Questions

• How fast can my computer run?
  – What limits this speed?
  – What can I do to make it run faster?
• How can I extend the battery life on my gadget?
  – How much energy must my computation take?
• How small can I make a memory?
  – Why does DRAM need to be refreshed?

Questions

• How many bits/second can I send over a link?
  – What limits this?
  – How do I maximize?
• How does technology scaling change these answers?
  • What can I rely on technology to deliver?

Outline

• Motivating Questions
• What this course is about
• Objectives
• What you need to know
• Structure
• Policies
• Content

Deconstruction

• Circuit-Level
  Modeling, Design, and Optimization
  for Digital Systems
  
  Look below the gates
  …transistors, resistance, capacitance, inductance…
  Abstract and predict
  Create
  Make efficient
  (fast, low energy, small)
  Compute, store, transmit binary values (0s, 1s)

What course about

• What Computer Engineers need to know about the physical properties in order to design efficient digital circuits
  • Physical Properties
    – Delay, Energy (Power), Area, Reliability
  • Efficient
    – Fast, Low Energy, Small, Won’t Fail (very often)
  • Digital Circuits
    – Computation, Storage, Communication
What course is about

• Modeling and abstraction
  – Predict circuit behavior
  – Well enough to know our design will work
  – …with specific properties
    • Speed, energy, ….
  – Well enough to reason about design and optimization
    • What knob can I turn to make faster?
    • How much faster can I expect to make it?

• Speed, energy, ….
• Well enough to reason about design and optimization
• What knob can I turn to make faster?
• How much faster can I expect to make it?

You will learn

• disciplines for robust digital logic and signaling
  – (e.g., restoration, clocking, handshaking)
• where delay, energy, area, and noise arises in gates, memory, and interconnect
• how to model these physical effects
  – back-of-the-envelope design
    • (e.g. RC and Elmore delay)
  – detailed simulation (e.g. SPICE)

Objectives

You will learn

• the nature of tradeoffs in optimization
  – Among delay, energy, area, noise
• how to design and optimize
  – logic, memory, and interconnect structures
  – at the gate, transistor, and wire level
• how technology scales
  – and its impact on digital circuits and computer systems
What you Need to Know

coming in to this course

What you need to know

• See pages linked from course page
• CIS170
  – Gates, Boolean logic, DeMorgan’s, optimization
• CIS215
  – RLC circuit analysis
  – Diagnostic Quiz next time – end of class

Structure

• MWF Lecture
• Reading from text
• 4 lecture periods → Lab
  – 3 Detkin
    • See phenomena first hand before simulate
  – 1 Ketterer → SPICE Intro

SPICE

• Simulation Program with Integrated Circuit Emphasis
  – Industry standard analog circuit simulator
  – Non-linear, differential equation solver specialized for circuits
• Integrated circuits – simply impractical to build to debug
  – Must simulate to optimize/validate design

Structures

• Homeworks – week long (6 total)
• Projects – two weeks long (3 total)
  – Design oriented
  – On three main topics
    • Computation
    • Storage
    • Communication
• Two midterms
• Final
Admin

- Won’t bring printouts to class
- Use course calendar
  - Lectures online before class
    - (most of the time)
  - Homeworks linked
    - Homework 1 out soon (tied to first lab visit)
    - Reading specified

Policies

See web page for details

- Turnin homework on blackboard
  - No handwritten homework
  - Use CAD Tools for circuit drawings
- Late homework penalty
- Individual work (HW & Project)
  - CAD drawings, simulations, analysis, writeups
  - May discuss strategies, but must acknowledge

Content

- Logic (Computation) [7 weeks]
  - Combinational
  - Sequential
- Storage [2 weeks]
- Communication [3 weeks]

- Logic
  - Transistors $\rightarrow$ Gates
  - In Lab: build gate, measure delay, restore
  - Restoration
  - Delay
  - Area (no layout $\rightarrow$ ESE570)
  - Energy
  - Synchronous (flip-flops, clocking, dynamic)
  - Project: fast multiplexor or ripple-carry adder
Content

• Memory
  – No Lab component
  – RAM Organization
  – Driving Large Capacitances
  – Signal amplification/restoration
  – Project: design a Register File

Content

• Communication
  – In Lab
    • Measure inductive ground bounce, crosstalk
    • Experiment with transmissions lines, termination
  – Noise
    • Crosstalk
    • Inductive
    • Ionizing particles, shot
  – Transmission Lines
  – Project: Chip-to-chip signaling

Wrapup

• Admin
  – Find web, get text, assigned reading…
  – http://www.seas.upenn.edu/~ese370
• Big Ideas / takeaway
  – Model to enable design
• Diagnostic Quiz next time
  – Review as needed
• Remaining Questions?