

# HSPICE Tutorial

Prepared by Dongwan Ha

Oct 21, 2008

## 1 Introduction

SPICE is a general purpose analog electronic circuit simulator. It is a powerful program that is used in IC and board-level design to check the integrity of circuit designs and to predict circuit behavior.

SPICE began as an innovative class project under the direction of Ron Rohrer at UC Berkeley in the academic year 1969-1970. It was originally named CANCER(Computer Analysis of Nonlinear Circuits, Excluding Radiation), and was developed by a class project team lead by Larry Nagel. He changed the name of it into SPICE and released to the public domain. It became very successful circuit simulator. From then to now, SPICE have given birth of many variations including PSPICE, HSPICE, SPICE-3 and so on.

HSPICE is one of SPICE variations commercialized by a company Meta Software ( merged with Avant! which is now part of Synopsys). Although there were many free and powerful SPICE variations, HSPICE became a standard of accurate circuit simulator. Since it costs a lot of money and effort for putting a circuit design into fabrication, accuracy and reliability of circuit simulator is a prime factor. Moreover, HSPICE was better supported since it is not free. Presumably, it was worth paying money for circuit design companies for this great simulator.

Anyway, SPICE is known for having several meanings, including

- Simulation Program with Integrated Circuit Emphasis
- Simulation Program Intended to Complement Engineering
- Sick Persons Idea of Cerebral Entertainment
- Special Program Inteded to Confuse Engineers
- Society for the Prevention of Intelligence in otherwise Competent Engineers

## 2 Netlist: input and output

### 2.1 Input file(.sp)

HSPICE needs well written input file. Input file includes network description of your circuit, and all options and analyses you want HSPICE to do. This input file may have following structure,

```

*<TITLE> (must have at the first line of your script)
( information you want to contain - usually title, author and revision information )
<declaration> $ declare global variables
<NETLIST> $ specify of your circuit
<SOURCES> $ stimulus
<options and misc> $ options, temperature and so on
<ANALYSIS> $ specify which analysis you want to perform (AC, DC, TRAN, etc.)
<MODELS> $ device model description
<measure> $ measurement statement ( '.measure', '.plot' )
<alter> $ alternative conditions you wish to run
.END

```

Note '<TITLE>' always comes first and '.end' always comes last. For the rest of the script, the order among items does not matter. You may need to contain items which contain capital letters. For example, we may not be able to run simulation without netlist or component model. For more details, consult Avant! Star-Hspice manual on the class website.

Check out this structure in the following example.

```

*<title>

*<declaration>
.global vcc gnd

*<netlist>
q1 collector base gnd npn_1
rc vcc collector 3k

*<sources>
v1 vcc gnd dc=1.8

vin base gnd dc=0 $ for dc analysis
*<analysis>
.dc vin 0 1.8 .01

*vin base gnd dc=0.637 ac=1$ for ac analysis
*.ac dec 20 1 10g

*vin base gnd sin(0.637 2u 20k) $ for transient analysis
*.tran 50u 100u

*<options>
.op
.options accurate
.options nomod post

*<models>
.model npn_1 npn is=5f bf=200 br=0.0005 vaf=1000000000 tf=100p

.end

```

## 2.2 Output listing(.lis)

Output listing file contains plenty of information on circuit analysis. This file lists all results obtained from the simulation. It includes

- HSPICE licensing information
- Error and Warning Report during the simulation
- Listing of the circuits
- Results from the analysis of the circuit

## 2.3 Graphical output(.tr#, .sw#, .ac#)

Graph data file are created by '.options nomod post' command. Using 'Avanwave' you can access to these graphical results from the simulation.

# 3 Input file Notation

## 3.1 Naming conventions

Every node and element within the HSPICE netlist must have its own unique name. These names can exist of the following,

- Node and Element identification
  - Either letters and numbers (e.g. ab, a1, 1a, 10)
  - Nodes with the same numbers treated as the same (e.g. 1a, 1b are the same)
  - 0 is ALWAYS ground
  - Case insensitive
  - Designated initial characters
    - \* resistor : Rxxx N1 N2 Value
    - \* capacitor : Cxxx N1 N2 Value
    - \* inductor : Lxxx N1 N2 Value
    - \* bjt : Qxxx C B E Model
    - \* mosfet : Mxxx D G S B Model
    - \* diode : Dxxx N+
    - \* independent voltage source : Vxxx N+ N- (DC) Value
    - \* independent current source : Ixxx N+ N- (DC) Value;
- Units
  - f = 1e-15
  - p = 1e-12
  - n = 1e-9
  - u = 1e-6

- m = 1e-3
- k = 1e3
- x = 1e6
- g = 1e9
- t = 1e12
- Miscellaneous
  - full line comment : \*
  - end of the line comment : \$
  - continuation of the line : +
- Parameterization
  - Use single quotation mark : 'expression'
  - \* example #1
 

```
.param VCC2 = 'VCC1 + 1'
vcc vcc gnd VCC2
```
  - \* example #2
 

```
vcc vcc gnd 'VCC1 + 1'
```
  - \* example #3
 

```
.measure v_p2p PARAM = 'v_p - v_n'
```

## 3.2 Analysis

Using HPSICE, circuit simulation including AC, DC, Transient and Noise analysis can be done.

- .DC : DC analysis
 

```
.DC Src_name Start Stop Increment <sweep ...>
.DC Vin 0 1.8 0.1
```
- .AC : AC analysis
 

```
.AC Scale nPoints Start Stop <sweep ...>
.AC DEC 20 1 10g $ 20 points per decade from 1 Hz to 1 GHz
```
- .TRAN : Transient analysis
 

```
.TRAN TimeStep StopTime <sweep ...>
```

You can attach `<sweep par start stop increment>` statement to obtain additional results with varying parameters.

## 3.3 Options

- .op : Print operating point in listing file
- .option nomod post : Require to obtain graphical result for Avanwave.
- .option dccap=1 : Invoke calculation of capacitance in DC analysis. This generates C-V plot. If not set, MOS device or voltage-variable capacitance values will not be evaluated.
- .option captab=1 : Print capacitance of each node at each operating point.

### 3.4 Measure and Plot

- .PRINT : This defines output from a SPICE analysis. The output will be listed in listing file as either voltage or current.

```
.PRINT DC V(Node1) V(Node1,Node2) I(Node3) VAR=PAR('Expression')
```

- .PLOT : This gives an ASCII plot instead of listing. The usage of this command is the same as we did in '.print' command.

```
.PLOT DC V(Node1) V(Node1,Node2) I(Node3) VAR=PAR('Expression')
```

## 4 More Information

for more details on HSPICE, please refer to the HSPICE Simulation and Analysis User Guide located at class website.