# EECS150 Components and Design Techniques for Digital Systems

John Wawrzynek
(Warznek)
Professor of EECS
johnw@eecs.berkeley.edu
643-9434
631 Soda Hall
Office hours: Tu, Th 1-2

Teaching Assistants:
Gabriel Eirea
Mark Feng
Yury Markovskiy
Yatish Patel
Jack Sampson
Norm Zhou



## Outline

- Enrollment
- Course Content
- Course Structure & Grading
- Cheating
- Questionnaire
- A Few Basic Principles of Digital Design



## Enrollment

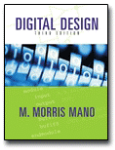
- If you are enrolled and plan to take the course you must attend your lab section next week, if not you will be dropped from the class roster. No exceptions!
- Please note: Thursday morning lab section will not be held. If you are enrolled in that lab section, please change (using Telebears) to a different lab before next week.
- If you are on the wait list and would like to get into the class you must:
  1. Turn in an appeal for on third floor Soda.
  2. Attend lectures and do the homework, the first two weeks.
  3. In the second week of classes, go to the lab section in which you wish to enroll. Give the TA your name and student ID.
  4. Later, we will process the waitlist based on these requests, and lab section openings.

→ The final class roster will be posted at noon on Friday Feb 1.



## Course Materials

- Textbook:
  Digital Design, Third Edition
  M. Morris Mano
  Prentice Hall, 2002




- Laboratory Materials:
  - Xilinx tools and design libraries available online.
  - Verilog reference online.
- Class notes, homework & lab assignments, and other documentation will be available on the CS150 homepage: http://www-inst.eecs.berkeley.edu/~cs152/index.html
  - Check the class homepage and newsgroup often!
  - You are responsible for checking the class web page at least once every 24 hours.

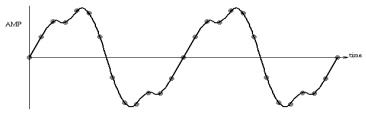


## Course Content

Components and Design Techniques for **Digital Systems**

**Synchronous Digital Hardware Systems**

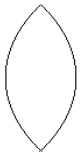
- Synchronous: "Clocked" - all changes in the system are controlled by a global clock (not asynchronous)
- Digital: All inputs/outputs and internal values (signals) take on discrete values (not analog).
  - Example: sound waveform




  - **A series of numbers is used to represent the waveform, rather than a voltage or current, as in analog systems.**



## Course Content



- Digital Architectures
- Arithmetic units, controllers
- Memory elements, logic gates, busses
- Transistor-level circuits
- Transistors, wires

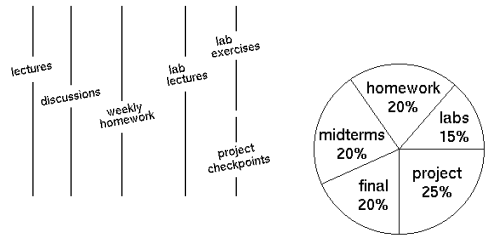
- Not a course on transistor physics and transistor circuits. Although, we will look at these to better understand the primitive elements for digital circuits.
- Not a course on computer architecture or the architecture of other systems. Although we will look at these as examples.

## Significant Changes from Fall 2001

- Lower enrollment
- Verilog replaces schematics
- New laboratory facility, 125 Cory. Thanks to National Semiconductor!
- New project board, virtex based. Thanks to Xilinx, Inc.
- Mano book replaces Katz.
- Changes in order and emphasis in course content (you will probably not notice.)
- Not Changed:
  - large project
  - lots of work
  - lots of fun!



## Course Structure & Grading






## Course Structure & Grading

***A week in the life of a EECS150 student***

| | |
|---|---|
| Monday (for example): | |
| Discussion section | 1 |
| Tuesday: | |
| Lecture 2-3:30 | 1.5 |
| Wednesday (for example): | |
| Lab section | 3 |
| Thursday: | |
| Lecture 2-3:30 | 1.5 |
| Friday: | |
| Lab Lecture | 1 |
| Reading book, reviewing notes | 3 |
| Homework | 4 |
| TOTAL | 15 hours/week |



## Cheating

- Any act that gives you unfair advantage at the expense of another classmate.
- Examples:
  - copying on exams, homework,
  - copying design data,
  - modifying class CAD software,
  - modifying or intentionally damaging lab equipment.
- If you ever have a question about what will be considered cheating, please ask me.
- We will use software to automatically compare your submitted work to others.
- If we detect you are involved in cheating you will be:
  - turned over to the Office of Student Judicial Affairs, for investigation and sanctions, additionally,
  - if you are found to have cheated, you will receive an F in the course.

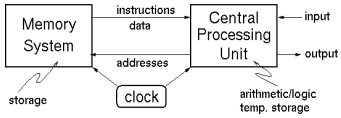


## Questionnaire



## Example Digital Systems

- Digital Computer




  - Usually design to maximize performance. "Optimized for speed"
- Handheld Calculator


- Usually designed to minimize cost. "Optimized for low cost"

- Of course, low cost comes at the expense of speed.

## Example Digital Systems

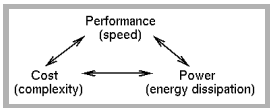
- Digital Watch


Designed to minimize power.
Single battery must last for years.

  - Low power operation comes at the expense of:
    - lower speed
    - higher cost



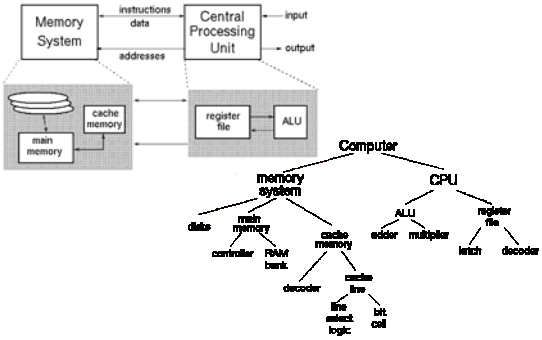
## Basic Design Tradeoffs




- You can improve on one at the expense of worsening one or both of the others.
- These tradeoffs exist at every level in the system design - every sub-piece and component.
- Design Specification -
  - Functional Description.
  - Performance, cost, power constraints.
- As a designer you must make the tradeoffs necessary to achieve the function within the constraints.



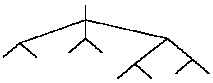
## Design Representation






## Hierarchy in Designs

- Helps control complexity -
  - by hiding details and reducing the total number of things to handle at any time.
- Modulalizes the design -
  - divide and conquer
  - simplifies implementation and debugging



- Top-Down Design
  - Starts at the top (root) and works down by successive refinement.
- Bottom-up Design
  - Starts at the leaves & puts pieces together to build up the design.
- Which is better?
  - In practice both are needed & used.
    - Need top-down divide and conquer to handle the complexity.
    - Need bottom-up because in a well designed system, the structure is influence by what primitves are available.



## Digital Design

*Given a functional description and performance, cost, & power constraints, come up with an implementation using a set of primitives.*

- How do we learn how to do this?
  1. Learn about the primitives.
  2. Learn about design representation.
  3. Learn formal methods to optimally manipulate the representations.
  4. Look at design examples.
  5. Use trial and error - CAD tools and prototyping.
- *Digital design is in some ways more an art than a science. The creative spirit is critical in combining primitive elements & other components in new ways to achieve a desired function.*
- Unlike art we have objective measures of a design:

  ***performance cost power***