

SPICE ‘Quick’ Reference Sheet

THE GENERAL ANATOMY OF A SPICE DECK

SPECIFYING CIRCUIT TOPOLOGY: DATA STATEMENTS

Basic Components

Resistors

Capacitors and Inductors

Voltage and Current Sources

Independent DC Sources

Independent AC Sources

Transient Sources

Sinusoidal Sources

Piecewise Linear Source (PWL)

Pulse

Dependent Sources

Semiconductor Devices

Diode

Bipolar Transistor

MOSFET

Ideal Operational Amplifiers

Subcircuits

SPECIFYING ANALYSIS: CONTROL STATEMENTS

Operating Point (.OP)

DC Statement (.DC)

AC Analysis (.AC)

Transient (.TRAN)

Initial Conditions (.IC)

Transfer Function (.TF)

LOOKING AT YOUR DATA: OUTPUT STATEMENTS

Output to your *.lis or *.lst file (.PRINT and .PLOT)

Outputting to MWaves/AWaves (.OPTION POST)

Searching for a Value (.MEASURE)

THE GENERAL ANATOMY OF A SPICE DECK

SPICE input file, called source file, consists of three parts. First, Data Statements describe the components and the interconnections. Then, Control Statements tell SPICE what type of analysis to perform on the circuit. Finally, Output Statements specify what outputs are to be printed or plotted.

Although these statements may appear in any order, it is recommended that they be given in the above sequence. Two other statements are required: the title statement and the end statement. The title statement is the first line and can contain any information, while the end statement is always .END. This statement must be a line by itself, followed by a carriage return! In addition, you can insert comment statements, which must begin with an asterisk (*) and are ignored by SPICE.

```
TITLE STATEMENT
ELEMENT STATEMENTS
.
COMMAND (CONTROL) STATEMENTS
OUTPUT STATEMENTS
.END <CR>
```

The statements have a free format and consist of fields separated by a blank. If one wants to continue a statement to the next line, one uses a "+" sign (continuation sign) at the beginning of the next line. Numbers can be integers, or floating points. For example,

```
RES1 1 0 3500 or RES1 1 0 3.5E3
```

One can also use the following scale factors:

Suffix	Scale		
T	E+12	1,000,000,000,000	Tera
G	E+09	1,000,000,000	Giga
X or MEG	E+06	1,000,000	Mega
K	E+03	1,000	Kilo
M	E-03	0.001	Milli
U	E-06	0.000001	Micro
N	E-09	0.000000001	Nano
P	E-12	0.000000000001	Pico
F	E-15	0.000000000000001	Femto

Both upper and lower case letters are allowed in HSPICE (SPICE uses only uppercase characters). So could specify a capacitor of 225 picofarad in the following ways:

```
225P, 225p, 225pF; 225pFarad; 225E-12; 0.225N, etc.
```

The Basic Components

Resistors

Rname N+ N- Value

- N+ represents the positive terminal, N- represents the negative terminal.
- Value is the resistance value
- The ultimate in simplicity.

Inductors and Capacitors

Cname N+ N- Value <IC=Initial Condition>

Lname N+ N- Value <IC=Initial Condition>

- IC is the initial condition (DC voltage for capacitors or DC current for inductors). It is optional
- The symbol <> means that the field is optional. If not specified, it is assumed to be zero. In case of an inductor, the current flows from N+ to N- .

Voltage and Current Sources:

Independent DC Sources

Voltage source: vname N+ N- <DC=> DCValue

Current source: Iname N+ N- <DC=> DCValue

- N+ is the positive terminal
- N- is the negative terminal
- DCValue gives the value of the source
- The name of a voltage and current source must start with V and I, respectively.

Independent AC Sources

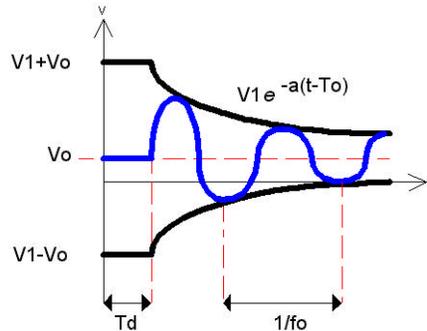
Voltage source: vname N+ N- AC ACValue, Phase

Current source: Iname N+ N- AC ACValue, Phase

- N+ is the positive terminal
- N- is the negative terminal
- ACValue gives the value of the source
- ACPhase gives the phase in degrees
- The name of a voltage and current source must start with V and I, respectively.

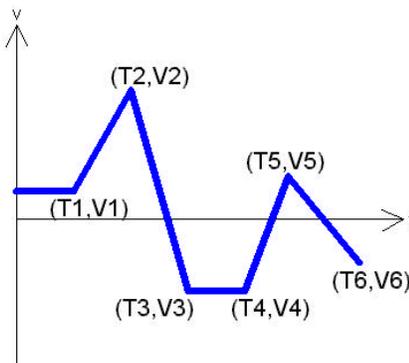
Transient Sources

Sinusoidal



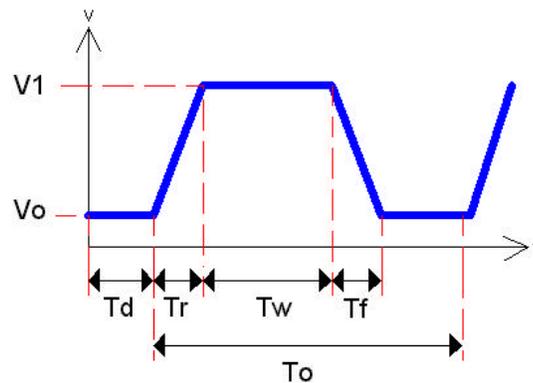
$Vname\ N+\ N-\ SIN(VO\ VA\ fo\ <TD>\ <a>\ <PHASE>)$	
<ul style="list-style-type: none"> • $Vname = VO + VA \exp[-a.(t - TD)] \sin[2\pi.f (t - TD) + (PHASE/360)]$ • VO - offset voltage in volts. • VA - amplitude in volts. • fo -the frequency in hertz. • TD - delay in seconds • a - damping factor per second • Phase - phase in degrees <p><i>(If TD, a and PHASE are not specified, it is assumed to be zero.)</i></p> <ul style="list-style-type: none"> • A cosine can be generated by shifting a sin by 90 degrees 	

Piece-Wise Linear



$Vname\ N+\ N-\ PWL(T1\ V1\ T2\ V2\ T3\ V3\ \dots\ Tn\ Vn)$	
<ul style="list-style-type: none"> • $(Ti\ Vi)$ specifies the value Vi of the source at time Ti 	

Pulse



<code>Vname N+ N- PULSE(Vo V1 Td Tr Tf Tw To)</code>
<ul style="list-style-type: none">• <code>Vo</code> - initial voltage• <code>V1</code> - peak voltage• <code>Td</code> - initial delay time• <code>Tr</code> - rise time• <code>Tf</code> - fall time• <code>Tw</code> - pulse-width• <code>To</code> - period of wave

Dependent Sources

Voltage controlled voltage source: <code>Ename N+ N- NC1 NC2 Value</code>
Voltage controlled current source: <code>Gname N+ N- NC1 NC2 Value</code>
Current controlled voltage source: <code>Hname N+ N- Vcontrol Value</code>
Current controlled current source: <code>Fname N+ N- Vcontrol Value</code>
<ul style="list-style-type: none">• <code>N+</code> and <code>N-</code> are the positive and negative terminals of the dependent source, respectively.• <code>NC1</code> and <code>NC2</code> are the positive and negative terminals of the controlling voltage source, respectively.• <code>Vcontrol</code> is the zero value voltage source used to measure the controlling current (the positive current flows into the positive terminal of the controlling voltage source!).• <code>Value</code> is a multiplier value. In math terms: $Value * (NC1 - NC2) = (N+ - N-)$

Semiconductor Devices

Diode

Element line (What appears in your circuit)

Dname N+ N- MODName

Model statement: (Appears in your deck to describe circuit element)

```
.MODEL MODName D (IS= N= RS= CJO= Tt= BV= IBV=)
```

```
.model D1N4148 D (IS=0.1PA, RS=16 CJO=2PF TT=12N BV=100 IBV=0.1PA)
```

- The element name starts with D to indicate that the element is a diode,
- N+ is the positive end and N- is the negative end (where the arrow points)
- MODName is the name of the model of the diode specified in the model line.
- IS - saturation current, (default=1E-14A),
- N - the emission coefficient, (=1),
- RS - the series resistance, (=0 ohm),
- CJO - junction capacitance, (=0F),
- TT - transit time, (=0sec),
- BV - reverse bias breakdown voltage, (=infinite) and
- IBV - the reverse bias breakdown current, (=1xE-10A).

If a parameter is not specified the default value (given in parenthesis) is assumed.

Bipolar Transistor

Element: (BASIC MODEL)

Qname C B E BJT_modelName

Model statement:

```
.MODEL BJT_modName NPN (BF=val IS=val VAF=val)
```

```
.MODEL BJT_modName PNP (BF=val IS=val VAF=val)
```

```
.model Q2N2222A NPN (IS=14.34F XTI=3 EG=1.11 VAF= 74.03 BF=255.9  
NE=1.307 ISE=14.34F IKF=.2847 XTB=1.5 BR=6.092 NC=2 ISC=0 IKR=0 RC=1  
CJC=7.306P MJC=.3416 VJC=.75 FC=.5 CJE=22.01P MJE=.377 VJE=.75  
TR=46.91N TF=411.1P ITF=.6 VTF=1.7 XTF=3 RB=10)
```

- BF is the common emitter current gain β ,
- IS is the saturation current
- VAF is the Early voltage.
- If no values are specified, the default values are assumed ($\beta=100$; IS=1E-16A, and VAF=[infinite]).
- As an example, the model parameters for the 2N2222A NPN transistor is given above. As you can see, there are many other parameters that can be specified. A more detailed list can be found in the SPICE manual

MOSFET

Element: (BASIC MODEL)

```
Mname ND NG NS <NB> ModName <L=VAL> <W=VAL>
```

Model:

```
.MODEL ModName NMOS (<LEVEL=val> <keyname=val> ... )
```

```
.MODEL ModName PMOS (<LEVEL=val> <keyname=val> ... )
```

- Like the BJT model, one can use a simplified circuit call, and simply specify width and length. The model being called will have additional parameters already specified. In this class, you will not have to worry about the modeling parameters.

Ideal Operational Amplifiers

We've actually already seen this! An op-amp can be modeled as a voltage controlled voltage source. Look back to the Dependent Sources section and see how you can match the leads up.

Subcircuits

But what if you want a realistic op-amp model? That's where subcircuits become useful. SPICE allows you to define a collection of elements as a subcircuit (e.g. an operational amplifier) and to insert this description into the overall circuit (as you would do for any other element).

Defining a subcircuit

A subcircuit is defined by a .SUBCKT control statement, followed by the circuit description as follows:

```
.SUBCKT SUBNAME N1 N2 N3 ...  
Element statements  
.  
.  
.  
.ENDS SUBNAME
```

in which SUBNAME is the subcircuit name and N1, N2, N3 are the external nodes of the subcircuit. The external nodes cannot be 0. The node numbers used inside the subcircuit are strictly local, except for node 0 which is always global.

Using a subcircuit

The element statement for a subcircuit is similar to any other element. The format is as follows:

```
Xname N1 N2 N3 . . . SUBNAME
```

in which Xname refers to the element (subcircuit) being used; N1, N2, N3 are the nodes to which the external nodes of the subcircuit are being connected, and SUBNAME is the name of the subcircuit being used.

An example of an inverting opamp circuit using the subcircuit of the uA741 (see operational amplifiers above) is given below. The subcircuit is called x1.

```
vs 1 0 dc 5  
r1 1 2 200  
rf 2 3 1k  
x1 0 2 3 opamp741  
.dc vs 0 10 1  
.plot dc v(3)  
.end
```

SPECIFYING ANALYSIS: CONTROL STATEMENTS

By now you should have a basic understanding of the vocabulary SPICE uses to describe the physical circuit. Now we turn to analysis.

.OP Statement

This statement instructs Spice to compute the DC operating points:

- voltage at the nodes
- current in each voltage source
- operating point for each element

HSpice does not give the DC voltages unless you have specified a certain analysis type, such as `.TRAN`, or `.AC` analysis (Spice automatically does a DC analysis before doing a transient or AC analysis). Thus, if you are only interested in the DC voltages in HSpice, you should specify the `.OP` option. This output will appear to the output (which is usually piped to a `*.lis` or `*.lst` file)

.DC Statement

This statement allows you to increment (sweep) an independent source over a certain range with a specified step. The format is as follows:

```
.DC SRCname START STOP STEP
```

in which SRC name is the name of the source you want to vary; START and STOP are the starting and ending value, respectively; and STEP is the size of the increment.

Example: `.DC V1 0 20 2`

When the `Start` and `Stop` values are identical (and the `Step` is non-zero), the `.DC` command produces only one value. This may be useful in HSpice when you do not want all the DC voltages and currents to be printed (with the `.OP` option), but are interested in a limited number of DC voltages and currents. In the `.PRINT` statement (see further) you would then also specify the node voltages you like to know (ex. `.DC Vs 3 3 1` and `.PRINT DC V(2), V(3), I(Vmeas)`).

You can nest the DC sweep command. This is often used to plot transistor characteristics, such as the Drain current i_{ds} versus the Drain-source voltage (V_{ds}) for different gate voltages (V_{gs}). This can be done as follows:

```
.DC SRCname1 START STOP STEP SRCname2 START STOP STEP
```

Example: `.DC Vds 0 5 0.5 Vgs 0 5 1`

In the example above, the voltage V_{ds} will be swept from 0 to 5V in steps of 1V for every value of V_{gs} .

.AC Statement

This statement is used to specify the frequency (AC) analysis. The format is as follows:

```
.AC LIN NP FSTART FSTOP  
.AC DEC ND FSTART FSTOP  
.AC OCT NO FSTART FSTOP
```

in which LIN stands for a linear frequency variation, DEC and OCT for a decade and octave variation respectively. NP stands for the number of points and ND and NO for the number of frequency points per decade and octave. FSTART and FSTOP are the start and stopping frequencies in Hertz

Example: `.AC DEC 10 1000 1E6`

.TRAN Statement

This statement specifies the time interval over which the transient analysis takes place, and the time increments. The format is as follows:

```
.TRAN TSTEP TSTOP <TSTART <TMAX>> <UIC>
```

- TSTEP is the printing increment.
- TSTOP is the final time
- TSTART is the starting time (if omitted, TSTART is assumed to be zero)
- TMAX is the maximum step size.
- UIC stands for Use Initial Condition and instructs HSPICE not to do the quiescent operating point before beginning the transient analysis. If UIC is specified, HSPICE will use the initial conditions specified in the element statements.

.IC Statement

This statement provides an alternative way to specify initial conditions of nodes (and thus over capacitors).

```
.IC Vnode1 = value Vnode2 = value etc.
```

.TF Statement

The `.TF` statement instructs HSPICE to calculate the following small signal characteristics:

- the ratio of output variable to input variable (gain or transfer gain)
- the resistance with respect to the input source
- the resistance with respect to the output terminals

```
.TF OUTVAR INSRC
```

in which `OUTVAR` is the name of the output variable and `INSRC` is the input source.

Example: `.TF V(3,0) VIN`

The `.TF` statement can be used to find the Thevenin small signal equivalent resistance. (The Thevenin voltage is given by the node voltage at the open circuit terminal, as a result of the `.OP` statement).

LOOKING AT YOUR DATA: OUTPUT STATEMENTS

.PRINT and .PLOT

These statements will instruct HSPICE what output to generate. If you do not specify an output statement, HSpice will always calculate the DC operating points. The two types of outputs are the .PRINTs and .PLOTs . A print is a table of data points and a plot is a low-resolution graphical representation. The format is as follows:

```
.PRINT TYPE OV1 OV2 OV3 ...  
.PLOT TYPE OV1 OV2 OV3 ...
```

in which TYPE specifies the type of analysis to be printed or plotted and can be DC , TRAN or AC .

The output variables are OV1, OV2 and can be voltage between nodes, the voltage between a node and ground. With currents, you can also specify the currents between nodes and more importantly, the currents running through a particular voltage source, which is useful for power consumption. In addition, you can define the type of output by simply putting a suffix after V or I. The suffixes are:

- M: Magnitude
- DB: Magnitude in dB (decibels)
- P: Phase
- R: Real part
- I: Imaginary part

Examples:

```
* Plot the DC voltage between nodes 1 and 2, the voltage at node  
* 3, and the current through the voltage source, Vmeas  
.PLOT DC V(1,2) V(3) I(Vmeas)  
  
* What does this ask for?  
.PRINT TRAN V(3,1) I(Vmeas)  
  
* How about this?  
.PLOT AC VM(3,0) VDB(4,2) VM(2,1) VP(3,1) IR(V2)
```

Outputting to Mwaves/Awaves (.OPTION POST)

.PRINT and .PLOT output values to your screen (unless you pipe it into a *.lis or *.lst file). But to really produce the nice plots, you need to use MWaves.

MWaves requires that you place `.option post` somewhere in your spice deck. HSPICE will then generate automatically the files that MWaves looks for when creating graphs and charts. If you don't, MWaves will only be able to plot a single point at best.

Searching for a Particular Value (.MEASURE)

.MEASURE is often used in circuit optimization. With it, you can find when a certain event occurs as you sweep various parameters.

You can use .MEASURE for finding:

- Rise, Fall and Time Delay
- Average, RMS, min, max, peak-to-peak and integral
- Find X when Y occurs
- Derivative and Integral Evaluations
- Equation Evaluations
- Relative Error (See Manual for Examples)

.MEASURE is a complex command. To explain all the nuances would make this quite a long and dull read. Instead, we are going to post some sample .MEASURE commands that you might find useful. We encourage you to take a look at Chapter 4 of the Spice Manual for more information

Rise and Fall

```
.MEAS TRAN rise TRIG V(1) VAL=.2 RISE=1  
+           TARG V(1) VAL=.8 RISE=1
```

Gives the time it takes for node 1 to go from 20% to 80% of the maximum voltage (assuming a max voltage of 1V)

Time Delay

```
.MEAS TRAN tdelay TRIG V(1) VAL=2.5 TD=10n RISE=1  
+           TARG V(2) VAL=2.5           FALL=1
```

This command takes a look between two points and calculates the the time delay as a signal pushes the voltage up at first node 1 and then node 2. The crossing does not count unless it lasts longer than 10 ns. It only counts the first time it rises and the first time it falls.

Average (and RMS, MIN, MAX and Peak to Peak)

```
.MEAS TRAN avgval AVG V(1) FROM=10ns TO=55ns
```

This takes the average value of node 1 from 10ns to 55ns and outputs it as `avgval` in the *.lis or *.lst file.

If you replace the keyword AVG with RMS, MIN, MAX and PP HSPICE will calculate that function for the time given.

Find and When

```
.MEAS TRAN DesiredCurr FIND I(Vmeas) WHEN V(1)=1V
```

Output to DesiredCurr the current through the voltage supply Vmeas when node 1 reaches 1V.

Derivatives and Integrals

```
.MEAS TRAN slewrate DERIV V(out) AT=25ns
```

This calculates the derivative of V(out) at 25ns. Derivative is always calculated with respect to the sweeping parameter (which is time in this case)

Using Equation Evaluations

```
.MEAS TRAN slew DERIV v(1) WHEN v(1)='0.9*vdd'
```

In this case, we're calculating slew rate when v(1) is equal to 0.9*vdd.

```
.MEAS AC delay DERIV 'VP(output)/360.0' AT=10kHz
```

This calculates the delay which is equal to the derivative of the phase normalized by 360 degrees.

SOME FINAL REMARKS

Please pardon us if this reference sheet seems a little long. Believe it or not, we've probably compressed several hundred pages of manual reading into 14 measly pages. But we hope we've done a good job of giving you a feel for what HSPICE can do and that you will be motivated toward investigating the manual a little more.