropy.

Landauer's Principle in a Nutshell (4 of 4)

- 4. Entropy/information is measured in logarithmic units.
  - Two equiprobable computational states → Entropy/information content of computational state is one factor-of-two logarithmic unit

$$H(c) = [\log 2] = [\log \mathrm{e}] \log_{\mathrm{e}} 2 = k_{\mathrm{B}} \ln 2$$

- 5. If entropy $\Delta S = H(c)$ ends up in a thermal environment at temperature T, this requires adding heat $\Delta Q = T\Delta S$ to the heat bath, by the definition of thermodynamic temperature:

$$\frac{1}{T} = \frac{\partial S}{\partial Q}$$

- ∴ Merging two equally-likely computational states implies that we must dissipate this amount of energy to the heat bath:

$$\Delta E_{\mathrm{diss}} = k_{\mathrm{B}} T \ln 2 \quad \leftarrow \text{Landauer limit}$$

7

Here is the "Landauer limit" in its usual form. As we can see from this and the previous 3 slides, it's a simple logical consequence of the bijectivity of physics, the notion of a computational state, the definition of thermodynamic temperature, and the definitions of a bit and of Boltzmann's constant, as logarithmic units.

The expressions in square brackets represent "indefinite logarithm" quantities, which are dimensioned in generic logarithmic units.

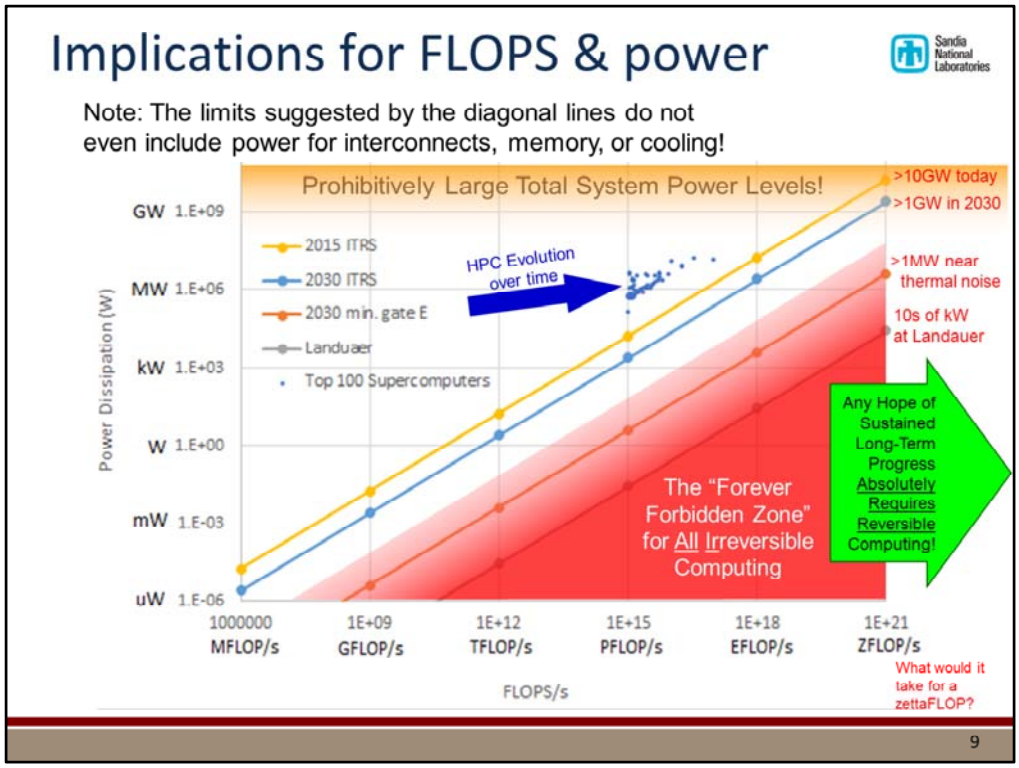
Enter Reversible Computing…

- **Problem:** Landauer's Principle teaches us that losing computational information (merging computational states) implies unavoidable energy dissipation.
- **Solution:** *Compute without losing information!*
  - Don't ever try to erase bits / merge two distinct computational states.
  - Instead, *transform* computational states one-to-one into new states.
    - No decrease in computational entropy
    - No need to eject computational entropy to the physical state
  - This is what we mean by reversible computing.
- Bennett (1973) showed that reversible computations can still compute any function…
  - To get rid of temporary results that are no longer needed, you can always *decompute* them instead of erasing/overwriting them

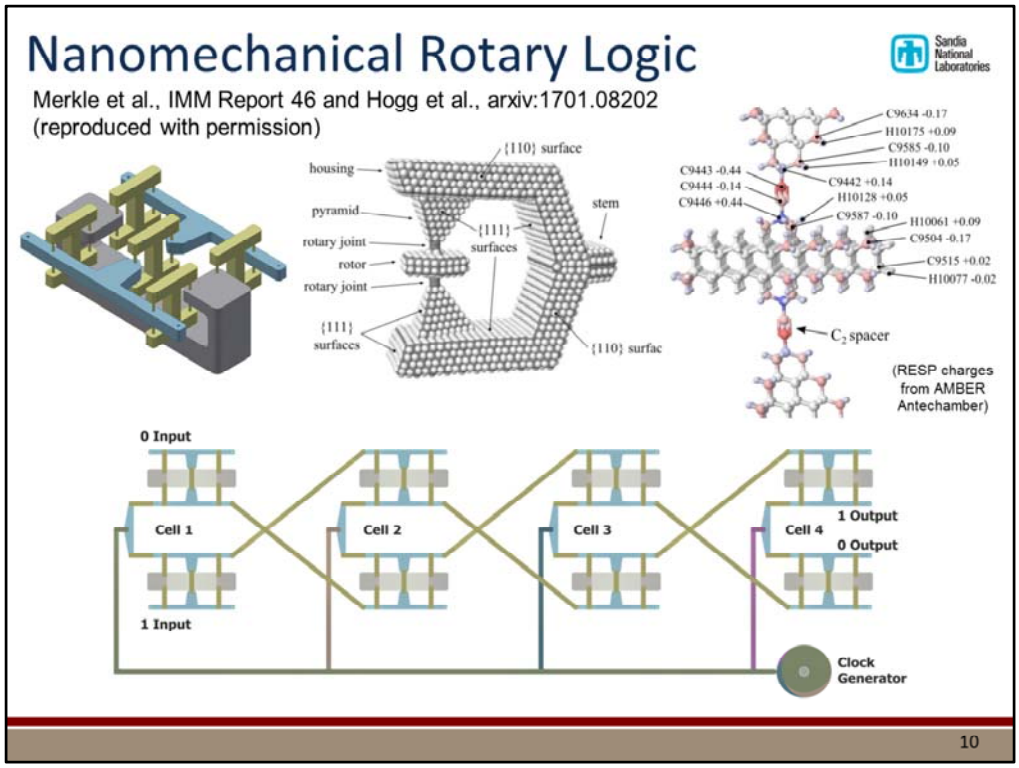
8

Reversible computing is really the only way to get around Landauer's principle. The principle itself applies to *any* physical system that we may consider as representing information. It doesn't matter if we use some nonstandard information encoding (such as spike timing), or if we think of the device as analog or digital (quantum mechanics guarantees that any finite system always still has some discrete set of distinguishable states). Whatever model of computation we use, in order for it to avoid the energy dissipation limits implied by Landauer's principle, it must also be reversible.

A question was asked at the workshop about reversible analog computing. It's worth noting that one way of achieving reversibility involves "analog" dynamical systems that are operating in a strongly chaotic regime. This is because a conservative dynamical system whose dynamical orbit has converged onto a strange attractor is essentially already in a thermodynamic equilibrium state, and thus incurs no further dissipation. (This is what I discussed in my Chaotic Logic talk last October at ICRC.) Another example of a form of analog computation that is at least partially reversible is reservoir computing, since the dynamical evolution of the reservoir system is typically at least somewhat reversible, that is, not dissipating its dynamical energy rapidly, relative to its rate of internal transformation along its trajectory).

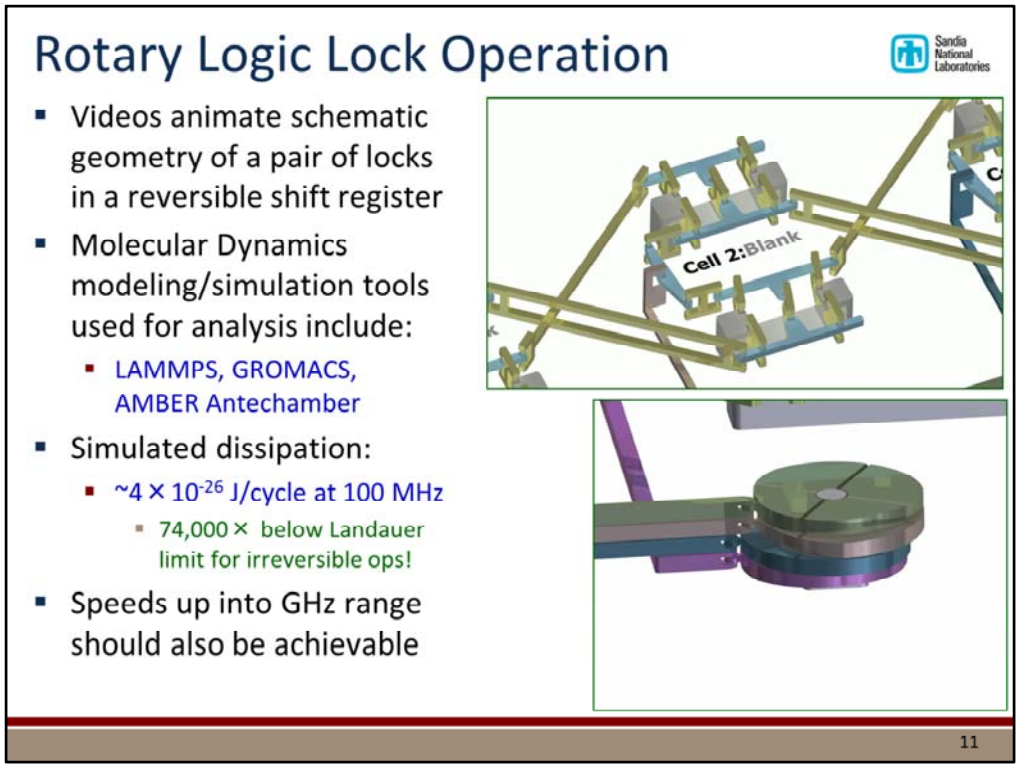


If we ever want to progress down into the red area of the chart, we have to move to some form of reversible computing.



An example concept illustrating a general point that the energy-delay product (which determines energy efficiency at a given speed) of reversible computing is highly technology-dependent. We can do many orders of magnitude better than CMOS in a well-designed reversible technology. Merkle, by the way, also invented the concepts of public-key cryptography, and cryptographic hash functions.

Of course, we don't yet have the atomic precision manufacturing technology required to build these structures, but the point is that there is nothing fundamental in the laws of physics that prevents far more efficient, high-performing versions of reversible computing from being possible. That is, the limits of reversible computing are only technology-dependent.



The two videos here can be played in the PowerPoint version of the talk. The top one shows a stage of a reversible shift register. The bottom one shows a set of cam wheels that can be used to drive the adiabatic mechanical transitions.

Note that based on the dissipation analysis, this technology would be several orders of magnitude more energy-efficient than any possible irreversible technology, despite still being able to run at GHz speeds.

One critique of this approach is that the signal propagation speed is limited to the speed of sound in these nanostructured diamondoid rods (the speed of sound in bulk diamond is about 12 km/s, which is 12 microns per nanosecond)… Faster communication speeds would be desirable… Maybe a hybrid of mechanical components for the logic, and electrical signaling for long-distance communication.

Later in the talk (if I have time) I will mention another implementation idea involving superconducting circuits. (We didn't end up getting to this.)

Structure of the Talk

1. Explain the motivations for this work
   - Overcoming limitations of traditional reversible computing theory
   - Dispelling confusion and facilitating technological progress
2. Develop **Generalized Reversible Computing** (GRC) theory
   - Starting point: Properly understanding Landauer's principle
   - *Logically reversible computations*: The correct general concept
3. Show how GRC can be used to model adiabatic circuits
   - Adiabatic transitions are conditionally-reversible computational ops
   - Building simple designs for truly, fully reversible AND and OR gates
4. Show how GRC makes possible a novel quasi-asynchronous ballistic style of reversible logic with reduced clocking needs
   - I call this **Asynchronous Reversible Computing** (ARC)
5. Conclusion

A related paper, "Foundations of Generalized Reversible Computing," was just submitted to RC'17.

12

This is the outline of the original talk. I did not have time to get to this material during my presentation, but perhaps the preface material I showed was more interesting to this audience.

Motivations for this Work

- Transcend the limitations of the traditional model of reversible logic networks
  - This model was originated by Landauer (1961)
  - Developed in detail by Fredkin and Toffoli (1980)
- Problems with the traditional model:
  - Insufficiently general to express all reversible computations
  - Inadequate to represent real reversible physical mechanisms
  - Leads to overly complex, hard to implement "primitive" gate ops
  - Restricted to synchronous logic → clocking overheads
- The traditional model has engendered a lot of confusion…
  - "Is logical reversibility really required for physical reversibility?"
- Has been a roadblock holding back progress in the field…
  - It's high time to adopt a more comprehensive theoretical model!

13

● We want to show how to transcend the limitations of the traditional (Landauer-Fredkin-Toffoli) theoretical model of reversible logic networks, which:

- Are insufficiently general to express the full range of truly logically- and physically-reversible computations that are in fact possible!
- Are inadequate to represent the inherent computational structure of the real-world adiabatic logic mechanisms that we can actually build!
- Lead to overly complex "primitive" gate operations that don't map easily to device-level implementations! → Resulting designs are inefficient
- Are restricted to synchronous logic schemes requiring extensive clocking overheads! → Creates an additional level of inefficiencies

● Due to these issues, the traditional reversible logic model has arguably engendered a lot of confusion, and has unfortunately been somewhat of a roadblock holding back progress in the field…

- It's high time that we adopt a more comprehensive theoretical model!

Landauer – What he got right!

- Information expelled from the computational state can't be destroyed, due to the reversibility of fundamental physics
  - Therefore ("Landauer's principle"), it ends up in environment
  - For an operation applied *in a given statistical context*, the entropy ΔS expelled from the device is simply given by the initial state entropy S^{I} minus the final state entropy S^{F}
    - Calculate from initial and final state probability distributions $p_i^{\mathrm{I}}, p_i^{\mathrm{F}}$

$$\Delta S = S^{\mathrm{I}} - S^{\mathrm{F}} = \left(\sum_i p_i^{\mathrm{I}} \log \frac{1}{p_i^{\mathrm{I}}}\right) - \left(\sum_i p_i^{\mathrm{F}} \log \frac{1}{p_i^{\mathrm{F}}}\right)$$

  - Logically rigorous consequence of unitarity of quantum physics!

14

● Information expelled from the computational state cannot be destroyed, due to the reversibility of fundamental physics

- Therefore ("Landauer's principle"), it ultimately ends up as thermal entropy in the environment, if it's not explicitly preserved somewhere
- For a computational operation applied *in a given statistical context*, the amount ΔS of entropy that must be expelled from the device is simply given by the initial state entropy S^{I} minus the final state entropy S^{F}
  - Computable from initial and final state probability distributions p_i^{I}, p_i^{F}

$$\Delta S = S^{\mathrm{I}} - S^{\mathrm{F}} = \left(\sum_i p_i^{\mathrm{I}} \log \frac{1}{p_i^{\mathrm{I}}}\right) - \left(\sum_i p_i^{\mathrm{F}} \log \frac{1}{p_i^{\mathrm{F}}}\right)$$

  - There was actually an arithmetic error in the specific numerical example Landauer computed in his 1961 paper, but his formulas were correct.
- I emphasize: The validity of this formula follows *immediately* from the unitarity of quantum time-evolution, and is *absolutely unquestionable*!!

Landauer – What he got less right…

- Defines logical (ir)reversibility for an *N*-bit *device*
- Assumes operation on the *entire* space of 2^N *combinatorially possible* initial states (or "inputs")

> … 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.

- But, the initial state probabilities are also important!
  - DeBenedictis & Frank previously pointed this out, at ICRC 2016 (http://bit.ly/2hYWLdV)
- **Crucial:** If some initial states *have probability 0*, then *not* all of the 2^N combinatorially-possible initial states are *statistically* possible
  - In such contexts, device operation can map full combinatorial space of 2^N initial states onto a *smaller* set of final states, *while retaining* $\Delta S = 0$ (reversibility)!
    - Landauer's "logical reversibility" tragically obscured this critically important fact!

15

● Here, he defines logical (ir)reversibility for an *N*-bit *device*, which he assumes implicitly to operate on the *entire* space of 2^N *combinatorially possible* initial states (or "inputs")

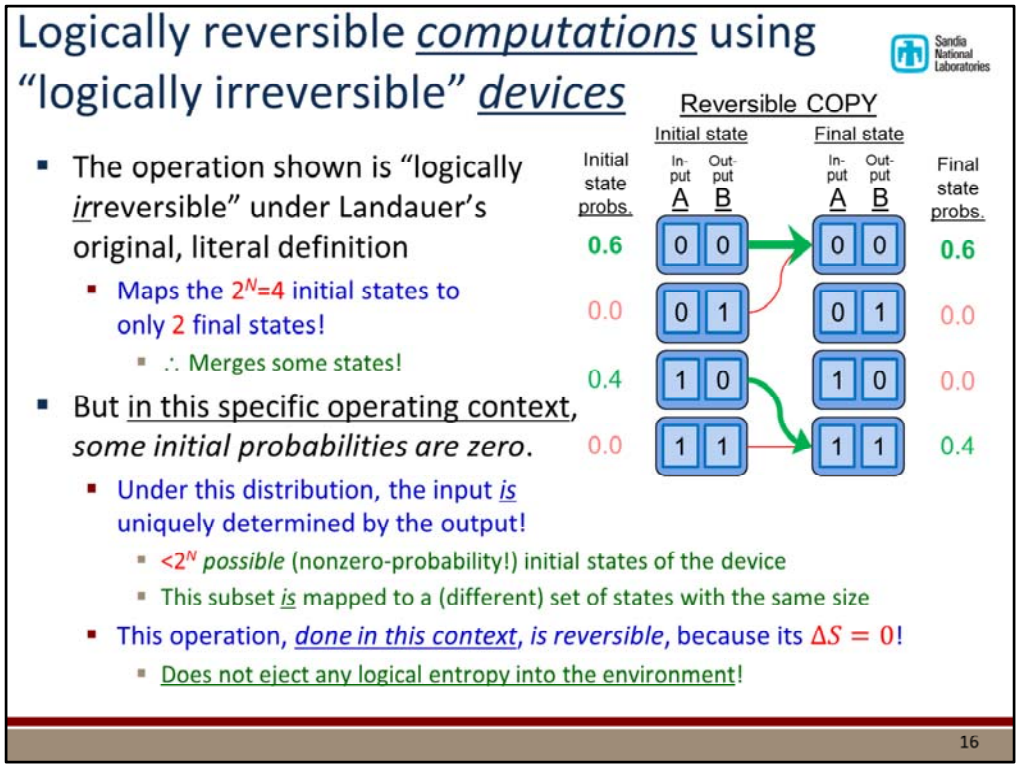
● However (c.f. prev. slide), what's actually important for determining the entropic reversibility of a *computation* is *not just* the choice of operation (state mapping) implemented by the devices, but also the statistical *operation context* – the probability distribution over the initial states!

- Landauer appears to be forgetting, here, that in the actual entropy-ejected formula (as shown on the previous slide), ΔS depends not just on the operation, but also on p^{I}, the initial state probabilities!!
  - DeBenedictis & Frank previously pointed this out, at ICRC 2016 (http://bit.ly/2hYWLdV)

● **Crucial:** If some initial states *have probability 0*, then *not* all of the 2^N combinatorially-possible initial states are *statistically* possible, in that context, but it's the *statistical* characteristics that are *the actual thing that really matters* for entropy purposes!

- Thus, in such contexts, a device operation can, perfectly consistently with known physics, map the full combinatorial space of 2^N initial states onto a *smaller* set of final states, *while retaining the property that the entropy ejected* $\Delta S = 0$ (reversibility)!
  - Landauer's definition of "logical reversibility" tragically obscured this critically important fact!

● Landauer also got a few other important things wrong at first, like not realizing that computed information that is no longer needed can later be reversibly *decomputed* instead of erasing it (as Bennett later showed), but that's already widely known.



● This diagram illustrates a state mapping or "device operation" that is normally assumed to be "logically *ir*reversible" under Landauer's original, literal definition

▪ Maps the 2^N=4 initial states to only 2 final states!

∴ Merges some states!

● However! Note that, crucially, in the specific operation context shown here, *some of the initial state probabilities are zero*.

▪ Under this distribution, the identity of the input, out of the actually-possible (that is, nonzero-probability!) inputs, *is* uniquely determined by the output!

◦ Note there are *less* than 2^N *possible* (nonzero-probability) initial states of the device, *given this distribution*, and this subset of states *is* mapped onto a (different) set of states with the same size (*i.e.*, smaller than the full set of 2^N states).

▪ We *ought* to say that this operation, *done in this context*, *is logically reversible*, because its $\Delta S = 0$! It does not eject any logical entropy into the environment!

What's the implication?

- The traditional concept of "logically reversible computation" we've used ever since Landauer is ***the wrong one***!
  - Significantly more restrictive than necessary.
- Need to *reconstruct* reversible computing theory
  - Many applications of the theory will end up changing!
- Conceptual elements:
  - Distinguish devices, operations, and computations.
  - Conditionally-reversible operations.
  - Devices supporting conditioned reversible operations.
  - Devices and circuits using the new model.

17

● The concept of "logically reversible computation" that has been used throughout the bulk of the reversible computing literature, from Landauer on, is simply ***the wrong one***!

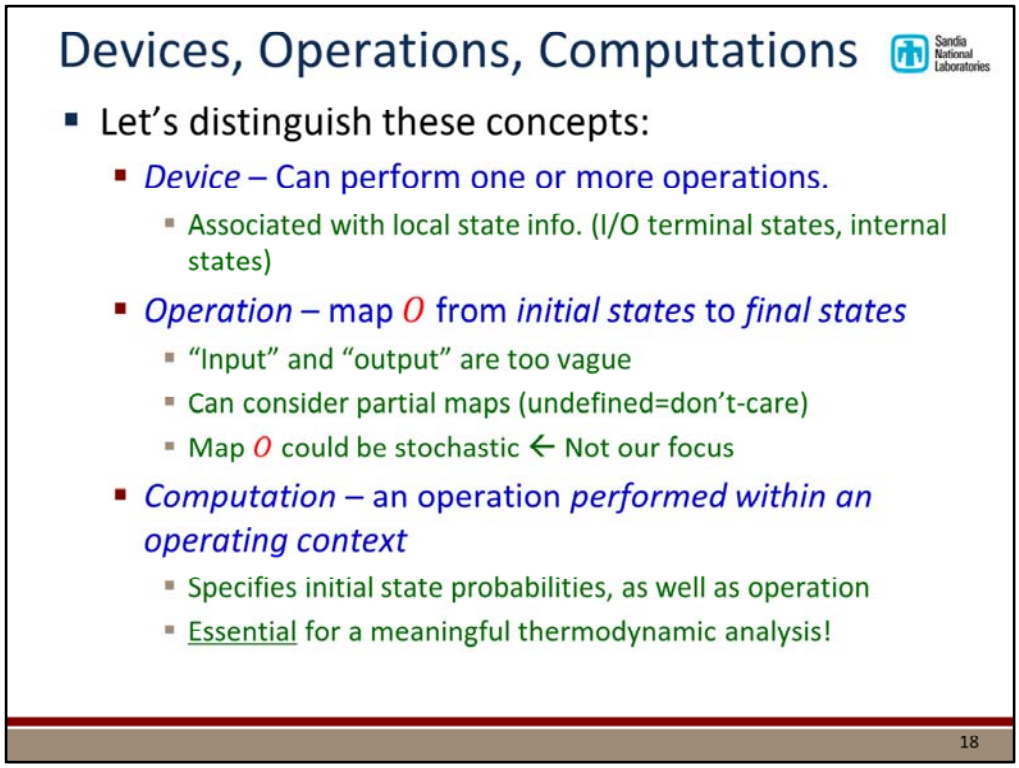
- In the sense that, it is significantly more restrictive than necessary.

● We need to *reconstruct* reversible computing theory from scratch, on top of a new, less restrictive foundation.

- Many applications of the theory will end up changing as a result!

● Some elements of the necessary conceptual progression:

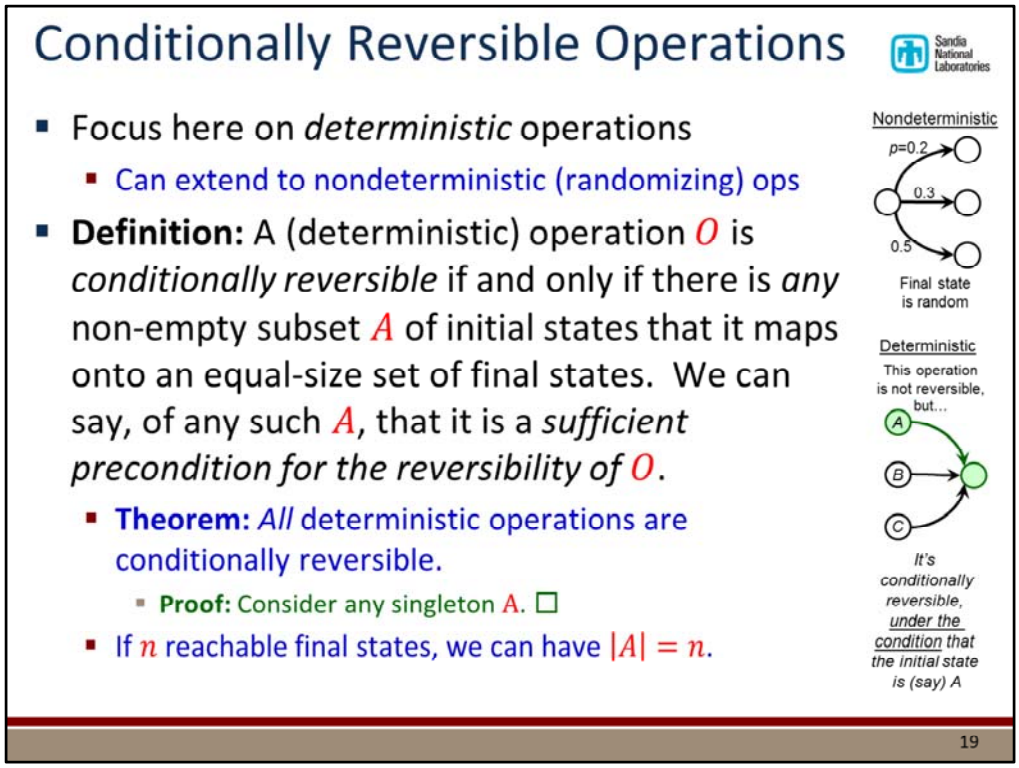
- Distinctions between devices, operations, and computations.
- Concept of devices supporting conditionally-reversible operations.
  - Crucially, the correct *general* concept of reversible computing *includes* computing with conditionally-reversible operations, when operated in design contexts in which their assumed preconditions are met.
- Thence, we can develop devices and circuits using the new model.
  - We'll see that it makes designs much simpler, and enables completely new styles of reversible circuits, such as asynchronous styles…



● One thing that is very helpful in understanding this issue, is distinguishing several fundamentally distinct concepts:

- A *device* – physical artifact that can perform one or more operations.
  - Associated with some local state info. (I/O terminal states, internal states)
- An *operation* – a mapping O transforming *initial states* to *final states*
  - The terms "input" and "output" are really too vague for many purposes
    Since real hardware devices may use some of their I/O terminals for *both* input and output functions (bidirectional), and some for neither, at times
  - We'll also consider partial maps (don't-care behavior in undefined cases)
  In general, the map O may be stochastic, but we'll not focus on that case
- A *computation* – an operation *performed within an operating context*
  - Specifies the initial state probabilities, as well as the operation performed
  - The probabilities are <u>essential</u> for a meaningful thermodynamic analysis!
    · Note that "entropy" always implicitly means "weighted-average entropy!" It's the expectation value of the log-improbability of the state:

$$S(p) = \mathrm{Ex}_p\left[\log\frac{1}{p}\right] = \sum_i p_i \log\frac{1}{p_i}$$



- We restrict attention, in this talk, to *deterministic* operations.
  - Nondeterministic (randomizing) operations raise other issues:
    - Carrying them out can actually *absorb* entropy from the environment
      Physical example: Paramagnetic cooling
    - Computations using operations that are *both* nondeterministic and logically irreversible can thus be thermodynamically reversible overall
      In the case where the initial state information was already truly random
    - Those points are all very interesting, but are not our present focus…
- **Definition:** A (deterministic) operation is *conditionally reversible* if and only if there is *any* non-empty subset S of initial states that it maps onto an equal-size set of final states. We say that such an operation is *conditionally reversible under the precondition that the initial state is in* S.
  - **Theorem:** *All* deterministic operations (that are defined over *any* non-empty set of initial states) are conditionally reversible.
    - **Proof:** Consider the singleton set consisting of any one initial state, out of those that the operation is defined over. Since the operation is deterministic, this set necessarily maps onto a singleton final state.
- This definition may therefore seem a bit vacuous at first,
  - but we'll see that in fact, it has enormous utility…

Operation Contexts, Computations, and Logical Reversibility (done right!)

- *Operation context* - a prob. Dist. p over initial comp. states.
  - A statistical situation in which op may be performed.
  - Has an associated entropy $S(p)$.
- A (deterministic) *computation* C is defined by a pair (O, p) of a deterministic operation O, and an operation context p.
  - This represents, performing the operation O *within the context* p.
  - O must be defined over at least all the nonzero-probability initial states
- **Definition:** A deterministic computation $C = (O, p)$ is *logically reversible* if and only if the set of all initial states that are assigned nonzero probability within the operating context p is a sufficient precondition for the reversibility of O.
  - **Theorem:** $C = (O, p)$ is logically reversible if and only if the entropy $S(p)$ is not changed under the state transformation O.

20

● An *operation context*, for our purposes, simply means a probability distribution p over initial computational states.

▪ It's just a statistical situation in which a given operation may be performed. It has an associated entropy $S(p)$.

● A (deterministic) *computation* C is defined by a pair (O, p) of a deterministic operation O, and an associated operation context p.

▪ This represents, performing the operation O *within the context* p. O must be defined over at least all the nonzero-probability initial states

● **Definition:** A deterministic computation $C = (O, p)$ is *logically reversible* (new definition!) if and only if the operation O is conditionally reversible under the precondition that the initial state is contained in the set of all states that are assigned nonzero probability within the operating context p.

▪ **Theorem:** $C = (O, p)$ is logically reversible (according to this definition) if and only if the entropy $S(p)$ is not changed under the state transformation O.

◦ As mentioned previously, the "if" part of this theorem wouldn't always hold in the nondeterministic case – since there are nondeterministic, irreversible operations that also don't change entropy in some operation contexts

◦ The "if" part also wouldn't hold under the conventional definition of "logically reversible," which fails to recognize that unconditional reversibility isn't required.

Now, we can say this:

- **Theorem:** A deterministic computation $C = (O, p)$ can be carried out in a thermodynamically reversible way (by some appropriately-designed mechanism) if <u>and only if</u> C is logically reversible (according to our new, corrected definition).
  - Proof by construction using known physical procedures
- The classic definition of logical reversibility is the ***<u>wrong one</u>***, because it does *not* actually satisfy the above theorem!!

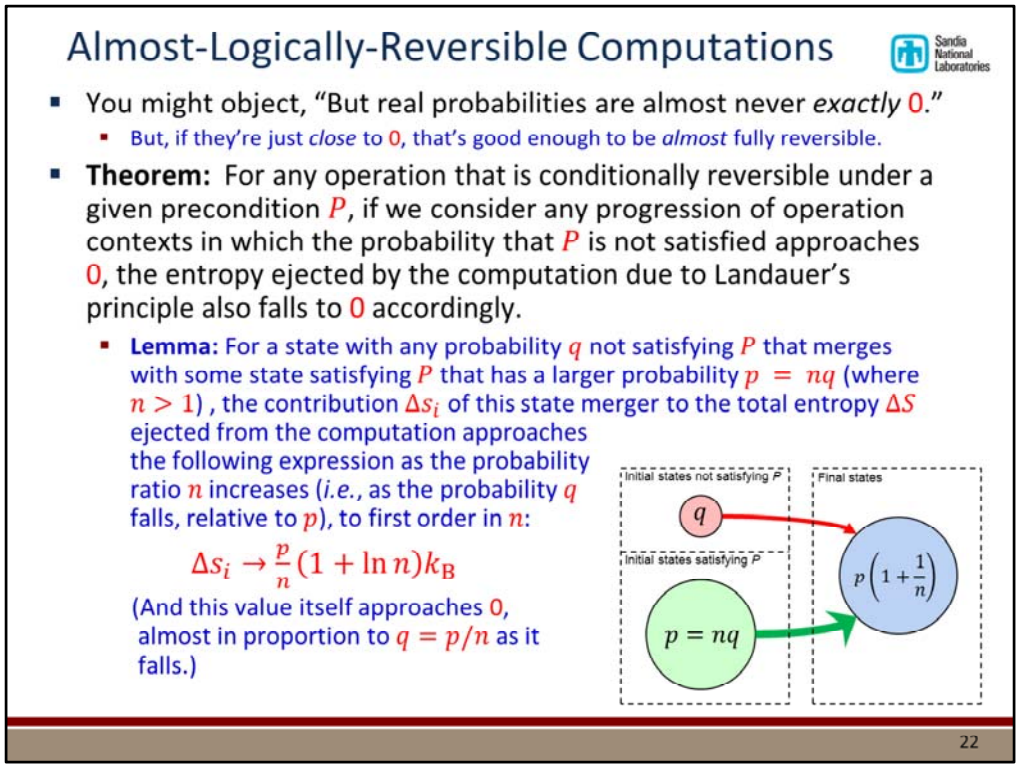
21

● **Theorem:** A deterministic computation $C = (O, p)$ can be carried out in a thermodynamically reversible way (by some appropriately-designed mechanism) if <u>and only if</u> C is logically reversible (according to our new, corrected definition).

- Proof is by construction using known abstract physical procedures
  - Still need to design specific concrete mechanisms with highest efficiency

● Note that the classic definition of logical reversibility that has been used throughout most of the reversible computing theory literature, starting with Landauer, is the ***<u>wrong one</u>***, because it does *not* actually satisfy the above theorem!!

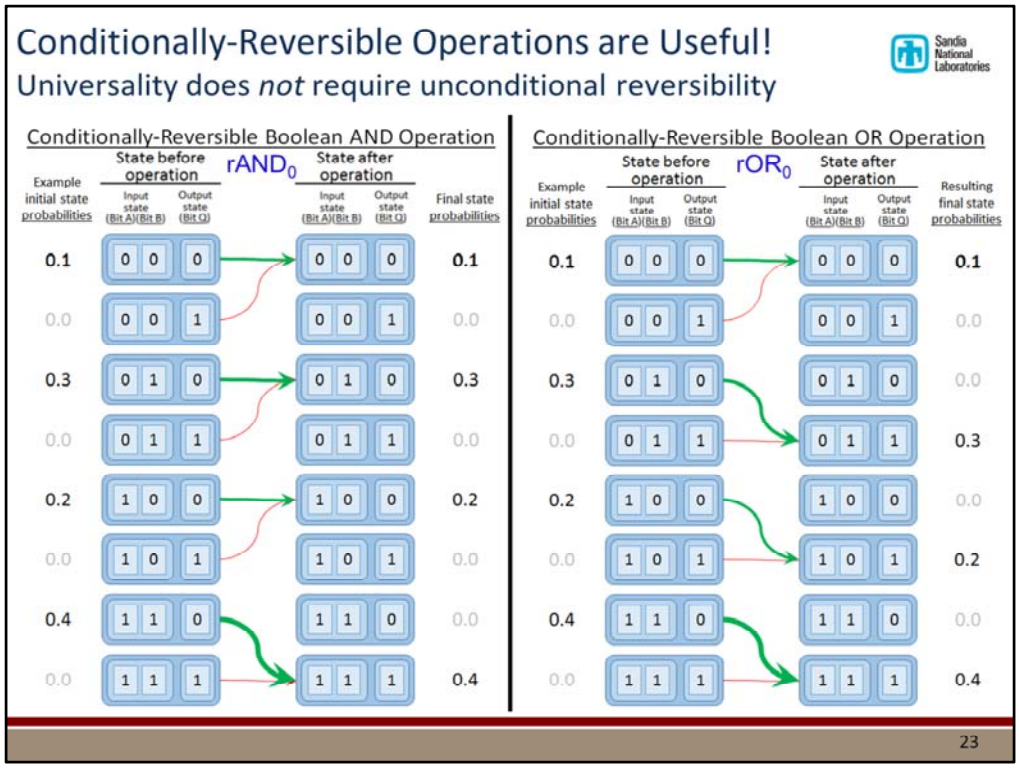
- The "only if" part of the theorem would not hold, because the traditional definition of logical reversibility fails to recognize that even operations that are only conditionally reversible can *also* be carried out in a thermodynamically reversible way, in operation contexts in which their precondition has probability 1 of being met.



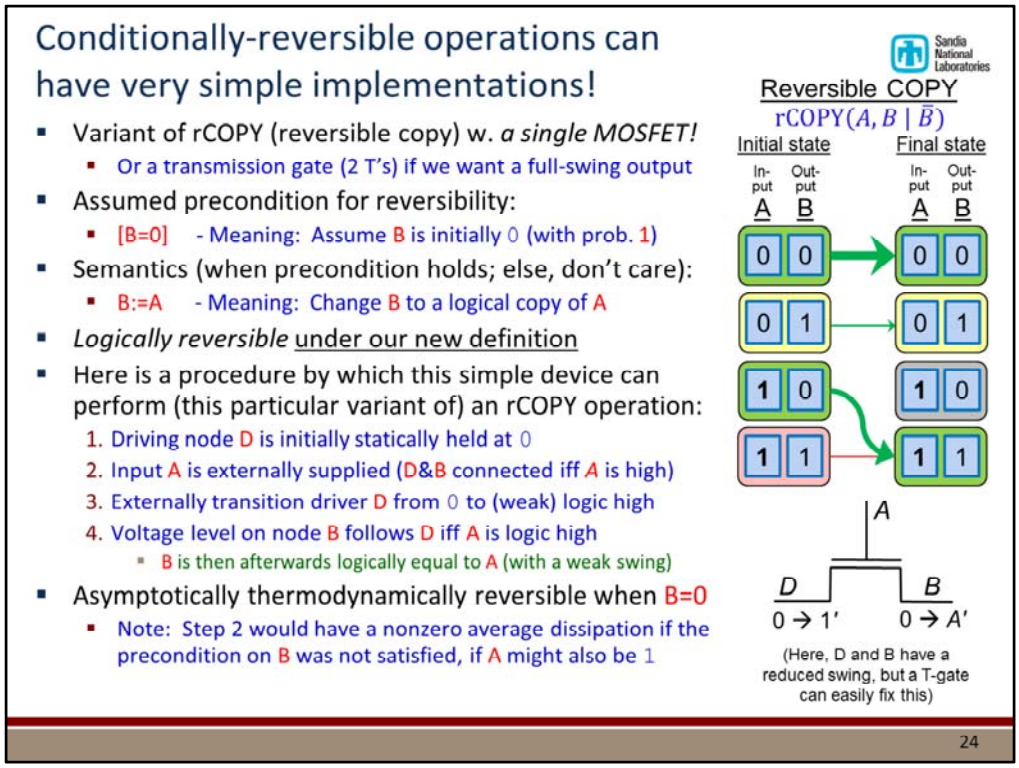
- You might object, "But real probabilities are almost never *exactly* 0."
  - But, that's OK… If they are *close* to 0, that's good enough to be *almost* fully reversible.
- **Theorem:** For any operation that is conditionally reversible under a given precondition P, if we consider any progression of operation contexts in which the probability that P is not satisfied 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 satisfying P that merges with some state satisfying P that has a larger probability $p = nq$ (where $n > 1$), the contribution Δs_i 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:

$$\Delta s_i \to \frac{p}{n}(1 + \ln n)k_B$$

    - (And this value itself approaches 0, almost in proportion to $q = p/n$ as it falls.)
  - Similarly, states with probability q not satisfying P that merge with each other (but not with any state satisfying P) each contribute $\Delta s_i \leq q \log q^{-1}$ to ΔS, and this also goes to 0 as $q \to 0$.
  - Thus, as $\sum q \to 0$, so does ΔS. □



Conditionally-reversible versions of Boolean AND and OR operations. In both of these examples, the assumed precondition for reversibility is that the output bit is initially 0.



● A variant of the rCOPY (reversible copy) operation from earlier can be performed by *a single MOSFET!*

▪ Or a transmission gate (2 T's) if we want a full-swing output

● Assumed precondition for reversibility:

▪ [B=0] - Meaning: Assume B is initially 0 (with probability 1)

● Operation semantics (when precondition holds; else, don't care):

▪ B:=A - Meaning: Change B to a logical copy of A

● A computation "[B=0] B:=A" (designating, doing this operation in any context satisfying the precondition in brackets) is *logically reversible* <u>under our new definition</u>

● Here is a procedure by which this simple device can perform (this particular variant of) an rCOPY operation:

1. Driving node D is initially statically held at 0
2. Input A is externally supplied (D&B are connected if and only if *A* is high)
3. Externally transition driving node D from 0 to (weak) logic high
4. Voltage level on node B follows D iff A is logic high

   B is then afterwards logically equal to A (with a weak swing)

● Note this process is asymptotically thermodynamically reversible in all operating contexts that satisfy the precondition [B=0], in the limit of relatively large devices (low leakage) and slow transitions

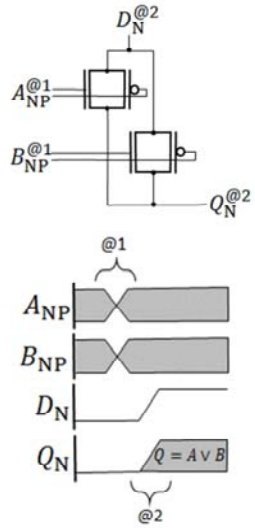
▪ The traditional definition of logical reversibility <u>fails</u> to account for the fact that this process is, in fact, physically reversible in this context!

● Note: Step 2 would have a nonzero average dissipation if the precondition on B was not satisfied, if A might also be 1

Another example: Reversible OR

- In this design, we use CMOS transmission gates (parallel complementary MOSFETs) to ensure the output levels are full-swing
  - All signals are dual-rail (complementary wires)
  - Use circuit twice to produce dual-rail output
- Computation sequence:
  1. Precondition: Output signal *Q* initially at logic 0
  2. Driving signal *D* is also initially logic 0
  3. 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
  4. 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

(This is also AND if we flip polarities)



25

● In this design, we use CMOS transmission gates (parallel complementary MOSFETs) to ensure the output levels are full-swing

▪ All signals are dual-rail (complementary wires)

Use circuit twice to produce dual-rail output

● Computation sequence:

1. Precondition: Output signal *Q* initially at logic 0
2. Driving signal *D* is also initially logic 0
3. 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
4. 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

Unconditionally-Reversible Operations are only a special case!

More critiques of Landauer '61…

- Landauer introduced what's now called a *Toffoli gate operation*, or controlled-controlled-NOT, an unconditionally logically reversible operation:
  $r = r \oplus pq$.
- Landauer describes (correctly) that AND can be embedded into this operation. (Given initial $r = 0$)
- However, his statement here that the AND operation "is not, in itself, reversible" is somewhat misleading!
  - That would only be true if:
    - The input bits were *consumed*…
      - But, CMOS gates never consume inputs!
    - Or, if the output bit was *destructively overwritten* with the result

> Consider, for example, a particular three-input, three-output device, i.e., a small special purpose computer with three bit positions. Let p, q, and r be the variables before the machine cycle. The particular truth function under consideration is the one which replaces r by $p \cdot q$ if $r = 0$, and replaces r by $\overline{p \cdot q}$ if $r = 1$. The variables p and q are left unchanged during the machine cycle. We can consider r as giving us a choice of program, and p, q as the variables on which the selected program operates. This is a logically reversible device, its output always defines its input uniquely. Nevertheless it is capable of performing an operation such as AND which is not, in itself, reversible.

- The approach Landauer takes here, of XOR'ing the result into the output bit, is indeed one that is logically reversible in *all* operation contexts.
  - But, it is rather complex to implement…
    - The simpler, conditionally-reversible setting of the output also works fine, in suitably restricted contexts!

26

More critiques of Landauer '61…

● Here, Landauer introduced what we now refer to as the *Toffoli gate operation*, or controlled-controlled-NOT, an unconditionally logically reversible operation:

$r = r \oplus pq$.

● Landauer describes (correctly) that AND can be embedded into this operation. (Given initial $r = 0$)

● However, his statement here that the AND operation "is not, in itself, reversible" is somewhat misleading!

- That would <u>only</u> be true if:
  - The input bits were *consumed*…
    - But, CMOS gates never consume inputs!
  - Or, if the output bit was *destructively overwritten* with the result
    - Merging nonzero-probability states
    - As opposed to, being transformed, in a logically reversible way, to the result
    - When the output bit is destructively overwritten, even NOT is irreversible!

● The approach Landauer takes here, of XOR'ing the result into the output bit, is indeed one that is logically reversible in *all* operation contexts.

But, it is rather complex to implement…

The simpler, conditionally-reversible setting of the output also works fine, in suitably restricted contexts!

Landauer did not consider this.

All truly, fully adiabatic circuits are (at least) conditionally reversible!

- "Dry switching" rules for designing truly adiabatic circuits:
  - Never close a switch when there's a voltage ≠0 between its terminals
    - E.g., don't turn on a transistor when $V_{DS} \neq 0$.
  - Never open a switch when there's a current passing through it.
    - E.g., don't turn off a transistor when $I_{DS} \neq 0$.
    - Only exception to this rule: If there's an alternate path for the current.
  - Never pass current through diodes (which have a voltage drop)
- Violating any of these rules leads to significant dissipation!
- **Theorem:** The operation of a switching circuit carries out a (conditionally) logically reversible computation, in any operation context where the above rules are always satisfied.
  - *It's impossible to erase information in any truly, fully adiabatic logic operation.* → Logically-reversible computing is key to adiabatic design
    - But the *right* definition of "logically reversible" <u>is our generalized one</u>!

27

- "Dry switching" rules for designing truly adiabatic circuits:
  - Never close a switch when there's a voltage ≠0 between its terminals
    - E.g., don't turn on a transistor when $V_{DS} \neq 0$.
  - Never open a switch when there's a current passing through it.
    - E.g., don't turn off a transistor when $I_{DS} \neq 0$.
    - Only exception to this rule: If there's an alternate path for the current.
  - Never pass current through diodes (which have a voltage drop)
- Violating any of these rules leads to significant dissipation!
- **Theorem:** The operation of a switching circuit carries out a (conditionally) logically reversible computation, in any operation context where the above rules are always satisfied.
  - *It's impossible to erase information in any truly, fully adiabatic logic operation.* → Logically-reversible computing is key to adiabatic design
    - But the *right* definition of "logically reversible" <u>is our generalized one</u>!

The "synchrony curse" in traditional reversible design

- All traditional, unconditionally-reversible operations (*e.g.*, Toffoli gates) are implicitly *synchronous* in their design…
  - Assume that all gate inputs are available at the same time
  - Requires extensive clock distribution (in adiabatic implementations), or unrealistically-precise timing (in ballistic implementations)
    - Failure to meet timing assumptions generally leads to irreversibility
  - ∴ … Even our supposedly "unconditionally reversible" gate designs are *actually* only *conditionally* reversible, because we always implicitly assume the precondition that *their timing assumptions are in fact met.*
- Since our real-world reversible gate implementations are *really* only conditionally-reversible anyway,
  - can we come up with timing-related preconditions that are easier to meet than the usual full-synchrony assumption?

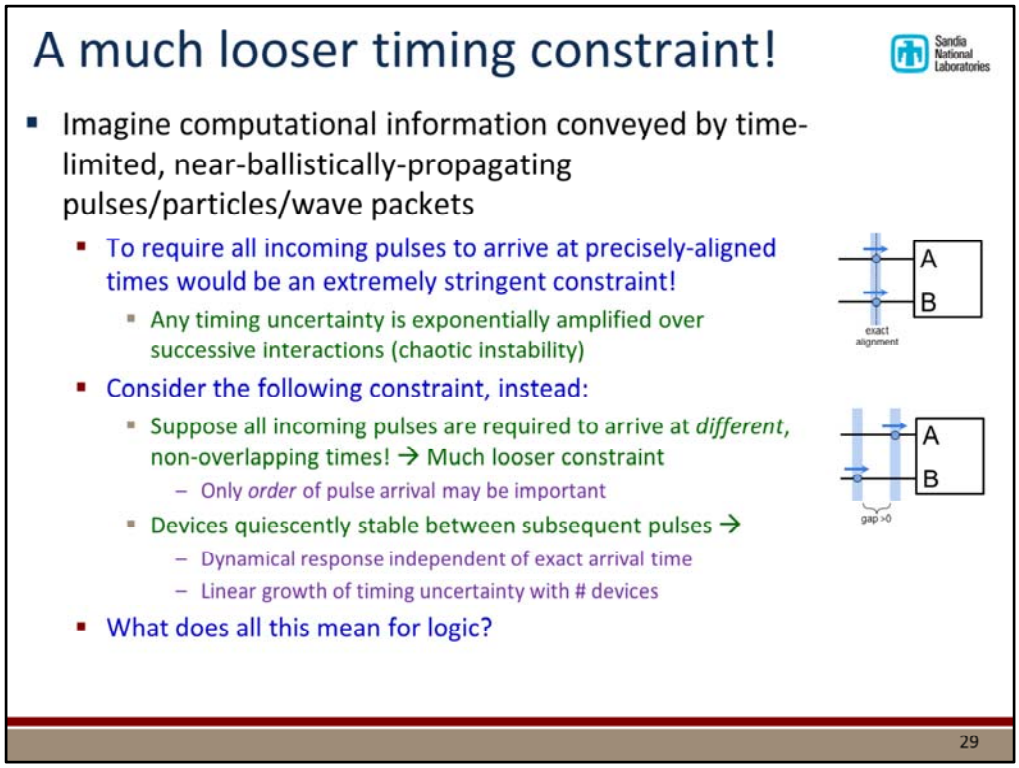
28

● All of the traditional, unconditionally-reversible operations (*e.g.*, Fredkin & Toffoli gates) are implicitly *synchronous* in their design…

- Assume that all gate inputs are available at the same time
- Requires extensive clock distribution (in adiabatic implementations), or unrealistically-precise timing (in ballistic implementations)
  - Failure to meet timing assumptions generally leads to irreversibility
- ∴ … Even our supposedly "unconditionally reversible" gate designs are *actually* only *conditionally* reversible, because we always implicitly assume the precondition that *their timing assumptions are in fact met.*

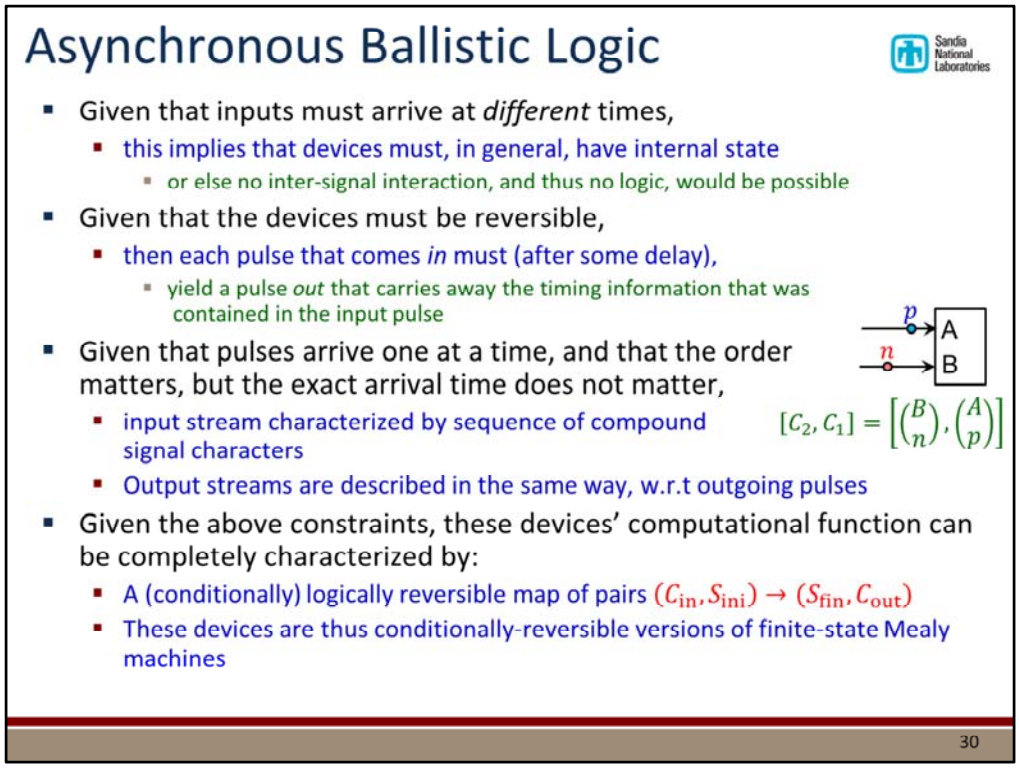
● Since our real-world reversible gate implementations are *really* only conditionally-reversible anyway,

- can we come up with timing-related preconditions that are easier to meet than the usual full-synchrony assumption?



● Imagine a scenario in which computational information is conveyed by time-limited, near-ballistically-propagating pulses/particles/wave packets

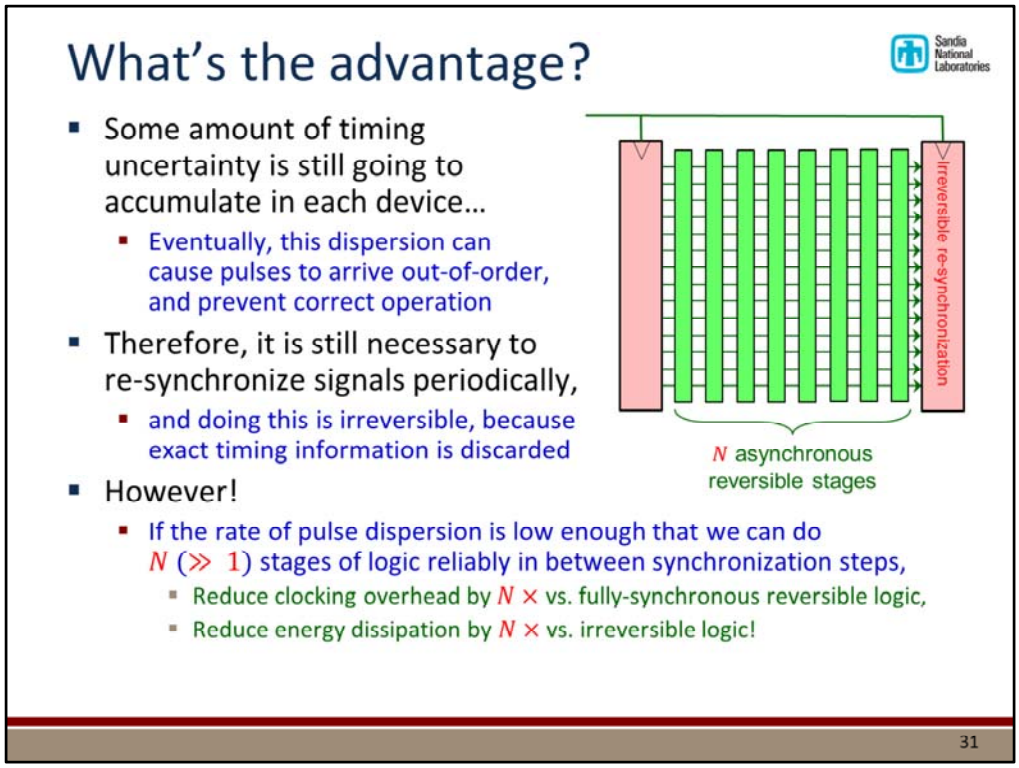
- To require all incoming pulses to arrive at precisely-aligned times would be an extremely stringent constraint!
  - Any uncertainty in the relative arrival times of pulses would generally lead to exponential amplification of uncertainties over successive interactions (chaotic instability)
    - Thus, entropy would increase during the interaction (historically, this observation goes all the way back to Boltzmann's H-theorem)
- Consider the following constraint, instead:
  - Suppose all incoming pulses are required to arrive at *different*, non-overlapping times!
    - Much looser constraint
    - Instead, we might only require a specific relative *order* of arrival of pulses (e.g., first a pulse on terminal A, then one on terminal B)
  - If devices are just quiescently stable in between subsequent pulses, then the dynamical response of a device to a pulse's arrival is independent of the exact arrival time of the pulse
    - We may plausibly expect that any increases in the timing uncertainty can be more easily constrained to be limited (growing linearly rather than geometrically, say) due to this time-independence
- What are the logical implications of this new, looser constraint?



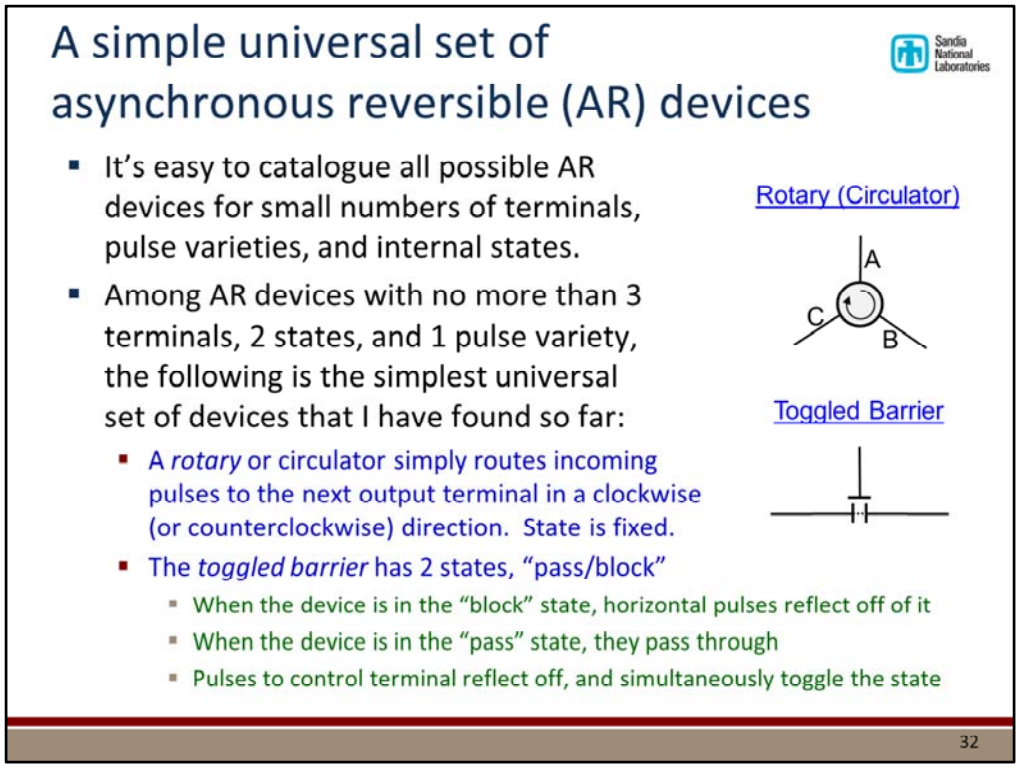
- Given that inputs must arrive at *different* times,
  - this implies that devices must, in general, have internal state
    - or else no inter-signal interaction, and thus no logic, would be possible
- Given that the devices must be reversible,
  - then each pulse that comes *in* must (after some delay),
    - yield a pulse *out* that carries away the timing information that was contained in the input pulse
- Given that pulses arrive one at a time, and that the order matters, but the exact arrival time does not matter,
  - an input stream can be characterized by a sequence of compound signal characters (read here from right to left, imagining the data is flowing towards the right)

$$[C_k, \dots, C_2, C_1] = \left[\binom{T_k}{V_k}, \cdots, \binom{T_2}{V_2}, \binom{T_1}{V_1}\right]$$

  where each C_i designates which terminal T_i the next incoming pulse arrives on, and (if there is more than one variety of pulse), which variety V_i of pulse is arriving. Output streams are described in the same way, w.r.t outgoing pulses
- Given the above constraints, these devices' computational function can be completely characterized by:
  - A (conditionally) logically reversible map of pairs $(C_{\text{in}}, S_{\text{ini}}) \to (S_{\text{fin}}, C_{\text{out}})$
    - Where the S's are device states (initial vs. final), and the C's are signal characters (incoming vs. outgoing). (Note that we're allowing here that terminals may, in general, be bidirectional)
  - These devices are thus conditionally-reversible versions of finite-state Mealy machines



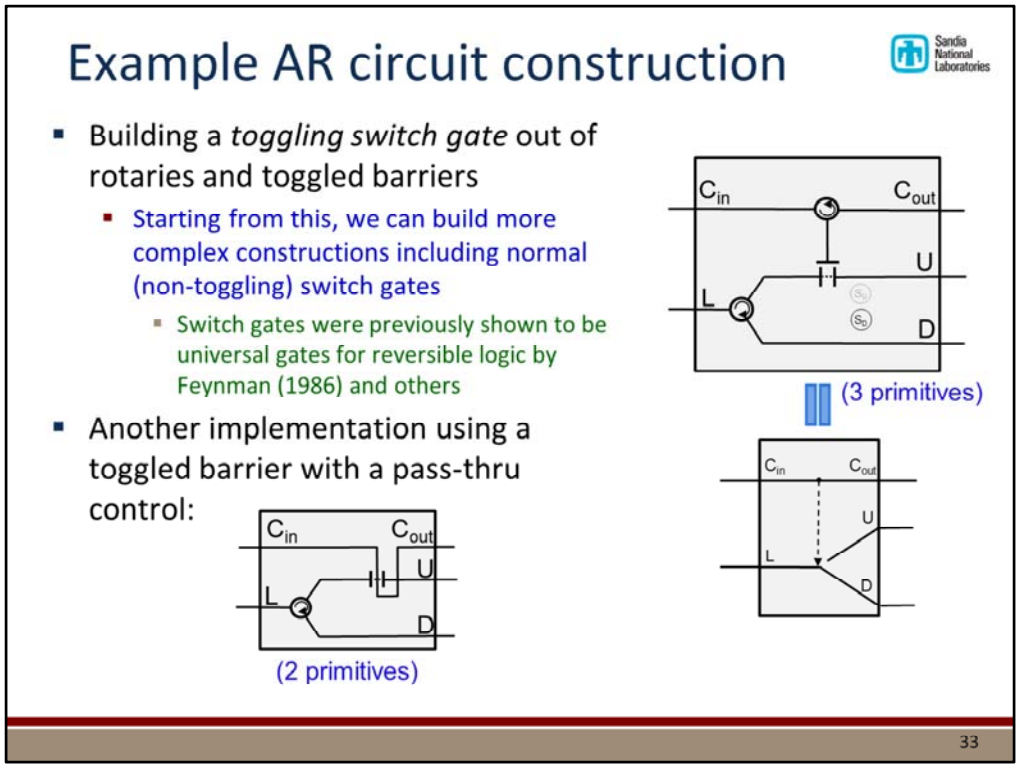
- Some amount of timing uncertainty is still going to accumulate in each device…
  - Eventually, this dispersion can cause pulses to arrive out-of-order, and prevent correct operation
- Therefore, it is still necessary to re-synchronize signals periodically,
  - and doing this is irreversible, because exact timing information is discarded
- However!
  - If the rate of pulse dispersion is low enough that we can do N ($\gg 1$) stages of logic reliably in between synchronization steps,
    - then we can reduce clocking overhead by a factor of $N \times$ compared to fully-synchronous reversible logic,
    - while reducing energy dissipation by $N \times$ compared to irreversible logic!



● It's easy to catalogue all possible AR devices for small numbers of terminals, pulse varieties, and internal states.

● Among AR devices with no more than 3 terminals, 2 states, and 1 pulse variety, the following is the simplest universal set of devices that I have found so far:

- A *rotary* or circulator simply routes incoming pulses to the next output terminal in a clockwise (or counterclockwise) direction. State is fixed.
- The *toggled barrier* has 2 states, "pass/block"
  - When the device is in the "block" state, horizontal pulses reflect off of it
  - When the device is in the "pass" state, they pass through
  - Pulses to control terminal reflect off, and simultaneously toggle the state



- Building a *toggling switch gate* out of rotaries and toggled barriers
  - Starting from this, we can build more complex constructions including normal (non-toggling) switch gates
    - Switch gates were previously shown to be universal gates for reversible logic by Feynman (1986) and others
- Another implementation using a toggled barrier with a pass-thru control:

Possible implementation technologies

- The pulses in ARC might be implemented by things like:
  - SFQ pulses in passive long Josephson junction (LJJ) transmission lines
  - More generally, soliton-like excitations of any nonlinear medium
  - Single particles or quasiparticles (*e.g.* excess electrons or excitons) propagating ballistically in suitable media (vacuum or crystal)
  - Optical pulses in some suitable medium
  - Electrical pulses in coaxial transmission lines
- Circulators already exist for microwave circuits and SFQ
- Still-open problem:
  - How exactly to build a toggled barrier, or any other simple AR device(s) sufficient for universal reversible logic?
- For any AR devices, an engineering challenge will be to get the pulse dispersion as low as possible in devices and wires
  - Reliability of the logical operation is also important to maximize
    - Error correction can be done, but is inherently irreversible

34

- The pulses in ARC might be implemented by things like:
  - SFQ pulses in passive long Josephson junction (LJJ) transmission lines
  - More generally, soliton-like excitations of any nonlinear medium
  - Single particles or quasiparticles (*e.g.* excess electrons or excitons) propagating ballistically in suitable media (vacuum or crystal)
  - Optical pulses in some suitable mediuma
  - Electrical pulses in coaxial transmission lines
- Circulators already exist for microwave circuits and SFQ
- Still-open problem:
  - How exactly to build a toggled barrier, or any other simple AR device(s) sufficient for universal reversible logic?
- For any AR devices, an engineering challenge will be to get the pulse dispersion as low as possible in devices and wires
  - Reliability of the logical operation is also important to maximize
    - Error correction can be done, but is inherently irreversible

Conclusions

- Traditional formalizations of reversible logic are an inadequate theoretical foundation for engineering real reversible hardware
  - The classic definition of *logical reversibility* is unnecessarily restrictive
    - Must be extended to more general notion of *conditionally logically reversible operations*, and *logically reversible computations* meeting the conditions
- The resulting new theoretical model of Generalized Reversible Computing (GRC) offers many advantages over the old model:
  - Rigorous correspondence to asymptotically thermodynamically reversible physical computing mechanisms
  - Foundation for designing much simpler primitive operations out of which more efficient reversible logic architectures may be constructed
  - GRC also is general enough to allow us to build on top of it a new framework for Asynchronous Reversible Computing (ARC),
    - Avoids many of the overheads incurred in clocking synchronous designs
- GRC is an important step towards making reversible computing practical, which is essential for the future of computing

35

● The traditional formalizations of reversible logic going all the way back to Landauer, and further developed by Fredkin and Toffoli, do not, in their existing form, comprise an adequate theoretical foundation for the engineering of real reversible hardware

- The classic definition of *logical reversibility* is unnecessarily restrictive
  - The classic concept of "logically reversible device operations" must be extended to encompass the more general notion of *conditionally logically reversible operations*, and *logically reversible computations* that meet the conditions
    - These offer more flexibility for hardware implementations, while still avoiding incurring any minimum dissipation from Landauer's principle

● The resulting new theoretical model of Generalized Reversible Computing (GRC) offers many advantages over the old model:

- It offers a precise, rigorous correspondence to the set of asymptotically adiabatic (thermodynamically reversible) physical computing mechanisms
- It provides a foundation for designing much simpler primitive operations out of which more efficient reversible logic architectures may be constructed
- GRC also is general enough to allow us to build on top of it a new framework for Asynchronous Reversible Computing (ARC),
  - which avoids many of the overheads incurred in clocking synchronous designs

● The reversible computing community really needs to embrace GRC (and models based on it) as the right foundation for further progress!