What We’re Doing

- Learn you a Cadence
- Learn simulation vocabulary
- Basic schematic guidelines
- Simulation results
Init

Before we begin, open a terminal:

- $ module load ece483
- $ cd ~/ece483.work
- $ virtuoso &
IC Design

- Circuit design is an iterative process
- We have models
- We estimate using math
- Implement estimations
- Simulate implementations
- Verify simulations vs specs
- Repeat till satisfactory
Cadence Design Systems

- Cadence is THE program that is used in industry

- Cadence has 3 levels of hierarchy:
  - Libraries
    - Attached to a certain technology node [250nm, 180nm]
    - Contain lots of cells, grouped by functionality
  - Cells
    - Represent circuit elements [MOSFET, AND gate, ALU]
    - Contains numerous views
  - View
    - Represents individual cell implementation
    - Schematic Symbol, Verilog, Layout
Getting Started

- Two windows pop up immediately
- Library manager
- Command Interface Window (CIW)
Making A Library

- **File → New → Library**

- “Attach to an existing technology library”

- **Most important step is CHOOSING THE RIGHT TECHNOLOGY**

- **250nm is tsmc03d**

- **180nm is tsmc02d**

- 130nm may be available later
Making a Cell

- **File → New → Cell View**

- Be sure to choose the library you made in the last step

- Set the view type to “schematic”

- Double click on the newly created View to open it
Schematic Shortcuts

This list is also on the website now

- **F** → zoom “full”, see everything in the schematic
- **I** → add instance
- **M** → move part
- **C** → copy part
- **W** → draw wire
- **L** → label wire
- **Q** → query property
- **U** → undo (shift+U is redo)
- **R** → rotate part (shift+R flips over y-axis)
Library: analogLib

- **Cell: res**
  - Ideal resistor
  - Only should need to change the resistance parameter

- **Cell: cap**
  - Ideal capacitor
  - Only should need to change the capacitance parameter

- **Cell: vdc/vsin/vpulse**
  - Ideal voltage source that can output DC, Sine, Pulse, etc

- **Cell: idc/isin/ipulse**
  - Ideal current source that can output DC, Sine, Pulse, etc

- **Cell: gnd**
  - It’s ground
Library: NCSU_Analog_Parts

Library: NCSU_Analog_Parts

- Cell: pmos4
  - Standard 4 terminal PMOS (DGS B)
  - Only change the width/length parameters
  - Can set to any value, but will autocorrect to quantized steps

- Cell: nmos4
  - Standard 4 terminal NMOS (DGS B)
  - Only change the width/length parameters
  - Can set to any value, but will autocorrect to quantized steps
Parts Specifics

- Always use the "symbol" view for all your parts
  - Other views will spit out an error

- Parameters can be either constants or variables
Drawing Schematics

- If you draw good schematics, you will have an easier time
- We encourage frequent use of labels
- Try not to overlap text with text
- You should be able to read everything without zooming out
- Try to align things vertically/horizontally
Bad Diff Amp
Good Diff Amp!
Simulation Types

- **DC Simulations**
  - Can do point-wise analysis (DC Operating Point)
  - X-axis is the voltage of a swept voltage source
  - Used for checking operating mode, bias currents

- **AC Simulations**
  - X-axis is the swept frequency of a voltage source (i.e. 1Hz -> 100MHz)
  - Used for checking gain, phase, distortion
  - Can lie! Uses small signal models, not full large signal ones

- **Transient Simulations**
  - X-axis is time
  - Used for measuring final performance, clipping, all of the above
  - If it works in transient, it works
Running Simulations

- Before you run ANY simulations, **Check+Save (F8)**
  - 90% of your errors will be “netlist not extracted due to changes”

- In schematic window, **Launch → ADE L**
ADE L Steps

- Add your design variables under **Variables → Copy From**

- Choose an analysis under **Analyses -> Choose**
  - TRANsient requires a stop time, in general use 10 * (Input Period)
  - DC sim around one point → “Save DC Operating Point”
  - DC sim sweep → select **Design Variable**, set params
  - AC sim sweep → select **Frequency**, set Fmin, Fmax

- Setup outputs under **Outputs → Setup**

- Big green button → **Netlist and Run**
ADE Example
DC Sim Example

- Checking operating regions
- Results → Print → DC Operating Points
- Click on a device

“region” is most common parameter used
- 0 → off
- 1 → triode
- 2 → saturation
- 3 → subthreshold
- 4 → breakdown
DC Sim Example

- Previous is kind of slow, especially iteratively
- Can use the calculator
- Tools → Calculator
  - Select OP
  - Select a device
  - Add to outputs
DC Sim Results
DC Annotations

- Sometimes you want to see all the I/V values on a schematic
- Can change the annotations (labels by devices)

- **Results → Annotations → DC Operating Points**

- Can go back to regular (name, W/L) by setting it to
- **Results → Annotations → Component Parameters**
AC Sim Example

- To use an AC sim, use an “vsin” or “isin” source
- Set the AC magnitude to 1V
  - We generally only care about gain, this normalizes everything

- Sweep Variable **Frequency**
- Choose reasonable limits, automatic points usually fine
AC Sim Example

- **Netlist + Run**
- **Results → Direct Plot → Main Form**

- Simple form for handling all options
  - Linear vs Log scale
  - Single vs differential

- Follow instructions on bottom of menu

- Add to ADE outputs if you want
Transient Sim Example

- Can use any source
  - Make sure to set **Amplitude** + **Frequency** params

- Stop time up to you

- Accuracy is a tuning knob:
  - Liberal/moderate OK for now
  - Conservative for final project

- Don’t worry about noise (for now)
Transient Sim Results

- Same as AC, Direct Plot is easiest
  - Results → Direct Plot → Main Form

- Same as AC, get voltage/current

- Can do single/differential

- Can save to outputs for repeated runs

- Tells you if something isn’t saved
Transient Sim Results

- All voltages saved by default
- Have to individually specify currents
- Or can save all:
  - Outputs ➔ Save All ➔ Select Device Currents
Parametric Analysis

- Sometimes you want to do nested sweeps
- Open window under **Tools → Parametric Analysis**
- Can sweep any variable given in ADE
Parametric Analysis Warnings

- Multiple variables go up as $O(m^n)$
  - This makes things extremely inefficient

- For example, one sim takes 1x times
  - Default # of steps is 5
  - One sweep = 5x
  - Two sweeps = 25x
  - Three sweeps = 125x

- **DO NOT USE PARAMETRIC ANALYSIS INSTEAD OF CALCULATIONS**
Summary

- We’ve covered approximately 2% of Cadence can do

- Using the tool is not optional, you will fail the final project

- “Can Cadence/Spectre do this?”
  - Almost always yes
  - Probably might be a little obtuse
  - Might have to dig deep into documentation
  - Might have to learn some SKILL code (Lisp derivative)