Intro to Cadence

Brady Salz

ECE483 Spring 17

ECE ILLINOIS

ILLINOIS

What We’re Doing

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

Init

Before we begin, open a terminal:

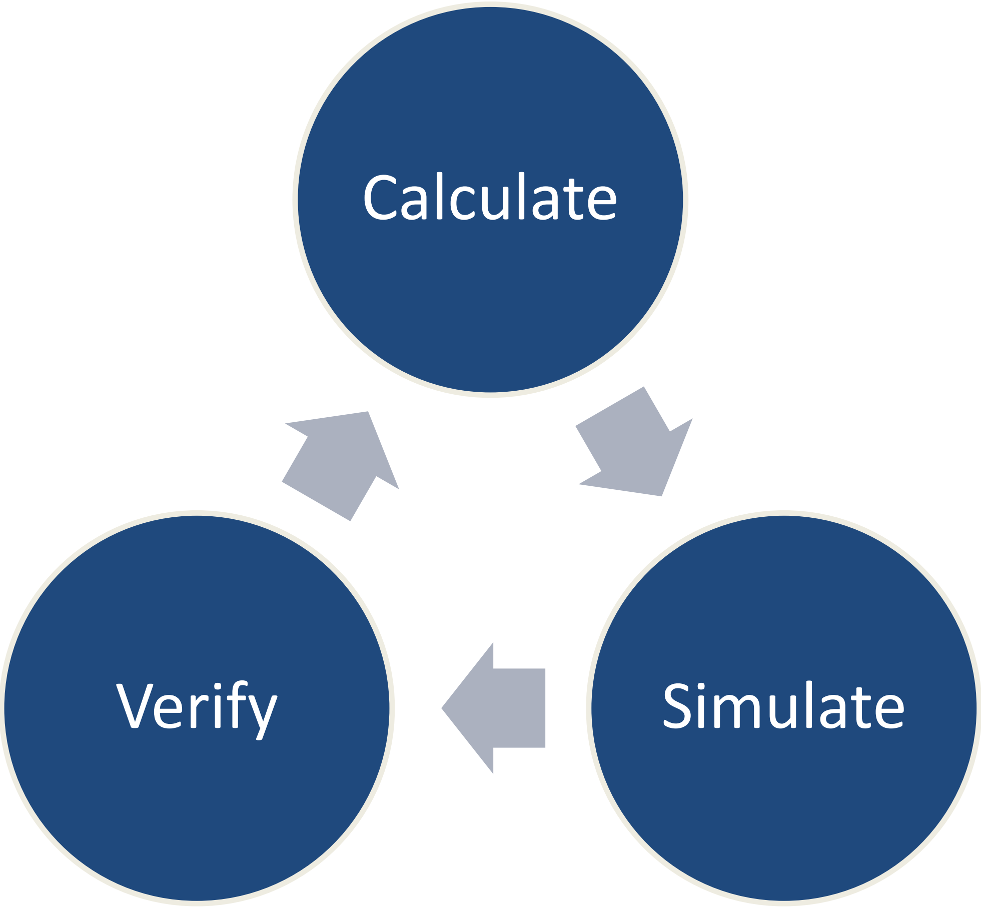
```
$ module load ece483
$ cd ~/ece483.work
$ virtuoso &
```

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



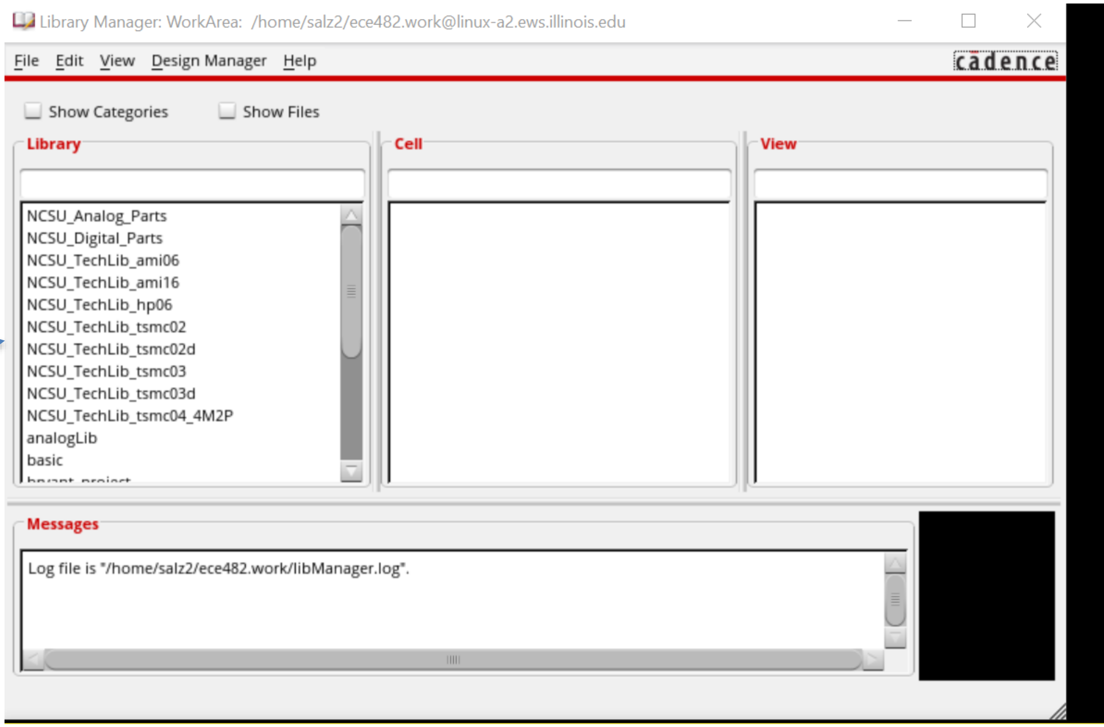
Cadence Design Systems

- 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

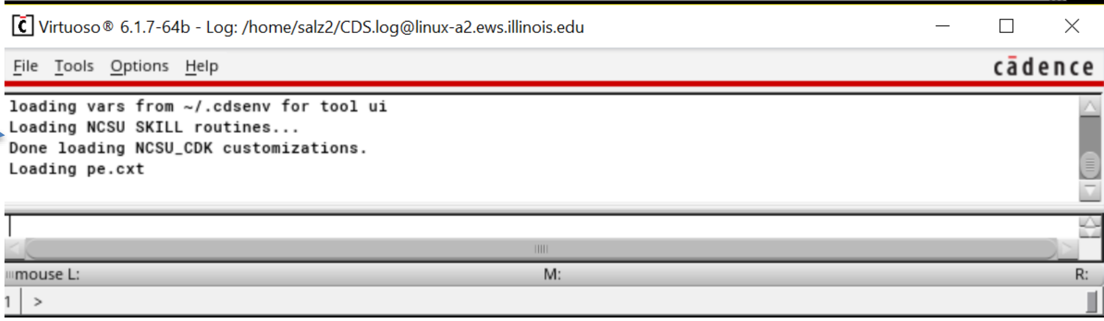
Getting Started

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


Library Manager: WorkArea: /home/salz2/ece482.work@linux-a2.ews.illinois.edu
File Edit View Design Manager Help
cadence
□ Show Categories
□ Show Files
Library
Cell
View
Messages

```
NCSU_Analog_Parts
NCSU_Digital_Parts
NCSU_TechLib_ami06
NCSU_TechLib_ami16
NCSU_TechLib_hp06
NCSU_TechLib_tsmc02
NCSU_TechLib_tsmc02d
NCSU_TechLib_tsmc03
NCSU_TechLib_tsmc03d
NCSU_TechLib_tsmc04_4M2P
analogLib
basic
```

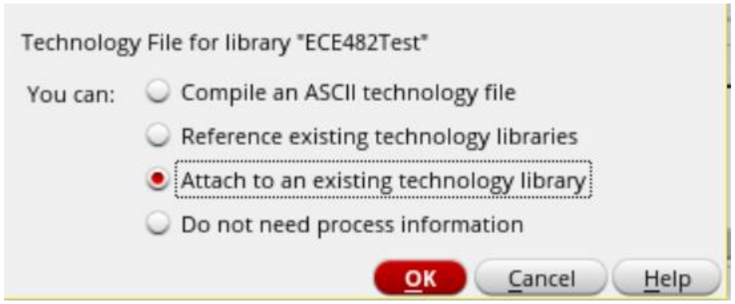
```
Log file is "/home/salz2/ece482.work/libManager.log".
```


Virtuoso® 6.1.7-64b - Log: /home/salz2/CDS.log@linux-a2.ews.illinois.edu
File Tools Options Help
cadence
mouse L:
M:
R:
1
>

```
loading vars from ~/.cdsenv for tool ui
Loading NCSU SKILL routines...
Done loading NCSU_CDK customizations.
Loading pe.cxt
```

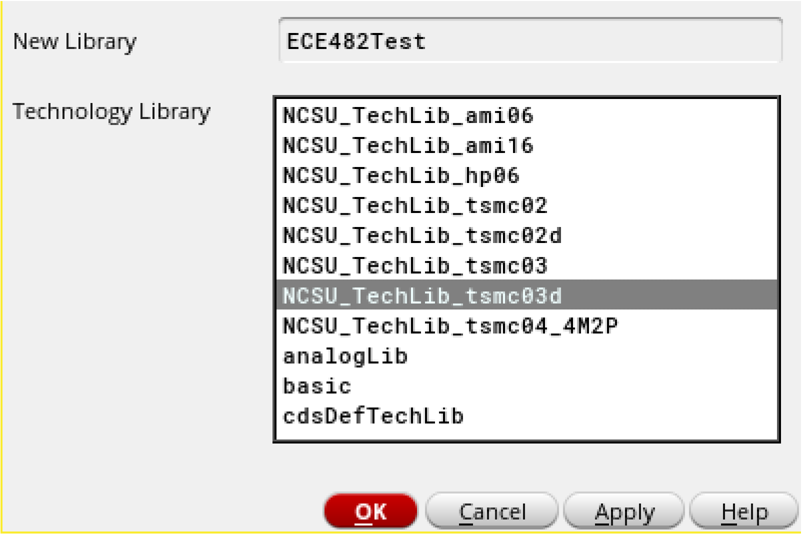
Making A Library

- **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



Technology File for library "ECE482Test"

You can:
○ Compile an ASCII technology file
○ Reference existing technology libraries
● Attach to an existing technology library
○ Do not need process information



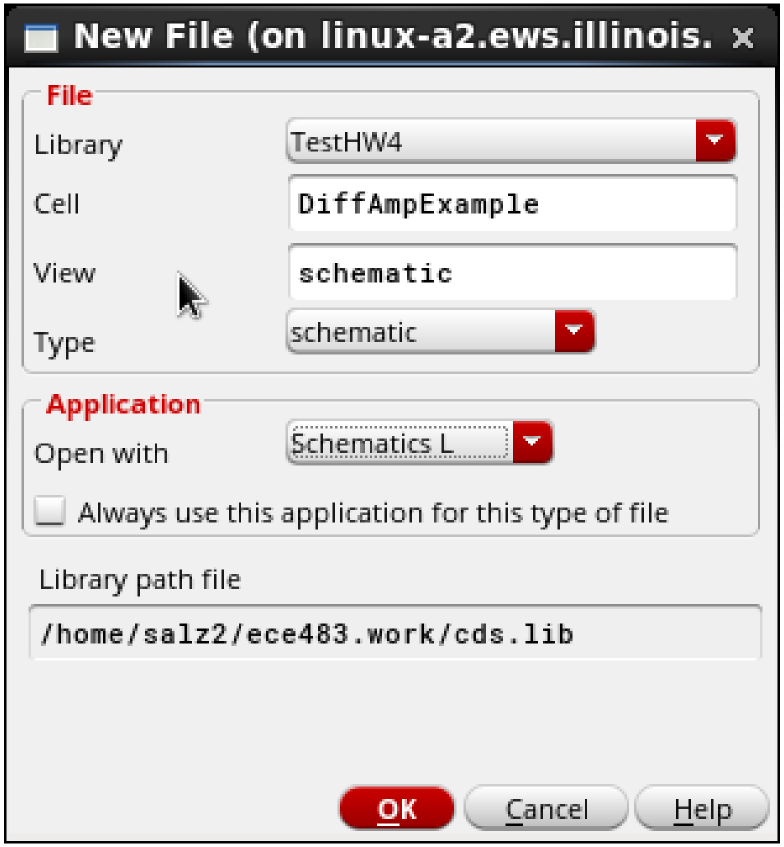
New Library: ECE482Test

Technology Library:

```
NCSU_TechLib_ami06
NCSU_TechLib_ami16
NCSU_TechLib_hp06
NCSU_TechLib_tsmc02
NCSU_TechLib_tsmc02d
NCSU_TechLib_tsmc03
NCSU_TechLib_tsmc03d
NCSU_TechLib_tsmc04_4M2P
analogLib
basic
cdsDefTechLib
```

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



New File (on linux-a2.ews.illinois.

File

Library: TestHW4

Cell: DiffAmpExample

View: schematic

Type: schematic

Application

Open with: Schematics L

□ Always use this application for this type of file

Library path file: /home/salz2/ece483.work/cds.lib

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**

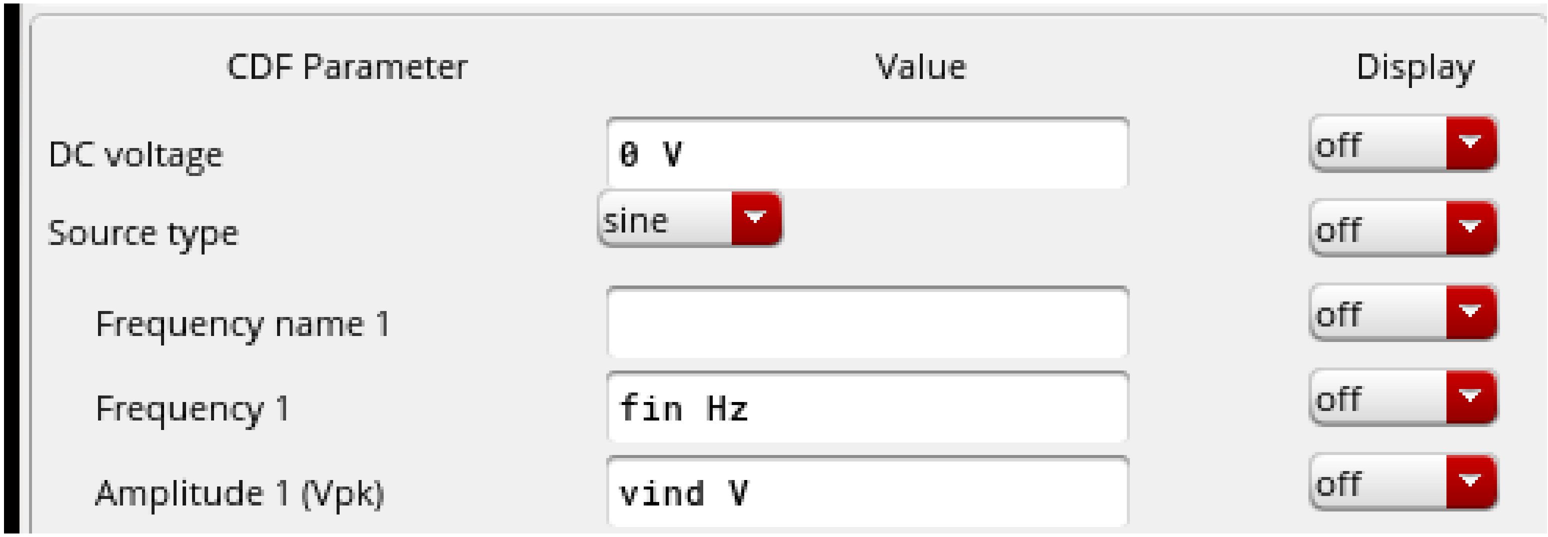
- Cell: **pmos4**
  - Standard 4 terminal PMOS (DGS B)
  - Only change the width/length parameters
  - Can set to any value, but will autocorrect to quantized steps

- 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

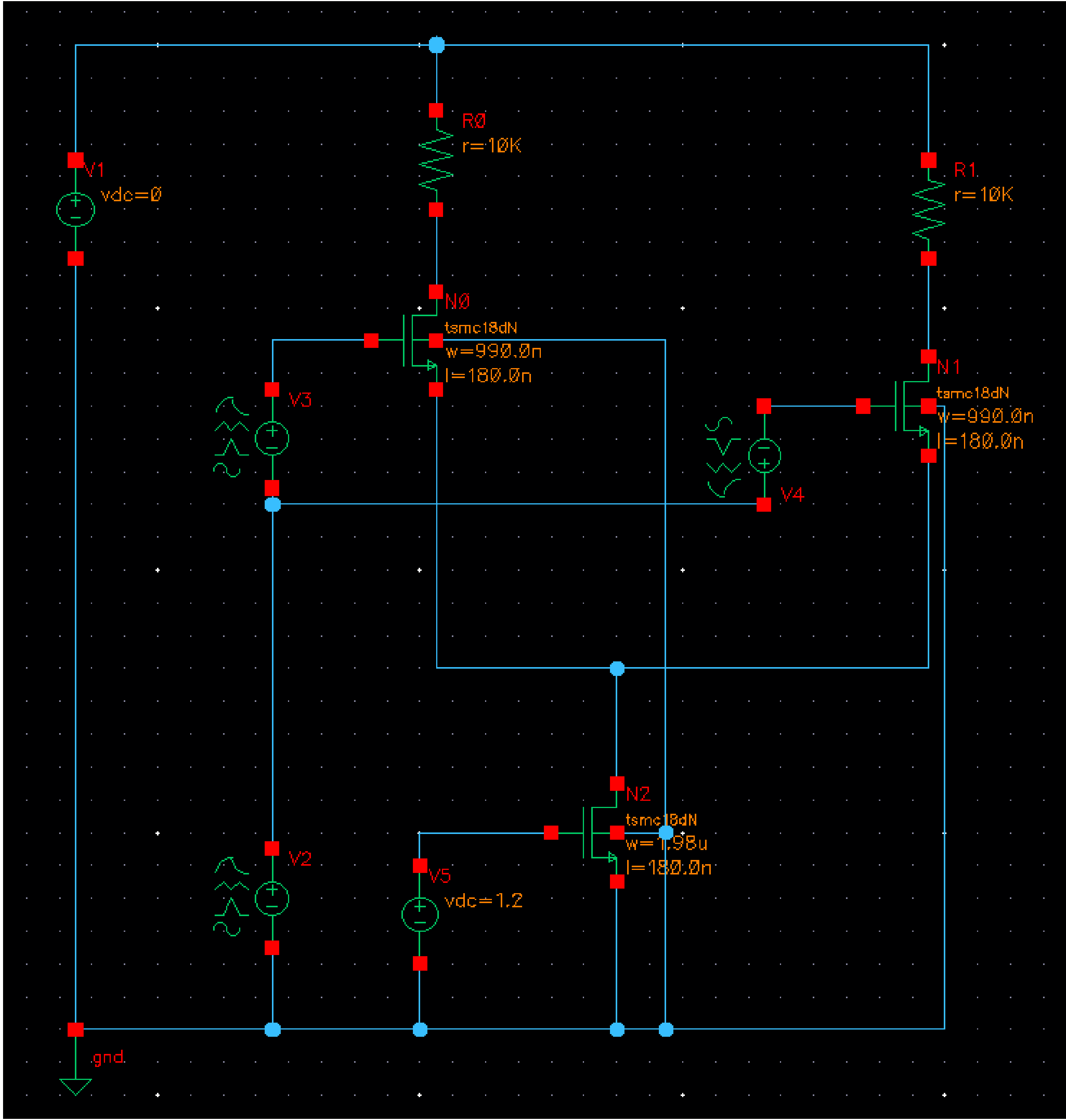


CDF Parameter	Value	Display
DC voltage	0 V	off
Source type	sine	off
Frequency name 1		off
Frequency 1	fin Hz	off
Amplitude 1 (Vpk)	vind V	off

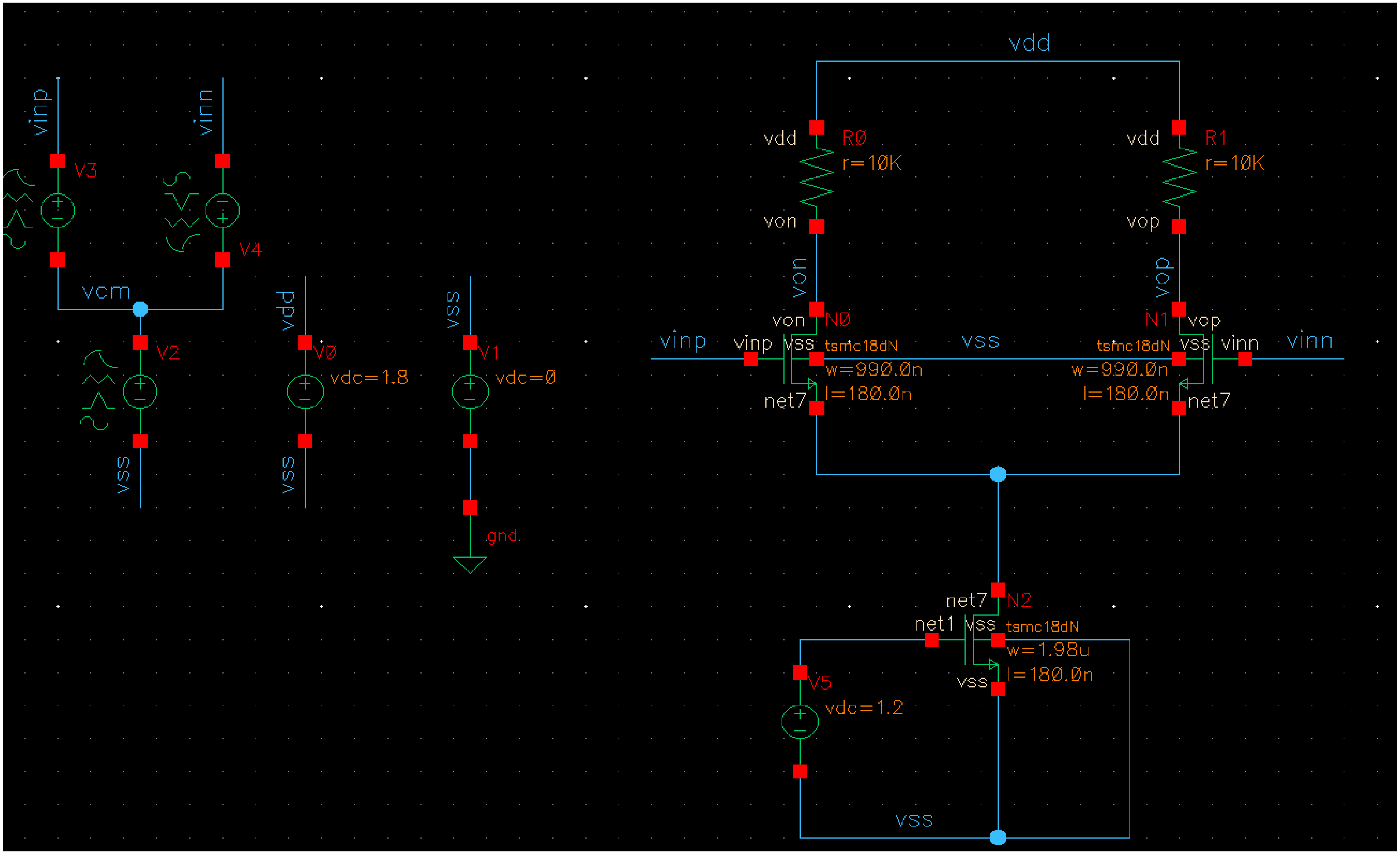
Drawing Schematics

- 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


R0
r=10K
R1
r=10K
V1
vdc=0
N0
tsmc18dN
w=990.0n
l=180.0n
N1
tsmc18dN
w=990.0n
l=180.0n
V3
V4
N2
tsmc18dN
w=1.98u
l=180.0n
V2
V5
vdc=1.2
gnd

Good Diff Amp!



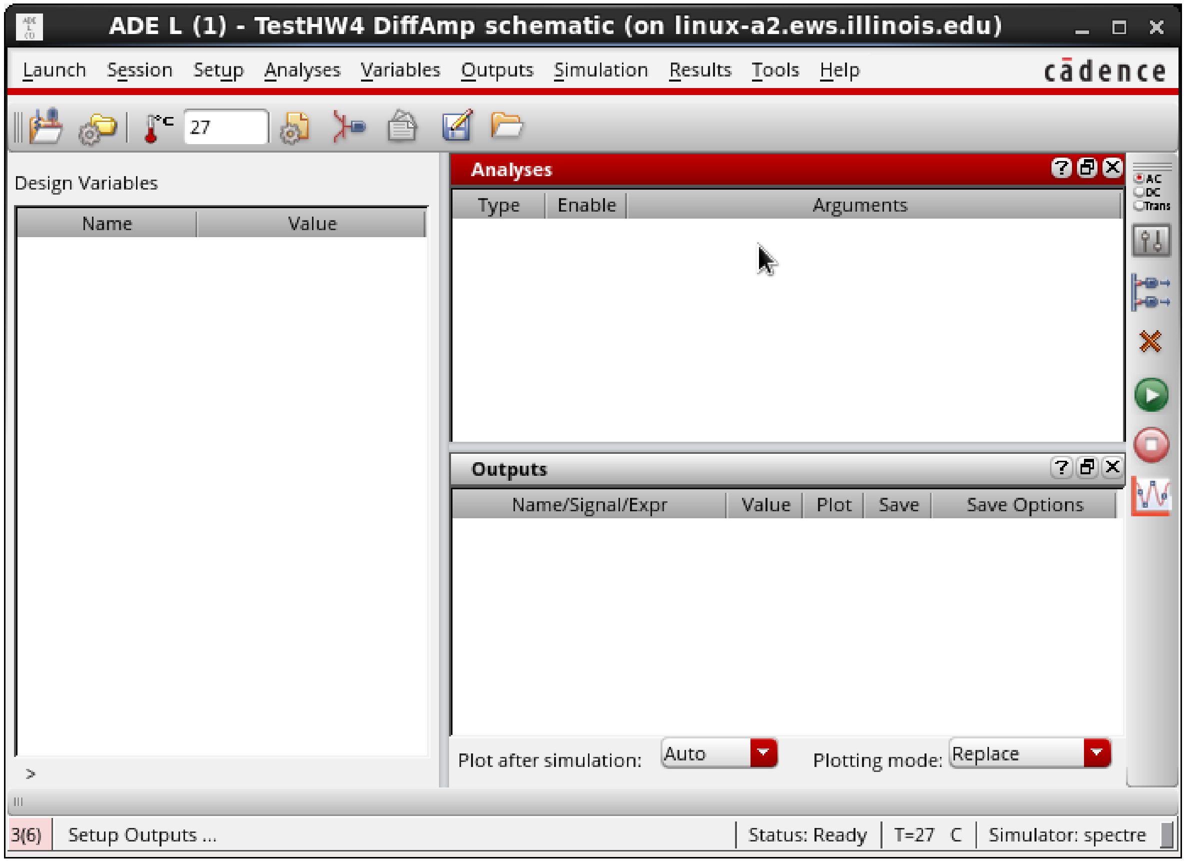
Simulation Types

- 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 (1) - TestHW4 DiffAmp schematic (on linux-a2.ews.illinois.edu)

Plot after simulation: Auto

Plotting mode: Replace

3(6) Setup Outputs ... | Status: Ready | T=27 C | Simulator: spectre

ADE L Steps

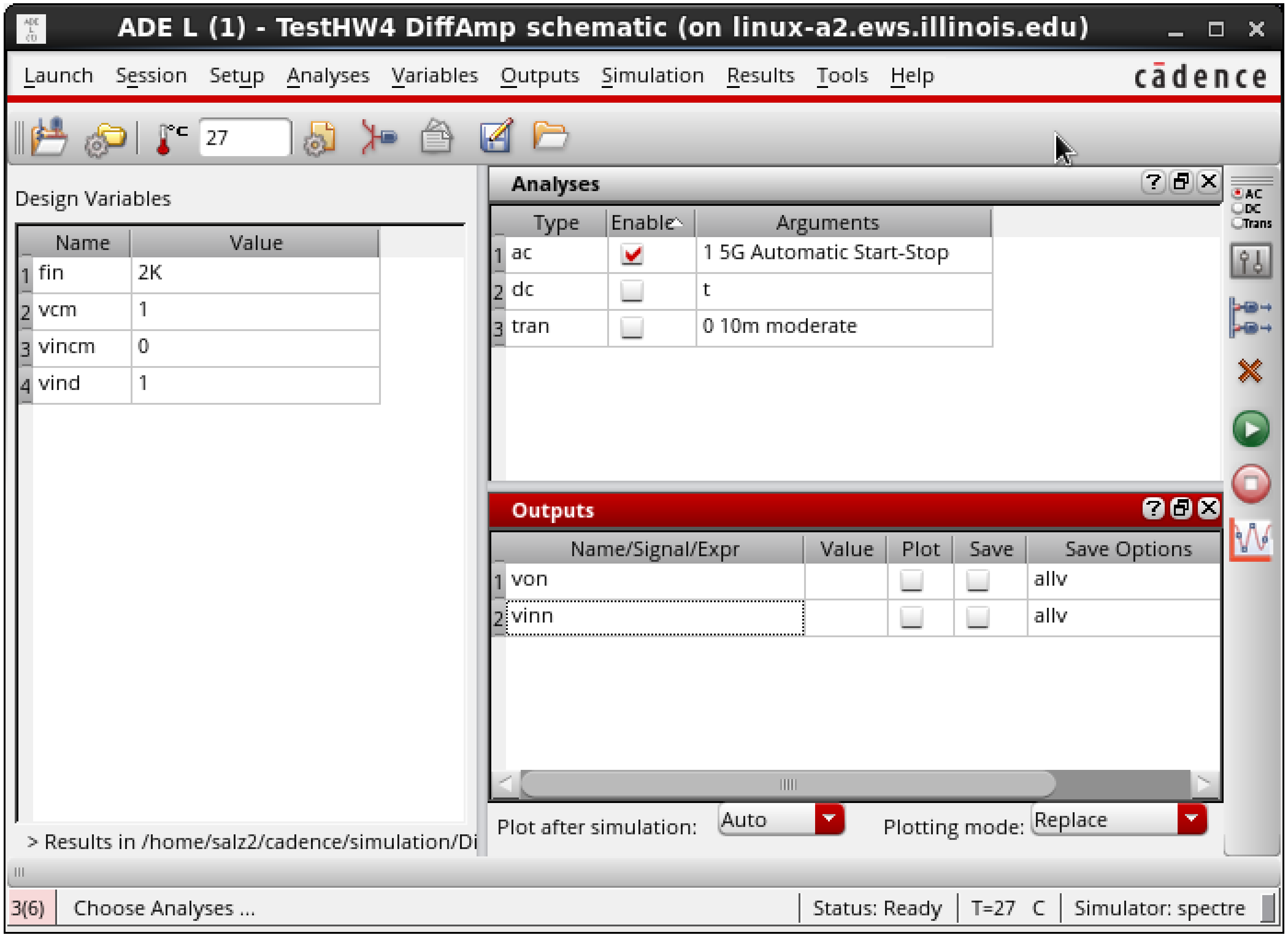
- Add your design variables under **Variables → Copy From**

- 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



ADE L (1) - TestHW4 DiffAmp schematic (on linux-a2.ews.illinois.edu)

Temperature: 27

Design Variables

	Name	Value
1	fin	2K
2	vcm	1
3	vincm	0
4	vind	1

Analyses

	Type	Enable	Arguments
1	ac	☑	1 5G Automatic Start-Stop
2	dc	□	t
3	tran	□	0 10m moderate

Outputs

	Name/Signal/Expr	Value	Plot	Save	Save Options
1	von		□	□	allv
2	vinn		□	□	allv

Plot after simulation: Auto

Plotting mode: Replace

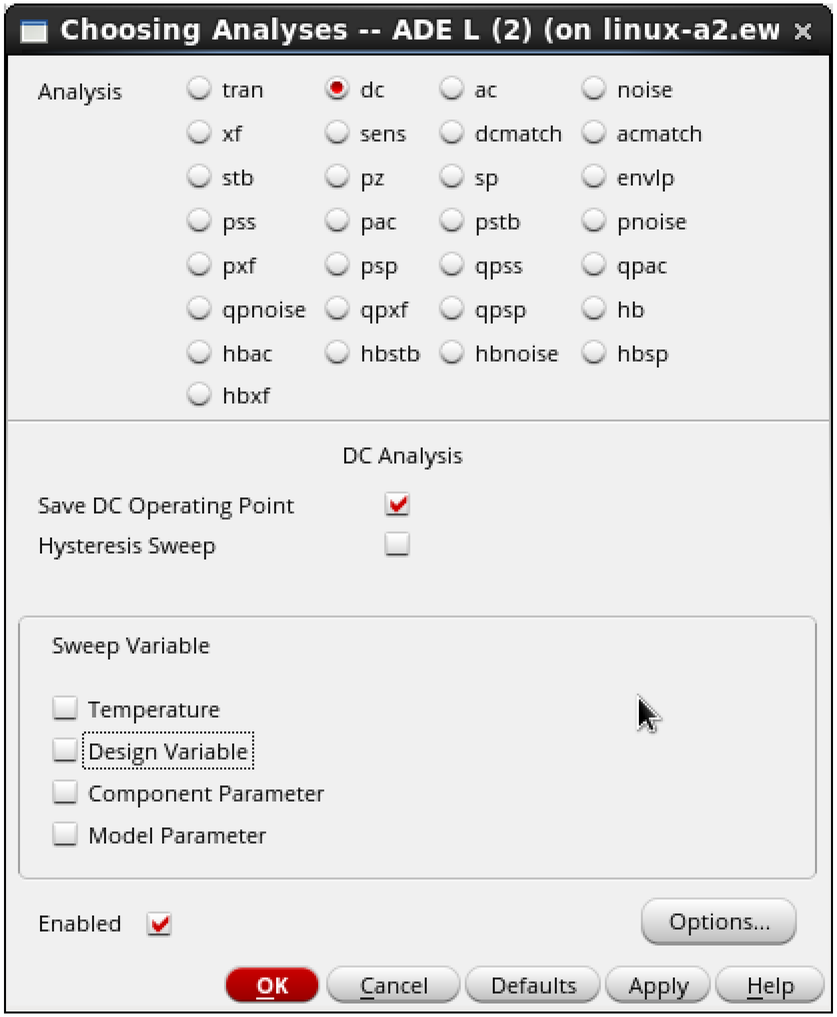
> Results in /home/salz2/cadence/simulation/Di

3(6) Choose Analyses ... Status: Ready T=27 C Simulator: spectre

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



Choosing Analyses -- ADE L (2) (on linux-a2.ew

Analysis ○ tran ● dc ○ ac ○ noise ○ xf ○ sens ○ dcmatch ○ acmatch ○ stb ○ pz ○ sp ○ envlp ○ pss ○ pac ○ pstb ○ pnoise ○ pxf ○ psp ○ qpss ○ qpac ○ qpnoise ○ qpxf ○ qpsp ○ hb ○ hbac ○ hbstb ○ hbnoise ○ hbsp ○ hbxf

DC Analysis

Save DC Operating Point ☑

Hysteresis Sweep □

Sweep Variable

□ Temperature

□ Design Variable

□ Component Parameter

□ Model Parameter

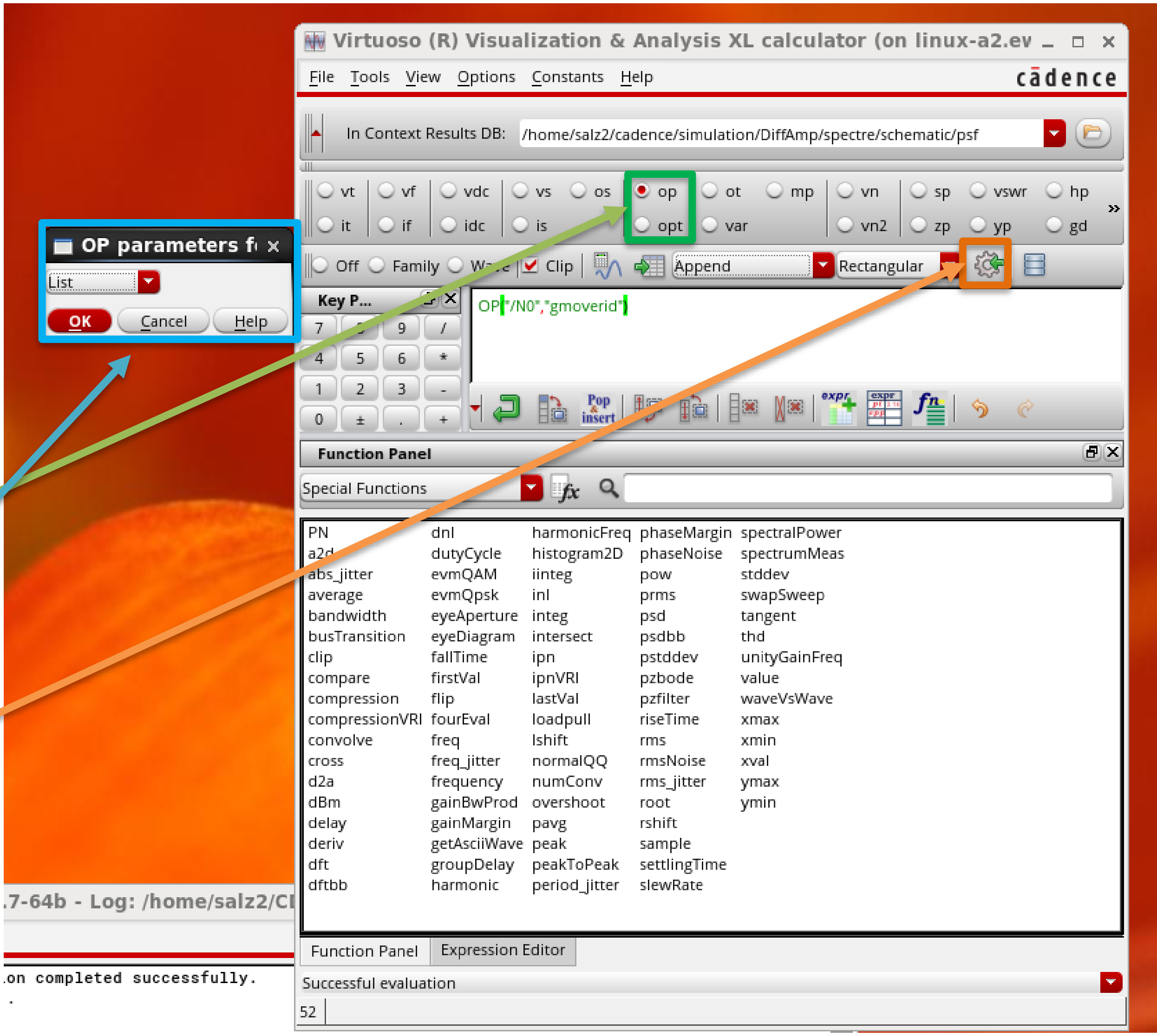
Enabled ☑

DC Sim Example

- Previous is kind of slow, especially iteratively

- Can use the calculator
- **Tools → Calculator**

- Select OP
- Select a device
- Add to outputs



In Context Results DB: /home/salz2/cadence/simulation/DiffAmp/spectre/schematic/psf

OP("/N0","gmoverid")

PN
a2d
abs_jitter
average
bandwidth
busTransition
clip
compare
compression
compressionVRI
convolve
cross
d2a
dBm
delay
deriv
dft
dftbb
dnl
dutyCycle
evmQAM
evmQpsk
eyeAperture
eyeDiagram
fallTime
firstVal
flip
fourEval
freq
freq_jitter
frequency
gainBwProd
gainMargin
getAsciiWave
groupDelay
harmonic
harmonicFreq
histogram2D
iinteg
inl
integ
intersect
ipn
ipnVRI
lastVal
loadpull
lshift
normalQQ
numConv
overshoot
pavg
peak
peakToPeak
period_jitter
phaseMargin
phaseNoise
pow
prms
psd
psdbb
pstddev
pzbode
pzfilter
riseTime
rms
rmsNoise
rms_jitter
root
rshift
sample
settlingTime
slewRate
spectralPower
spectrumMeas
stddev
swapSweep
tangent
thd
unityGainFreq
value
waveVsWave
xmax
xmin
xval
ymax
ymin

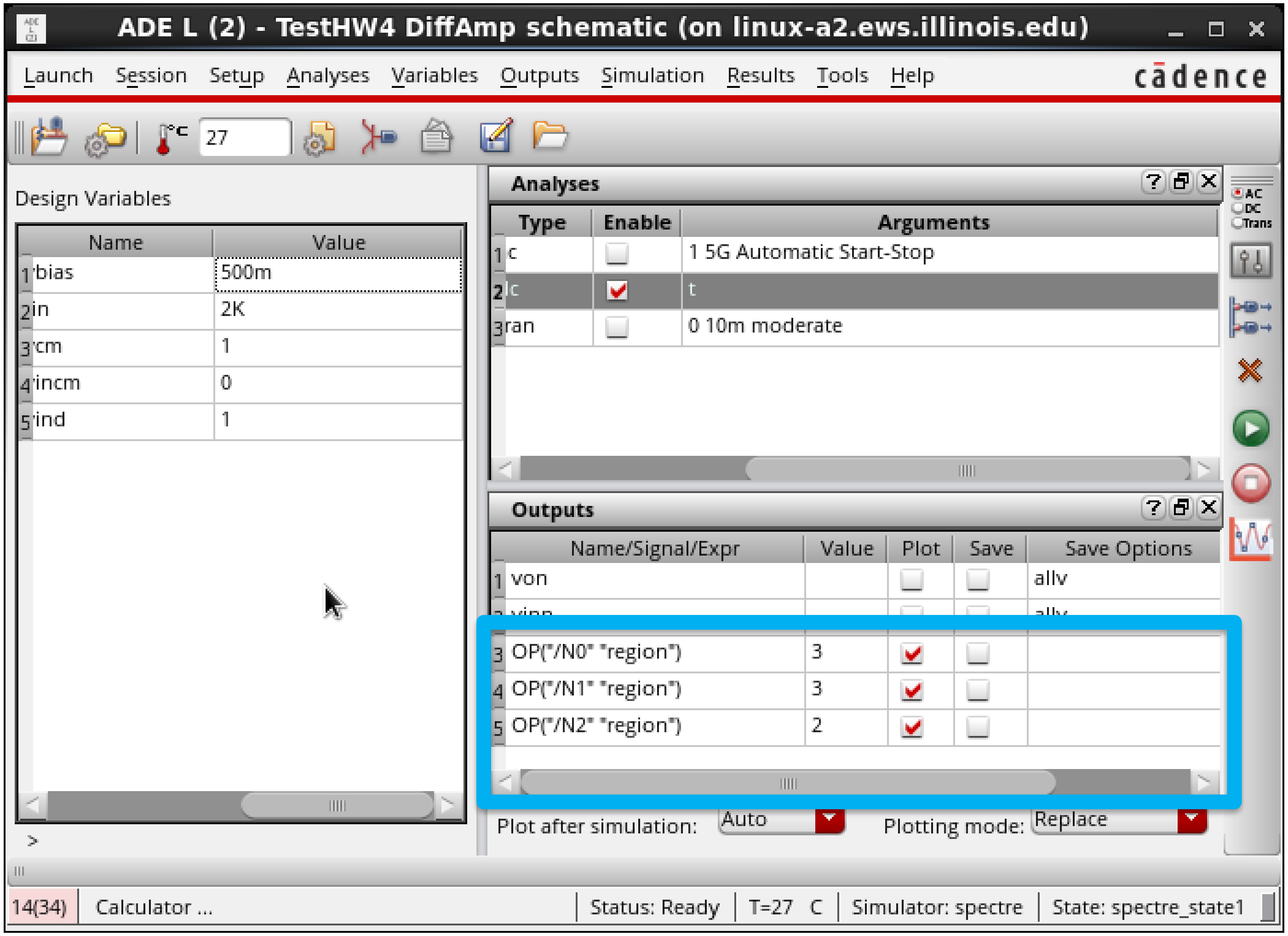
Successful evaluation

52

.7-64b - Log: /home/salz2/CI

```
on completed successfully.
.
```

DC Sim Results



ADE L (2) - TestHW4 DiffAmp schematic (on linux-a2.ews.illinois.edu)

Design Variables

	Name	Value
1	bias	500m
2	in	2K
3	cm	1
4	incm	0
5	ind	1

Analyses

	Type	Enable	Arguments
1	c	□	1 5G Automatic Start-Stop
2	lc	☑	t
3	ran	□	0 10m moderate

Outputs

	Name/Signal/Expr	Value	Plot	Save	Save Options
1	von		□	□	allv
2	vinn		□	□	allv
3	OP("/N0" "region")	3	☑	□	
4	OP("/N1" "region")	3	☑	□	
5	OP("/N2" "region")	2	☑	□	

Plot after simulation: Auto   Plotting mode: Replace

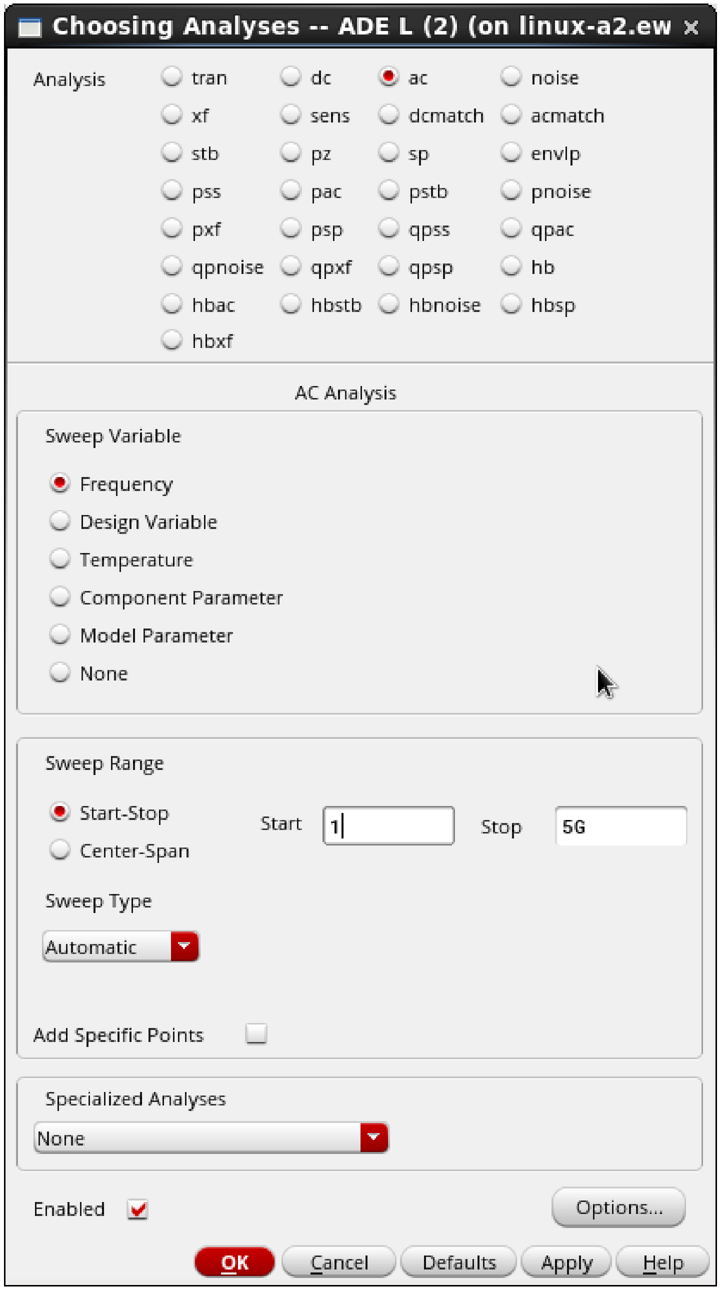
14(34) | Calculator ... | Status: Ready | T=27 C | Simulator: spectre | State: spectre_state1

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



Choosing Analyses -- ADE L (2) (on linux-a2.ew

Analysis ○ tran ○ dc ● ac ○ noise ○ xf ○ sens ○ dcmatch ○ acmatch ○ stb ○ pz ○ sp ○ envlp ○ pss ○ pac ○ pstb ○ pnoise ○ pxf ○ psp ○ qpss ○ qpac ○ qpnoise ○ qpxf ○ qpsp ○ hb ○ hbac ○ hbstb ○ hbnoise ○ hbsp ○ hbxf

AC Analysis

Sweep Variable

● Frequency
○ Design Variable
○ Temperature
○ Component Parameter
○ Model Parameter
○ None

Sweep Range

● Start-Stop ○ Center-Span

Start 1 Stop 5G

Sweep Type: Automatic

Add Specific Points ☐

Specialized Analyses: None

Enabled ☑

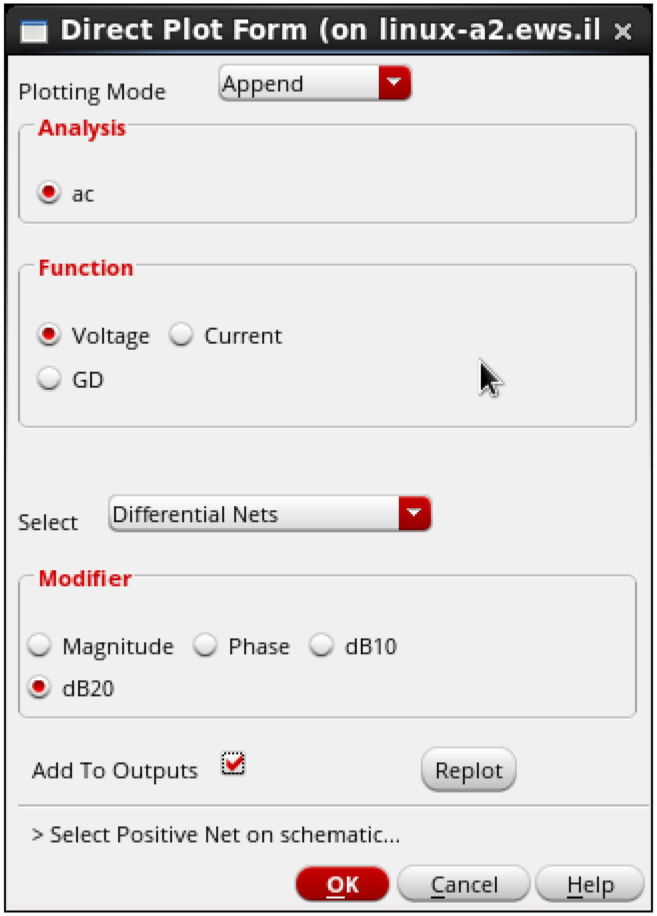
AC Sim Example

- **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



Direct Plot Form (on linux-a2.ews.il

Plotting Mode: Append

Analysis

● ac

Function

● Voltage ○ Current
○ GD

Select: Differential Nets

Modifier

○ Magnitude ○ Phase ○ dB10
● dB20

Add To Outputs ☑

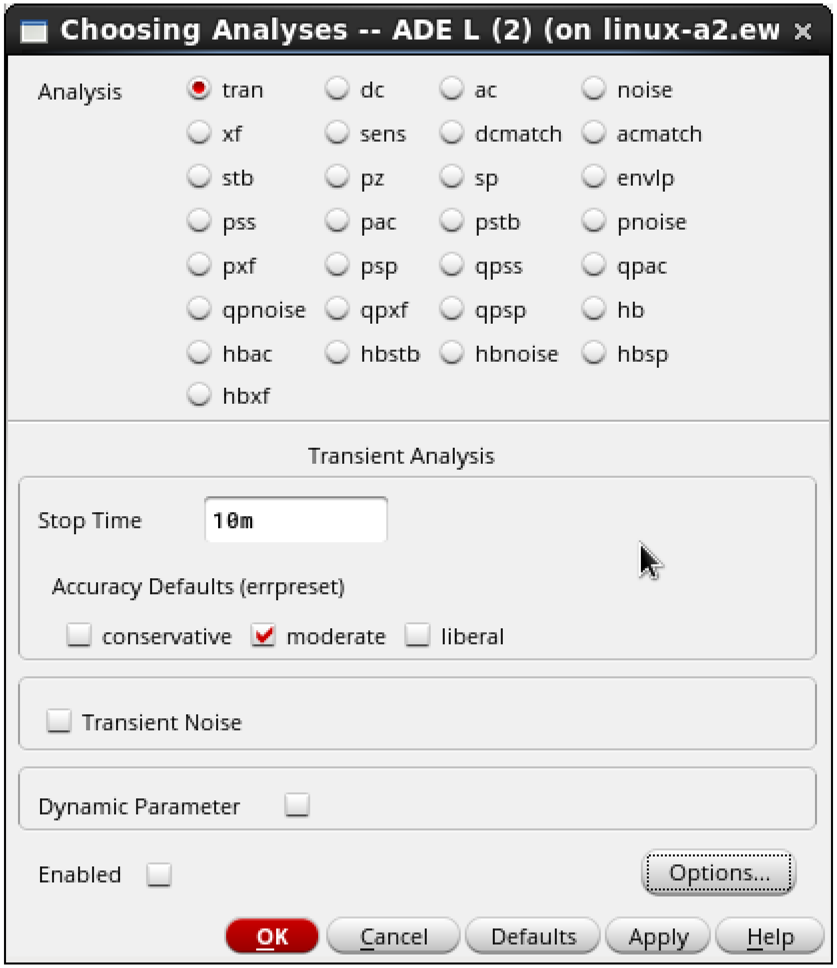
> Select Positive Net on schematic...

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)



Choosing Analyses -- ADE L (2) (on linux-a2.ew

Analysis
● tran ○ dc ○ ac ○ noise
○ xf ○ sens ○ dcmatch ○ acmatch
○ stb ○ pz ○ sp ○ envlp
○ pss ○ pac ○ pstb ○ pnoise
○ pxf ○ psp ○ qpss ○ qpac
○ qpnoise ○ qpxf ○ qpsp ○ hb
○ hbac ○ hbstb ○ hbnoise ○ hbsp
○ hbxf

Transient Analysis

Stop Time 10m

Accuracy Defaults (errpreset)
□ conservative ☑ moderate □ liberal

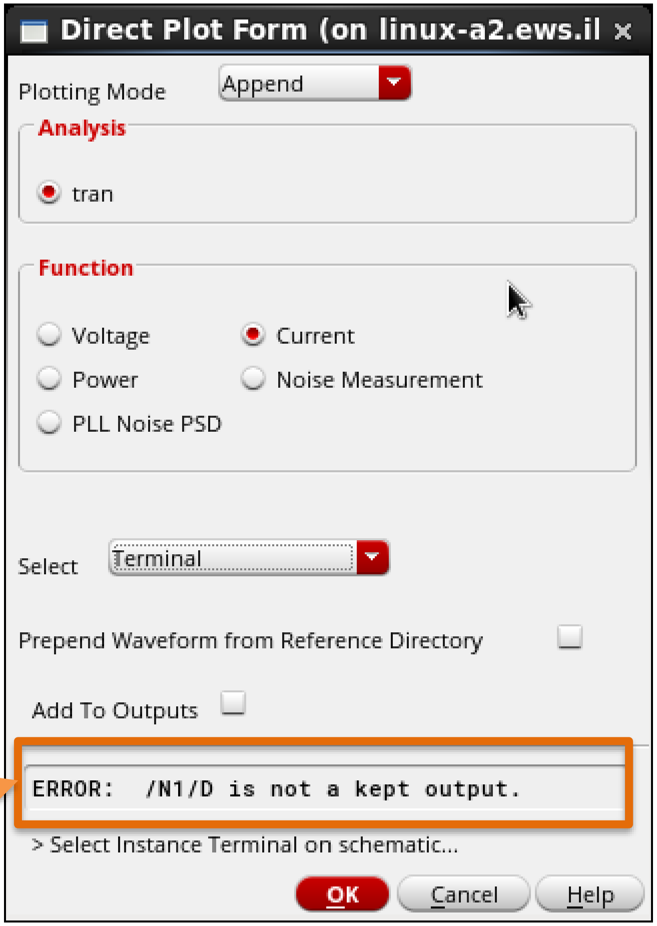
□ Transient Noise

Dynamic Parameter □

Enabled □

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



Direct Plot Form (on linux-a2.ews.il

Plotting Mode: Append

Analysis

● tran

Function

○ Voltage ● Current

○ Power ○ Noise Measurement

○ PLL Noise PSD

Select: Terminal

Prepend Waveform from Reference Directory □

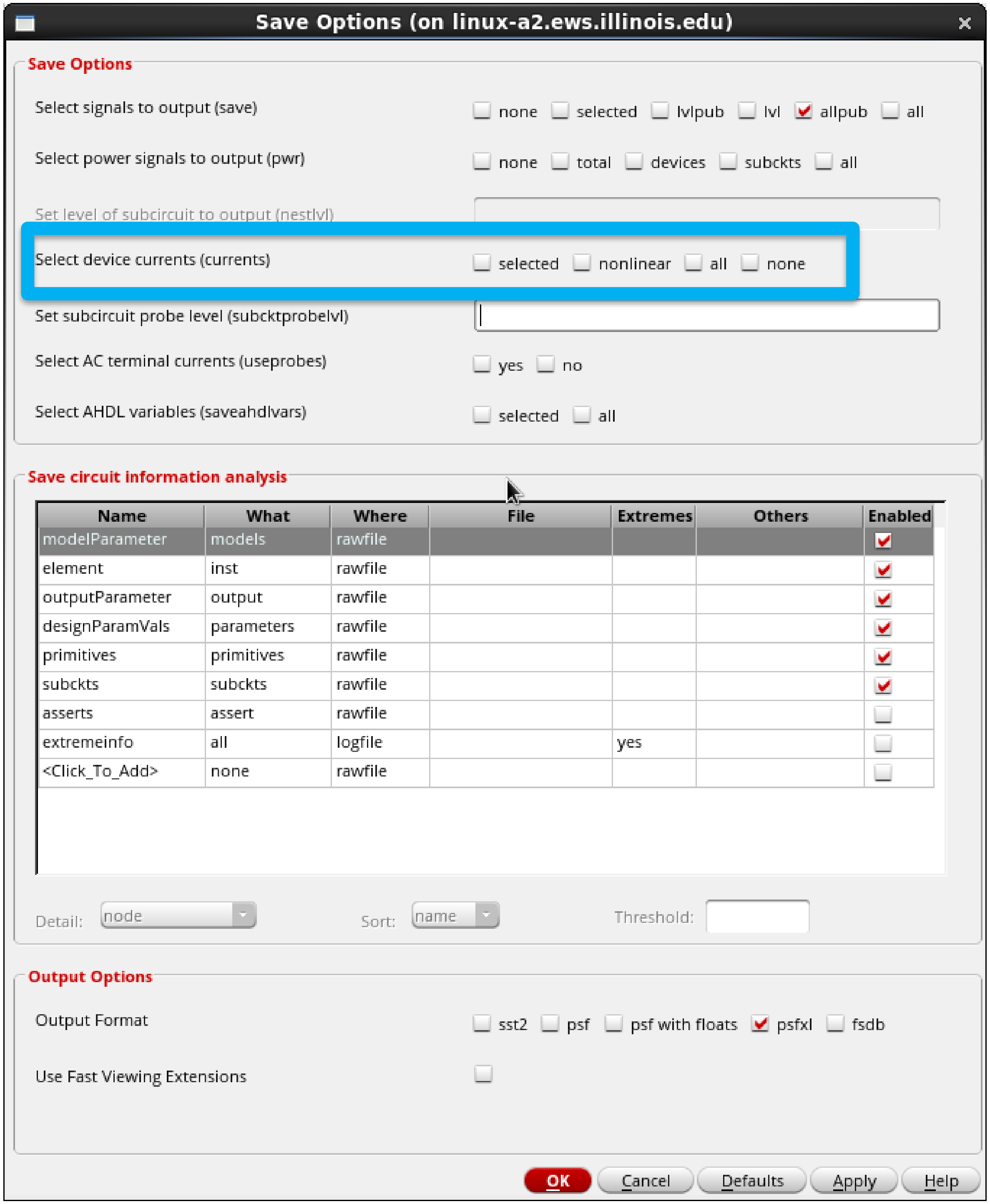
Add To Outputs □

```
ERROR:  /N1/D is not a kept output.
```

> Select Instance Terminal on schematic...

Transient Sim Results

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



Save Options (on linux-a2.ews.illinois.edu)

Save Options

Select signals to output (save) □ none □ selected □ lvlpub □ lvl ☑ allpub □ all

Select power signals to output (pwr) □ none □ total □ devices □ subckts □ all

Set level of subcircuit to output (nestlvl)

Select device currents (currents) □ selected □ nonlinear □ all □ none

Set subcircuit probe level (subcktprobelvl)

Select AC terminal currents (useprobes) □ yes □ no

Select AHDL variables (saveahdlvars) □ selected □ all

Save circuit information analysis

Name	What	Where	File	Extremes	Others	Enabled
modelParameter	models	rawfile				☑
element	inst	rawfile				☑
outputParameter	output	rawfile				☑
designParamVals	parameters	rawfile				☑
primitives	primitives	rawfile				☑
subckts	subckts	rawfile				☑
asserts	assert	rawfile				□
extremeinfo	all	logfile		yes		□
<Click_To_Add>	none	rawfile				□

Detail: node Sort: name Threshold:

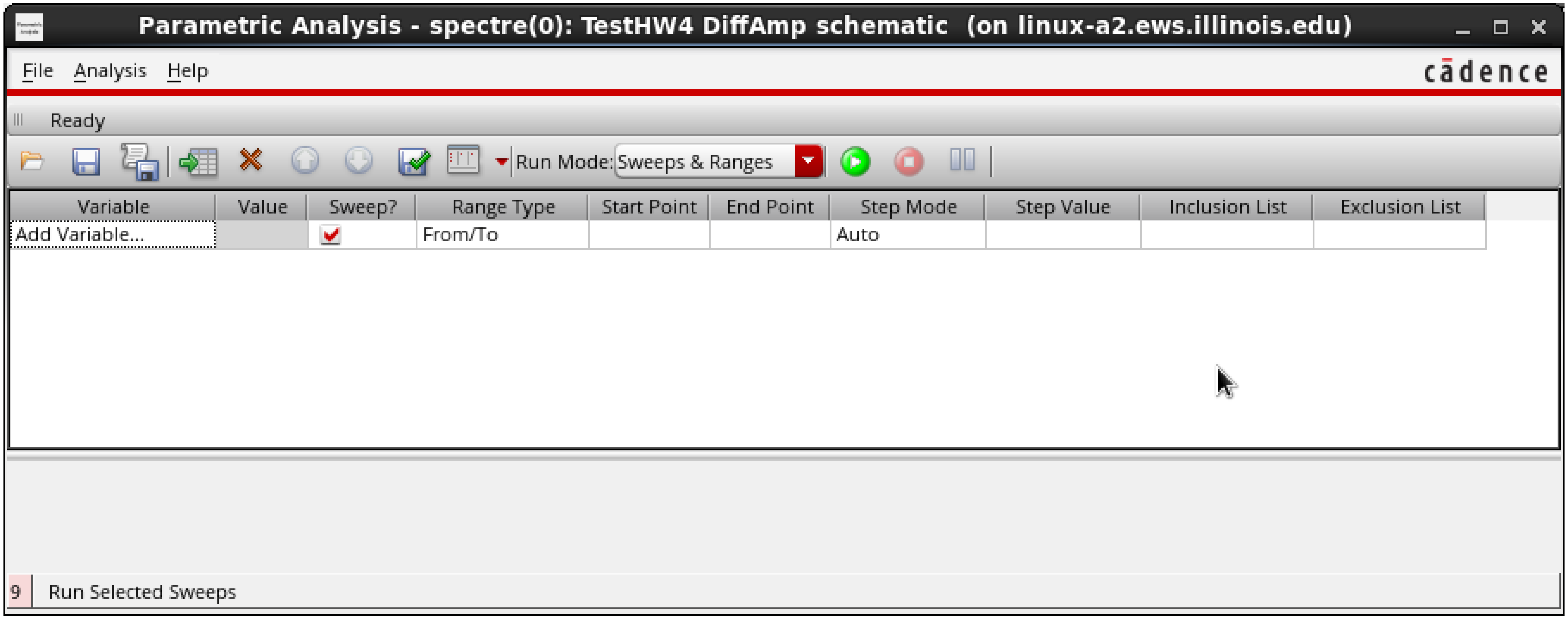
Output Options

Output Format □ sst2 □ psf □ psf with floats ☑ psfxl □ fsdb

Use Fast Viewing Extensions □

Parametric Analysis

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



Parametric Analysis - spectre(0): TestHW4 DiffAmp schematic (on linux-a2.ews.illinois.edu)

Ready

Run Mode: Sweeps & Ranges

Variable	Value	Sweep?	Range Type	Start Point	End Point	Step Mode	Step Value	Inclusion List	Exclusion List
Add Variable...		✓	From/To			Auto			

9 Run Selected Sweeps

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)