Concise SPICE

Ihsan Djomehri, Spring 1999

1 Introduction to Circuit Simulation

Developed at Berkeley in the 70s, SPICE has evolved into the tool of choice for circuit level simulation. This handout seeks to provide a concise, fairly comprehensive introduction to HSPICE (version 98.2 running on MIT's Athena network) and the graphing utility AvanWaves. I credit the authors of previous manuals for their information. In essence, SPICE reads in a list (called the "SPICE deck") of circuit nodes and the elements between them, generates a series of nodal equations, and solves for the voltages. The deck also contains statements identifying which mode(s) the circuit should solve for: DC, AC, or transient analysis. Words of caution: many hacks have been introduced into SPICE over the years; always check the output to make sure it is intuitively reasonable!

2 Creating a SPICE Deck

To start on Athena, make a new directory for your SPICE stuff, then type "add hspice" to add the HSPICE locker. Begin by sketching out your circuit on paper, labeling each node with a number, naming each circuit element and assigning properties to them. Note that node 0 is always ground and there can't be any floating nodes. Use *emacs* to edit any text file like *test.sp* that will become our SPICE deck. An * in front of any line causes it to be interpreted as a comment. Later on, we will be using the circuit schematic in Figure 1 as a demonstration.

The general format for entering in a circuit element is a one-letter identifier merged with a name, space, the node numbers that connect it, space, the properties of the element. Numerical property values are always in SI units (meters, seconds, volts, etc…) but can be modified by a suffix letter that acts as a multiplier:

Suffix:	t	g	x	k	m	u	n	p	f
Multiplier:	1e12	1e9	1e6	1e3	1e-3	1e-6	1e-9	1e-12	1e-15

Thus, for various circuit elements we write (you modify italics, <>s are optional):

Voltage:	V*name N+ N- <qualifier> value*
Current:	I*name N+ N- <qualifier> value*
Resistor:	R*name N+ N- value*
Capacitor:	C*name N+ N- value*
Inductor:	L*name N+ N- value*
Diode:	D*name N+ N- model_name*
BJT:	Q*name Nc Nb Ne model_name*

MOSFET: M*name Nd Ng Ns Nb model_name* L=*value* W=*value*
<+ AD=*value* AS=*value* PD=*value* PS=*value*>

where *N+*, *N-* are node numbers from the positive to negative node; *Nc*, *Nb*, *Ne* are the collector, base, and emitter node numbers of a bipolar transistor; *Nd*, *Ng*, *Ns*, *Nb* are the drain, gate, source, and body node numbers of a MOSFET; *model_name* refers to a user specified element model; L, W are the length and width of a MOSFET; AD, AS are the areas of the drain and source to body interface, while PD, PS are their perimeters.

Typically, start by defining all your voltage and current sources, and then introduce your passive elements and transistors. The *qualifer* input for the sources is DC, AC, or *Transient_Function* for the respective analyses, where the *Function* can be:

Square Wave: PULSE(*v1 v2 td tr tf pw per*)
Sine Wave: SIN(*vo va td* <*theta*>)

where *v1*, *v2* are the low and high voltages, *td* is the time delay before starting, *tr*, *tf* are the pulse rise and fall times, *pw* is the pulse width (time being at *v2*), *per* is the period; also, *vo* is the offset voltage, *va* is the sinusoidal amplitude.

After creating the network list, one needs to create models for various circuit elements that exhibit non-linearities, etc… The general format for the statements is:

Diode: .model *model_name* D <is=*value*> *etc*…
BJT: .model *model_name type* <bf=*value*> *etc*…
MOSFET: .model *model_name type* level=*lnum* <vto=*value*> <kp=*value*> *etc*…

where *type* is NPN or PNP for BJTs, and NMOS or PMOS for MOSFETs; *lnum* specifies the SPICE level number (model complexity) being used. The various parameters that go into a model will be given or estimated from device physics. Also, you can specify the ambient temperature in Centigrade with the command line:

.temp *temperature*

3 Performing Circuit Analysis

To complete the SPICE deck, one must add statements that determine the analysis to be undertaken. Always first put in the line:

.options post

which will display relevant facts about the circuit and output a *test.ic* file that contains the bias points of all nodes. Next, include lines from the next three paragraphs as appropriate. Finally, the deck must always conclude with a ".end" statement. To actually run the simulation, use “hspice *test.sp* > *test.out*” which dumps the results into *test.out* (view it).

DC Analysis: the following statements are useful when dealing with DC scenarios:

.nodeset V(*node#*)=*value*
.tf V/I(*node#out*) V/I*namein*
.dc V/I*name1 start1 stop1 step1* <V/I*name2 start2 stop2 step2*>

where you choose either *V* or *I* (for voltage or current) when V/I is indicated; *namein* is the name of the input source. The “.nodeset” command is used to set the “initial guess” voltage to a node in situations where SPICE has trouble converging to a solution. The “.tf” command performs a small signal analysis at the DC bias points given the output node and input source: the *test.out* file will give the gain, input resistance, and output resistance of the system. The “.dc” command does a DC sweep for plotting given the first source’s parameters, and an optional nested sweep of the second source; a file *test.sw0* is generated upon completing the simulation.

AC Analysis: the following statements are useful for frequency domain analysis:

.ac DEC *points_per_interval fstart fstop*
.pz V(*node#out*) V*namein*

The “.ac” command will produce Bode plot information of the circuit for the designated AC bias points in the given frequency range, taking data points per DECade (logarithm base 10); a file *test.ac0* is generated. The “.pz” command evaluates the poles and zeros of the transfer function given the output and input: the results are spit out in *test.out*.

Transient Analysis: the following statements are useful for time domain studies:

.ic V(*node#*)=*value*
.tran *tstep tstop*
.measure tran *tname* trig V(*node#*) val=*value* rise=1
\+ targ V(*node#*) val=*value* fall=1
.measure tran *pname* avg p(*element_name*) from=*tstart* to=*tstop*

The “.ic” command sets optional initial conditions for transient analysis at the specified nodes. The “.tran” command performs a time dependent analysis starting from time=0s for plotting; a file *test.tr0* is generated. The “.measure” command is used to output various user-named quantities to a file *test.mt0*: e.g., you can find the time between the trigger of a rising edge and the target of a falling edge; or you can find the power dissipation in an element.

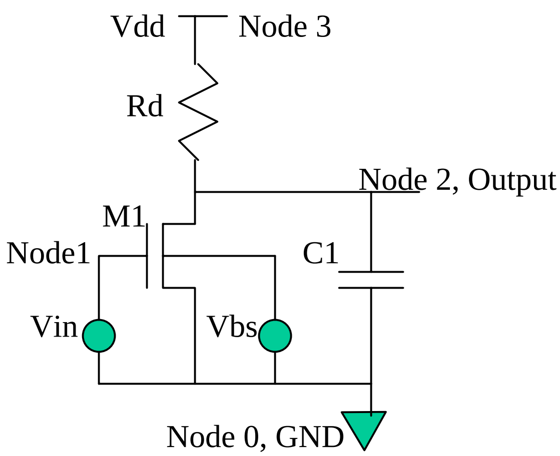
4 Graphical User Interface

In order to generate the waveforms from the aforementioned analyses, run the graphics program by typing “awaves”. There are many options to play with; the main ones are described below. To plot, open your HSPICE design and the Results Browser. Choose whichever analysis you wish to view. Make sure the X-Axis is appropriate (usually input Voltage for DC, Frequency for AC, and Time for Transient). Click on which type of data should be plotted on the Y-Axis. Right click on axes to change to log scale. Choose the node number for the type of data to be plotted. Enjoy!

Design	Open:	Open a HSPICE simulation
Panels	Add:	Add a graphing panel
	Delete:	Delete a panel or selected curve
Window	Zoom In/Out:	Zoom in/out on the data
Measure	Point:	Click on point on a curve to find its values
	PointToPoint:	Click and hold to measure between points
	Delete:	Delete a measurement
Tools	Results Browser:	Open the Results Browser
	Expressions:	Generate user defined expressions to graph
	Print:	Print out waveforms

5 Examples

Figure 1:



DC Analysis:

The schematic in Figure 1 can be constructed with the following deck (explanatory comments are in italics above each line):

The first line is a “commented-out” header
* SPICE test.sp

define a voltage source named “dd” with positive terminal at node 3, negative terminal at ground node 0, and providing 5 volts
Vdd 3 0 5
define a voltage source to be used for the backgate effect
Vbs 4 0 0

define an input source with 1.5 V for DC analysis
Vin 1 0 DC 1.5

define a resistor of 5000 ohms
Rd 3 2 5k
define a capacitor of 100 femtofarads hanging from the output
C1 2 0 100f

define a MOSFET named "1" from drain node 2 to gate node 1 to source node 0 to body node 4, with the model named NMOS and 2 micron gate length, width of 40 microns
M1 2 1 0 4 NMOS L=2u w=40u

create a SPICE level 1 model for an NMOS transistor named NMOS, with the specified characteristics
.model NMOS NMOS level=1 vto=.75 kp=8e-5 gamma=.5 phi=.4 lambda=.05

set the ambient temperature to 27 Centigrade
.temp 27

Now we need to add analysis commands to the deck:

this generates lots of useful data on the circuit, such as bias points
.options post

find the voltage gain (about –6), input and output resistance, where node 2 is the output and Vin is the small signal input source; we're using the DC bias value
.tf V(2) Vin
do a sweep of the source Vbs from 0 to -0.4 V in -0.2 V steps, while sweeping the source Vin from 0 to 5 V in .1 V steps, and save data for graphing
.dc Vin 0 5 .1 Vbs 0 -.4 -.2

essential command to end SPICE deck; must be followed by a carriage return
.end

AC Analysis:

* SPICE test.sp

Vdd 3 0 5
Vbs 4 0 0
define an input source with 1.5 V for AC analysis
Vin 1 0 AC 1.5

Rd 3 2 5k
C1 2 0 100f

M1 2 1 0 4 NMOS L=2u w=40u

.model NMOS NMOS level=1 vto=.75 kp=8e-5 gamma=.5 phi=.4 lambda=.05

.temp 27

.options post

do a sweep of AC input frequency from 1 MHz to 10 GHz on the source input; using the AC bias value
.ac DEC 10 1x 10g
find the poles (there's one at about 2e9 rad/s, approximately 1/RC) and zeros of the transfer function from node 2 as output and Vin as input
.pz V(2) Vin

.end

Transient Analysis:

* SPICE test.sp

Vdd 3 0 5
Vbs 4 0 0
the input will be a square wave from 0 to 5 V starting after 5 ns delay, with .1 ns rise and fall times, a pulse width of 10 ns, and a 20 ns period
Vin 1 0 PULSE(0 5 5n .1n .1n 10n 20n)

Rd 3 2 5k
C1 2 0 100f

M1 2 1 0 4 NMOS L=2u w=40u

.model NMOS NMOS level=1 vto=.75 kp=8e-5 gamma=.5 phi=.4 lambda=.05

.temp 27

.options post

solve for the voltages at circuit nodes as a function of time (with the given input) from 0 to 25 ns with .05 ns steps
.tran .05n 25n
measure the average transient total circuit power dissipation from 5 ns to 25 ns
.measure tran pdiss avg power from=5n to=25n

.end

Figure 2 displays the results of analyses done on the demonstration circuit. The part (a) plot shows the DC sweep characteristics (Vout versus Vin) for various Vbs biases; it acts as a RTL (resistor-transistor logic) inverter. The part (b) plot gives the output magnitude versus frequency from AC analysis. The part (c) plot show the transient characteristics of the output for the given square wave input; it acts as an inverter, but when the output returns to Vdd, it must charge the capacitor, so we see a solution decay time of about RC ~ 0.5 ns.