Hspice Netlists Common Problems & Suggestions

Common Problems from Lab 3, Part I

 Algebraic equations must always be enclosed by single quotes. Correct: L='2*lambda' Incorrect: L=2*lambda Hspice will read this incorrect version as a transistor length of 2 meters.

- 2. Macro statements (defined between .macro and .eom) should only include the components included in the desired circuit block. For instance, an inverter has two transistors so an inverter macro should not contain anything other than the two transistors.
- 3. Incorrect perimeter and area calculations can lead to problems in the transient simulations. A common mistake was to define a single set of AD, AS, PD, PS parameters that apply to all transistors. This makes it impossible to have different sized transistors within the same netlist. See the suggested transistor macros below for a better way to handle this.
- 4. Circuit nodes should be named logically. Nodes 1 9 give no indication of what part of the circuit it is or where things connect. For instance, clk is a very logical name for the clock signal and vdd is common for the power supply voltage. Typically, an inverted signal is indicated by an n (clock_bar is described as clkn). Before even attempting to write a circuit netlist, draw the complete schematic and label all nodes. This will save tremendous amounts of time in debugging. In addition, the TAs will expect to see your labeled schematic with any question regarding your netlist.
- 5. Be sure to understand what the simulation you are running is supposed to accomplish. For instance, a sweep of the inverter input voltage to generate the VTC is easily accomplished with a DC analysis. DC analysis statements allow you to specify a voltage source and sweep ranges. Simulations with respect to time, however, require a transient simulation. The transient simulation defines an end time and time step size.
- 6. It is very important that you understand what the circuit is supposed to do before running a simulation. Circuit simulators are extremely useful for checking and fine-tuning circuits. They are NOT useful if you do not understand the circuit. There are many ways a circuit simulator can be wrong and without an idea of the correct behavior, you will not know if the output is valid. Asking a TA if something is working will get the response, "What do you expect it to do?" Think about it first and ask for help if you can't explain a discrepancy between your circuit behavior and what you think it should do.

Suggested Transistor Macros

.macro pfet d g s Wi='5*lambda' Le='2*lambda' Mp d g s vdd pmos W=Wi L=Le AD='5.5*lambda*Wi' PD='11*lambda+2*Wi' + AS='5.5*lambda*Wi' PS='11*lambda+2*Wi' .eom

.macro nfet d g s Wi='5*lambda' Le='2*lambda' Mp d g s gnd nmos W=Wi L=Le AD='5.5*lambda*Wi' PD='11*lambda+2*Wi' + AS='5.5*lambda*Wi' PS='11*lambda+2*Wi' .eom

These two macros define single pmos and nmos transistors. There are several advantages to using a macro for the transistors. Bulk connections can be handled automatically such that pmos bulk is always connected to vdd and nmos bulk is always connected to gnd. The only connections made in each transistor instance are the drain, gate, and source. By default, the transistors are defined to be minimum size (W / L = 5*lambda / 2*lambda). When a different size is desired, simply include Wi and/or Le on the instance lines. An example of an inverter using these macros is shown below.

XMp0 out	in	vdd	pfet	Wi='15*lambda'
Xmn0 out	in	gnd	nfet	

Notice that in this example the nmos is minimum size and the pmos is 3 x minimum width. Both are minimum length devices. Finally, the macros calculate the area and perimeter of the source and drain regions for every transistor automatically.