What This Class is All About

- Designing, optimizing, and realizing **synchronous digital systems**
  - Both critical components and design techniques

- What you will learn:
  - Importance of making things work
  - Using abstraction to deal with complexity
    - And to use the “right” abstraction for the problem you are solving
  - What specifications are important in digital systems
    - And how they trade off with each other
Where This Is Useful

Consumer Products

Communications Infrastructure

Aerospace and Military

Automotive

Course Content

Programming Languages
Asm / Machine Lang
Instruction Set Arch
Machine Organization
HDL
FlipFlops
Gates
Circuits
Devices
Transistor Physics
IC processing

Deep Digital Design Experience

Fundamentals of Boolean Logic
Synchronous Circuits
Finite State Machines
Timing & Clocking
Device Technology & Implications
Controller Design
Arithmetic Units
Encoding, Framing
Testing, Debugging
Hardware Architecture
Hardware Design Language (HDL)
Design Flow (CAD)
Course Content - Design Layers

Not a course on transistor physics and transistor circuits. Although, we will look at these to better understand the primitive elements for digital circuits.

High-level Organization: Hardware Architectures
System Building Blocks: Arithmetic units, controllers
Circuit Elements: Memories, logic blocks
Transistor-level circuit implementations
Circuit primitives: Transistors, wires

Not a course on computer architecture or the architecture of other systems. Although we will look at these as examples.

Practical Information

• Instructor:
  – Professor Elad Alon
    • 519 Cory Hall, 642-0237, elad@eecs.berkeley.edu
    • Office hours: Tu./Th. 11am-12pm

• TA’s:
  – Daiwei Li, daiweili@hkn.eecs.berkeley.edu
    • Office hours: Tues. 4-5pm, 125 Cory
  – James Parker, james.parker@berkeley.edu
    • Office hours: Mon. 2-3pm, 125 Cory
  – Dan Yeager, yeagerd@gmail.com
    • Office hours: Wed. 4-5pm, 125 Cory

• Webpage: http://www-inst.eecs.berkeley.edu/~cs150
Enrollment

- Working on getting a larger room
  - When/if this happens should be able to accommodate everyone on the waiting list

- Some important points:
  - This class will not be easy
  - Workload will be heavy
    - Definitely not the class for you if looking for an easy grade
  - But, you will learn a lot

- Class will be videotaped/webcast
  - But please don't skip the lectures

Discussions

- Two discussion sessions (125 Cory)
  - Fri. 2-3pm
  - Mon. 11am-12pm
  - Identical material in both sessions

- No discussions this week
  - First discussion will be held on Fri. Sept. 2nd 2-3pm
Labs

• Schedule/staffing (125 Cory):
  • Tues. 5-8pm: Daiwei, James
  • Wed. 9am-12pm: Dan, Daiwei
  • Wed. 5-8pm: James, Dan
  • Pick one lab and stick with it

• No “lab lecture” on Fridays, 2-3pm.

• No lab sections this week; labs start next week
  • (1st lab Tues. Aug. 30th)

Assignments

• 9 homeworks
• 5 labs
  • Can work together on these
  • But must turn in your own solution

• Design project (split into multiple phases)
  • Done in pairs – find a partner now!

• 1 midterm, 1 final
  • Midterm: Thurs. Oct. 27th, evening (TBD)
  • Final: Tues. Dec. 13th, 8-11am
Course Grading

- Homeworks: 10%
- Labs: 5%
- Midterm: 20%
- Final: 25%
- Project: 40%

- Homeworks due Thursdays 5pm
- Labs due beginning of next lab session
- Project checkoffs due no later than Wed. 8pm
  - Will probably have sign-up sheets for check-offs during lab times

A Couple of Notes

- No late assignments will be accepted
  - Homeworks, labs, and project checkpoints are crucial for you to understand the material
    - We promise to minimize busy-work
  - Course is fast-paced – you will not have time to catch up

- Don’t even think about cheating
  - We have software that automatically compares submissions
  - Penalties for cheating will be stiff – see website for policy
  - If you’re not here to learn, this isn’t the class for you
A Few More Tips

• Attend lectures and office hours, ask questions
  • Helps to make sure you understand the material and can apply it in new situations

• Be well organized and neat in all of your written submissions

• In lab/project, add complexity one step at a time
  • Always have a working design

• Planning is crucial
  • But don’t be afraid to abandon your original plan and start over if it becomes clear it won’t work

Course Materials

Textbook: Harris & Harris
Publisher: Morgan Kaufmann

• Class notes, homework & lab assignments, solutions, and other documentation will be available on the class webpage:
  http://www-inst.eecs.berkeley.edu/~cs150
  – Check the class webpage and newsgroup often!
  – You are responsible for checking the class webpage at least once every 24 hours (in case we need to post changes/corrections.)

piazza For online Q/A.
http://www.piazza.com/
More info later.
Evolution of Digital Design As Evidenced by CS150

• Final project circa 1980:
  – Example project: pong game with buttons for paddle and LEDs for output.
  – Few 10’s of logic gates
  – Gates hand-wired together on “bread-board” (protoboard).
  – No computer-aided design tools
  – Debugged with oscilloscope and logic analyzer

CS150 ca. 1995

• Final project:
  – Example project: MIDI music synthesizer
  – Few 1000’s of logic gates
  – Gates wired together internally on field programmable gate array (FPGA) development board with some external components.
  – Circuit designed “by-hand”, computer-aided design tools to help map the design to the hardware.
  – Debugged with circuit simulation, oscilloscope and logic analyzer
CS150 ca. 2000

- Final project:
  - Ex: eTV – decode and display streaming video over Ethernet
  - Few 10,000’s of logic gates
  - Gates wired together internally on FPGA development board and communicate with standard external components.
  - Circuit designed with logic-synthesis tools, computer-aided design tools to help map the design to the hardware.
  - Debugged with circuit simulation, logic analyzer, and in-system debugging tools.

Moore’s Law – 2x stuff per ~2yr
CS150 Today

- Xilinx XUPV5 development board (a.k.a ML505)
- Could enable very aggressive final projects
  - More than would be feasible to complete
- Project debugging with simulation tools and with in-system hardware debugging tools.

- LX110T FPGA: ~1M logic gates.
  - Interfaces: Audio in/out, digital video, ethernet, on-board DRAM, PCIe, USB, ...

Final Project Fall 2011

- Executes most commonly used MIPS instructions.
- Pipelined (high performance) implementation.
- Serial console interface for shell interaction, debugging.
- Ethernet interface for high-speed file transfer.
- Video interface for display with 2-D vector graphics acceleration.
- Supported by a C language compiler.
Hierarchy & Abstraction in Designs

• Helps control complexity -
  – by reducing the total number of things to handle at once
• Modulalizes the design -
  – divide and conquer
  – simplifies implementation and debugging

Design Styles

• 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.

• Classic debate: which is better?
Key Digital Design Metrics and Tradeoffs

- Tradeoffs between these three metrics exist at all levels of hierarchy
- Basic goal as a designer:
  - Implement a design with given functionality using available primitives while meeting and/or balancing cost, power, and performance constraints
- How do we learn how to do this?

Learning Digital Design

1. Learn about the primitives and how to use them.
2. Learn about design representations.
3. Learn formal methods to optimally manipulate the representations.
4. Look at design examples.
5. Practice! → CAD tools and prototyping.

- Digital design is often called an art since creativity is critical when combining components in new ways
- But, unlike art, we have quantitative measures of success:
  
  Cost    Power    Performance