VLSI II - Spring 2004
EE 382M (*14975*)

HSPICE TUTORIAL

A} *Tutorials on the Class Website*

1. ***General information on Hspice*** –Describes how to access hspice on the ECE LRC machines and useful commands to read up from the hspice manual.

2. ***Digital circuit simulation using Hspice*** - Recommended starting tutorial. Introduces through examples most of the commands required for the HW's.

3. ***Getting started with Hspice-A tutorial*** – Gives an example hspice netlist. Suggested reading in addition to the second tutorial.

4. ***hspice_2000_2.pdf*** **–** Detailed Star-Hspice Manual

B} ***Setting up Account***

Location:

Hspice version: **2001.2**
Servers: **sunfire.ece.utexas.edu (sunfire1/sunfire2)**
Access terminals: **Sunray or Sunblade terminals in ENS 507**

Setting up Account:

*For **ksh**, add the following lines to the end of your .profile*

PATH=/usr/local/packages/hspice/2001.2/bin:$PATH
LM_LICENSE_FILE="/usr/local/packages/hspice/2001.2/license.dat"

*For **csh**, add the lines below to the end of your .cshrc*
set path (/usr/local/packages/hspice/2001.2/bin $path)
setenv LM_LICENSE_FILE /usr/local/packages/hspice/2001.2/license.dat

C} ***About Hspice***

- Hspice is a powerful circuit simulator for simulation of electrical circuits in steady-state, transient and frequency domains.
- We will use Avant!'s Star-Hspice and Awaves (for plotting waveforms)
- It accurately simulates DC to microwave frequencies greater than 100 GHz.

D} Description of Hspice Files and Commands:

The main hspice file (input file) is a file with the .sp extension.

General Structure of the input file (*.sp)

Number	Section	Description
1	Title	Required first line of input file
2	Setup	To set conditions of simulation, initial values etc
3	Library/Files	Attaching Library, subcircuit and other data files
4	Netlist	Netlist of circuit being simulated
5	Sources	Input sources to the circuit
6	Analysis	Statements specifying type and conditions of analysis
7	Output	Specifying output variables and measure statements
8	Alter blocks	Changing analysis points, libraries etc
9	End	Required last line of input file

Note:

Hspice is ***NOT*** case sensitive

Sample Hspice netlist

```
Homework 2 Problem 2  $ REQUIRED FIRST LINE (TEXT CAN VARY)
*EE 382M-VLSI II, Fall 2002

* COMMENTS: 1) * TO COMMENT THE ENTIRE LINE
            2) $ TO COMMENT THE PORTION BEYOND $


.options CONVERGE=1 GMINDC=1.0000E-12 accurate probe list
node $ post

*  .options – MECHANISM FOR CONTROLLING THE SIMULATION
              OUTPUTS AND TOLERANCE LEVELS

* Use these lines for 0.13u
.include "mos013.txt"  $ transistor models
*  .include – USED TO INCLUDE TEXT FROM ANOTHER FILE

.param Vdd=1.5V     $ Vdd value
*  .param – TO DEFINE A VARIABLE AND ITS VALUE

.param L=0.13u     $ transistor length

.global vdd
*  .global – VARIABLE IS ASSIGNED SAME VALUE IN MAIN AND
             SUBCIRCUITS

.global L

.include "library-013.txt" $ the library of circuit elements
.include "block1-013.txt"  $ the circuit for the 1st pipeline stage
.include "block2-013.txt"  $ the circuit for the 2nd pipeline stage

*--------------------NETLIST------------------------------*
x1 a0_in a0 clk flop      $ input flipflop 1
x2 b0_in b0 clk flop      $ input flipflop 2
x3 a0 b0 g17_00_3r_in block1      $ first cycle logic
x4 g17_00_3r_in g17_00_3r clk flop      $ intermediate
stage flipflop
x5 g17_00_3r pre_xnor_in block2      $ second cycle logic
x6 pre_xnor_in pre_xnor clk flop     $ output flipflop
mload 0 pre_xnor 0 0 nmos W=14u L=L     $ output load

*--------------------NETLIST END--------------------------*

Vdd Vdd 0 Vdd
```

```
* you will need to modify the following lines to apply an
input stimulus
* and clock to the circuit

*--------------------STIMULUS-----------------------------*
Va0_in a0_in 0 DC 0
Vb0_in b0_in 0 DC 0
Vclk clk 0 DC 0
*--------------------STIMULUS END-------------------------*

.END $ REQUIRED LAST STATEMENT IN ALL HSPICE FILES
```

Explaining Parts of the input file:

1) **TITLE Statement and COMMENTS**

--

--

.TITLE Statement

Description:
Required first line of input file.

Format:
.TITLE <string>
or
<string>

Example:
Homework 1 Problem 2

--

COMMENTS

Description:
* - Comments entire line
$ - Comments part of line after $

Example:
* Vdd 2 0 DC 5V *(entire line commented)*
   Vdd 2 0 DC 5V $ 3V *(part after $ commented)*

--

2) SETUP Statements

--

--

.GLOBAL statement

Description:

Variable is assigned same value in Main and Subcircuits

Format:

.GLOBAL name1 name2

Example:

.GLOBAL Vdd Vss

--

.PARAM statement

Description:

To define a variable and its value

Format:

.PARAM var_name =var_value

Example:

.PARAM Vdd=1.5V *(unit has to be specified)*

--

.OPTIONS Statement

Description:

- Control options are set in .options statement. Any number of options can be set in one statement.
- If there is more than one .options statement the settings of the last one are taken.

Format:

.OPTIONS opt1 opt2…

Example:

.OPTIONS GMINDC=1.0000E-12 plot

.OPTION KEYWORDS (i.e opt1…)

(Page 318 of the hspice manual gives a keyword list for .option)

1. ***General control Keywords***

LIST (use this in your .options statement)

Produces an element summary listing of the input data to be printed.

NODE (use this in your .options statement)

Causes a node cross reference table to be printed. The table lists each node and all elements connected to it.

MEASOUT

Outputs .Measure statement values and sweep values into the <design>.mt# file; where # depends on the run specified by the .ALTER statement. *(explained later)*.

If .ALTER is not used then the file is <design>.mt0

PROBE
By default Hspice reports all node voltages and currents. The PROBE statement in .options along with the .PROBE / .PLOT / .PRINT / .GRAPH statements restricts the recorded values to those specified in these statements.

POST
Enables saving of results to be later analyzed using Awaves.
----

2. ***DC operating point Keywords***

For Convergence
CONVERGE=1
Specifies a methods to handle convergence problems

GMINDC=x
Specifies the magnitude of conductance placed in parallel with all pn junctions and MOSFET devices. **Increase** value if required to overcome convergance problems. Default value: 1e-12
----

3. ***Transient Analysis Keywords***

For Accuracy
ACCURATE
Sets some parameters governing accuracy to pre specified values.

.OP statement

Description:
Used to calculate the DC operating point of the circuit both in DC and transient analysis.

Format:
.op (Stores all voltage, current, conductance and capacitance)

.IC statement

Description:
Sets transient initial conditions.

Format:
.IC V(n1)= val1 V(n2)=val2

.NODESET statement

Description:
Initializes specified nodes for DC operating point analysis. Speeds up convergence.

Format:
.NODSET V(n1)= val1 V(n2)=val2
----

3) LIBRARY AND FILE INCLUDE statements

.INCLUDE statement

Description:
To include a file in the data file

Format:
.INCLUDE “<filepath> filename”

Example:
.INCLUDE “block1”

.LIB statement

Description:
Commonly used commands, analysis statements etc can be placed in a library file. Each such group is enclosed inbetween .LIB & .ENDL statements with a unique entryname.

Format:
.LIB “<filepath> filename” entryname

Example:
.LIB “MODEL” cmos1

---

4) NETLIST statements

--

--

Instantiating elements *(MOSFET, R, C, Voltage Source etc)*

<u>*Description:*</u>
The Hspice netlist can be composed of instances of many elements joint together. Each element is instantiated in the following format.

<u>*Format:*</u>
instance_name <node1 node2…> <element_name> <para1=val1> <par2=val2> <M=val>
where;
node# - the node names
element_name – name of the original object
instance_name – name of the instance
par# - parameters associated with the object
M – multiplier; replicates the element val times in parallel

The element_name **must** begin with the following letters for the mentioned circuit elements
C – Capacitor
I – Current source
M – MOSFET
R – Resistor
V – Voltage source
X – Sub circuit

--

.SUBCKT or .MACRO statements

<u>*Description:*</u>
The design is generally built hierarchically by splitting elements into smaller functional macros (e.g. gates)
.SUBCKT and .MACRO end in .ENDS and .EOM

<u>*Format:*</u>
.SUBCKT subname n1 <n2 n3…> <par=val ….>
or
.MACRO subname n1 <n2 n3…> <par=val ….>

To access elements within subcircuits use dot (.)
X1.inst_gate2.c (Node c of inst_gate2 in element X1)

<u>*Example:*</u>
Description of an inverter
.macro inv IN OUT size0=0.4u size1=0.3u
M1 OUT IN vdd pmos L=0.13u W=size0
M2 OUT IN 0 0 nmos L=0.13u W=size1
.eom

Instantiation of macro
X1 out_1 a inv size0=0.6u size1=0.5u
Note here that size0=0.6u, i.e. the value mentioned in the instantiation overrides the value mentioned at a lower level

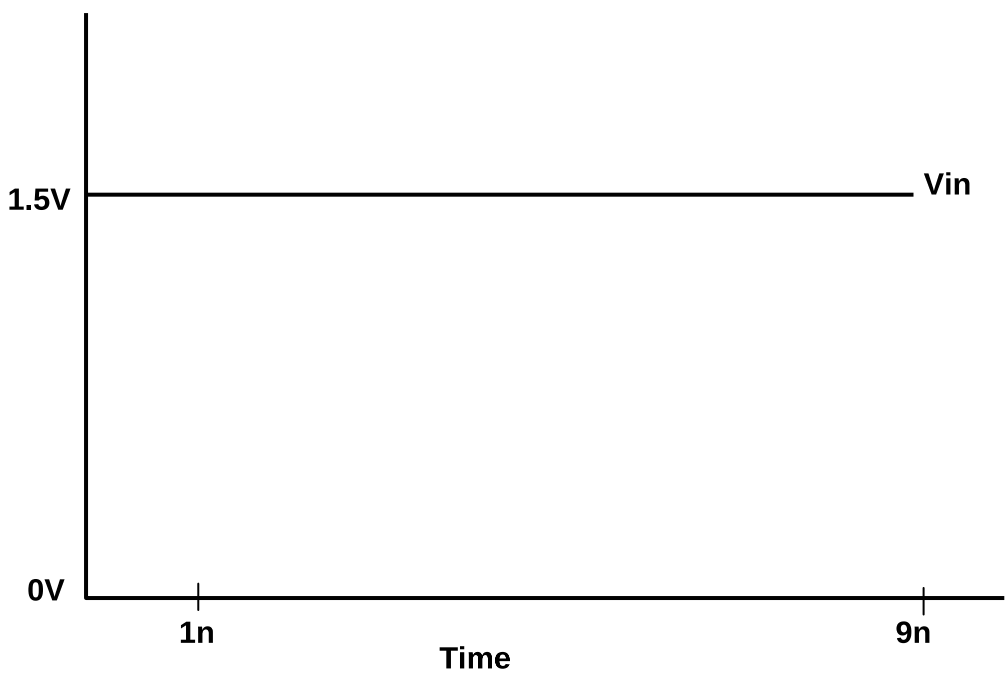
5) SOURCE statements

We will discuss about the following independent sources

DC Voltage Source

<u>*Format:*</u>
Vin node1 node2 **DC** value

<u>*Example:*</u>
Vin a0_in b0_in **DC** 1.5V



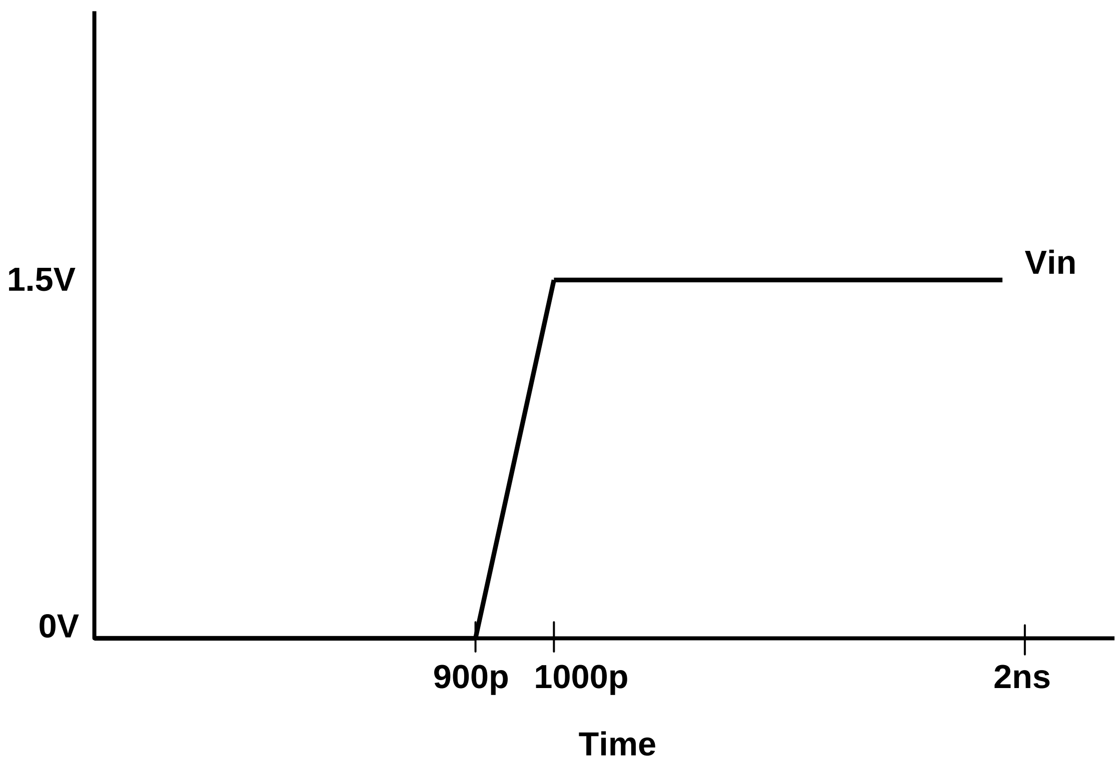
Piece-wise Linear (PWL)

Format:

Vin node1 node2 **PWL** time0 value0 time1 value1 time2 value2 …

Example:

Vin a0_in b0_in **PWL** 0 0V 900p 0V 1000p 1.5V



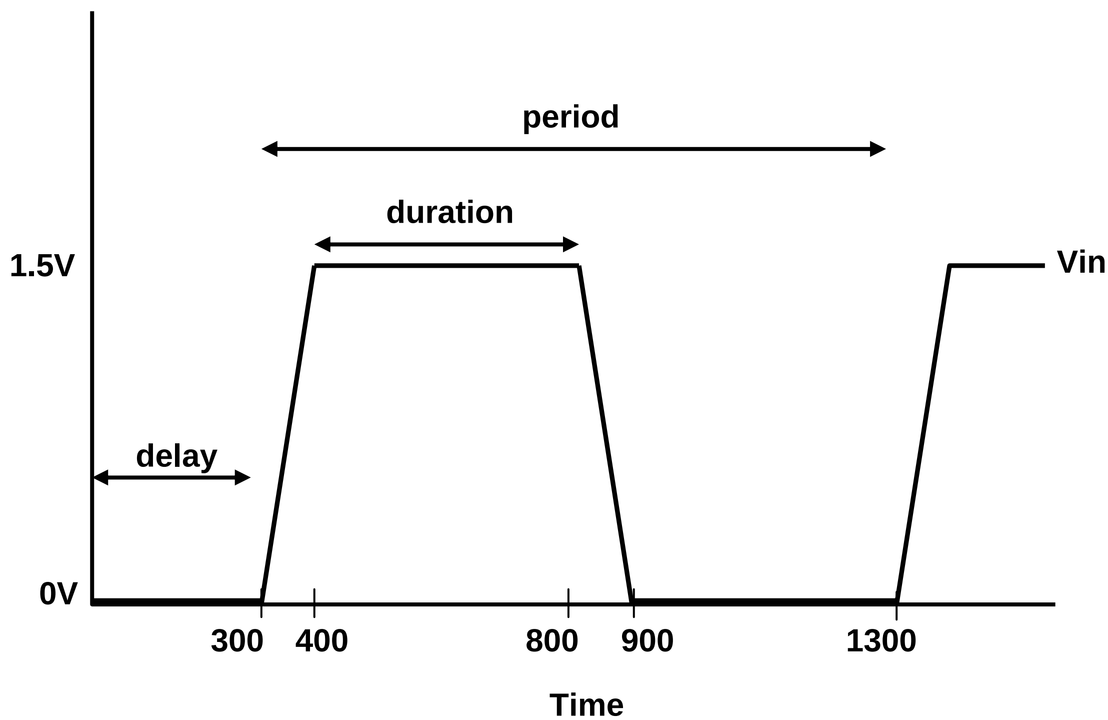
PULSE

Format:

Vin node1 node2 **PULSE** init_val pulse_val delay rise_time fall_time duration period

Example:

Vin a0_in b0_in **PULSE** 0V 1.5V 300p 100p 100p 400p 1000p



5) ANALYSIS

.***DC*** statement

Description:
Used to sweep DC values.

Format:
.DC var1 start1 stop1 incr1 <var2 start2 stop2 incr2>

Example:
.DC Vdd 1.2V 1.5V 0.3V

.***TRAN*** statement

Description:
Performs Transient simulation over the specified period of time.

Format1:
.**tran** 1ns 100ns
* Transient analysis every 1ns from 0 to 100ns.

Format 2:
.**tran** incr tran_stop **sweep** var start_val stop_val sweep_incr

* Sweeps “var” through range start_val to stop_val
* For each value of “var” performs transient analysis

Example:
.**tran** 0.1ns 10ns **sweep** period 500p 1000p 100p

Format 3:

.**tran** incr tran_stop **sweep** variable **LIN** n start_val stop_val

* “variable” is assigned “n” number of values uniformly distributed between start_val and stop_val.
* For each value of “variable” performs transient analysis

Example:
.**tran** 0.1ns 10ns **sweep** period LIN 6 500p 1000p

Format 4:

.**tran** incr tran_stop **sweep** variable **POI** n p_1 p_2 p_3 … p_n

* “variable” is assigned “n” values p_1 through p_n
* For each value of “variable” performs transient analysis

Example:
.**tran** 0.1ns 10ns **sweep** period POI 4 500p 600p 800p 1000p

---

.*TEMP* statement

Description:
Used to specify temperature

Format:
.TEMP val *(where val is in °C)*

Example:
.TEMP 25

---

6) OUTPUT statements

.PROBE statement

Description:
Specifies the data that is to be be recorded. Must be used along with the .OPTIONS PROBE statement.

Format:
.PROBE analysis_type var1 <var2…>
where; analysis_type could be DC, AC, TRAN, NOISE …
var# - variables; max allowed variables are 32 per .PROBE

Example:
.PROBE TRAN V(n_1) V(b)

.MEASURE statement (USER DEFINED ANALYSIS)

Description:
The .MEASURE statement is used to print ***user specified*** electrical specifications.
It can be used for DC, AC and Transient analysis.
Used to measure delay, power and other such parameters over the data points produced as a result of transient analysis.
Result of measurement placed in **.mt0** file.

Delay Measurement

* Calculate delay between two events occurring during transient analysis

Syntax:

.**measure tran** result
\+ **trig** trig_var **val**=trig_val <**td**=delay> <**rise/fall**=n>
\+ **targ** targ_var **val**=targ_val <**td**=delay> <**rise/fall**=n>

* “trig_var” – variable from which measurement will start
* “targ_var” – variable at which measurement will end
* “trig_val” – value of trig_var at which measurement will start
* “targ_val” – value of targ_var at which measurement will end
* “delay” – time that must elapse before starting (or stopping) measurement
* “n” – number of rising (or falling) transitions before starting (or stopping) measurement

Example:

.**measure tran** mydelay **trig** V(clk) **val**=’Vdd/2’ **td**=1ns **rise**=3
\+ **targ** V(out) **val**=’Vdd/2’ **td**=3ns **fall**=1

* “mydelay” will contain the time delay between the time at which the 3rd rising transition on the “clk” signal occurs and signal value is Vdd/2, and the time at which the 1st falling transition on the signal “out” occurs and signal value is Vdd/2.
* Start counting the number of rising transitions on “clk” only after simulation time=1ns.
* Start counting the number of falling transitions on “out” only after simulation time=3ns.

Power calculation

Syntax:

.**measure tran** result **AVG POWER** from=start_time to=end_time

* Returns the average power dissipated in the circuit during the transient analysis between simulation times start_time and end_time.

<u>Example:</u>

.**measure tran** avgpow **AVG POWER** from=1ns to=10ns
--

7) .ALTER statement (Pg 97)

<u>Description:</u>
The .ALTER statement can be used to rerun a simulation using different parameters and data.
The .ALTER block cannot include .PLOT , .PRINT, .GRAPH or any other IO statement.
All analysis statements .DC, .TRAN etc can be added, but only if it has not been used previously in the main program.

The .ALTER statement can include the following statements

- .DEL LIB
- .INCLUDE, .LIB
- .IC, .OP
- .OPTIONS
- .PARAM
- .TEMP
- .TRAN, .DC

Note:For more details check the manual. Be cautions about using this statement in the HW's. It might be better to narrow down to a small window where the answer lies rather than sweeping over large ranges for different conditions.
--

8) .END statement

Description:
Every input file must end with a .END statement.
A carriage return after .END is also required.

Format:
.END
--

E} INPUT AND OUTPUT FILES:

Input Files:

*.ini	Initialization file
*.ic	To set the DC operating point

Output Files:

Extension	
General files	
*.lis	Output listing, Plot outputs and Errors
*.st#	Output Status
*.pa#	Subcircuit cross-listing
In response to the .MEASURE statement	
*.mt#	Transient analysis output
*.ms#	DC analysis output
*.ma#	AC analysis output
In response to .OPTIONS POST statement	
*.tr#	Transient analysis output for Awaves
*.sw#	DC analysis output for Awaves
*.ac#	AC analysis output for Awaves

F} ABBREVIATIONS:

Unit	Abbreviation	Quantity	Abbreviation
Giga	G	Seconds	S
Mega	MEG	Watts	W
Killo	K	Farads	F
Milli	M		
Micro	µ		
Nano	N		
Pico	P		
Femto	F		

G} ***RUNNING HSPICE***

1. *From the Command Line:*
   In Prompting mode: (one by one prompts for the required inputs)

   hspice *(Enter)*

   In Non prompting mode:
   hspice <path>inputfile_name <<-o path> outputfile_name>
   e.g.
   hspice infile.sp
   hspice infile.sp –o outfile.lis

2. *From Awaves: (explained later in Awaves section)*

H} Using Awaves to view waveforms

Awaves is used to view waveforms generated by the spice simulation.
It can be used for running hspice as well.

Follow the given steps:

1. To open Awaves, on the command prompt type
   awaves & *(Return)*

2. Design -> Open
   Choose the .sp file of your design.

3. If you haven't run hspice on the file earlier it will give you an error. Close this error. A design browser window also opens up.

4. ***Running Hspice***
   If you haven't run hspice, run it from
   Tools -> Run Hspice
   Here select your design and press ***Run***. You can stop at any time by pressing ***Stop***.
   Hitting ***Run*** again continues the simulation.
   Listing opens up the .lis file. Errors are listed in this file.

5. ***Viewing Waveforms***
   The Results Browser should be open. Else open from Tools->Results Browser.
   Select the type of analysis and file from the top window.
   If you have used the .PROBE statement the probed values should appear directly in the ***Curves*** window.
   Else in ***Hierarchy*** select Top, ***Types*** select Voltages (or whatever you wish to plot) and the needed waveforms from the ***Curves*** window.

As mentioned earlier if you don’t mention specific electrical parameters in .PROBE all the currents and voltages are stored. Hence you can access nodes of internal macros by double clicking on ***Top*** and browsing to the required macro instance and then following the above mentioned procedure.

6 Always update the waveforms once you run hspice on a design.
Panels->Update

7 If nothing else works use the tried and tested method of shutting down awaves and invoking it again to update the results.