

HSPICE Files

Suffixes:

HSPICE has many files that it can take as input or produce. These files must contain these suffixes:

- **HSPICE Input**
 - *input netlist: .sp*
 - design configuration: .cfg
 - initialization: hspice.ini

Note: The italicized files those that you must have.

- **HSPICE Output**
 - *run status: .st0*
 - *output listing: .lis*
 - graph data, transient: .tr# (e.g. .tr0)
 - graph data, dc: .sw# (e.g. .sw2)
 - graph data, ac: .ac# (e.g. .ac1)
 - measure output: .m*# (e.g. .mt0)

Note: The italicized files are those that you will always have.

File Descriptions:

Netlist file (.sp)

The netlist file is your input file. This file contains your circuit description and all options and analysis you wish HSPICE deal with. The structure of the file looks like this:

title	-Implicit first line; becomes netlist file title
.options	-set conditions for simulation
ANALYSIS and TEMP	-statements to sweep variables
.print/.plot/Analysis	-set print, plot, and analysis variables
.initial conditions	-input state of system
sources	-stimulus
netlist	-circuit descriptions
.model libraries	-.lib and .inc
.end	-terminates the simulation

Note: This is just a suggested structure. Though the title always comes first and .end always comes last, the rest can come in any order in between. Another

popular structure that is more implicit is to put the analysis statements after the circuit descriptions.

An example Netlist structure:

```

This is a netlist                                     $must have a title line
.options post acct opts node                         $HPSICE options

l6 6      16      .05                                $inductor
c616     0      .05                                $capacitor
r16     16      0      40                          $resistor
c44     14      .1                                $another capacitor
l5 data  15      1                                $another inductor
c515    0      .2                                $another capacitor

v4      4      0      dc 0 ac 0 0 pulse 0 1 0 .15 .15 .4 2 $voltage source
vdata   data  0      sin(1.0 1.0 1.0 0.0 1.0)           $voltage source
v6      6      0      exp(1 0.1 .02 .6 .2)              $voltage source

.model ...                                           $models and subckts

.tran    .1 5                                         $transient analysis
.print   v(6) i(r16)                                  $print statement
.plot    v(4) v(14) v(data)                          $plot statement

.end                                               $Must have an end

```

Design configuration (.cfg)

Configuration files are used by HSPICE, GSI and HSPLOT to describe the available terminals and hardcopy devices. There is already a default .cfg file (meta.cfg) that already has all the printer and terminal types used at the ECE labs so it is unlikely that this file will need to be created.

Initialization (hspice.ini)

The initialization file deals primarily with setting up HSPICE itself and has no need to be modified by users.

Run status (.st0)

The run status file keeps track of all that HSPICE performs on a netlist file. Since no results are stored here, just execution steps, this file is not very useful and can be deleted without any great loss to a user. It does sometimes offer small benefit for debugging purposes as it can show which steps were not performed by HSPICE.

Output listing (.lis)

This is one of the most important files in HSPICE as this file lists all results obtained from the simulation. This file contains (in order of listing in the file):

1. HSPICE licensing information
2. Listing of the circuit
3. Results from the analysis of the circuit (.op, .print, .plot, .measure, .ac, and .tran in order of their appearance in the netlist file)

Graph data files (.tr#, .sw#, .ac#)

Graph data files are created by .OPTION POST command and contain the data to graphed by HSPLIT, GSI, or HSPICE. Basically these files retain the data to be graphed from the .lis file in a different graphing format.

Measure output (.m*#)

The measure output file hold the result from .MEASURE commands. These files aren't very useful either as the data is hard to read and already exists in a nicely formatted way in the .lis file.

Netlist Notation:

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 Names or Numbers (e.g. n1,33,in1,100)
 - Numbers: 1 to 99999999 (99 million)
 - Nodes with number followed by letter are all the same (e.g. 1a=1b)
 - 0 is **ALWAYS** ground
 - Capitals do not matter (e.g. node1=NODE1=Node1=nOdE1)
- Allowable Characters and Conventions (if nodes have names and not just numbers)
 - Begin with letter or /
 - Max of 16 characters (after 16 ignored)
 - May *contain*: +-* / ; \$ # . [] ! < > _ %
 - May *not contain*: () , = <space>
 - Ground may be either 0, gnd, or !gnd
- Every node must have at least 2 connections.

HSPICE units:

HSPICE is a smart program and will automatically assign ohms to resistors, Farads to capacitors, and Henries to inductors. However, it is not written that you must trust HSPICE and so the units can be manually assigned by:

- R - ohm
- C - Farad
- L - Henry

The scaling for the all units in HSPICE is done by:

- F=1e-15
- P=1e-12
- N=1e-9
- U=1e-6
- M=1e-3 *****
- K=1e3
- MEG=X=1e6 *****
- G=1e9
- T=1e12

Note: Capitals do not matter (e.g. F=f and M=m), therefore many people have problems with M and MEG. Please be careful with these two units!

Global Variables:

- Syntax
 - .GLOBAL node1 node2 node3Globally defined nodes
 - .GLOBAL VBIAS VCCGlobally defined sources
- Usage
 - When subcircuits are included in the data file.
 - Assigns common node name to subcircuit nodes
 - Power supply connection of all subcircuits often done this way
 - .GLOBAL VCC
 - Connects all nodes named VCC (all circuits have a common node)

Essentially globals are used most often for subcircuits, though they are great for sources. This way all power sources do not have to be separately defined.

Misc. Syntax:

In general, HSPICE notation is very simple. All statements beginning with a `<.>` are commands (e.g. `.OPTION`, `.PRINT`, `.AC`, etc.) while those that don't are treated as elements (e.g. `r51`, `cc`, `lload`, etc). There are, however, a few special commands:

- `<*>` - indicates a full line comment
- `<$>` - indicates a end of line comment
- `<+>` - indicates a continuation of the previous line

Examples of these can be found in the BJT amplifier example [bjtamp.sp](#). Here the `.OPTION` command is extended for four lines and the end of line comment is used after the `.MEASURE` commands. Notice too that by using tabs for the elements and their nodes, it makes the netlist much easier to read. It is not required that tabs be used, but it is very helpful.

Algebraic Syntax:

HSPICE also has the added benefit of being able to do some algebraic manipulation of some variables. The variables can be user defined or just the voltages and currents at nodes (e.g. `v(1)`, `I(q2)`, etc.). Also the algebra must be enclosed within single quotes (see examples below). The algebraic functions used by HSPICE in addition to `(+,-,*,/)` are as follows:

- `sin(x)`
- `cos(x)`
- `tan(x)`
- `atan(x)`
- `exp(x)`
- `log(x)` - this function will use the abs value of x and then apply the sign of x to the results.
- `pwr(x,y)` - x to power y, see also log.
- `sinh(x)`
- `cosh(x)`
- `tanh(x)`
- `sqrt(x)` - see also log
- `db(x)`
- `log10(x)` - see also log
- `abs(x)`
- `min(x,x)`
- `max(x,x)`

Some examples of using algebraic functions in HSPICE are:

- Parameterization (variable declaration)

```
.PARAM x='y+3'
```

- In elements

```
r1      1      0      r='abs(v(1)/i(m1))+10'
```

- In .MEASURE statements
- .MEAS vmax MAX V(1)
- .MEAS imax MAX I(q2)
- .MEAS ivmax PARAM='vmax*imax'

Topology Rules:

It is important to know when constructing the circuit description that HSPICE will not allow certain topologies. It will not allow:

- Every node must have a DC path to ground (i.e. all circuits must have at least one ground not in series with a capacitor).
- No dangling nodes (i.e. all nodes must have at least two connections).
- No voltage loops (i.e. no voltage sources in parallel with no other elements).
- No ideal voltage source in closed inductor loop.
- No stacked current sources (i.e. no current sources in series).
- No ideal current source in closed capacitor loop.

Running HSPICE:

Text Interface:

To run HSPICE in text interface nothing could be simpler, just follow these steps:

1. Create the nestlist (.sp) file using any available text editor (vi, pico, emacs, nedit, etc.).
2. To run the HSPICE simulation type in

```
hspice <filename>.sp > <filename>.lis
```

3. View the results (.lis file)