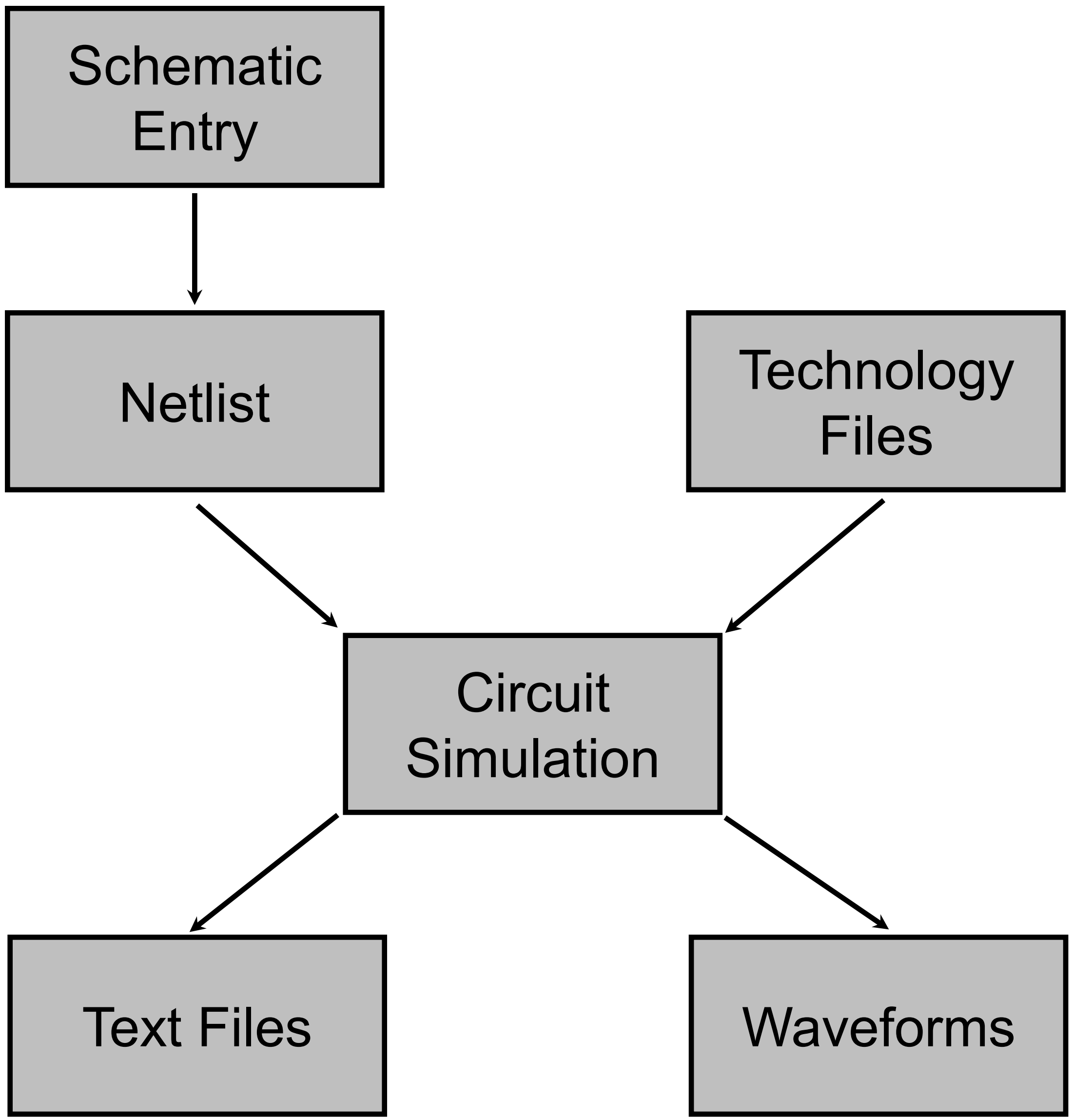
EECE 488: Short HSPICE Tutorial

Last updated by: Mohammad Beikahmadi
January 2013

SPICE?

- Simulation Program with Integrated Circuit Emphasis
- An open source analog circuit simulator
- Predicts circuit behavior, checks signal integrity, and optimizes circuit performance
- Why do we need it?
    - For IC, it’s impossible to breadboard before manufacturing
    - Extremely high fabrication cost
    - Complex mathematic equations

Circuit Simulation Process



Types of Simulation

- **DC simulation**: steady state, constant source, operating point, sweep
- **AC simulation**: frequency response, small signal analysis, noise
- **Transient simulation**: observe circuit behavior with respect to time

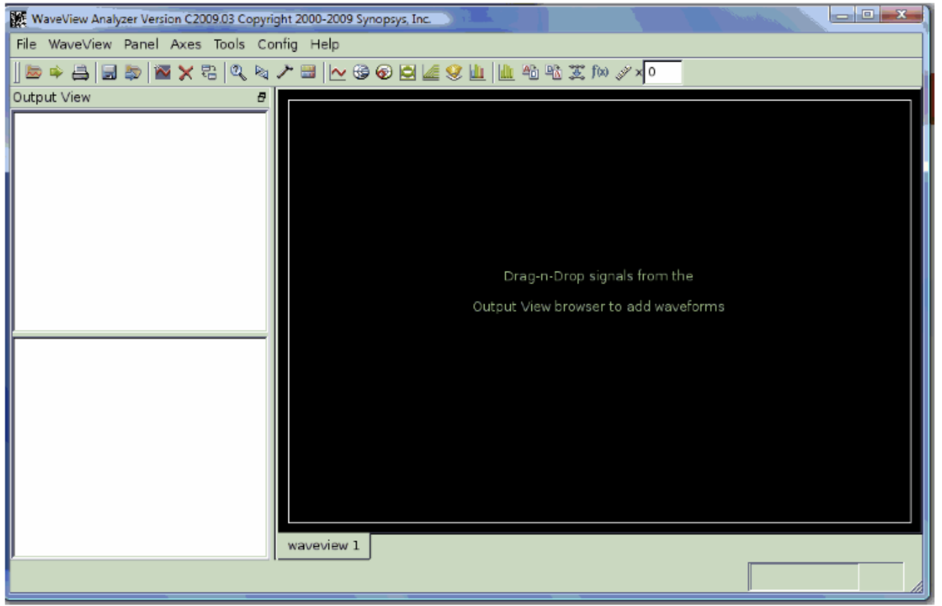
Setting it up (some stuff taken from EECE 481 tutorial made by Nima and Dipanjan)

- Let’s do the following:
  - Open a terminal
  - Create a directory eece488: **mkdir eece488**
  - Source the following file to set up HSPICE:
    - **source /CMC/scripts/setenv.synopsys.hspice.2008.09.csh**
  - Now we can use HSPICE!

Setting it up

- In order to check simulation results, we shall use **WaveView Analyzer.**
  - Type **wv** & and you should be able to see the following window:
  - If you need tutorials on how to use WaveView:

  From menu bar -> Help -> Wave View Analyzer Help



Setting it up

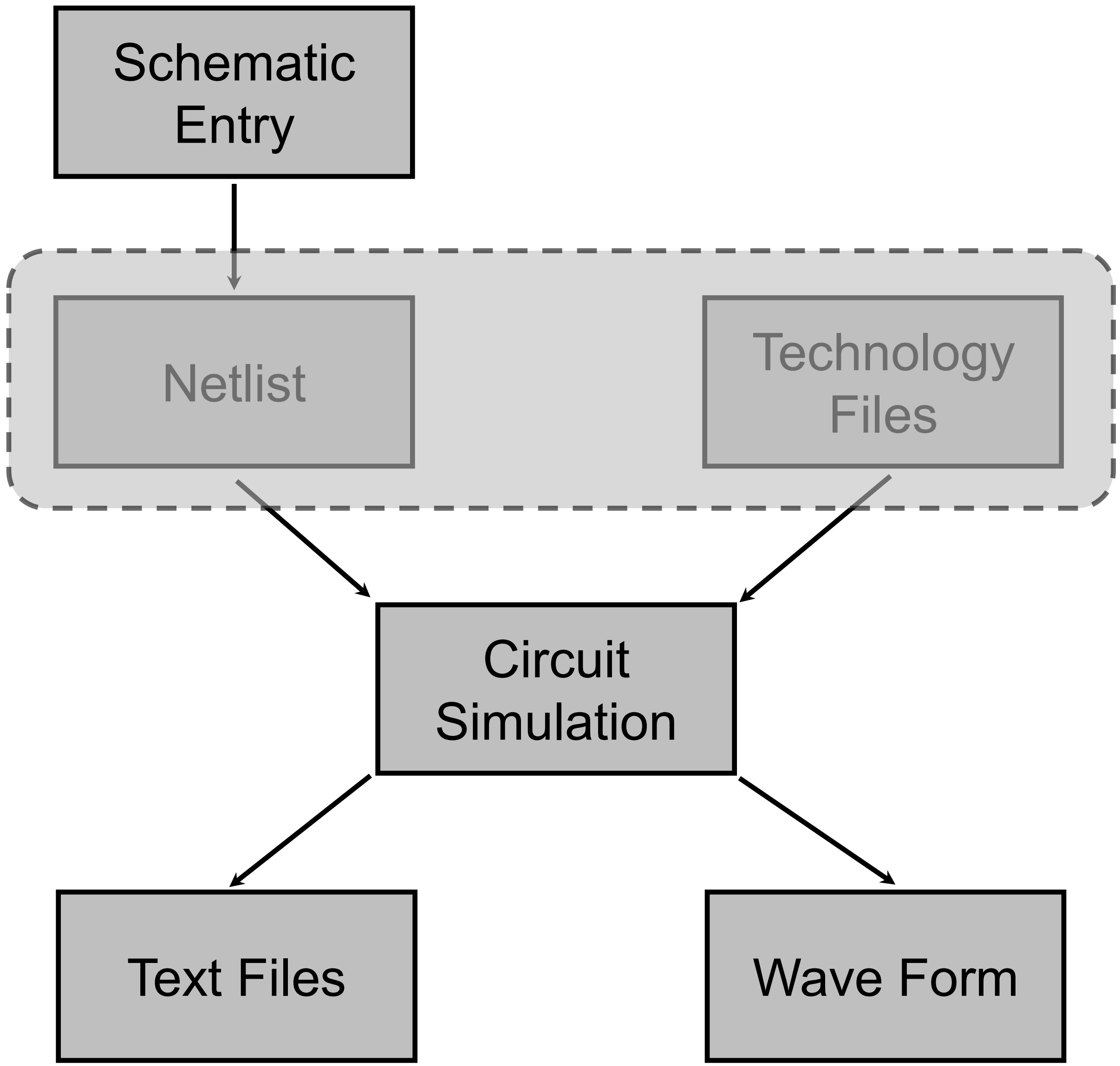
- HSPICE Manuals

Name of Manual	Description
HSPICE User Guide: Simulation and Analysis	Gives you the basics. How HSPICE works, the file types, how to setup a circuit, how to run transient, DC and AC analyses.
HSPICE Signal Integrity User Guide	Describes how to use HSPICE to maintain signal integrity in your chip design.
HSPICE Reference Manual: Elements and Device Models	Describes the mathematical models used for passive devices, diodes, JFET's, MESFET's and BJT's.
HSPICE Reference Manual: MOSFET Models	Describes the mathematical models used for the various MOSFET models.

- For this course, you should only need the **simulation and analysis** guide!

Type: **acroread /ubc/ece/data/cmc/experimental/tools/synopsys/hspice_vB-2008.09-SP1/hspice/docs_help/hspice_sa.pdf**

Use HSPICE - 1st, netlist!



Use HSPICE - 1st, netlist!

- Well, actually, before writing the code (netlist), you need to have a circuit in mind!

- HSPICE does not provide schematic entry...

- Draw the circuit first

- Open a text file...let’s use **gedit**, so type command **gedit &**

- From the schematic, we construct the circuit in HSPICE code in the text file -> Netlist

Use HSPICE - 1st, netlist!

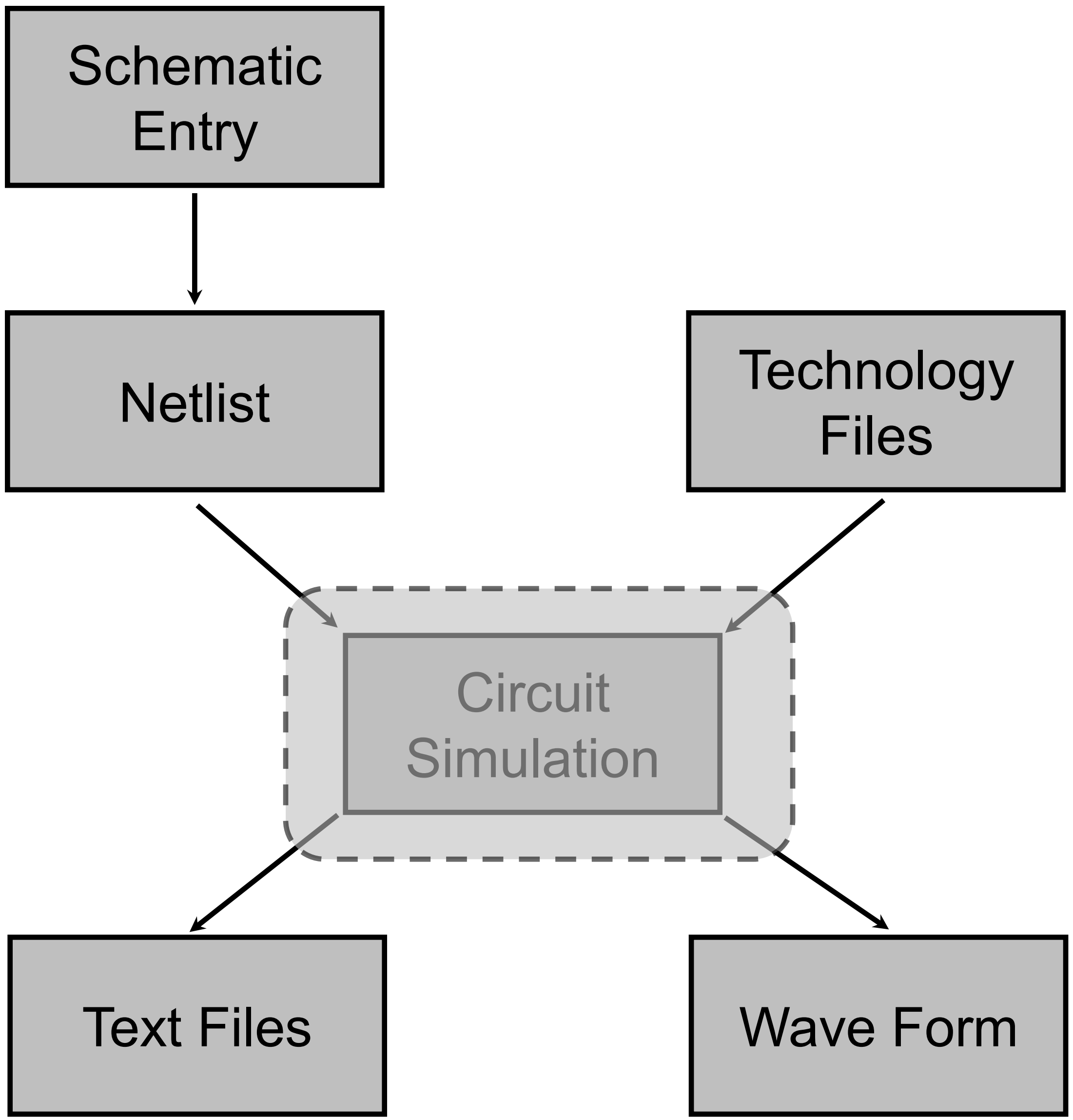
- For a normal netlist (file ends with extension .sp):
  - Title
  - Model library
  - Circuit structure
  - Stimuli
  - Type of simulation
  - .option post
  - .end

Use HSPICE - 1st, netlist!

- Super simple HSPICE netlist example:
  - One constant voltage source in series with 2 resistors of same value
    - General component format: Name Positive_node Negative_node Value
  - Instruct HSPICE that it will be a DC sweep simulation
    - Syntax: .dc <source> <start> <stop> <step>
  - Finish the netlist and name it as **simple_dc.sp**
  - Ready to simulate!

```
*simple dc circuit
R1 X Y 1k
R2 Y 0 1k
V1 X 0 1
.DC V1 0 1 0.1
.option post
.end
```

Use HSPICE - 2nd, run HSPICE to simulate!



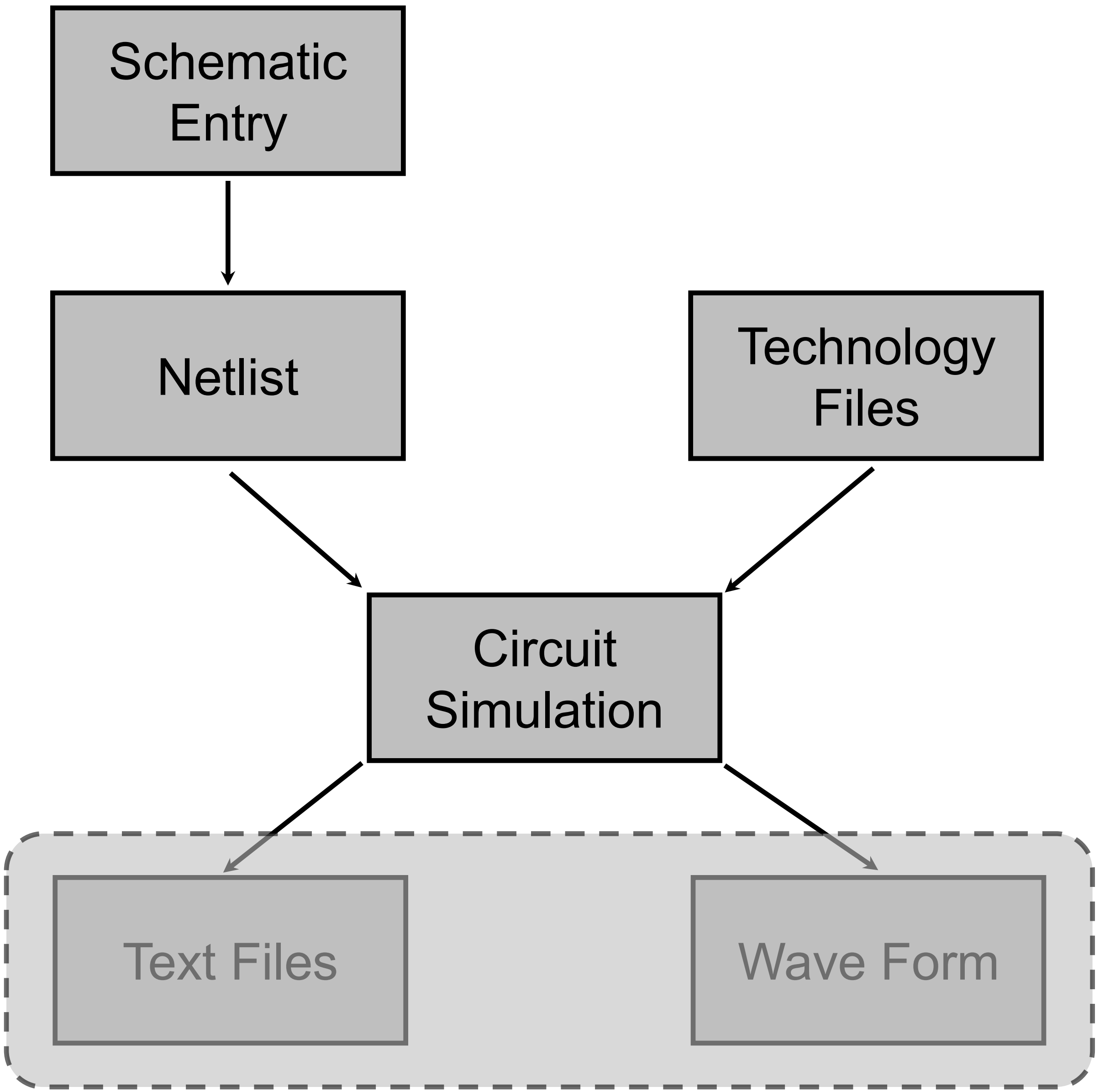
Use HSPICE - 2nd, run HSPICE to simulate!

- Command to run HSPICE:
  - **hspice simple_dc.sp >! temp.lis**
  - **hspice** calls the program
  - **simple_dc.sp** is the name of netlist, extension is required
  - **>** tells HSPICE to output the results in the file following the symbol
  - **!** tells HSPICE to replace the file if file of same name exists
  - **temp.lis** is the output file, you can change the name if you want

Use HSPICE - 2nd, run HSPICE to simulate!

- After you run HSPICE:
  - If you get the message saying hspice job concluded...it means that it has been compiled successfully without syntax error
  - If you get the message saying job aborted, then you have to go to the output file (.lis) to figure out your syntax error

Use HSPICE - 3rd, check results!



Use HSPICE - 3rd, check results!

- Now we can either check results through text file (.lis) or Wave View Analyzer
- Let’s check our previous simple example using wv
    - Type **wv &**
- The analyzer has two sub-windows, the black one will show all of the important graphical information
- Check online documentation to get familiar with the program.

Use HSPICE - 3rd, check results!

- In the Wave View Analyzer window:
    - Click on File on top left corner
    - Click on “Import Waveform File”
    - Choose simple_dc.sw0 which is a waveform file produced after we run HSPICE
    - simple_dc.sw0 should show up on the Output View Window
    - Double click on simple_dc.sw0
    - A sub branch “top level view” should show and let’s click it
    - Now you should be able to plot the things that you want to see, i.e. node voltages, etc

Another DC Simulation Example (with transistor)

- Plot I_D vs. V_{GS} for an NMOS device with W/L=5um/1um belonging to a 0.35um technology
  - We need to call the model library by including the following in the netlist:
    - .lib '/ubc/ece/home/courses/eece488/hspice/cmosp35/mm0355v.l' TT
  - Syntax for transistors:
    - Name Drain Gate Source Substrate Model_name Length Width
    - **M**1 Out In Gnd Gnd NCH l = 1um w = 5um

Another DC Simulation Example (with transistor)

- Draw the circuit and name all the nodes
- Generate netlist: nmos_sweep.sp
- Sweep the gate voltage
- Plot drain currents
- Look familiar?

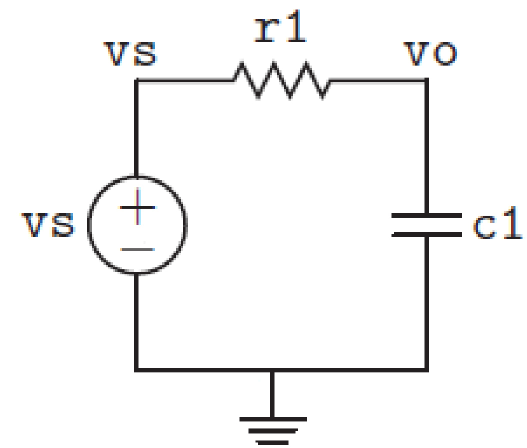
Transient Simulation 1

- Dealing with time domain
- We want to look at the circuit in a time window
    - Syntax: .tran <t step> <t stop>
- Use of sinusoidal or pulse stimuli
- To construct a voltage pulse source:
    - PULSE V_lower V_higher delay rise_time fall_time pulse_width period
    - PULSE 0 5 0.01ps 0.01ps 0.01ps 2ns 4ns

Transient Simulation 1

- Simple Low pass RC circuit as an example
  - R1=1K Ohms, C1=1 uF
  - Vs: Periodic pulse

    (pulse width= 5ms, period=10 ms, pulse height: 1 V)
  - Netlist name: quick_tran.sp
- Showing the charging and discharging of capacitor



Transient simulation 2

- Let’s take a look at a CMOS inverter
- Draw the circuit
- Enter the netlist (inverter.sp)
- Simulate!

AC Analysis Example

- When we do AC analysis, we are generally interested in the following
  - Gain/Magnitude measured in dB of power (20*log(#) -> dB)
  - Phase (degrees or radians)
- Sweep frequency -> Frequency response
- To do this in HSPICE
  - .AC DEC 5 1 100MEG
    (DEC = decade, 5 is # of points per decade, then range of frequency of interest)
  - To plot gain: .PRINT AC VDB(OUT, IN)

AC Analysis Example

- We shall do an ac analysis on a simple common source amplifier
- Common source means source is connected to the ground
- Draw the circuit, and label the nodes
- Enter the netlist (ac_analysis.sp)
- Simulate!

Subcircuits

- Use subcircuits to define a collection of elements and to create reusable circuits
- Syntax:

```
.SUBCKT SUBNAME Node1 Node2 Node3 ...

….Element statements….

.ENDS SUBNAME
```

- Using a subcircuit:

```
Xname Node1 Node2 Node3 ... SUBNAME
```

Subcircuit Example

- Define an inverter as a subcircuit and then use it to create and simulate a buffer (buffer.sp)

- Write a netlist for an inverter and then use .include statement to import the netlist in order to create the buffer’s netlist (buffer2.sp)

Some Tips

- How to connect to school Linux machines
  - You will need SSH secure shell client
  - **http://help.ece.ubc.ca/How_To_Tunnel_Connections_Through_SSH**
  - An enhanced terminal for Windows: http://mobaxterm.mobatek.net/
- Open a terminal, and type ssh -X your_ece_id@ssh.ece.ubc.ca
- Enter password
- Boom you are connected!

Questions??!?!!

- No? ok have a nice day.