

Spice Tutorial

EE 115A Analog integrated Circuits I

Pedro Irazoqui-Pastor

SPICE stands for Simulation Program with Integrated-Circuit Emphasis and was developed at UC Berkeley in the '70s. There are many different versions of circuit simulators and most trace their roots to the free UNIX based Berkeley distribution. These include HSPICE, PSpice (what you'll be using), Cadence, 3f5 and many more. The use of Spice has become so ubiquitous that the word has gone from being an acronym to just being a name, like lasers and radars before it.

In this tutorial, you will learn the basics of Spice including:

- How to describe a circuit to Spice
- How to set the initial conditions for a circuit
- How to run a simulation in Spice, as well as some different sorts like:
 - o Transient analysis
 - o DC analysis
 - o AC analysis
- How to store and plot the various outputs of your simulation
- Why, even with a good simulator, you still need to use your brain

Circuits are described to Spice in a text file that is called the Spice deck. This file contains a title line, a netlist, a series of commands and definitions preceded by a period called cards, and a .end line, in that order. Spice is best learnt by example, and by doing so I recommend reading this tutorial while seated at a computer with Spice running, and actually performing the commands as you go along.

Example 1

First, sit down in front of any one of the PC's in one of the Boelter Hall computer clusters and log in. You will need a SEASnet account for this, if you do not have one you can get one on the second floor of Boelter Hall.

- From the Start menu select All Programs -> Orcad Family Release 9.2 -> Pspice
- Enlarge the program window to fill the workspace
- From the menu select File -> New -> Text File
- Enter the following Spice deck EXACTLY as shown:

*****BEGIN BELOW THIS LINE*****

The title of my sample spice deck must be the first line

- * Any comments are preceded by an asterisk and ignored
- * by Spice
- ** Circuit Description **
- * First I will describe my circuit in a netlist giving each
- * element, the nodes it connects to and it's parameter
- * values node 0 "zero" must always be ground. All other

* nodes should bear maninful names not just numbers.

* Input Voltage

* NAME NODE1 NODE2 SIN(OFFSET AMPLITUDE

* FREQ) DC OFFSET AC MAG

Vin in 0 sin(0V 1V 1Hz) DC 0V AC 1V

* RC Network

R1 in out 1k

C1 out 0 1uF

** Analysis Requests **

*Transient Analysis: TRAN (TSTEP TSTOP)

.tran 1ns 10s

*AC Analysis: AC (TYPE NOPOINTS START STOP)

.ac dec 10 1Hz 1MEG

** Output Requests **

* A .probe card allows you to plot any of the node signals

* later

.probe

.end

*****END ABOVE THIS LINE*****

- Save the file as rc.cir or any other name you like with .cir at the end
- From the menu select View -> Simulation Queue
- From the simulation queue menu select File -> Add Simulation
- Set type to .cir and select your spice deck
- From the menu select Simulation -> Run
- From the menu select View -> Simulation Results
- Select the [Transient] button
- From the menu in the PSpice window select Trace -> Add Trace
- Click on V(in) then on V(out) then on the [OK] button
- From the menu select Plot -> AC
- From the menu select Trace -> Add Trace
- In the right column click on LOG() and then in the left column click on V(in), then click on LOG() again, and then on v(out), finally click on the [OK] button

This deck described the circuit in Figure 1 with the nodes named in red where the ground node is ALWAYS node zero:

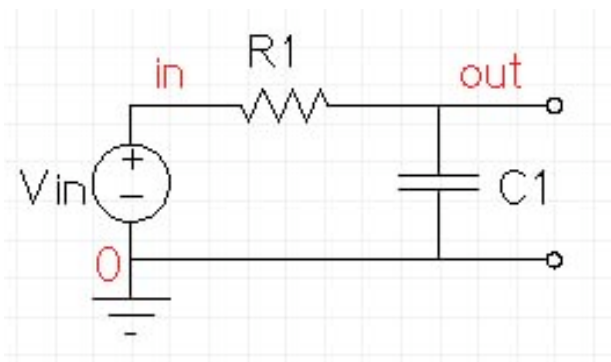


Figure 1: A Simple RC Circuit

That's it! Not so hard was it? You can also interactively plot voltages and currents after you've loaded your spice deck. Note that in this example we performed two types of analysis, transient and ac. The first calculates the time-dependent (hence the name transient) response of the circuit and plots the output as a function of time. The second calculated the frequency-dependent response of the circuit and plots the output, you guessed it, as a function of frequency. There is one more type of analysis which we will see in Example 3, it is called a dc analysis and it calculates the response of the circuit to a varying dc input voltage, and plots all outputs as a function of that dc input.

Any time you wish to do something new in Spice but realize that you don't know how to, do not despair, help is available. Simply click select Help from the menu and a very useful help menu will appear explaining most basic, and not-so-basic commands, to you. Alternatively buy, borrow, or sign-out the book called "Spice" by G. W. Roberts and A. S. Sedra.

Example 2

Remember the voltage doubler we saw in class a long time ago? In this example you will learn to use the online help, set the initial conditions, and simulate the doubler shown in Figure 2.

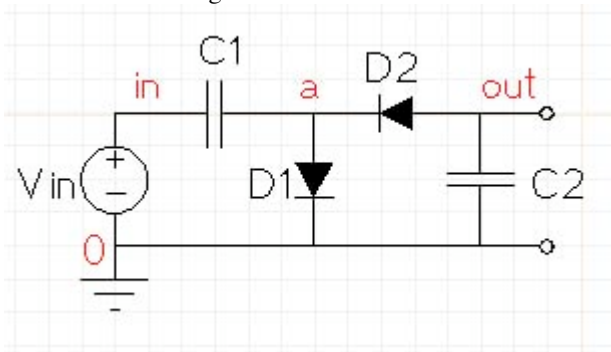


Figure 2: A Voltage Doubler

You will also learn how to define what are called models for diodes and transistors. First use help to find out how to set the initial voltages at nodes 2 and 3 to zero. This is done using the .ic card, but it's up to you to figure out how. Trust me, it isn't very difficult. Next define each diode like this in your spice deck:

D1 a 0 diode

Then after the netlist you will place a model card that will look like this:

```
.model diode D (IS=1e-14)
```

This says: "anything in the netlist called 'diode' is really of type D with parameter IS=1e-14. We have defined the saturation current, but many more parameters are available for us to play with. For a list, use the help. Those that we do not define remain at default values. Help will tell you what those values are. Finally, perform a transient analysis of the circuit from time 0s to 10s in 1ms increments. What do the voltages at nodes 2 and 3 look like? Does it work? Do you know how you might change this circuit to get an output equal to +2Vin instead of -2Vin? If you do, make the change and simulate it to prove to yourself this works.

Example 3

In this example you will create a spice deck to calculate the dc analysis of the circuit shown in Figure 3.

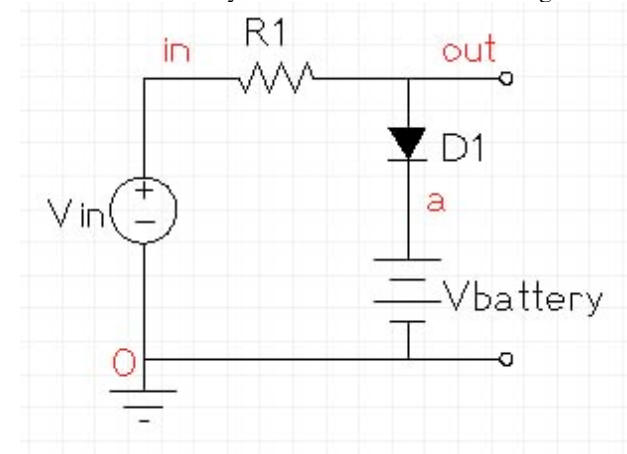


Figure 3: R-Diode Circuit

A dc analysis call in a spice deck looks like:

```
* DC Analysis (DC VSOURCE START STOP  
* INCREMENT)
```

```
.dc Vin -10V 10V 1mV
```

Try this out for several different values of battery voltage. If the battery voltage represents the forward voltage drop of a real diode, what can you say about the desirability of keeping that voltage high or low? After completing Example 4, and if you want a challenge, come back to this and tell me: If I implement this diode using a MOSFET, how might I design that FET so that the forward voltage drop would be low. Can you show me how this works in a plot?

Example 4

In this example we will introduce the common-source MOS amplifier, how to describe one, model one and implement one in the circuit shown in Figure 4.

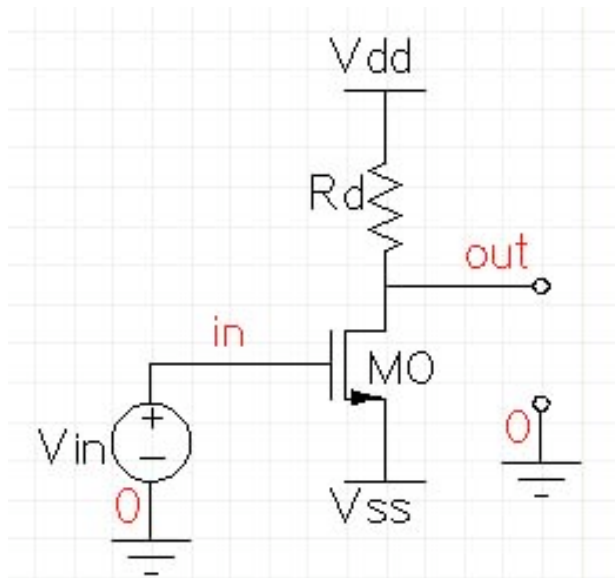


Figure 4: Common-Source MOS Amplifier

We will define each transistor like this:

M0 drain gate source body pch L=10u W=10u

M1 drain gate source body nch L=10u W=10u

And our model cards will look like this:

```
.model pch pmos (LEVEL=1 TOX=1e-7 U0=600
+VT0=0.7 LAMBDA=0.02)
```

```
.model nch nmos (LEVEL=1 TOX=1e-7 U0=600
+VT0=0.7 LAMBDA=0.02)
```

Where level defines the complexity of the model, Tox the oxide thickness, U0 the mobility, and Lambda the channel-length modulation. The spice deck which simulates this circuit is given below for your convenience:

*****BEGIN BELOW THIS LINE*****

Common-Source MOS Amplifier Circuit

** Circuit Description **

* Power Supplies

Vdd Vdd 0 DC 5V

Vss Vss 0 DC -1.25V

* Input Voltage

Vin in 0 DC 0V AC 1V

* Common-Source Circuit

* MOS D G S B

M0 out in Vss Vss nch L=10u W=10u

Rd Vdd out 1MEG

* Model Definitions

* TOX = Oxide thickness in Meters

* U0 = Surface mobility in cm/Vs

* VT0 = Zero-bias threshold voltage in V

* LAMBDA = channel-length modulation in 1/V

* CBD = base-drain junction capacitance

* CBS = base-source junction capacitance

```
.model nch nmos (LEVEL=1 TOX=1e-7 U0=600
+VT0=0.7 LAMBDA=0.02 CBD=20fF +CBS=20fF)
```

** Analysis Request **

* DC Analysis (SOURCE START STOP INCREMENT)

```
.dc Vin -10V 10V 1mV
```

* AC Analysis (AC TYPE NOPOINTS START STOP)

```
.ac dec 10 1Hz 1GHz
```

** Output Request **

```
.probe
```

```
.end
```

*****END ABOVE THIS LINE*****

Run the simulation and comment on the results. What effect does varying the drain resistor have and why. Change Vss and comment on the change in AC gain. How can you use the DC analysis to restore the gain modifying only the dc offset in the Vin line and nothing else? Finally, set CBD and CBS to zero, run the dc analysis, adjust the offset to maximize the gain and then run the ac analysis again, what has changed and why.

Example 5

Finally we are going to take a quick look at a MOSFET circuit called an inverter. Create a spice deck to simulate the MOS inverter in Figure 4.

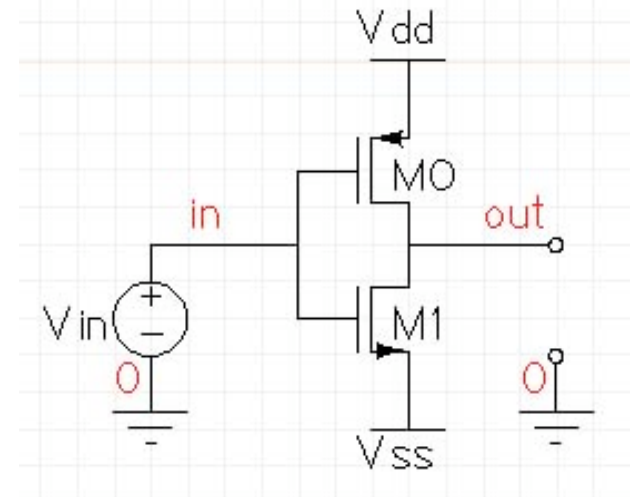


Figure 4: A MOS Inverter

Show me in a plot why they call this an inverter? What sort of analysis will show this best? Also tell me how you could use this as an amplifier and over what range of inputs this would work, and what the gain would be.

That's it, you now know the basics of Spice. Anything more, all the way to analog IC design just builds on what you've learned here. The key is to keep using the tool, and you will get better at it, and the better you get the more useful and powerful it will become.

Summary of Commands

Resistor	RNAME NODE1 NODE2 VALUE
Capacitor	CNAME NODE1 NODE2 VALUE
Voltage Source	VNAME POSNODE NEGNODE SIN(OFFSET AMPLITUDE FREQ) DC OFFSET AC MAGNITUDE
Current Source	INAME FROMNODE TONODE VALUE
Diode	DNAME ANODENODE CATHODENODE MODELNAME
Bipolar	QNAME COLLECTORNODE BASENODE EMITTERNODE MODELNAME
MOSFET	MNAME DRAINNODE GATENODE SOURCENODE BODYNODE MODELNAME W=WIDTH L=LENGTH
***	***
AC Analysis	.AC TYPE NOPOINTS STARTFREQ STOPFREQ
DC Analysis	.DC SOURCE START STOP INCREMENTSIZE
Tran Analysis	.TRAN TSTEP TSTOP
***	***
Allow probing	.PROBE