COSC 494/594
Unconventional Computation

Introduction

Course Information

- Instructor: Bruce MacLennan
- Course website: web.eecs.utk.edu/~mclennan/Classes/494-UC or 594-UC
- Email: maclennan@eecs.utk.edu
- Prereqs: linear algebra (basic CS, physics)
- Grading:
  - Homework (every week or two)
  - Project or two
  - Term paper (on some kind of unconventional computation)
  - Presentation (for 594)

Course Outline

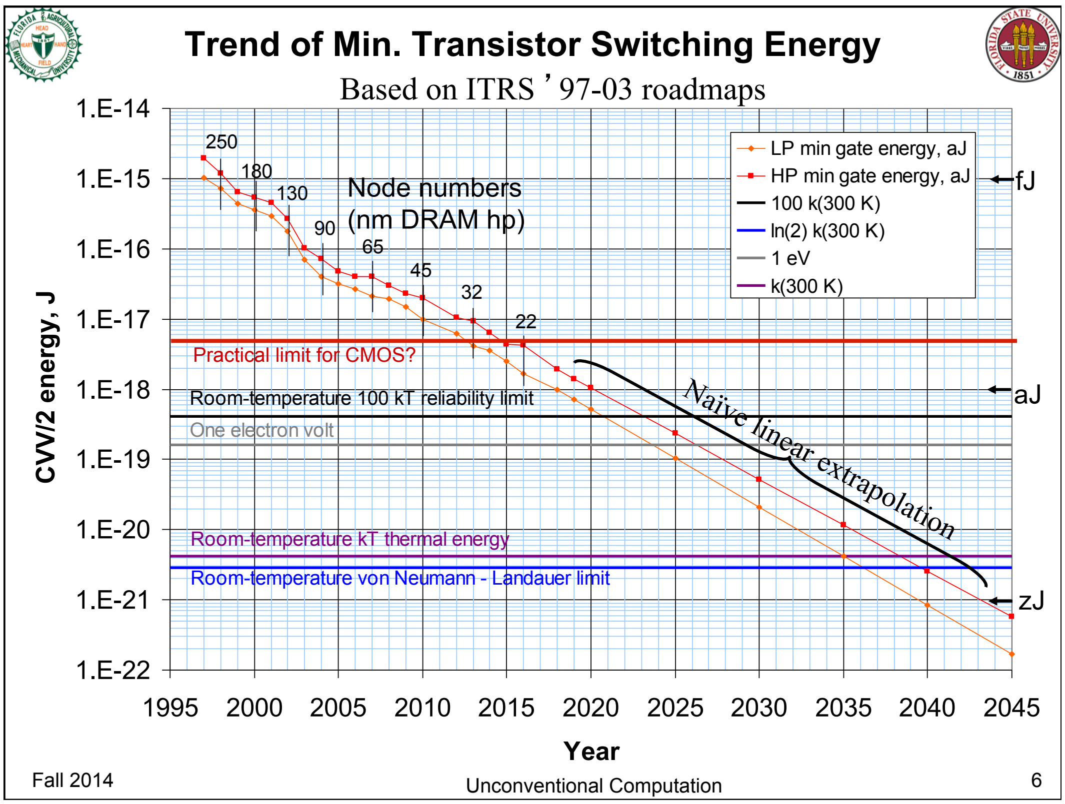
I. Introduction
II. Physics of computation
III. Quantum computation
IV. Molecular computation
V. Analog computation?
VI. Grad presentations on other unconventional computing paradigms

Unconventional Computation

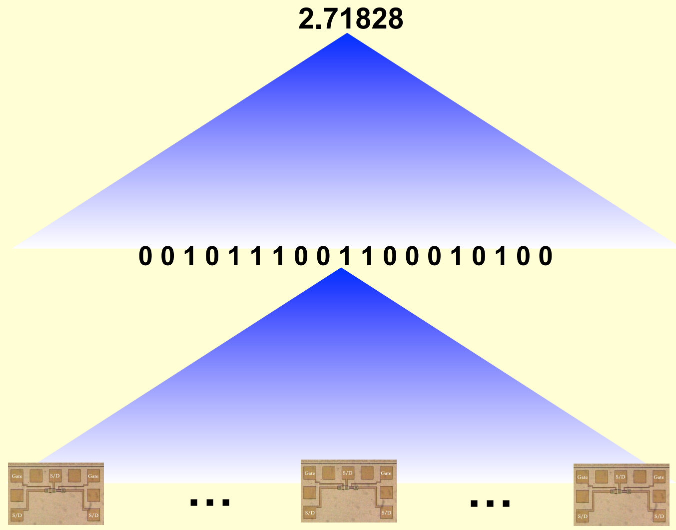
- Unconventional (or non-standard) computation refers to the use of non-traditional technologies and computing paradigms
  - Why would you want to do this?
- Hypercomputation or super-Turing computation refers to computation "beyond the Turing limit"
  - Is this possible?

Post-Moore's Law Computation

- The end of Moore's Law is in sight!
- Physical limits to:
  - density of binary logic devices
  - speed of operation
- Requires a new approach to computation
- Significant challenges
- Will broaden & deepen concept of computation in natural & artificial systems

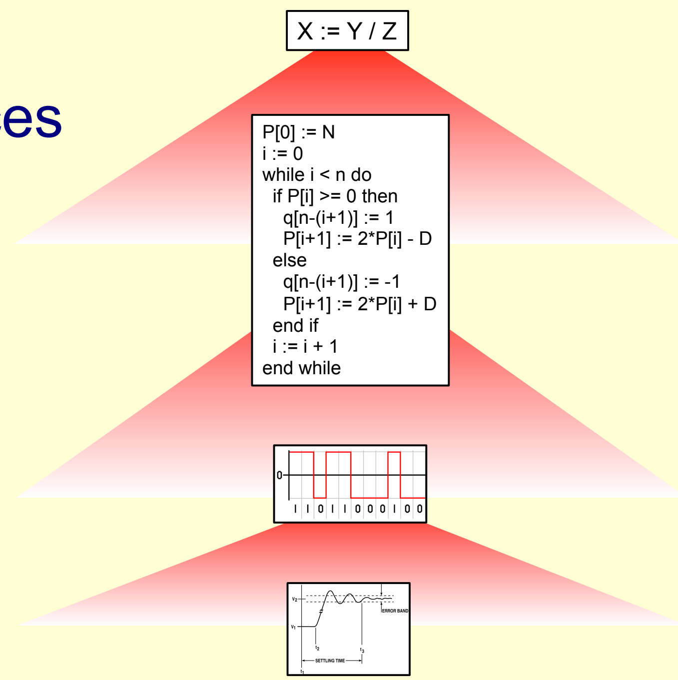


Differences in Spatial Scale



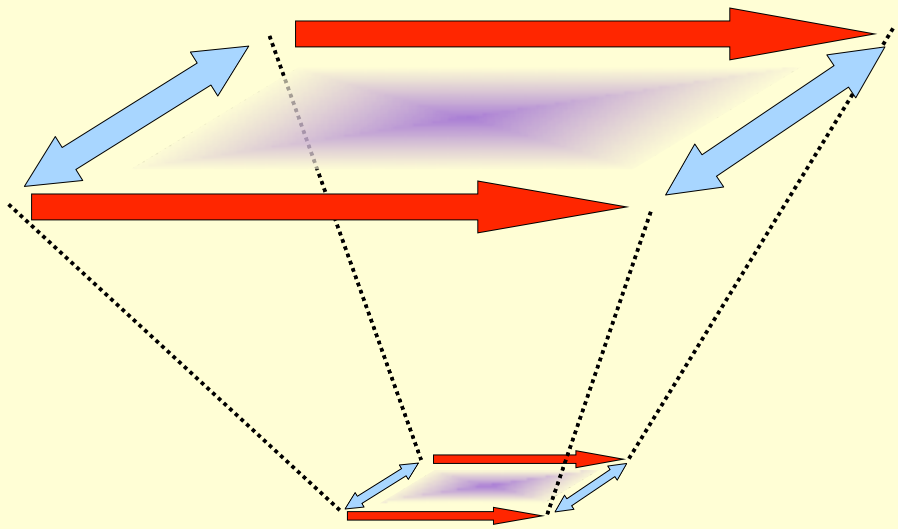
(Images from Wikipedia)

Differences in Time Scale



(Images from Wikipedia)

Convergence of Scales



Implications of Convergence

- Computation on scale of physical processes
- Fewer levels between computation & realization
- Less time for implementation of operations
- Computation will be more like underlying physical processes
- Post-Moore’s Law computing ⇒ greater assimilation of computation to physics

Computation is Physical

"Computation is physical; it is necessarily embodied in a device whose behaviour is guided by the laws of physics and cannot be completely captured by a closed mathematical model. This fact of embodiment is becoming ever more apparent as we push the bounds of those physical laws."

— Susan Stepney (2004)

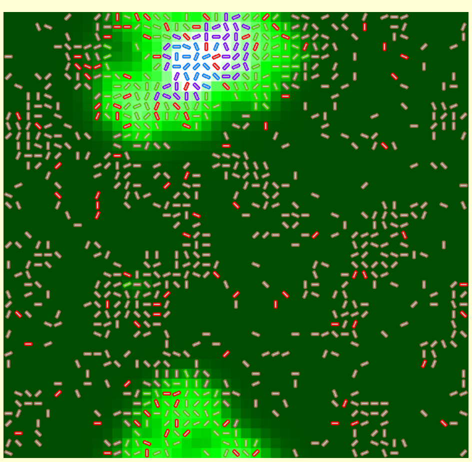
Cartesian Duality in CS

- Programs as idealized mathematical objects
- Software treated independently of hardware
- Focus on *formal* rather than *material*
- Post-Moore's Law computing:
  - less idealized
  - more dependent on physical realization
- More difficult
- But also presents opportunities…

Strengths of "Embodied Computation"

- Information often implicit in:
  - its physical realization
  - its physical environment
- Many computations performed "for free" by physical substrate
- Representation & information processing emerge as regularities in dynamics of physical system

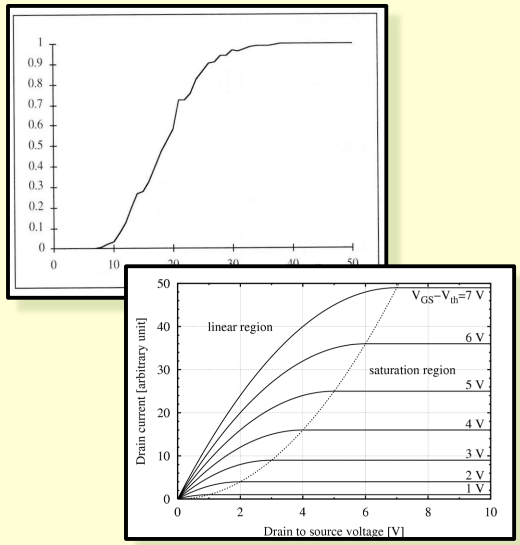
Example: Diffusion



- Occurs naturally in many fluids
- Can be used for many computational tasks
  - broadcasting information
  - massively parallel search for optimization, constraint satisfaction etc.
- Expensive with conventional computation
- Free in many physical systems

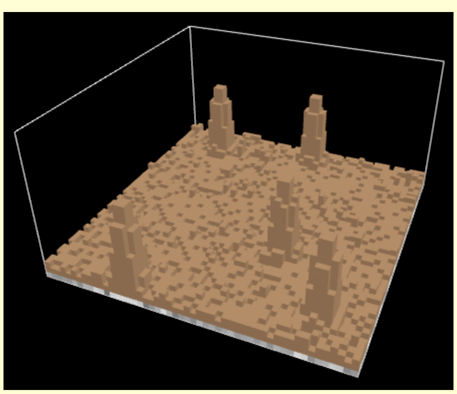
Example: Saturation

- Sigmoids in ANNs & universal approx.
- Many physical systems have sigmoidal behavior
  - Growth process saturates
  - Resources become saturated or depleted
- Embodied computation uses free sigmoidal behavior



(Images from Bar-Yam & Wikipedia)

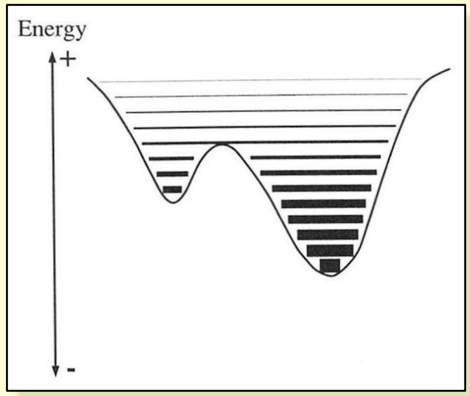
Example: Negative Feedback



- Positive feedback for growth & extension
- Negative feedback for:
  - stabilization
  - delimitation
  - separation
  - creation of structure
- Free from
  - evaporation
  - dispersion
  - degradation

Example: Randomness

- Many algorithms use randomness
  - escape from local optima
  - symmetry breaking
  - deadlock avoidance
  - exploration
- For free from:
  - noise
  - uncertainty
  - Imprecision
- "Free variability"



(Image from Anderson)

"Respect the Medium"

- Conventional computer technology "tortures the medium" to implement computation
- Embodied computation "respects the medium"
- Goal of embodied computation:

  Exploit the physics, don't circumvent it

But is it Computing?

Some Non-Turing Characteristics of Embodied Computation

- Operates in real time and real space
  - with real matter and real energy
  - and hence non-ideal aspects of physical realization
- Often does not terminate
- Often has no distinct inputs or outputs
- Often purpose is not to get an answer from an input
- Often purpose is not to control fixed agent
- Different notions of equivalence and universality

Is EC a Species of Computing?

- The Turing Machine provides a precise definition of computation
- Embodied computation may seem imprecise
- & difficult to discriminate from other physical processes
- Expanding concept of computation beyond TM requires an expanded definition

Non-Turing Computation

Frames of Relevance

- CT computation is a *model* of computation
- All models have an associated *frame of relevance*
  - determined by model's simplifying assumptions
  - by aspects & degrees to which model is similar to modelled system
- Determine questions model is suited to answer
- Using outside FoR may reflect model & simplifying assumptions more than modelled system

Models & Simplifying Assumptions

- Turing computation is a *model* of computation
- A model is like its subject in relevant ways
- Unlike it in irrelevant ways
- A model is suited to pose & answer certain classes of questions
- Thus every model exists in a *frame of relevance* (FoR)
- FoR defines domain of reliable use of model

Example: FoR of Maps

The FoR of Turing Computation

- *Historical roots*: issues of formal calculability & provability in axiomatic mathematics; hence:
  - finite number of steps & finite but unlimited resources
  - computation viewed as function evaluation
  - discreteness assumptions

Idealizing Assumptions

- Finite but unbounded resources
- Discreteness & definiteness
- Sequential time
- Computational task = evaluation of well-defined function
- Computational power defined in terms of sets of functions

Alternate Frames of Relevance for Expanded Notions of Computation

- Natural Computation
  - applying natural processes in computation
  - alternative realizations of formal processes
- Nanocomputation
  - direct realizations of non-Turing computations
  - unique characteristics
- Quantum & Quantum-like Computation

Natural Computation

- *Natural computation* = computation occurring in nature or inspired by it
- Occurs in nervous systems, DNA, microorganisms, animal groups
- Good models for robust, efficient & effective artificial systems (autonomous robots etc.)
- Different issues are relevant

Relevant Issues Outside TC FoR

- Real-time control
- Continuous computation
- Robustness
- Generality, flexibility & adaptability
- Non-functional computation

Relevant Issues Outside TC FoR

- Error, noise & uncertainty are unavoidable
  - must be part of model of computation
  - may be used productively
- Microscopic reversibility may occur
  - e.g., reversible chemical reactions
  - want statistical or macroscopic progress
- Computation proceeds asynchronously in continuous-time parallelism

Real-Time Control

- Real-time (RT) response constraints
- Asymptotic complexity is usually irrelevant
  - Input size typically constant or of limited variability
  - Computational resources are bounded
- Relevant: relation of RT response rate to RT rates of its components

Continuous Computation

- Inputs & outputs often:
  - Are continuous quantities
  - Vary continuously in real time
- Computational processes often continuous
- More or less powerful than TMs?
- Obviously can be approximated by discrete quantities varying at discrete times, *but* …

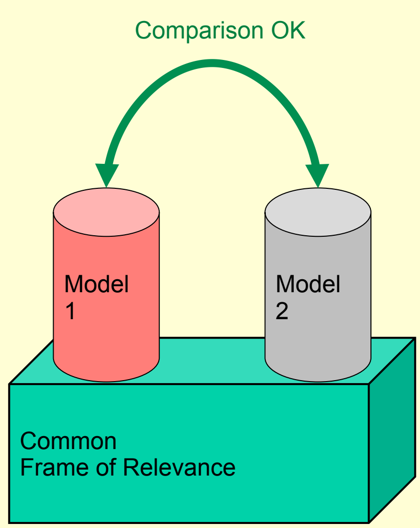
"Metaphysics" of Reals

- "Metaphysical issues":
  - Turing-computable reals vs. standard reals
  - Standard reals vs. non-standard reals
- Results depend on "metaphysical issues" ⇒ outside FoR of model
- Naïve real analysis is sufficient for models of natural computation

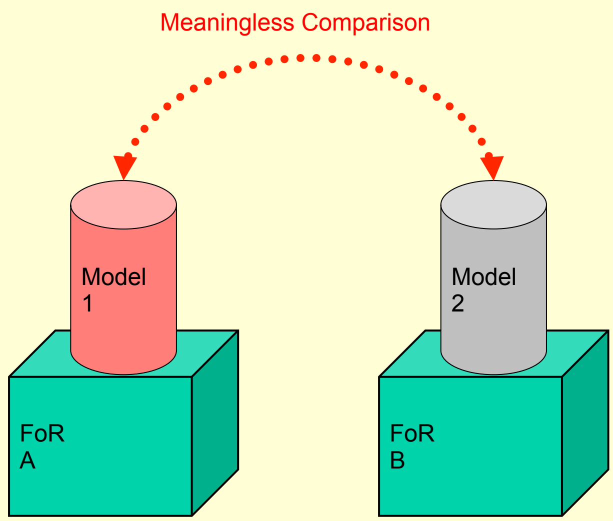
Cross-Frame Comparisons

- Can we compare models with different FoRs?
- Yes: can translate one to other's FoR
- Typically make incompatible simplifying assumptions
- Results may depend on specifics of translation
- E.g., how are continuous quantities represented in TC?

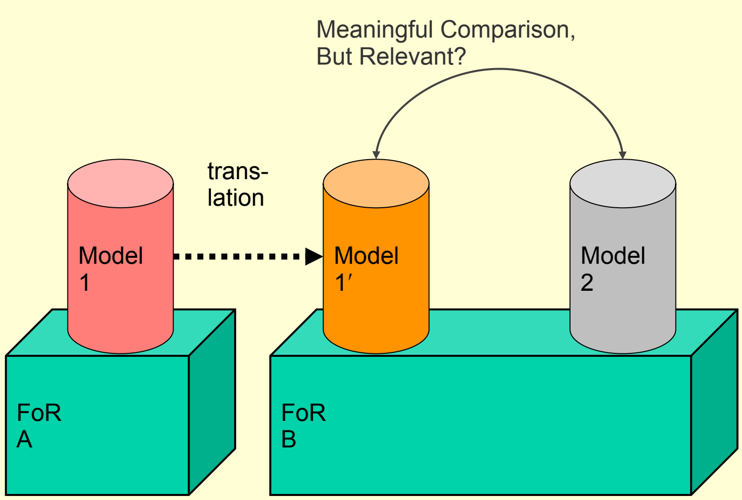
Within-Frame Comparison



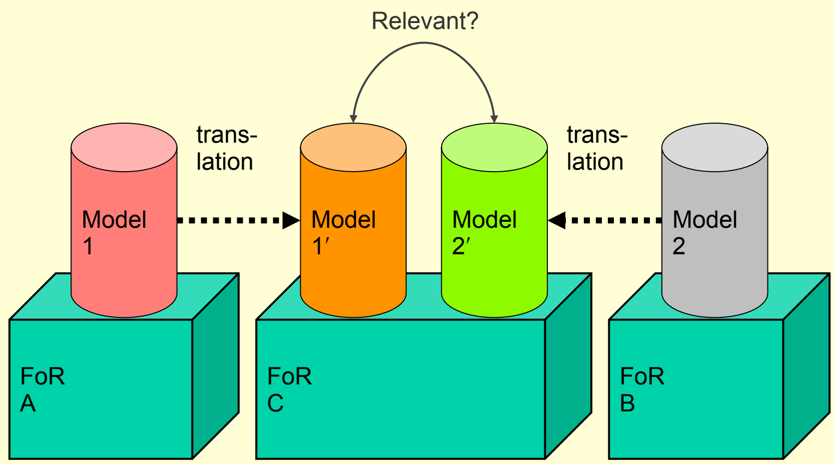
Cross-Frame Comparison



Translated Comparison



Translation to Third Frame



Super-Turing vs. Non-Turing

- Notion of Super-Turing computation is relative to FoR of Turing computation
- Super-Turing computation is important, but so is *Non*-Turing computation

Some Issues in Non-Turing Computation

- What is computation in broad sense?
- What FoRs are appropriate for non-Turing computation?
- Models of non-Turing computation
- How fundamentally to incorporate error, uncertainty, imperfection, reversibility?
- How systematically to exploit new physical processes?

Expanding the Range of Physical Computation

- Digital VLSI becoming a vicious cycle?
- A limit to the number of bits and MIPS
- Alternative technologies are surpassed before they can be developed
- False assumption that binary logic is the only way to compute
- How to break out of the vicious cycle?

What is Computation?

- What distinguishes computing (physically realized information processing) from other physical processes?
- Computation is a *mechanistic process,* the purpose or function of which is the *abstract manipulation* (processing) of *abstract objects*
- Purpose is *formal* rather than *material*
- Does not exclude embodied computation, which relies more on physical processes

Possible Physical Realizations of Computation

- Any abstract manipulation of abstract objects is a potential computation
  - *de novo* applications of math models
  - applications suggested by natural computation
- But it must be physically realizable
- *Any reasonably controllable, mathematically described, physical process can be used for computation*

Some Requirements

- Speed, but:
  - faster is not always better
  - slower processes may have other advantages
- Feasibility of required transducers
- Accuracy, stability & controllability as required for the application
  - natural computation shows ways of achieving, even with imperfect components

Matching Computational & Physical Processes

- Familiarity of binary logic maintains vicious cycle
- Natural computation shows alternate modes of computation, e.g.:
  - information processing & control in brain
  - emergent self-organization in animal societies
- Openness to usable physical processes
- Library of well-matched computational methods & physical realizations

General-Purpose Computation

- Value of general-purpose computers for all modes of computation
- "Universality" is relative to frame of relevance
- E.g., speed of emulation is essential to real-time applications (natural computation)
- Merely computing the same function may be irrelevant

Conclusions

- Turing model of computation exists in a frame of relevance
  - not appropriate to natural computation, nanocomputation, quantum / quantum-like computation
  - central issues of these include continuity, indeterminacy, parallelism
- Broader definition of computation is needed
- Facilitates new implementation technologies
- Improves understanding of computation in nature