EECS151 : Introduction to Digital Design and ICs

Lecture 1 – Introduction

Bora Nikolić

Mondays and Wednesdays 11am-12:30pm
Cory 540AB and on-line

Class Goals and Expected Outcomes

What This Class is All About?
- Introduction to digital integrated circuit and system engineering
  - Key concepts needed to be a good digital system designer
  - Discover your own creativity!
- Learn models that allow reasoning about design behavior
  - Manage design complexity through abstraction and understanding of automated tools
  - Allow analysis and optimization of the circuit’s performance, power, cost, etc.
- Learn how to make sure your circuit and the whole system work
  - There are way more ways to mess up a chip than to get it right.

Digital design is not twitch.com!
Learn by doing!

Course Focus

Possible Course Sequences
- Computer Engineering/Architecture/Hardware
  - CS 61C → EECS 151 → CS 152 → ...
  - EE 16A/B
  - CS 61C → CS 152 → EECS 151 → ...
  - EE 16A/B
  - With a 151/152 background should be able to take any graduate-level course in architecture/digital systems
- Circuits: EECS 151 + EE 140 is a springboard into integrated circuits

Prerequisites
- CS 61C
  - C, Boolean logic, RISC-V ISA
  - We will review combinational and sequential logic and RISC-V datapath, pipelining (but go much more in depth)
- EE 16A/B
  - Digital gates, RC networks
  - We will review transistor operation and design of CMOS logic

CS 61C Background – RV32I

Processor Design Project in CS61C
- Designed in LogiSim
- From Fall’20 onwards, able to run RV32I compliance tests
  - Exported Verilog, ran synthesis and simulation (R. Lund, et al, WCAE’21)

Deep Digital Design Experience
- Fundamentals of Boolean Logic
- Synchronous Circuits
- Finite State Machines
- Timing & Clocking
- Device Technology & Implications
- Controller Design
- Arithmetic Units
- Memories
- Testing, Debugging, Verification
- Hardware Architecture
- Hardware Description Languages (HDL)
- Design Flows (CAD)
At the end of EECS 151
• Should be able to build a complex digital system

Berkeley chip in 2021
of IEEE Solid-State Circuits Conference

The Tapeout Class (EE194/290C)
• In Spring 2021, 19 students completed a 28nm chip design in a semester (14 weeks)
  • Just returned from fabrication
  • Prerequisites: Either EECS151 (ASIC lab preferred) or EE140

Careers in Hardware
• Apple’s New Silicon Initiative event tonight
  • Please come (it is on-line) and ask questions
  • Your chance to meet Apple’s Fellows
  • Fellowships and scholarships available
• Undergraduate scholarships for students interested in System-on-a-Chip design
  • Applications next week!

EECS151/251A Crew
Professor Borivoje Nikolić
(Bora)
SOV Cory Hall
bora@berkeley.edu
Office Hours:
TBD

Alisha Menon
ASIC Labs
Office Hours:
M 2-3pm

Vignesh Iyer
FPGA Labs
Office Hours:
M 10am-11am

Daniel Grubb
ASIC Labs
Office Hours:
Tu 11-12pm

Charles Hong
FPGA Labs
Office Hours:
W 5-6pm

Zhenhao Liu
FPGA Labs
Office Hours:
W 5-6pm

Zhaokai Liu
ASIC Labs
Office Hours:
F 2-3pm

Alisha Menon
FPGA Labs
Office Hours:
M 2-3pm

Nayiri Krazytstetowicz
ASIC Labs
Office Hours:
Th 11:12pm

Reader: Bob Zhou

Course Information
• Basic Source of Information, class website:
  http://inst.eecs.berkeley.edu/~eecs151/fa21/
  • Lecture notes and video modules
  • Assignments and solutions
  • Labs and project information
  • Exams
  • Piazza Discussion Forum
  • Many other goodies …

Print only what you need; Save a tree!

Health and Safety
• You need to have a green badge to come to in-person lectures, labs discussions!
  • If you are feeling sick, please stay home!
    • Everything is available on-line
    • And I will switch to on-line lectures if I get sick

Class Organization
• Lectures
  • Discussion sessions (M 3-4pm, W 3-4pm, W 8-9am (online))
  • Office hours (8 hours per week!)
  • Problem Sets (~1 per week)
  • Labs – FPGA or ASIC
    (or both, but only if you are sure that you have time)
  • Design project
  • 2 Midterms + 1 Final
Lectures

• Slides available on website before the lecture
• Lectures are webcast!
  • Recordings available as well
• But please do come to lectures!
• We like interactive lectures!

Class Textbooks

No Required Book this semester

Recommended (previously required)

Recommended (previously required)

Useful

• Useful LA lab reference (EE151/251LA):
  • Erik Brunvand: Digital VLSI Chip Design with Cadence and Synopsys CAD Tools

Discussion Sessions

• Start next week!
• Review of important concepts from lecture (remember - no required textbook)
• Help with problem sets

Problem Sets

• Approximately 10 over the course of the semester (one per week)
• Posted on Thursday, due on Friday 11:59pm, 8 days later
• Essential to understanding the material
• Ok to discuss with colleagues but need to turn in your own work / write-up
• Late turn-in: 20% point deduction per day, except with documented medical excuse
• Solutions posted the week after due date

Labs

• Choose either FPGA or ASIC or both
• 6 FPGA / 6 ASIC lab exercises, done solo
  • Lab report (check off) due by next lab session
• Design project lasts 7 weeks, done with partner
  • Project demo/interview RRR week
  • Project report due RRR week
• All labs held in 111/117 Cory
  • ASIC: M 5-8PM, W 8-11am (online + 111 Cory), F 11am-2pm
  • FPGA: W 6-9PM, Th 11am-2pm, F 8-11AM (fourth lab may be opened)
• All labs start next week!

Midterms and Final

• Midterms scheduled in evening
• Review session in advance
  • Midterms:
    • Th. Oct. 7, 7-9pm (tentative)
    • Th. Nov. 4, 7-9pm (tentative)
  • Final: M, Dec 13, 11:30am-2:30PM
All exams are closed book – with one double sided 8.5x11 sheet of notes

Clobber Policy

• The clobber policy allows you to:
  • Override your Midterm 1 score with the score on the final if you perform better on the final, and
  • Override your Midterm 2 score with the score on the final exam if you perform better on the final.
• Note that the reverse is not true - you must take the entire final exam, regardless of your Midterm 1 and Midterm 2 scores.

Course Information

• For interactions between faculty, GSIs and fellow students – we are using Piazza
  For fastest response post your questions on Piazza.
  (make sure to register ASAP if you don’t want to miss any of the action)
  https://piazza.com/berkeley/fall2021/eecs151251a
Policy on Academic Dishonesty

- Details of our cheating policy on the class web site. Please read it and ask questions.
- If you turn in someone else's work as if it were your own, you are guilty of academic dishonesty. This includes problem sets, answers on exams, lab exercise checks, project design, and any required course turn-in material.
- Also, if you knowingly aid in cheating, you are guilty.
- However, it is okay to discuss with others lab exercises and the project (obviously, okay to work with project partner). Okay to discuss homework with others. But everyone must turn in their own work.
- Do not post your work on public repositories like github (private o.k.)
- If we catch you cheating, you will get negative points on the assignment: It is better to not do the work than to cheat!
- If it is a midterm exam, final exam, or final project, you get an F in the class.
- All cases of cheating reported to the Center for Student Conduct.

Grading Breakdown

<table>
<thead>
<tr>
<th>Component</th>
<th>Grade Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final Exam</td>
<td>40%</td>
</tr>
<tr>
<td>Midterm Exam</td>
<td>20%</td>
</tr>
<tr>
<td>HWs</td>
<td>20%</td>
</tr>
<tr>
<td>Project</td>
<td>25%</td>
</tr>
<tr>
<td>Labs</td>
<td>12.5%</td>
</tr>
<tr>
<td>EECS 151/251A</td>
<td></td>
</tr>
<tr>
<td>LA – ASIC</td>
<td>3 units</td>
</tr>
<tr>
<td>LA – FPGA</td>
<td>2 units</td>
</tr>
<tr>
<td>Total</td>
<td>100%</td>
</tr>
</tbody>
</table>

Tips on How to Get a Good Grade

- The lecture material is not the most challenging part of the course.
  - You should be able to understand everything as we go along.
  - Do not fall behind in lecture and tell yourself you “will figure it out later from the notes or book”.
  - Notes will be online before the lecture (usually the night before). Study them before class.
  - Ask questions in class and stay involved in the class – that will help you understand. Come to office hours to check your understanding or to ask questions.
  - Complete all the homework problems - even the difficult ones.
  - The exams will test your depth of knowledge. You need to understand the material well enough to apply it in new situations.

- You need to enroll in both the lab and the course.
  - Take the labs very seriously. They are an integral part of the course.
  - Choose your project partner carefully. Your best friend may not be the best choice!
  - Most important (this comes from 30+ years of hard-wire design experience):
    - Be well organized and neat with homework, labs, project.
    - In lab, add complexity a little bit at a time - always have a working design.
    - Don’t be afraid to throw away your design and start fresh.

Getting Started

- Discussions and labs start next week
- Lab 1 assigned later this week – complete the setup before coming to the lab!
- Midterm prepared to the lab session
- Register on Piazza as soon as possible
- Register for your EECS151 class account at inst.eecs.berkeley.edu/webacct
- If you are registering through concurrent enrollment:
  - See us in person this week

Diversifying Applications

- Healthcare and wellness
- Teledmedicine
- Education
- Autonomous driving
- Gaming
- Entertainment, VR/AR
- Visualization
- Cloud
- Personal Computers
- WWW
- Scientific computing
- Data processing
- Mainframes
- Smartphone Apps
- Machine learning

Digital Integrated Circuits and Systems
Past, Present and Future

- It is being customized as well!
Moore’s Law

- In 1965, Gordon Moore noted that the number of transistors on a chip doubled every 12 months.
- He made a prediction that semiconductor technology will double its effectiveness every 12 months.

“The complexity for minimum component costs has increased at a rate of roughly a factor of two per year. Certainty over the short term, this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000.”

Gordon Moore, Cramming more Components onto Integrated Circuits, (1965).
**Recent Cost Trend**

Cost nearly doubled!

L. Su, HotChips, August 2019.

**The Other Trends**

- **Dennard Scaling (1974)**
  - Voltages (and currents) should be scaled proportionally to the dimensions of the transistor
  - If so, delay and power should scale
    \[ \text{Delay} \approx C \cdot V/\text{avg} \]
    \[ P \approx C \cdot V^2/\text{Delay} \]
  - And, in theory, power density constant!
  - We are not following Dennard's scaling since ~2005


**Power Dissipation**

Has been > doubling every 2 years

Has to stay ~constant

**Cost Of Developing New Products**

- These are non-recurring (NRE) costs, need to be amortized over the lifetime of a product
- We will attempt to dismantle this...

**Frequency**

Frequency Trends in Intel's Microprocessors

Has been doubling every 2 years, but quickly turned flat

**Power and Performance Trends**

What do we do next?
Solutions
• Software faced the complexity issues as well
• Solutions: Increase abstraction level, be modular, improve reuse, rely on open source, be agile...
• Apply to hardware design

Abstraction
• How to design a Pong game
  • Hand layout
  • Gate-level design (semi custom)
  • Describe in HDL Synthesis, place and route
  • HLS, HCL
  • “Computer, design a pong game”

Hierarchy in Designs – Complexity Control
• Design Abstraction
  • Hide details and reduce number of things to handle at any time
• Modular design
  • Divide and conquer
  • Simplifies implementation and debugging
• Regularity
  • Instantiate identical modules

Digital Design: What’s it all about?
Given a functional description and performance, cost, & power constraints, come up with an implementation using a set of primitives.

• How do we learn to do this?
  1. Learn about the design primitives and how to use them.
  2. Learn about design representations.
  3. Learn formal methods and tools to manipulate the representations.
  4. Study design examples.
  5. Have robust approach to verification.
  6. Use trial and error – CAD tools and prototyping. Practice!
• Digital design is a bit an art as well as a science. The creative spirit is critical in combining primitive elements & other components in new ways to achieve a desired function.
• However, unlike art, we have objective measures of a design: