
Document Outline

- A. Accessing PSpice
 - B. Using PSpice with Netlist Circuit Descriptions
 - C. Fundamentals of SPICE Netlist
-

A. Accessing PSpice

OrCAD PSpice is installed in the Division of Engineering Computer Services (DECS) labs, which have PCs for your use in the Engineering Building rooms 1307, 1312, 1318, 1320, 1328, 2200 and 2314. Please read the signs on the doors to see when the rooms are reserved.

PSpice 9.1 student version is available for free download at many sites on the internet including:
<http://www.engr.uky.edu/~cathey/pspice061301.html>
<http://www.electronics-lab.com/downloads/schematic/013/> (avoid advertising and scroll to bottom for the correct download. This site also has links (untested) to help files and tutorials).

A detailed discussion of PSpice (and SPICE in general) can be found at <http://www.uta.edu/ee/hw/pspice/>

B. Using PSpice with Netlist Circuit Descriptions

Dr. Wierzba has a brief walkthrough for PSpice on any of his websites for ECE 201/202/203 or ECE 302/303 at <http://www.egr.msu.edu/~wierzba/index.html>.

You can use any text editor to create a SPICE netlist. PSpice has a built-in text editor so this document will assume that editor is being used.

Launching PSpice

DECS Labs

Start >> Programs >> Cadence SPB >> ASM Simulator >> ASM Simulator

The exact location of the program may change. Please inform Dr. Mason if you discover it is in a different path.

Home Install

Start >> Programs >> PSpice Student >> PSpice AD Student

The exact path could vary depending on how you install, but you want the PSpice AD Student application rather than any of the others that are installed.

Basic Simulation with PSpice (an example)

The following borrows heavily from Dr. Wierzba's [PSpice Handout](#) to provide an introduction by example.

1. Create a new circuit:
File >> New >> Text File
2. Create a circuit netlist
Enter the circuit netlist to the right into the blank sheet that was created.
3. Save circuit netlist
File >> Save As
and choose an appropriate location on your PC or "M" drive.
Under *File name* type *ex1.cir*.
Use the .cir extension so PSpice will know it is a circuit file. If you use another text editor, be user to save as an ASCII text file; other file types may have character codes PSpice will not recognize.
4. Close file and reload

```
Example 1
V1 1 0 100
V2 3 4 50
R1 1 3 10
R2 1 2 3
R3 3 2 2
R4 5 4 4
IS 5 2 5
R5 5 0 6
.PROBE
.TRAN 1 200 0 1
.END
```

You must close the file once after it is created and reload it before it will simulate properly.

File >> Close

File >> Open

set the *Files of type* drop box to *.cir and then select your file, ex1.cir

5. Run simulation

At the top of the screen under the main menu you should see your file name (ex1).

Press the blue triangle right of the filename (▶) to simulate your circuit based on the analysis line(s) specified in your netlist.

If there are errors in your netlist, the simulation will not run. Check the status window in the lower left corner to see if it says "Simulation complete".

6. Fix errors

If there are errors, you can get more information about the errors by looking at the "output" file that is saved in the same location as your .cir file.

File >> Open

set the *Files of type* drop box to *.out and then select your file, ex1.out

After identifying the error(s), return to the .cir file by selecting the ex1.cir tab near the bottom of the screen. You can fix the errors directly in this .cir file without reloading.

Save the revised .cir file.

Press ▶ to simulate your circuit again.

7. Display simulation results

When the simulation is successful, a black window will open where you can display *traces*, graphs of signals in your circuit vs. time, frequency, etc. depending on the analysis mode.

Trace >> Add Trace

To display the voltage at node 1, you can either

a) select $v(1)$ from the Simulation Output Variables or

b) enter $v(1)$ into the *Trace Expression* dialog box

Note that all signals listed in the output list are node voltages. For example, V(R3) does not show the voltage "across" element R3; rather it shows the voltage at the first node assigned to R3, in this case node 3.

The check boxes on the right side of the *Simulation Output Variables* section permit you to toggle on/off different types of signals. For more complex circuits, you may want to disable some elements, such as Currents, Alias Names, or Subcircuit Nodes to simplify the list.

The *Trace Expression* dialog box allows you to enter mathematical expressions using node voltages and can be very useful. For example, to display the voltage drop from node 3 to node 4, you can enter $V(4)-V(3)$ and click OK. For gain plots vs. frequency (.AC analysis), you may want to show the signal in units of dB, which can be done by entering $DB(V(out))$, where 'out' is the output node.

8. Cursor and measurement functions

To read precise values from a trace, first activate the cursor.

Trace >> Cursor >> Display

This will place two cursors (crosshairs) on the trace. The "reference" cursor is set by right-clicking on the trace and the "measurement" cursor is set by left-clicking or pressing the left/right arrow keys.

The reference cursor can be move in small steps by holding down SHIFT and using the left/right arrow keys.

The trace X and Y values at the point of the measurement cursor will be displaced in a box at the bottom of the screen.

For the example circuit, if you move the cursor on the V(1) trace to 60s on the x-axis, you should see a value (y-axis) of 100.

You can remove the cursors, for example to print, by repeating the display cursor command.

Trace >> Cursor >> Display

9. Add labels

You add a label of the X and Y values at the point of the cursor using a label mark.

Plot >> Label >> Mark

You can also add text, lines, etc to the plot using the Plot >> Label commands.

10. To print the active window pane (select a tab at the bottom to change active window panes)

File >> Print

11. To exit

File >> Exit

You add a label of the X and Y values at the point of the cursor using a label mark.

If you are in a DECS lab, be sure to log off before leaving your computer.

Important Notes About PSpice

- PSpice requires a .PROBE statement in the netlist or it will not generate a trace (.dat) window.
- Ground must be assigned to node '0'. If you assign ground to a different node name, include a DC voltage source of 0V connecting your ground node to node 0.

C. Fundamentals of SPICE Netlist

- 1: Circuit Description
- 2: Analysis Methods
- 3: Subcircuit Definitions
4. Semiconductor Models

C.1. Circuit Description

Although most SPICE simulators today use a graphical interface, an understanding of SPICE *netlist* concepts is very useful. A netlist is a complete text description of the circuit which includes the *element lines*, *analysis statement(s)*, and a list of *model parameters* for semiconductor devices. Whether using a graphical interface or a text netlist, all of the parameters discussed below are necessary to describe the circuit and desired simulation. This section explains the element lines, and model and analysis statements are covered below.

Circuits for SPICE simulation are described the *elements* of the circuit and a list of the *nets* (nodes) each element is connected to. Elements include *sources* (voltage, current, etc.) and devices (resistors, capacitors, transistors, etc.). Elements must be assigned unique names starting with a designated letter as shown below. The element name is the first part of the *element line* which also contains a list of the nets each terminal of the element is connected to and key data describing the device. Nets, or nodes, can be assigned names or numbers depending on the specific software used. Data in the element line describes the element, e.g. the DC voltage of a voltage source or the resistance of a resistor. Because semiconductor elements are more complex to model than linear elements, modeling parameters for semiconductor devices are contained in a separate part of the netlist called a .MODEL statement. To indicate which .MODEL statement should be used for each element, semiconductor element lines must contain a model name (which also appears in the .MODEL statement). Although most of the parameters for semiconductors are in the model statement, the parameters that can be changed by designers (i.e. non-process specific parameters) are assigned in the element line. In general, element line parameters for semiconductor devices relate to the size of the device, e.g. the length and width of an MOS transistor.

Although graphical SPICE programs do not use element lines, the data entered in the element line of a netlist must also be specified. For example, in many SPICE programs you can double-click a device and then assign the appropriate parameters. For semiconductor devices, you may have to open the model description to alter those parameters.

A nonfunctional example showing many of the common circuit elements is shown in the netlist below.

```
Vdd      1  0  5
R1 1     2  1K
R2 2     0  2K
Mn1      2  3  0      0      cmosn  L=2e-6 W=4e-6
Mp1      2  3  1      1      cmosp  L=2e-6 W=4e-6
C1 3     0  1n
```

Note: some (but not all) units are implied and do not need to be specifically written. For example, Vxx is assumed to be in volts, Rxx is assumed to be in ohms, etc.

Element Line Descriptions (for *some* common elements –others are available)

Voltage Source

Vxx +node -node DC value

Current Source

Vxx +node -node DC value

Resistor

Rxx node1 node2 value

Capacitor

Cxx node1 node2 value

Diode

Dxx +node -node model_name <area>

plus other optional parameters

Bipolar Transistor

Qxx collector base emitter <bulk> model_name <area>

plus other optional parameters

MOS Transistor

Mxx drain gate source bulk model_name <L=value> <W=value> <AD=value> <AS=value>

plus other optional parameters

Sub-Circuits

Xxx node1 node2 node3 ... node_n subcircuit_name

subcircuit_name must match name in .Subcircuit definition statement

Note: parameters in <> are optional and not required for simulation.

Device and Node Name Formats

Traditionally, devices and circuit nodes were labeled using numbers. However, most SPICE programs used today permit devices and nodes labels using alphanumeric characters. It is generally preferable to use descriptive text names like 'Vout' rather than, say, '5' because it makes your circuit easier to identify. But check that your chosen SPICE simulator permits alphanumeric characters or try using only numbers if you get errors you can't explain. **Important:** Some simulators, including OrCAD PSpice, require ground to be defined as node '0' and may not converge otherwise.

Scientific Notation and Metric Abbreviations

SPICE supports scientific notation where "e" (for "exponent") denotes the power of ten to be applied.

SPICE also supports the abbreviations listed below. For example:

$$0.0123 = 1.23e-2 = 12.3e-3 = 12.3m \text{ (where m is for milli)}$$

$$1230000 = 1.23E6 = 1.23MEG = 1.23E3K = 0.00123G$$

Note that SPICE is not case sensitive so "m" or "e" is the same as "M" or "E".

The following metric abbreviations are supported by SPICE

m	milli	10^{-3}	k	kilo	10^3
u	micro	10^{-6}	meg	mega	10^6
n	nano	10^{-9}	g	giga	10^9
p	pico	10^{-12}	t	tera	10^{12}
f	femto	10^{-15}			

C.2. Analysis Methods

SPICE can perform a variety of types of analysis on a circuit. Some of these are briefly described below.

AC Analysis

To sweep all AC sources across a range of frequencies, use the .AC command.

`.AC <Lin/Dec/Oct> N FSTART FSTOP`

Select either Lin, Dec, or Oct for a linear, decade, or octave scaled sweep. DEC is most common in frequency response analysis.

N is the number of points analyzed per scale unit (Lin/Dec/Oct).

FSTART and FSTOP are the beginning and ending frequencies of the AC analysis.

Transient Analysis

A transient analysis will tell us what our circuit is doing as a function of time. A transient analysis has the following properties which must be defined.

`.TRAN TSTEP TSTOP <TSTART> <TMAX>`

TSTEP is the time step used between analysis points, and the smaller the value is, the more resolution you will get in your simulation. A typical value is 1nsec.

TSTOP is the time when you want your transient analysis to stop. This value will depend on the frequency of your input signal(s) and the time it takes for your circuit to produce its final output.

TSTART is used if you want the analysis to start at some time other than Time=0.

TMAX is used to set the largest step time that will be used. This parameter is typically not needed.

DC Analysis

Use a DC analysis if you want to view the response of the circuit when a source (voltage or current) sweeps between two DC values. The command and parameters are:

`.DC Source_name START STOP STEP`

Source_name is the name of the source that will be swept.

START and STOP are the beginning and ending values of the sweep.

STEP is the increment value used during the sweep.

Operating Point Analysis

.OP is the operating point command and it will produce a list of the initial operation conditions including node voltages and device currents. It is useful to view bias values without doing a full simulation.

SHOW Device Parameters

.SHOW will produce a list of the bias currents and voltages as well as small signal parameter values for each device in your circuit. You can get information for only select devices by typing the name of the device in the box after clicking on the .SHOW option. This analysis is very useful for ensuring all of your semiconductor devices are operating in the active/saturation region.

.SHOW may not be available in all simulators.

See SPICE manuals for more information and additional analysis types.

C.3. Subcircuit Definitions

For complex circuits, it is often preferable to form a *subcircuit* of a functional circuit block and then link that circuit into the main circuit as a single subcircuit element. This is especially useful when the subcircuit will be included multiple times in a circuit, for example creating a subcircuit defining an opamp that will be included three times in an instrumentation amplifier.

There are two primary components to subcircuits: the element line description and the subcircuit definition. The element line description is placed within the main circuit netlist and contains a list of nodes

defining how the subcircuit is connected within the main circuit and a subcircuit ID that refers to a specific subcircuit.

Example subcircuit element line description

```
Xopamp1 vin vref vdd gnd vout opamp741
      ( node names ) (subcircuit ID)
```

The subcircuit definition can appear anywhere within the main circuit netlist and is usually placed after the main circuit is listed. A subcircuit must always begin with a .SUBCKT statement that includes the subcircuit ID and defines the externally accessible nodes of the subcircuit. Following the .SUBCKT statement is a sub-netlist describing the subcircuit, including node names that define the externally accessible nodes of the subcircuit. The subcircuit must end with a .ENDS statement. Note that many commercial electronic components have defined subcircuits that can be obtained from the internet.

The node names used in the subcircuit element line description do not need to match those used in the subcircuit definition. However, the node order (e.g., input = first, output = second, etc.) is set by the subcircuit definition and must be matched by the element line description.

Example subcircuit definition

```
.SUBCKT lpfilter in out
R1 in out 1k
C1 out gnd 1u
.ENDS
```

C.4. Semiconductor Models

Semiconductor devices are nonlinear, and as a result several parameters are required to model their operation in circuit simulators such as SPICE. The required parameters are specific to each device type and to the equations that are used to model the performance of each device. For example, there are a wide range of equation-models used to describe MOSFETs, and the equations we learn in class are for the most basic model.

Semiconductor Device .MODEL Statements

The .MODEL statements for all devices contain the name of the model (referred to by the element line), the device type (D for diode, etc.) and a list of parameters specific to each device type. Parameters not provided in the .MODEL statement are assigned default values. Note the parameters are enclosed in parentheses.

PN Junction Diodes

```
.MODEL Model_Name D <(parameter 1=value1 parameter 2=value2 . . . )>
```

Bipolar Junction Transistors

```
.MODEL Model_Name NPN <(parameter 1=value1 parameter 2=value2 . . . )>
.MODEL Model_Name PNP <(parameter 1=value1 parameter 2=value2 . . . )>
```

MOS Field Effect Transistors

```
.MODEL Model_Name NMOS <(parameter 1=value1 parameter 2=value2 . . . )>
.MODEL Model_Name PMOS <(parameter 1=value1 parameter 2=value2 . . . )>
```

Semiconductor Device Parameters

For each device type there is a list of parameters that define the device. In general, these parameters are the same as those seen in the equations used to model the device. However, there are additional parameters that model effects not described by the simple large signal models covered in class. Device parameters are determined by the processes used to fabricate the device, and accurate model parameters can generally be obtained from manufacturer datasheets or from integrated circuit foundries. The following is a brief list of the parameters you will need to consider for each device type.

PN Junction Diodes

Name	Parameter	Units	Default
IS	saturation current	A	1.0E-14
N	emission coefficient	-	1
BV	reverse breakdown voltage	V	infinite
RS	diode series resistance	Ω	0
CJO	zero-bias junction capacitance	F	0
VJ	junction potential	V	1
M	grading coefficient	-	0.5

Bipolar Junction Transistors

Name	Parameter	Units	Default
IS	saturation current	A	1.0E-16
BF	ideal max. forward beta	-	100
VA	forward Early voltage	V	infinite

(there are many other parameters for BJT that are beyond the scope of this class)

MOS Field Effect Transistors (only a few of many possible)

Name	Parameter	Units
LEVEL	model identification	-
TOX	gate oxide thickness	meter
NSUB	substrate impurity concentration	cm^{-3}
XJ	source/drain junction depth	meter
VTO	equilibrium threshold voltage	V
LAMBDA	channel length modulation (level 1 only)	V^{-1}
PHI	surface potential at strong inversion	V
KP	transconductance parameter	A/V^2
GAMMA	bulk threshold parameter	$\text{V}^{1/2}$
LD	lateral diffusion	meter
CGDO	gate-drain overlap capacitance/channel width	F/m
CGSO	gate-source overlap capacitance/channel width	F/m
CGBO	gate-bulk capacitance/channel length	F/m
CJ	zero bias s/d junction (bottom) capacitance/area	F/m^2
CJSW	zero bias s/d sidewall capacitance/length	F/m
MJ	bulk junction bottom grading coef.	
MJSW	bulk junction sidewall grading coef.	
PB	bulk junction potential	

Example Model Statements

```
.MODEL pn D (IS=10E-15 N=1 BV=50)
```

```
.MODEL npnbt NPN (IS=10E-15 BF=150 VAF=200)
```

```
.MODEL pnpbt PNP (IS=10E-15 BF=150 VAF=200)
```

```
.MODEL cmosn NMOS (LEVEL=1 UO=669.5 TOX=2.92E-08 NSUB=1.8050E+16
+ PHI=0.7 VTO=0.5940 KP=7.9174E-05 GAMMA=0.6546 LAMBDA=0.0134
+ XJ=0.2U TPG=1 LD=1.0530E-07
+ CGDO=1.8679E-10 CGSO=1.8679E-10 CGBO=4.3907E-10 CJ=2.8446E-04 MJ=5.2989E-01
+ CJSW=1.4208E-10 MJSW=1.0000E-01 PB=9.8338E-01)
```

```
.MODEL cmosp PMOS (LEVEL=1 UO=181.5 TOX=2.9200E-08 NSUB=4.4510E+15
+ PHI=0.7 VTO=-0.8135 KP=2.1464E-05 GAMMA=0.3250 LAMBDA=0.02
+ XJ=0.2U TPG=-1 LD=1.11E-09
+ CGDO=5.0000E-11 CGSO=5.0000E-11 CGBO=4.0071E-10 CJ=3.0160E-04 MJ=4.4794E-01
+ CJSW=1.8785E-10 MJSW=1.0476E-01 PB=7.6778E-01)
```