Course Focus

- Focus is on analog design methodology

- Methodology = how to translate a set of specs into a circuit (topology + sizing)
  - Note that also need to understand where the specs came from

- Especially in analog, some things are much “easier” to do than others
  - Concrete methodology helps to make tradeoffs more clear
  - Sometimes (often) the right thing to do is change the specs
Course Goal

• **Learn how to create systematic methodologies to analog design**
  • Based on fundamental principles
  • For a wide variety of applications
  • And that can be captured within an executable program (generator – more later)

• **Will develop increasingly more complete design methodology examples**
  • And will introduce additional topologies/circuits as we realize the need for them
Teaching Staff

• Elad’s office hours
  • 519 Cory Hall
  • Office hours TBA

• GSI: Eric Chang
  • Office hours TBA
Administrative

• **Course web page:**
  https://inst.eecs.berkeley.edu/~ee240b/sp18/

• **Lecture videos**
  • Volunteers for recording?

• **All announcements made through piazza**
  • In case you weren’t already enrolled:
    http://www.piazza.com/berkeley/spring2018/ee240b
Lecture Notes

• Compilation from offerings by multiple faculty/instructors:
  • Prof. Bernhard Boser, Prof. Ali Niknejad, Dr. Simone Gambini, Dr. Lingkai Kong, and myself

• Primary source of material for the class
  • No required text – reference texts on next slide

• Notes posted on the web
Reference Texts


Grading

• **HW: 10%**
  - One HW roughly every two weeks
  - You will be “graded” purely by on-time submission
    - You should “self-grade and make sure you understand the solutions – falling behind/not doing this will doom you to failure everywhere else.”

• **Project: 30%**
  - Groups of 2 – find a partner ahead of time

• **Midterm: 25%**

• **Final Exam: 35%**
Homework

• Can discuss/work together
  • But write-up must be individual

• Submission via gradescope
  • Further details will be announced later

• Generally due 5pm on Thursdays

• **No** late submissions
  • Start early!
Schedule Notes

• ISSCC Week: 2/12 - 2/15 (no lectures)
• Midterm: March 8 (tentative)
• Spring break: 3/26 – 3/30
• Project (tentative)
  • Part 1 due Apr. 10
  • Part 2 due Apr. 19
  • Part 3 due May 3
• Final: Wed., May 10, 8am – 11am
Course Material Introduction
Analog and Mixed-Signal Circuits

Physical World

EE 247A
...

EE 240A
EE 240B
EE 242A

Signal Conditioning

EE 240B
EE 240C

A/D

DSP

EE 241A/B
EE 251A
...

D/A

Signal Conditioning
Why Analog Circuits?

• The “real” or “physical” world is analog
  • Analog is required to interface to just about anything
    • Digital signals have analog characteristics too…

• In many applications, design of analog components is in the critical path
  • More later
Example: RF Transceiver

http://www.ti.com/product/CC110L
Another Example

- Power is once again the key motivating factor

From P. Upadhyaya, ISSCC 2015
Not Just Communications

[Lemkin, JSSC 4/1999]
Some Important Context

Image from moorinsightsstrategy.com

Image from chiprebel.com

Image from chiprebel.com
What You Will Therefore Be Doing

- You will be tasked with building many different variants of the same function/block
- You will be tasked with building many different blocks
- You will be tasked with putting many different blocks together to realize a (sub-)system
- How do you do this efficiently without (re-)introducing any known errors?
Re-Use is the Key

• Today: Integrate pre-designed blocks (IP)
  • But re-use is still limited – IP is blackbox, so if ever need to extend/modify, usually end up building your own

• Berkeley view: Capture designer’s knowledge (methodology) as an executable generator
  • Good methodologies will be parameterized (i.e., support variants)
  • New features supported by incrementally extending the code
In Other Words…

• Your goal as an analog designer **should not** be to deliver a specific *instance*.

• Instead, you **should** strive to realize the best *generator* that you can
  • So that the generator can be executed to realize any instance you are tasked with building
  • And so that you can actually effectively re-use your colleagues work (and they can re-use yours)
Berkeley Analog Generator (BAG)

- Hierarchical, Python-based framework allowing executable specification of design procedure
  - I.e., BAG takes care of the “plumbing”

- Will not require you to use BAG in this class
  - But forcing yourself to codify your methodology is an outstanding way to check and develop your understanding

J. Crossley et al., ICCAD Nov. 2013
BAG Example

SOI

FinFET

Bulk

Gain (dB)

Phase (Degrees)

Frequency (Hz)
Course Outline (approx.)

• Module 1: Analog design core
  • “Modeling” MOS transistors
  • Electronic noise and noise analysis
  • GBW- and noise-limited amplifier design

• Module 2: MOS amplifier implementation
  • OTA topologies and design
  • Time-domain behavior (settling)
  • Interference mitigation
  • Common-mode feedback
Course Outline (approx.)

• Module 3: AFE system (Photonic Link) design
  • Link circuit components and analysis
  • Comparators
  • Layout and matching effects
  • Offset cancellation

• Module 4: Wrap-up
  • Discrete time analog circuits
  • Sampling
  • Biasing and references
  • Design strategies/motifs