

Brief Spice Tutorial
ECE 3110, University of Utah, Fall 2002

By now, you have used SPICE in at least one other class. As a result, some familiarity is assumed. This tutorial will focus on the usage of input files for netlists. Some students may have experience using only the schematic capture version of PSPICE, but this tutorial should enable the transition to be less troublesome.

Since you already know the basics, a detailed example of a differential CMOS amplifier will be simulated and used as the basis of this tutorial.

The following circuit, shown in the figure below, will be simulated in the following ways: 1) the DC operating points, 2) the time domain response to a sinusoid (transient test), 3) the AC response (Bode Plot), 4) and the step response.

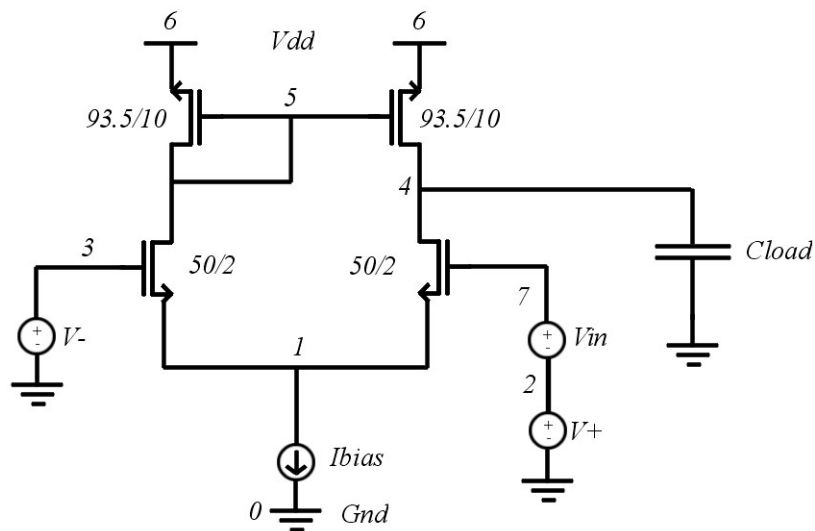


Figure One: CMOS Differential Amplifier

The first step will be to create a new .cir file. The first line of a netlist file *must* be a comment, which is any line beginning with the symbol “*”. The next thing needed is

the model to use for the transistors. In this case the models were downloaded from the Sedra & Smith textbook website. By now, your .cir file should contain the following (note the single * at the beginning):

```
*
* Differential-pair simulation for SPICE tutorial
* ECE 3110, Fall 2002, Dan Clement
*
* Spice MOSFET model from Sedra and Smith book:
*
.MODEL NMOS NMOS( level=2 vto=1 nsub=1e16 tox=8.5e-8 uo=750
+ cgso=4e-10 cgdo=4e-10 cgbo=2e-10 uexp=0.14 ucrit=5e4 utra=0 vmax=5e4 rsh=15
+ cj=4e-4 mj=2 pb=0.7 cjsw=8e-10 mjsw=2 js=1e-6 xj=1u ld=0.7u )
*
.MODEL PMOS PMOS( level=2 vto=-1 nsub=2e15 tox=8.5e-8 uo=250
+ cgso=4e-10 cgdo=4e-10 cgbo=2e-10 uexp=0.03 ucrit=1e4 utra=0 vmax=3e4 rsh=75
+ cj=1.8e-4 mj=2 pb=0.7 cjsw=6e-10 mjsw=2 js=1e-6 xj=0.9u ld=0.6u )
*
```

The next thing to be added are the DC supplies. In this circuit there is one DC current supply and two DC voltage supplies.

```
* Define the bias current.
Ibias 1 0 200u
*
* Define the DC voltage sources.
Vdd 6 0 5V
V- 3 0 2.5V
V+ 2 0 2.5V
Vin 7 2 0V
*
```

Note the syntax for the DC sources: *Ixxx (or Vxxx) node+ node- value*

Note also that we use a “u” to denote microamps since a lowercase “u” resembles the Greek letter mu (μ).

Now that the bias sources are taken care of, add in the devices and components in the circuit. For this circuit there are four transistors and one load capacitor.

```

* Add the transistors in
*input pair first:
M1 5 3 1 0 NMOS L=2u W=50u
M2 4 7 1 0 NMOS L=2u W=50u
*Now the current mirror devices
M3 5 5 6 6 PMOS L=10u W=93.5u
M4 4 5 6 6 PMOS L=10u W=93.5u
*
* Capacitive load
Cload 4 0 5p
*

```

Note the syntax: for the transistor,

Mxxx drain gate source bulk MODEL_NAME L=x W=x

Note that for MOSFET transistors, we must specify their length (L) and width (W). Since their dimensions are expressed in micron (micrometers), we again use a “u”.

For the capacitor (same for resistors and inductors, except change C to R or L)

Cxxx node+ node- value

Now the circuit is defined. The operating point can be simulated with just a couple more lines of code. The calculate operating points command, .OP, needs to be included. Also, every SPICE input file (.cir file) needs to have the .END command at the end. Add the following lines of code and save the file:

```

* Request the DC operating point to be output to .out file
.OP
*
.END

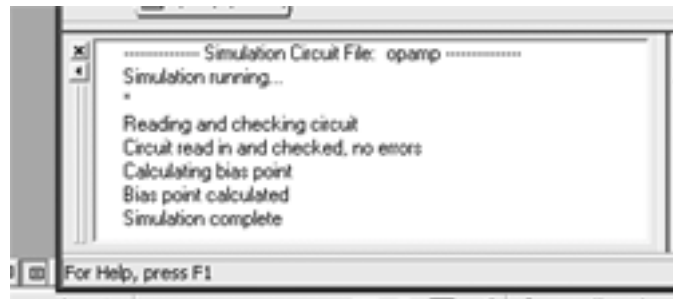
```

At this point the circuit can be loaded and simulated. For the PSPICE used in the lab, open the PSPICE A/D Lite program. Open your file as type .cir. Make sure the type associates as .cir because it won't run the simulation if you don't. You should be able to see your .cir file in the program now. To run the simulation, click on the blue “play” button as shown below:



Simulation for SPICE tutorial
on Clement

Note that the name of the input file will be shown in the text box to left of the “play” button. In this case it was called opamp. The simulation should run. If there are errors you will get a warning dialogue box that pops up. In the bottom left, a summary is displayed that gives status reports of the simulation. Shown in the figure below is the result of a successful operating point analysis. There is some information to the right also, but this information is really only needed for specialized situations which will not be encountered in this class.



All the simulation data is written to the .out file. The .out file will include whatever output you specify using .OP or .PRINT commands, a copy of your input file, and other things. It resembles a log file and you have to look through it to get what you want. In this case, the operating points will be shown. Here is what it should look like (might be formatted slightly differently):

```

**** 09/02/02 22:48:40 ***** PSpice Lite (Mar 2000) *****
*
****      SMALL SIGNAL BIAS SOLUTION      TEMPERATURE = 27.000 DEG C
*****
NODE   VOLTAGE   NODE   VOLTAGE   NODE   VOLTAGE   NODE   VOLTAGE
(  1)   2.0150  (  2)   2.5000  (  3)   2.5000  (  4)   2.5010

(  5)   2.5010  (  6)   5.0000  (  7)   2.5000

VOLTAGE SOURCE CURRENTS
NAME           CURRENT

Vdd            -2.000E-04
V-             0.000E+00
V+             0.000E+00
Vin            0.000E+00

TOTAL POWER DISSIPATION 1.00E-03 WATTS

```

Note that the DC voltages are printed, along with the node number. The power consumption is also summarized.

Now that the circuit is up and working, the rest of the analysis is easy. The next item to simulate is the transient (time-domain) response to a sinusoid. The Vin voltage source will be defined as a sine wave at 10kHz. The following code defines the voltage source and type of analysis desired:

```

* Define the input sine wave (transient, time domain)
*Vin 7 2 sin(0 15mV 10kHz)
*
* Run the transient test:
*.tran .1u 400u 0 .1u
*
* request that all nodes be probed
*.probe

```

Here is the syntax for the sine wave:

Vxxx node+ node- sin(DC_VALUE PEAK FREQUENCY)

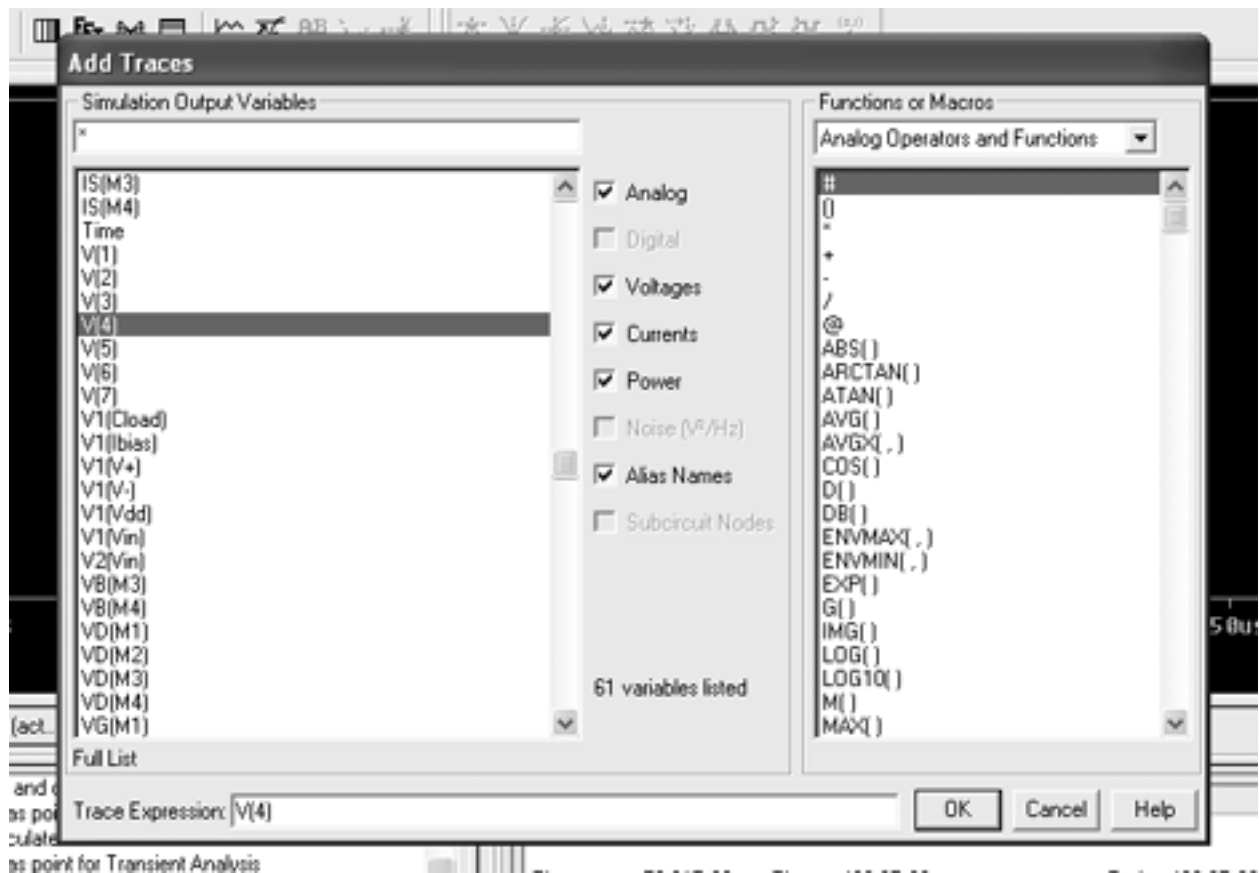
The syntax for the .TRAN analysis command:

.tran plot_time_step time_stop time_to_start_plot_at calculation_time_step

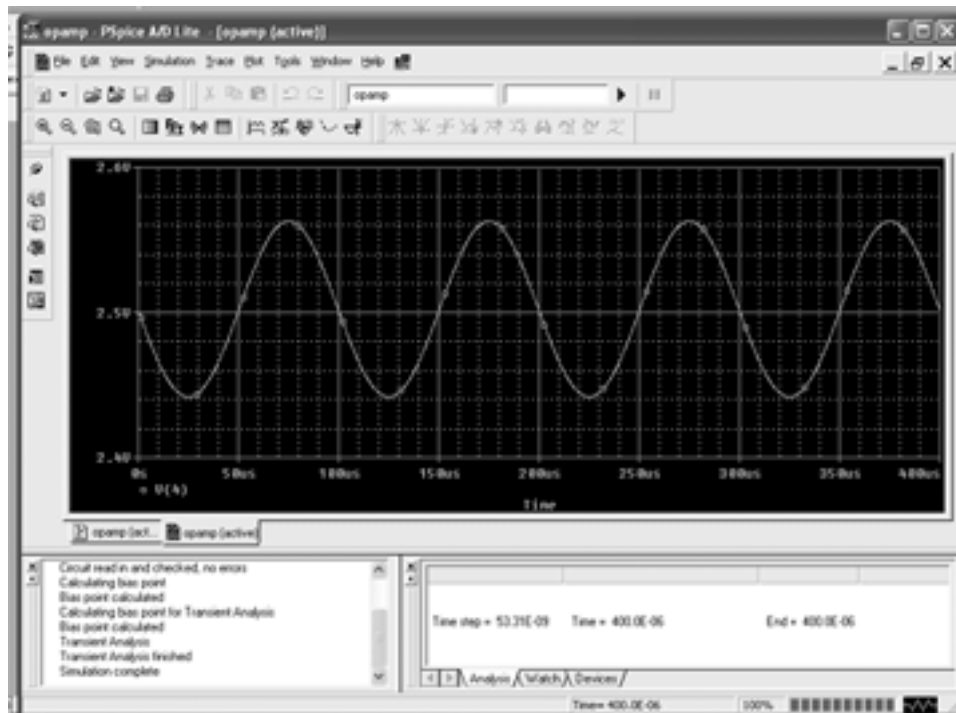
Request that the data be available for plotting after the simulation:

.PROBE

Add the code above, save the file, and re-simulate. When the simulation is finished, an empty graph should show up. To plot the data, go to the trace menu and click on add trace. The following window will pop up:



Single click on the V(4) as shown, and click OK. This should produce a plot in the previously empty graph. That's it for the transient test. Here is what it should look like:



The AC response will be simulated next. It's basically the same procedure as before except for an AC sinusoidal source and a .ac command. Note that an input voltage amplitude of 1V is used so that the transfer function equivalent to the output voltage ($V_{out}/1 = V_{out}$). Here is the code:

```
* Define the input sine wave (ac, frequency domain, use 1 Volt to make H(s) easier)
Vin 7 2 AC 1
*
* Run the AC test:
.ac dec 500 1 10G
*
* request that all nodes be probed
.probe
*
.end
```

Once again, here is the syntax:

AC sine wave:

Vxxx node+ node- AC magnitude

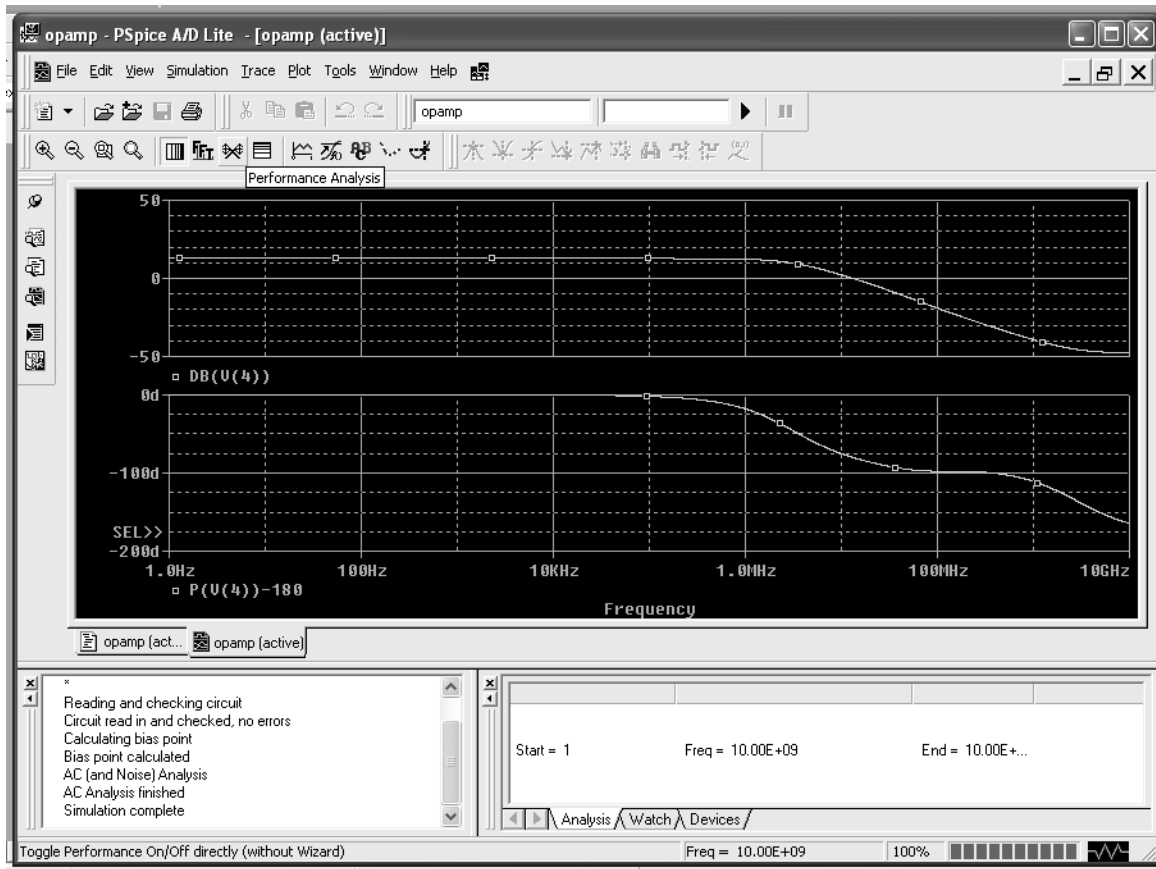
.AC command:

.ac analysis_type(dec for log space) number_points_per_decade start_f end_f

Add the above code to the .cir file, but this time comment out the transient sinusoidal source, and the .tran command line. Only one type of analysis can be requested in a simulation, with the exception of .op. To comment out a line, put a * character in front of the comment. If you use this method of commenting out the other analyses, you can use one netlist for each simulation type, making it very fast and easy to finish and to switch back and forth.

Run the simulation. When the blank plot is displayed, go to the plot menu and select “add plot to window”. Now there are two sets of graphs available. Click the mouse on the top graph so that SEL>> is displayed on the y-axis. Open the add trace dialogue as before, except this time click on the DB() function from the right hand side list. After you click on DB(), click on V(4). This will apply the DB() function to the output voltage, V(4). Click OK. This will plot the magnitude portion of your Bode plot.

To plot the phase, click on the lower graph so that SEL>> is shown on the y-axis. Open the add trace dialogue again and select the P() function and then on V(4). This will plot the phase of your Bode plot. Before you click ok, in the bottom text area, after P(V(4)), type in -180. This will subtract 180 degrees from the phase. This is needed because the amplifier is an inverting amplifier, and the true phase response needs to start at 0 degrees. You should always report the phase this way. Click OK. The graph should look like this:



The graphs can be printed from this environment at any time. Just click on the printer symbol as in most Windows programs.

There are features to add text labels, titles, and cursor markers to the graphs. These are pretty easy to figure out and will be left to you. You may need to polish up the graphs to make them suitable to turn in and/or to report quantities such as the -3dB point.

The final type of simulation is also a transient test, except this time a square wave will be input. This is done to test the slew rates of the differential pair, an important quantity that you won't have to worry about for a while. Just note the syntax of the square wave.

Change V_{in} to the following:

```
Vin 7 2 pulse(0 15m 0 1n 1n 50u 100u)
```

The syntax is:

Vxxx node+ node- pulse(DC_value PEAK_value delay risetime falltime pulse_width period)

The transient portion will stay the same as the sine wave was. Everything else is the same.

For reference, the entire input file used is repeated here for clarity. Note that it is currently set up for the square wave simulation.

```
*
* Differential-pair simulation for SPICE tutorial
* ECE 3110, Fall 2002, Dan Clement
*
* Spice MOSFET model from Sedra and Smith book:
*
.MODEL NMOS NMOS( level=2 vto=1 nsub=1e16 tox=8.5e-8 uo=750
+ cgso=4e-10 cgdo=4e-10 cgbo=2e-10 uexp=0.14 ucrit=5e4 utra=0 vmax=5e4 rsh=15
+ cj=4e-4 mj=2 pb=0.7 cjsw=8e-10 mjsw=2 js=1e-6 xj=1u ld=0.7u )
*
.MODEL PMOS PMOS( level=2 vto=-1 nsub=2e15 tox=8.5e-8 uo=250
+ cgso=4e-10 cgdo=4e-10 cgbo=2e-10 uexp=0.03 ucrit=1e4 utra=0 vmax=3e4 rsh=75
+ cj=1.8e-4 mj=2 pb=0.7 cjsw=6e-10 mjsw=2 js=1e-6 xj=0.9u ld=0.6u )
*
* Define the bias current.
Ibias 1 0 200u
*
* Define the DC voltage sources.
Vdd 6 0 5V
V- 3 0 2.5V
V+ 2 0 2.5V
*
* Add the transistors in
*input pair first:
M1 5 3 1 0 NMOS L=2u W=50u
M2 4 7 1 0 NMOS L=2u W=50u
*Now the current mirror devices
M3 5 5 6 6 PMOS L=10u W=93.5u
M4 4 5 6 6 PMOS L=10u W=93.5u
*
* Capacitive load
Cload 4 0 5p
*
* Define the input sine wave (transient, time domain)
*Vin 7 2 sin(0 15mV 10kHz)
Vin 7 2 pulse(0 15m 0 1n 1n 50u 100u)
*
* Define the input sine wave (ac, frequency domain, use 1 Volt to make H(s) easier)
```

```

*Vin 7 2 AC 1
*
* Request the DC operating point to be output to .out file
.OP
*
* Run the transient test:
.tran .1u 400u 0 .1u
*
* Run the AC test:
.ac dec 500 1 10G
*
* request that all nodes be probed
.probe
*
.end

```

That concludes the basic tutorial for SPICE. Hopefully the example is enough to get you started. If you need more in depth syntax help, look in your textbook. Appendix C has a good summary of the syntax, with the exception of voltage sources. Sinusoidal and pulse sources were covered in the tutorial so this shouldn't be a problem.

A good web site covering SPICE may be found at:

<http://www.seas.upenn.edu/~jan/spice/spice.overview.html>

Another great source is the book:

Introduction to Pspice, A supplement to ELECTRIC CIRCUITS, FIFTH EDITION, Susan A. Riedel and James W. Nilsson. ISBN 0-201-89582-X, 1997 Addison Wesley Longman, Inc.

SPICE Tutorial Addendum

One more comment on the usage of the output file: If the .PRINT command is invoked, the variables listed will be output in array format to the output file. For example, if the transient simulation is performed and V(4) is your output voltage, typing .PRINT TRAN

V(4) in your .cir file will cause the time and amplitude points of the simulated output voltage to be printed to your output file. These data points can be copied and pasted into their own document and saved. In MATLAB, the file can be opened and the data can be read directly into a matrix. (Example: “load data_only.dat”)

For more information on how to format your output using .PRINT, see your textbook, Appendix C.