

ECE 3410 – Microelectronics I

SPICE Handout II

Dependent Sources

We previously examined independent voltage and current sources in SPICE. It is also possible to represent independent sources using the following statements:

Voltage controlled voltage source:	<code>Ename N1 N2 NC1 NC2 Value</code>
Voltage controlled current source:	<code>Gname N1 N2 NC1 NC2 Value</code>
Current controlled voltage source:	<code>Hname N1 N2 Vcontrol Value</code>
Current controlled current source:	<code>Fname N1 N2 Vcontrol Value</code>

For voltage-controlled sources: N1 and N2 are the positive and negative terminals, respectively, of the source. NC1 and NC2 are the positive and negative terminals of the control voltage. “Value” refers to the constant by which the control value is scaled.

Example: `E2 2 3 1 0 10`

This statement declares a dependent voltage source, connected between nodes 2 and 3, whose value is 10 times the voltage between nodes 1 and 0.

For current-controlled sources: the dependent source sources a current which flows from N1 to N2, and is controlled by the current through a voltage source named “Vcontrol,” scaled by the constant “Value.” Usually Vcontrol is a voltage source of 0V, referred to as a “meter.”

Transient Sources

SPICE is also capable of representing SIN, PULSE and PWL waveforms used for transient analyses. These are presented below using voltage sources. The same syntax is also valid for current sources.

Sinusoidal Sources:

`Vname N1 N2 SIN(V0 VA FREQ [TD THETA PHASE])`

where: V0 = offset voltage
VA = amplitude
FREQ = frequency
TD = delay (the source is a DC V0 until time TD)
THETA = damping factor (zero for undamped signals)
PHASE = phase in degrees.

For a damped sinusoidal source, the sin wave is multiplied by a factor of $\exp(-\text{THETA} \cdot (t - \text{TD}))$.
TD, THETA and PHASE can be omitted if they are zero.

Piecewise Linear Sources:

`Vname N1 N2 PWL(T1 V1 T2 V2 T3 V3 ...)`

This sources a signal which takes value V1 at time T1, value V2 at time T2, and so on. The complete signal is constructed by connecting straight lines between the specified points.

Pulse Sources:

`Vname N1 N2 PULSE(V1 V2 TD Tr Tf PW Period)`

where: V1 = initial voltage
V2 = peak voltage
TD = delay before pulse
Tr = rise time
Tf = fall time
PW = pulse width (time between rise and fall)
Period = period of repeating pulse.

Exponential Sources:

Vname N1 N2 EXP(Vi Vp [Dr Tr Df Tf])

where: Vi = initial voltage
Vp = peak voltage
Dr = delay before rise
Tr = time-constant for rise
Df = delay before fall
Tf = time-constant for fall

Transient and AC Analyses

We have already introduced Operating Point analysis and DC sweeps. The most powerful analyses SPICE can perform are the *Transient analysis*, which allows you to observe the circuit's behavior over an arbitrary range of time, and the *AC analysis* which allows you to compute Phase and Magnitude responses (and Bode plots) for a circuit.

Transient:

.tran step duration

where: step = maximum time-step
duration = stopping time for the simulation.

This analysis can be used with any of the transient sources described above. To print the results of the analysis into the output file, use the .print statement:

.print {AC | DC | TRAN} v(node_name) i(source_name)

The .print statement causes the output file to print the voltage on node "node_name" vs time, and/or the current through an independent voltage source "source_name" vs time. This produces a table of data in the output file similar to this one:

TIME	V(2)
0.000E+00	7.500E-01
1.000E-07	8.250E-01
2.000E-07	9.000E-01
3.000E-07	9.750E-01

AC:

.ac {LIN | OCT | DEC} num start stop

where: lin = use a linear scale for frequencies
oct = step frequencies by octaves
dec = step frequencies by decades
num = number of points per interval
for linear scale: num = points between start and stop
for oct scale: num = points per octave
for dec scale: num = points per decade
start = initial frequency
stop = ending frequency

To perform an AC analysis, you need to have an AC source. This is declared as follows:

Vname N1 N2 AC Amplitude

To print out AC results, the '.print AC' command is used. This results in a table in the output file, such as the following:

FREQ	V(2)
1.000E+00	2.000E+00
1.259E+00	2.000E+00
1.585E+00	2.000E+00
1.995E+00	2.000E+00

Probe

Pspice (along with most other SPICE flavors) has a built-in tool for displaying waveforms called *Probe*. In order to use Probe, you need a simple declaration in your SPICE file:

```
.probe
```

This tells SPICE to record *all internal voltages and currents* in a data file. When the simulation is complete, you can switch to the probe window and choose waveforms to display. The Probe tool also has the ability to display mathematical functions of one or more waveforms. We will explore some of these capabilities in the examples and exercises.

The probe statement may be invoked with other options to modify its behavior. These are some of the more useful examples:

```
.probe v(1) v(2) ... - tells SPICE to store only waveforms at nodes 1, 2, etc. in the data file.  
.probe/csdv         - tells SPICE to output to a CSV file instead of a probe file.
```

“CSV” stands for “Comma-Separated Values,” a generic text format which can be easily imported by other programs. This is a good way to transfer data from SPICE simulations to Matlab.

Examples

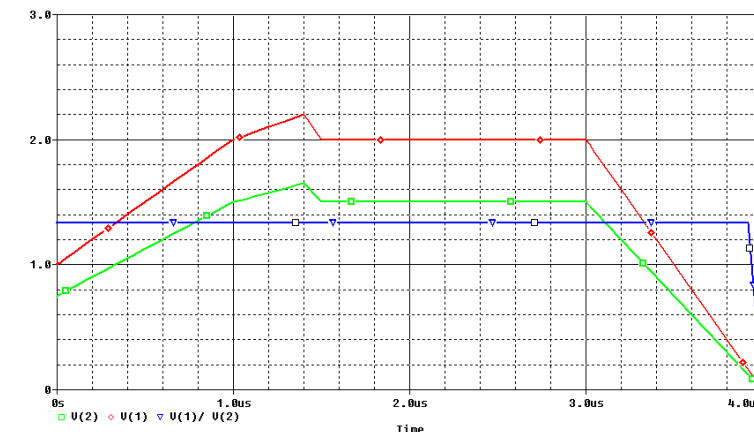
Example 1: Piecewise Linear Transient Simulation

```
Piecewise Linear Example  
.probe  
V1 1 0 PWL(0us 1v 1us 2v 1.4us 2.2v 1.5us 2v 3us 2v 4us 0v)  
R1 1 2 1k  
R2 2 0 3k  
.tran .1us 4us  
.print TRAN v(2)  
.end
```

To run this example, type the statements into a text file called “pwl.cir”. Run Pspice AD, and select File->”Open Simulation.” When the File Browser appears, navigate to your file's directory. Make sure you select the “.cir” file type. Open your file and press the “Run Simulation” button (it looks like a VCR's “play” button).

After the simulation finishes, the Probe display window appears with no traces displayed. To display a trace, select Trace->”Add Trace”. A menu of available waveforms appears. Select V(1) by clicking on it. This is the waveform at node 1, which is the output of the voltage source V1. Then select V(2) to choose the voltage at node 2. Press OK.

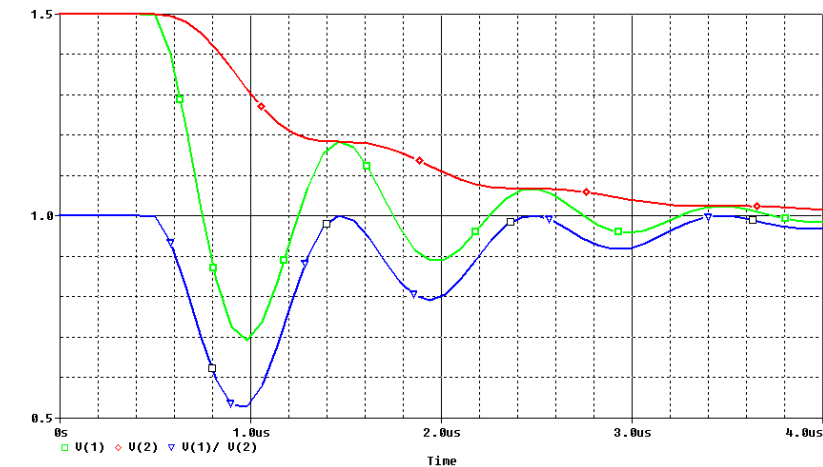
Now the input piecewise linear waveform is displayed along with the voltage in the resistor divider. These should be related to each other by a constant ratio. To confirm this, we can display the ratio V(1)/V(2). Go to the “Add Trace” menu and click on V(1). Then press “/”, and click on V(2). Notice that the expression “V(1)/V(2)” appears in the “Trace Expression” field at the bottom of the window. You can also type expressions directly into this field. Press okay, and you should see a display like this one:



Example 2: Sinusoidal Transient Simulation

```
Sinusoidal waveform Example
.probe
V1 1 0 SIN(1v 0.5v 1Meg 0.5us 1e6 90)
R1 1 2 1k
C1 2 0 1nF
.tran .1us 4us
.print TRAN v(2)
.end
```

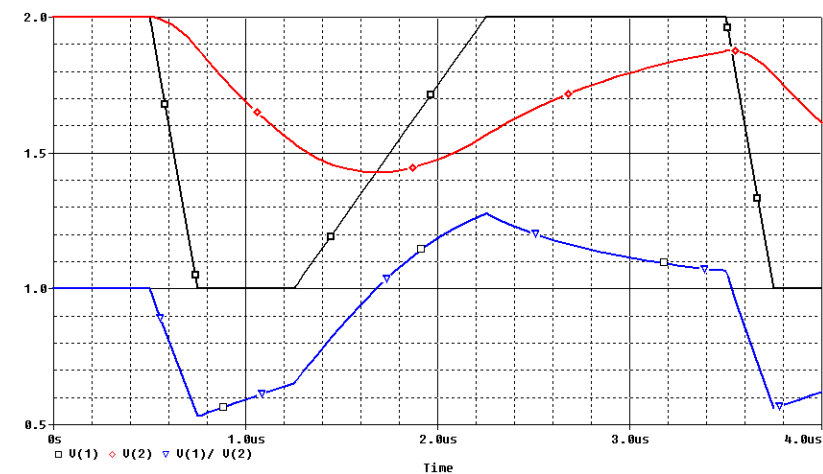
This circuit is a little bit different from the last one, in that this one represents an RC filter rather than a resistive voltage divider. Repeat the procedures used in the Piecewise Linear example, and display the waveforms for V(1), V(2) and V(1)/V(2). You should get a result that looks like this one:



Example 3: Pulse Transient Simulation

```
Pulse waveform example
.probe
V1 1 0 PULSE(2v 1v 0.5us 0.25us 1us 0.5us 3us)
R1 1 2 1k
C1 2 0 1nF
.tran .1us 4us
.print TRAN v(2)
.end
```

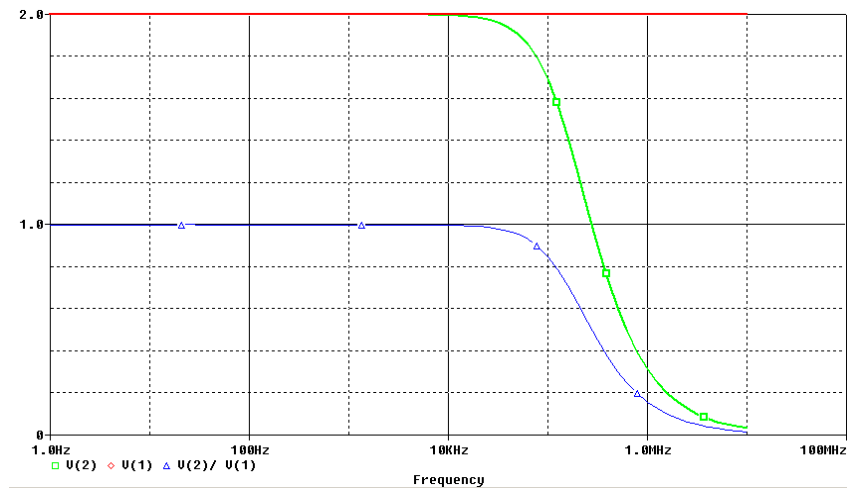
Simulate this example as before, selecting the same waveforms. You should get a result resembling this one:




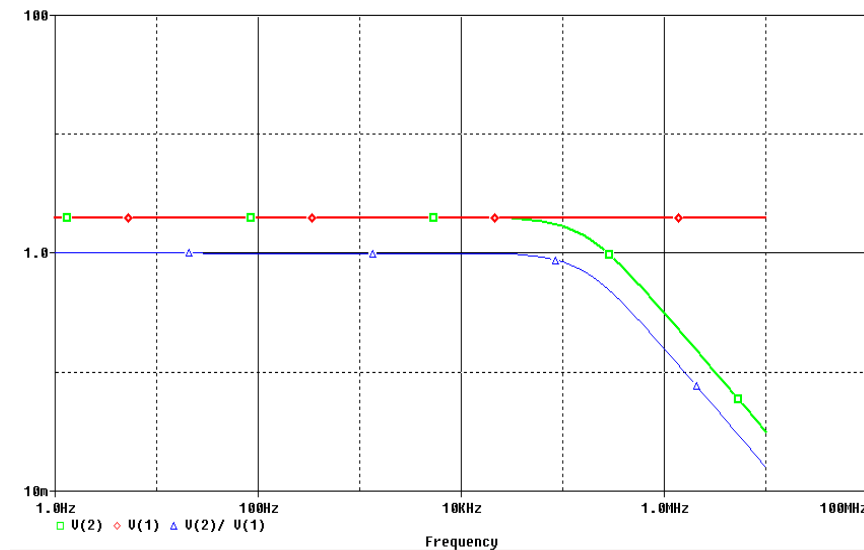
Example 4: AC Simulation

```
AC Analysis Example
.probe
V1 1 0 AC 2V
R1 1 2 1k
C1 2 0 1nF
.ac dec 10 1 10Meg
.print ac v(2)
.end
```

Simulate the above circuit. Display the waveforms for V(1), V(2) and V(2)/V(1). You should get a result that looks like this:

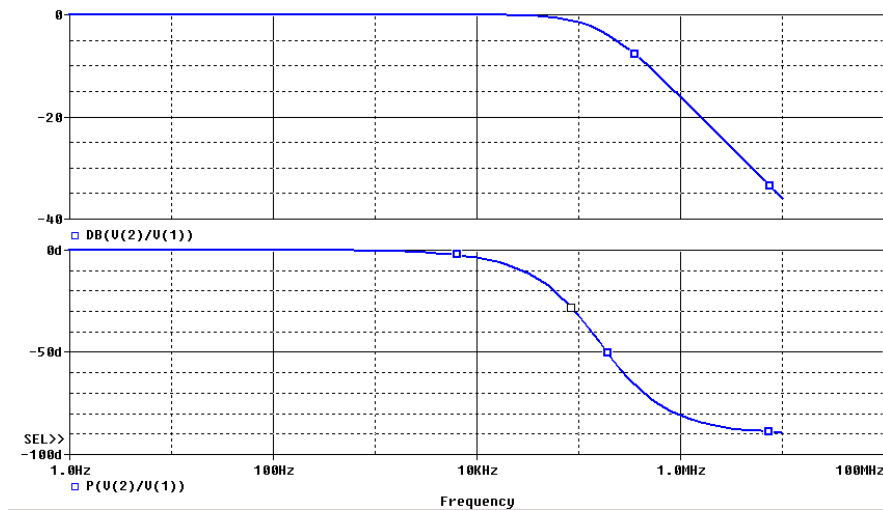


There is an option in the toolbar called “log Y axis”, which looks like this: . Press this button to convert the Y-axis display to a log scale. The results should look like this:



The traces are starting to look more like a Bode plot. To make a true Bode plot, we need two traces: the magnitude (in dB) and the phase (in degrees). Under the Trace menu, select “Delete All Traces.” Now go to the Add Trace menu. On the right is a list of mathematical expressions. One of these is called DB(). Click on this expression. In the “Trace Expression” field, complete the expression so that it reads “DB(V(2)/V(1))”. Click OK. You should see a magnitude plot for a low-pass filter.

Next we want to add the phase in a separate subplot. Under the “Plot” menu, select “Add Plot to Window”. This creates a new, blank subplot in the Probe window. Make sure the new subplot is selected, and go to the Add Trace menu. In the Trace Expression field, type “P(V(2)/V(1))”. Click OK. The resulting graph should look like a Bode plot, like this one:



Exercises:

Simulate all the examples in this handout. For each SPICE file, sketch the circuit that corresponds to the SPICE description. On your circuit diagrams, indicate the nodes and voltages which are plotted in the simulations. Turn in printouts of the output files (or portions of the output files proving that you did the simulations), and printouts of the traces obtained in the simulations.