

---

# Commands

## 1

---

## Overview

This chapter contains a table of naming and numerical conventions, having detailed descriptions of each Pspice “dot” command.



## Command Reference

[Table 1-1](#) lists all of the PSpice and PSpice A/D analysis “dot” commands. The “dot” command is contained in the circuit file. Schematics users can enter analysis specifications through the Analysis/ Setup dialog box.

**Table 1-1** *Command Summary*

Type	Corresponding Pspice Command	Description	Page
Standard Analyses	.AC	Frequency response	<a href="#">1-4</a>
	.DC	DC sweep	<a href="#">1-6</a>
	.FOUR	Fourier components	<a href="#">1-11</a>
	.NOISE	Noise	<a href="#">1-23</a>
	.OP	Bias point	<a href="#">1-25</a>
	.SENS	DC sensitivity	<a href="#">1-35</a>
	.TF	Small-signal DC transfer function	<a href="#">1-42</a>
	.TRAN	Transient	<a href="#">1-43</a>
Simple Multi-Run	.STEP	Parametric	<a href="#">1-36</a>
Analyses	.TEMP	Temperature	<a href="#">1-41</a>
Statistical Analyses	.MC	Monte Carlo	<a href="#">1-16</a>
	.WCASE	Sensitivity/Worst-Case	<a href="#">1-45</a>
Initial Conditions	.IC	Clamp node voltage for bias point calculation	<a href="#">1-12</a>
	.NODESET	Restored .NODESET bias point Store .NODESET bias point information	<a href="#">1-22</a>
Device Modeling	.END	End subcircuit definition	<a href="#">1-10</a>
	.DISTRIBUTION	Model parameter tolerance distribution	<a href="#">1-9</a>
	.MODEL	Modeled device definition	<a href="#">1-19</a>
	.SUBCKT	Start subcircuit definition	<a href="#">1-39</a>

Table 1-1 Command Summary

Type	Corresponding Pspice Command	Description	Page
Output Control	.PLOT	Analysis plot to output file (line printer format)	1-31
	 .PRINT	Analysis table to output file	1-33
	 .PROBE	Simulation results to Probe data file	1-34
	.WIDTH	The width of the output	1-48
Circuit File Processing	.END	End of circuit simulation description	1-10
	.INC	Include specified file	1-13
	.LIB	Reference specified library	1-14
	.PARAM	Parameter definition	1-29
Options	.OPTIONS	Sets all the options, limits.	1-26
		Analyses control parameters, and output characters	

## .AC (AC Analysis)

**Purpose** The .AC command is used to calculate the frequency response of a circuit over a range of frequencies.

**General Form** .AC <sweep type> <points value>  
+ <start frequency value> <end frequency value>

**Example** .AC LIN 101 100Hz 200Hz  
.AC OCT 10 1kHz 16kHz  
.AC DEC 20 1MEG 100MEG

<sweep type> The sweep type must be either LIN, OCT, or DEC.

Parameter *	Description	Description
LIN	linear sweep	The frequency is swept linearly from the starting to the ending frequency. The <points value> is the total number of points in the sweep.
OCT	sweep by octaves	The frequency is swept logarithmically by octaves. The <points value> is the number of points per octave.
DEC	sweep by decades	The frequency is swept logarithmically by decades. The <points value> is the number of points per decade.

\*One of the sweep types LIN, OCT, or DEC, must be specified.

<points value> The points value (an integer), is the number of points in the sweep.

<start frequency value> <end frequency value>

The end frequency values must not be less than the start frequency value, and both must be greater than zero. The whole sweep must include at least one point.

The simulator calculates the frequency response by linearizing the circuit around the bias point. All independent voltage and current sources which have AC values are inputs to the circuit.

**Note** A .PRINT, .PLOT, or .PROBE command must be used to get the results of the AC sweep analysis.

If a group delay (“G” suffix) is specified as an output, the frequency steps must be close enough together that the phase of that output changes smoothly from one frequency to the next. Group delay is calculated by subtracting the phases of successive outputs and dividing by the frequency increment.

During AC analysis, the only independent sources which have nonzero amplitudes, are those using AC specifications. The SIN specification does not count as it is used only during transient analysis.

AC analysis is a linear analysis. To analyze nonlinear functions, such as mixers, frequency doublers, and AGC, it is necessary to use transient analysis.

## .DC (DC Analysis)

**Purpose**

The .DC command performs a DC sweep analysis on the circuit.

**General Form**

.DC *<linear sweep type>* *<sweep variable name>*  
+ *<start value>* *<end value>* *<increment value>*  
+ *[nested sweep specification]*

.DC *<logarithmic sweep type>* *<sweep variable name>*  
+ *<start value>* *<end value>* *<points value>*  
+ *[nested sweep specification]*

.DC *<sweep variable name>* LIST *<value>\**  
+ *[nested sweep specification]*

**Example**

```
.DC VIN -.25 .25 .05
.DC LIN I2 5mA -2mA 0.1mA
.DC VCE 0V 10V .5V IB 0mA 1mA 50uA
.DC RES RMOD(R) 0.9 1.1 .001
.DC DEC NPN QFAST(IS) 1E-18 1E-14 5
.DC TEMP LIST 0 20 27 50 80 100 -50
```

The DC sweep analysis calculates the circuit's bias point over a range of values for *<sweep variable name>*. The first form, and the first four examples, are for doing a linear sweep. The second form, and the fifth example, are for doing a logarithmic sweep. The third form, and the sixth example, are for using a list of values for the sweep variable.

### Linear Sweeps

*<start value>* Can be greater or less than *<end value>*: that is, the sweep can go in either direction.

*<increment value>* **The value must be greater than zero.**

### Logarithmic Sweeps (DEC or OCT)

*<start value>* The value must be positive and less than *<end value>*.

*<points value>* The value is the number of points in the sweep, and must be an integer.

### Nested Sweep

For a nested sweep, a second sweep variable, sweep type, start, end, and increment values can be placed after the first sweep. In the nested sweep example, the first sweep is the “inner” loop: the entire first sweep is performed for each value of the second sweep.

The rules for the values in the second sweep are the same as for the first. The second sweep generates an entire .PRINT table or .PLOT plot for each value of the sweep. Probe allows nested sweeps to be displayed as a family of curves.

### Sweep Type

The sweep can be linear, logarithmic, or a list of values. .

Parameter *	Description	Meaning
LIN	linear sweep	The sweep variable is swept linearly from the starting to the ending value. <i>&lt;increment value&gt;</i> is the step size.
OCT	sweep by octaves	Sweep by octaves. The sweep variable is swept logarithmically by octaves. The <i>&lt;points value&gt;</i> is the number of steps per octave.
DEC	sweep by decades	Sweep by decades. The sweep variable is swept logarithmically by decades. The <i>&lt;points value&gt;</i> is the number of steps per decade.
LIST	list of values	Use a list of values. In this case there are no start and end values. Instead, the numbers that follow the keyword LIST are the values that the sweep variable is set to.

---

\*For [*linear sweep type*], the keyword LIN is optional, but either OCT or DEC must be specified for the *<logarithmic sweep type>*

## 1-8 Commands

---

*<sweep variable name>*

After the DC sweep is finished, the value associated with *<sweep variable name>* is set back to the value it had before the sweep started. The following items can be used as sweep variables in a DC sweep:

Parameter	Description	Meaning
<b>Source</b>	A name of an independent voltage or current source.	During the sweep, the source's voltage or current is set to the sweep value.
<b>Model Parameter</b>	A model type and model name followed by a model parameter name in parenthesis.	The parameter in the model is set to the sweep value. The following model parameters cannot be (usefully) swept: L and W for the MOSFET device (use LD and WD as a work around), and any temperature parameters, such as TC1 and TC2 for the resistor.
<b>Temperature</b>	Use the keyword TEMP for <i>&lt;sweep variable name&gt;</i> .	The temperature is set to the sweep value. For each value in the sweep, all the circuit components have their model parameters updated to that temperature.
<b>Global Parameter</b>	Use the keyword PARAM, followed by the parameter name, for <i>&lt;sweep variable name&gt;</i> .	During the sweep, the global parameter's value is set to the sweep value and all expressions are reevaluated.



## .DISTRIBUTION (User-Defined Distribution)

### Purpose

The .DISTRIBUTION command is used to define a user distribution for tolerances, and is only used with Monte Carlo. The curve described by a .DISTRIBUTION command controls the relative probability distribution of random numbers generated by PSpice to calculate model parameter deviations.

### General Form

.DISTRIBUTION <name> (<deviation> <probability>)\*

### Example

```
.DISTRIBUTION bi_modal (-1,1) (-.5,1) (-.5,0) (.5,0) (.5,1) (1,1)
.DISTRIBUTION triangular (-1,0) (0,1) (1,0)
```

The distribution curve is defined by (<deviation> <probability>) pairs, or corner points, in a piecewise linear fashion. Up to 100 value pairs are allowed.

### <deviation>

The deviation must be in the range (-1,+1), which matches the range of the random number generator. No <deviation> can be less than the previous <deviation> in the list, although it can repeat the previous value.

### <probability>

This represents a relative probability, and must be positive or zero.

# **.END** (End of Circuit)

### **Purpose**

The .END command marks the end of the circuit. All the data and every other command must come before it. When the .END command is reached, PSpice does all the specified analyses on the circuit.

### **General Form**

.END

There can be more than one circuit in an input file. Each circuit and each command are marked by a .END command. PSpice processes all the analyses for each circuit before going on to the next one.

Everything is reset at the beginning of each circuit. Having several circuits in one file gives the same results as having them in separate files and running each one separately. However, all the simulation results go into one “.out” file and one “.dat” file. This is a convenient way to arrange a set of runs for overnight operation.

### **Note**

*The last statement in an input file must be a .END command.*

### **Example**

```
* 1st circuit in file
... circuit definition
.END
* 2nd circuit in file
... circuit definition
.END
```

## **.FOUR** (Fourier Analysis)

**Purpose**                      Fourier analysis decomposes the results of a transient analysis into Fourier components.

**General Form**            `.FOUR <frequency value> [no. harmonics value] <output variable>`

**Example**                    `.FOUR 10kHz V(5) V(6,7) I(VSENS3)`  
`.FOUR 60Hz 20 V(17)`

The analysis results are obtained by performing a Fourier integral on the results from a transient analysis. The analysis must be supplied with specified output variables using evenly spaced time points. The time interval used is *<print step value>* in the `.TRAN` command, or 1% of the *<final time value>* (TSTOP) if smaller, and a 2<sup>nd</sup>-order polynomial interpolation is used to calculate the output value used in the integration. The DC component, the fundamental, and the 2<sup>nd</sup> through 9<sup>th</sup> harmonics of the selected voltages and currents, are calculated by default, although more harmonics can be specified. A `.FOUR` command requires a `.TRAN` command. Fourier analysis does not require a `.PRINT`, `.PLOT`, or `.PROBE` command. The tabulated results are written to the output file (".out") as the transient analysis is completed.

*<output variable>*        Is an output variable of the same form as in a `.PRINT` command or `.PLOT` command for a transient analysis.

*<frequency value>*        Is the fundamental frequency. Not all of the transient results are used, only the interval from the end, back to  $1/\text{<frequency value>}$  before the end is used. This means that the transient analysis must be at least  $1/\text{<frequency value>}$  seconds long.

**Note**        *The results of the .FOUR command are only available in the output file. They cannot be viewed in Probe.*

# .IC (Initial Bias Point Condition)

**Purpose** The .IC command is used to set initial conditions for both small-signal and transient bias points. Initial conditions can be given for some or all of the circuit's nodes.

**General Form** .IC < V(<node> [,<node>])=<value> >\*

**Example** .IC V(2)=3.4 V(102)=0 V(3)=-1V

The voltage between two nodes can be specified. During bias calculations, PSpice clamps the voltages to specified values by attaching a voltage source with a 0.0002 ohm series resistor between the specified nodes. After the bias point has been calculated and the transient analysis started, the node is "released."

<value> Is a voltage which is assigned to <node> for the duration of the bias point calculation.

**Note** *The .IC sets the initial conditions for the bias point only. It does not affect a DC sweep.*

If the circuit contains both the .IC command and .NODESET command for the same node, the .NODESET command is ignored (.IC overrides .NODESET).

Refer to your PSpice user's guide for more information on setting initial conditions.

## .INC (Include File)

**Purpose** The .INC command is used to insert the contents of another file.

**General Form** .INC <"file name">

**Example** .INC "SETUP.CIR"  
.INC "C:\LIB\VCO.CIR"

<"file name"> Can be any character string which is a legal file name for the computer system.

**Note** *For Unix based systems, file names are case sensitive. The file extension is not defaulted to ".inc". If a file name is specified, it must include its extension.*

Including a file is the same as bringing the file's text into the circuit file. Everything in the included file is actually read in. The comments of the included file are then treated just as if they were found in the parent file.

Included files can contain any legal PSpice statements, but the following notes must apply:

- The included files should not contain title lines unless they are commented
- .END command (if present), should mark only the end of the included file,
- Included files can be nested. Up to 4 levels of "including" are allowed.

**Comments** Every model and subcircuit definition, even if not needed, takes up space in the memory (RAM).

## .LIB (Library File)

**Purpose** The .LIB command is used to reference a model or subcircuit library in another file.

**General Form** .LIB ["file name"]

**Example** .LIB  
.LIB LINEAR.LIB  
.LIB "C:\LIB\BIPOLAR.LIB"

*file name* Can be any character string which is a legal file name for the computer system.

It may include a volume, directory, and version number. On some systems, such as the VAX, it can also be a logical name.

**Note** *For Unix based systems, file names are case sensitive.*  
*The file extension **is not defaulted** to ".lib". If a file name is specified, it **must include its extension**. If ["file name"] is left off it default to "NOM.LIB".*

Library files may contain comments, .MODEL statments, subcircuit definitions (including the .ENDS statement) and .LIB statement. No other statements are allowed.

Referencing a library is not the same as simply bringinig the file's text into the circuit file. Only those model or subcircuit definitions which are called by the circuit file are actually read in. So, only those model or subcircuit definitions which are needed take up space in main memory (RAM).

For examples of library statements, print out the file NOM.LIB.

**Note for IBM-PC versions only:**

library searches may be set-up automatically by using a DOS environment variable. By using the command form

SET PSPICELIB = <directory> [; <directory> ]\*

before *Pspice* is run , library files not found in the current directory may be found by searching the directories specified in the list. For example the command

```
set PSpiceLib=c:\dir1;..\jones;c:\pspice\lib
```

may be put in the AUTOEXEC.BAT file.

The directories will be searched in list order (left to right), similar to the SET PATH command in dos. The current values of environment variables may be seen by typing SET<cr>.

## .MC (Monte Carlo Analysis)

**Purpose** The .MC command causes a Monte Carlo (statistical) analysis of the circuit and multiple runs of the selected analysis (DC, AC, or transient) are performed.

**General Form** .MC <#runs value> <analysis> <output variable>  
+ <function> [option]\*

**Example** .MC 10 TRAN V(5) YMAX  
.MC 50 DC IC(Q7) YMAX LIST  
.MC 20 AC VP(13,5) YMAX LIST OUTPUT ALL

The first run uses nominal values of all components. Subsequent runs use variations on model parameters as specified by the DEV and LOT tolerances on each .MODEL parameter (see [.MODEL \(Model\) 1-19](#) command section for details on the DEV and LOT tolerances).

<#runs value> The total number of runs to be performed (for printed results the upper limit is 2,000, and if the output is to be viewed using Probe, the limit is 400).

The other specifications on the .MC command control the output generated by the Monte Carlo analysis.

<analysis> At least one DC, AC, or TRAN must be specified for <analysis>. This analysis is repeated in subsequent passes. All analyses that the circuit contains are performed during the nominal pass. Only the selected analysis is performed during subsequent passes.

<output variable> Identical in format to that of a .PRINT output variable; see [1-33](#) for .PRINT examples.



*<function>* Specifies the operation to be performed on the values of *<output variable>* to reduce these to a single value. This value is the basis for the comparisons between the nominal and subsequent runs.

The *<function>* must be one of the following.

Function	Definition
YMAX	Find the absolute value of the <b><i>greatest difference</i></b> in each waveform from the nominal run.
MAX	Find the <b><i>maximum value</i></b> of each waveform.
MIN	Find the <b><i>minimum value</i></b> of each waveform.
RISE_EDGE( <i>&lt;value&gt;</i> )	Find the <b><i>first occurrence</i></b> of the waveform crossing <b><i>above</i></b> the threshold <i>&lt;value&gt;</i> . The waveform must have one or more points at or below <i>&lt;value&gt;</i> followed by one above; the output value listed is the first point that the waveform increases above <i>&lt;value&gt;</i> .
FALL_EDGE( <i>&lt;value&gt;</i> )	Find the <b><i>first occurrence</i></b> of the waveform crossing <b><i>below</i></b> the threshold <i>&lt;value&gt;</i> . The waveform must have one or more points at or above <i>&lt;value&gt;</i> followed by one below; the output value listed is where the waveform decreases below <i>&lt;value&gt;</i> .

**Note** *<function>* and all *[option]s* (except for *<output type>*) have no effect on the Probe data that is saved from the simulation. They are only applicable to the output file.

## 1-18 Commands

---

[*option*]\* Can include zero or more of the following:

Option	Definition	Type Example
LIST	Lists, at the beginning of each run, the model parameter values actually used for each component during that run.	
OUTPUT <output type>	Asks for an output from subsequent runs, after the nominal (first) run. The output from any run is governed by a .PRINT, .PLOT, and .PROBE command in the file. If OUTPUT is omitted, then only the nominal run produces output. The <output type> is one of the ones shown in the Type Example column	ALL forces all output to be generated (including the nominal run).  FIRST <N> generates output only during the first n runs.  EVERY <N> generates output every n th run.  RUNS <N>* does analysis and generates output only for the listed runs. Up to 25 values can be specified in the list.
RANGE * (<low value>, <high value>)	Restricts the range over which <function> is evaluated. An “*” can be used in place of a <value> to show “for all values”. See the range examples in the Type Example column.	YMAX RANGE(*,.5) YMAX is evaluated for values of the sweep variable (e.g., time and frequency) of .5 or less.  MAX RANGE(-1,*) The maximum of the output variable is found for values of the sweep variable of -1 or more.

- If RANGE is omitted, then <function> is evaluated over the whole sweep range. This is equivalent to RANGE(\*,\*).

**Comments** For more information on Monte Carlo analysis, refer to your PSpice user’s guide.

# .MODEL (Model)

**Purpose** The .MODEL command defines a set of device parameters which can be referenced by devices in the circuit.

**General Form** .MODEL <model name> <model type>  
+ ([<parameter name> = <value> [tolerance specification]])\*

**Example**

```
.MODEL RMAX RES (R=1.5 TC1=.02 TC2=.005)
.MODEL DNOM D (IS=1E-9)
.MODEL QDRIV NPN (IS=1E-7 BF=30)
.MODEL MLOAD NMOS(LEVEL=1 VTO=.7 CJ=.02pF)
.MODEL CMOD CAP (C=1 DEV 5%)
.MODEL DLOAD D (IS=1E-9 DEV .5% LOT 10%)
.MODEL RTRACK RES (R=1 DEV/GAUSS 1% LOT/UNIFORM 5%)
```

<model name> The model name which is used to reference a particular model. It is good practice to make this the same letter as the device name ( e.g., D for diode, Q for bipolar transistor) but this is not required.

<model type> The <model type> is the device type and must be one of the types outlined in the following table:

Model Type	Instance Name	Type of Device
CAP	Cxxx	capacitor
CORE	Kxxx	nonlinear, magnetic core (transformer)
D	Desexed	diode
DINPUT	Nxxx	digital input device (receive from digital)
DOUTPUT	Oxxx	digital output device (transmit to digital)
GASFET	Bxxx	N-channel GaAs MESFET
IND	Lxxx	inductor
ISWITCH	Wxxx	current-controlled switch
LPNP	Qxxx	lateral PNP bipolar transistor
NJF	Jxxx	N-channel junction FET
NMOS	Mxxx	N-channel MOSFET

Model Type	Instance Name	Type of Device
NPN	Qxxx	NPN bipolar transistor
PJF	Jxxx	P-channel junction FET
PMOS	Mxxx	P-channel MOSFET
PNP	Qxxx	PNP bipolar transistor
RES	Rxxx	resistor
UDLY	Uxxx	digital delay line
UEFF	Uxxx	edge-triggered flip-flop
UGATE	Uxxx	standard gate
UGFF	Uxxx	gated flip-flop
UIO	Uxxx	digital I/O model
USUHD	Uxxx	setup and hold checker
UTGATE	Uxxx	tristate gate
UWDTH	Uxxx	pulse width checker
VSWITCH	Sxxx	voltage-controlled switch

Devices can only reference models of a corresponding type; e.g.,

- A JFET can reference a model of types NJF or PJF, but not of type NPN.
- There can be more than one model of the same type in a circuit, although they must have different names.

Following the *<model type>* is a list of parameter values enclosed by parentheses. None, any, or all of the parameters can be assigned values. Default values are used for all unassigned parameters. The lists of parameter names, meanings, and default values are found in the individual device descriptions.

[*tolerance specification*] Can be appended for each parameter, using the format:

[DEV [*track & dist*] *<value>*[%]] [LOT [*track & dist*] *<value>*[%]]

to specify an individual device (DEV) and the device lot (LOT) parameter value deviations. The tolerance specification is used by the .MC analysis only.

The LOT tolerance requires that all devices that refer to the same model use the same adjustments to the model parameter. DEV tolerances are independent, that is each device varies independently. The “%” shows a relative (percentage) tolerance. If it is omitted, *<value>* is in the same units as the parameter itself.

[*track & dist*] Specifies the tracking and non-default distribution, using the format:

[/*<lot #>*]/[/*<distribution name>*].

These specifications must immediately follow the keywords DEV and LOT (without spaces) and are separated by “/”.

*<lot #>* Specifies which of ten random number generators, numbered 0 through 9, are used to calculate parameter value deviations. This allows deviations to be correlated between parameters in the same model, as well as between models. The generators for DEV and LOT tolerances are distinct: there are ten generators for DEV tracking and ten generators for LOT tracking. Tolerances without *<lot #>* get individually generated random numbers.

*<distribution name>*

The distribution name is one of the following. The default distribution can be set by using the .OPTIONS command DISTRIBUTION parameter.

Distribution Name	Function
UNIFORM	Generates uniformly distributed deviations over the range $\pm<value>$ .
GAUSS	Generates deviations using a Gaussian distribution over the range $\pm 3\sigma$ and <i>&lt;value&gt;</i> specifies the $\pm 1\sigma$ deviation (i.e., this generates deviations greater than $\pm<value>$ ).
<i>&lt;user name&gt;</i>	Generates deviations using a user-defined distribution and <i>&lt;value&gt;</i> specifies the $\pm 1$ deviation in the user-definition; see the <a href="#">.DISTRIBUTION (User-Defined Distribution)</a> command (1-9).

**Comments** For more information refer to your PSpice user’s guide.

## **.NODESET** (Endostea)

### **Purpose**

The .NODESET command helps calculate the bias point by providing an initial best guess for some node voltages. Some or all of the circuit's node voltages can be given the initial guess, and in addition, the voltage between two nodes can be specified.

### **General Form**

`.NODESET < V(<node> [,<node>])=<value> >*`

### **Example**

`.NODESET V(2)=3.4 V(102)=0 V(3)=-1V`

This command is effective for the bias point (both small-signal and transient bias points) and for the first step of the DC sweep. It has no effect during the rest of the DC sweep, nor during a transient analysis.

Unlike the .IC command, .NODESET provides only an initial guess for some initial values. It does not clamp those nodes to the specified voltages. However, by providing an initial guess, .NODESET can be used to “break the tie” in, for instance, a flip-flop, and make it come up in a required state.

If both the .IC command and .NODESET command are present, the .NODESET command is ignored for the bias point calculations (.IC overrides .NODESET).

### **Comments**

For Schematics-based designs, refer to your PSpice user's guide for more information on setting initial conditions.

## .NOISE (Noise Analysis)

**Purpose** The .NOISE command causes a noise analysis of the circuit to be performed.

**General Form** .NOISE V(<node> [,<node>]) <name> [interval value]

**Example** .NOISE V(5) VIN  
.NOISE V(101) VSRC 20  
.NOISE V(4,5) ISRC

**Note** *A noise analysis is performed in conjunction with AC analysis and requires a .AC command.*

V(<node> [,<node>])

An output voltage. It has a form such as V(5), which is the voltage at the output node five, or a form such as V(4,5), which is the output voltage between two nodes four and five.

<name>

The name of an independent voltage or current source where the equivalent input noise is calculated. The <name> is not itself a noise generator, but only a place where the equivalent input noise is calculated.

The noise-generating devices in a circuit are the resistors and the semiconductor devices. For each frequency of the AC analysis, each noise generator's contribution is calculated and propagated to the output nodes. At that point, all the propagated noise values are RMS-summed to calculate the total output noise. The gain from the input source to the output voltage, the total output noise, and the equivalent input noise are all calculated. If

<name> is a voltage source then the input noise units are  
volt/Hertz<sup>1/2</sup>

<name> is a current source then the input noise units are  
amp/Hertz<sup>1/2</sup>

The output noise units are always volt/Hertz<sup>1/2</sup>.

## 1-24 Command

---

*[interval value]*

The interval value is an integer which specifies how often the detailed noise analysis data is written to the output file.

Every nth frequency, where n is the print interval, a detailed table is printed showing the individual contributions of all the circuit's noise generators to the total noise. These values are the noise amounts propagated to the output nodes, not the noise amounts at each generator. If *[interval value]* is not present, then no detailed table is printed.

The detailed table is printed while the analysis is being performed, and does not need a .PRINT command or a .PLOT command. The output noise and equivalent input noise can be printed in the output by using a .PRINT command or a .PLOT command.



## .OP (Bias Point)

**Purpose** The .OP command causes detailed information about the bias point to be printed.

**General Form** .OP

**Example** .OP

This command does not write output to the Probe data file. The bias point is calculated whether or not there is a .OP command. Without the .OP command, the only information about the bias point in the output is a list of the node voltages.

With a .OP command the currents and power dissipation of all the voltage sources are printed.

Using a .OP command can cause the small-signal (linearized) parameters of all the non-linear controlled sources and all the semiconductor devices to be printed in the output file.

The .OP command controls the output for the regular bias point only. The .TRAN command controls the output for the transient analysis bias point.

**Note** *If no other analysis is performed, no Probe data file can be created.*

## .OPTIONS (Analysis Options)

**Purpose** The .OPTIONS command is used to set all the options, limits, and control parameters for the simulator.

**General Form** .OPTIONS [*option name*]\* [ <*option name*>=<*value*> ]\*

**Example** .OPTIONS NOECHO NOMOD DEFL=12u DEFW=8u DEFAD=150p  
DEFAS=150p  
.OPTIONS ACCT RELTOL=.01

The options can be listed in any order. There are two kinds of options: those with values, and those without values. Options without values are flags which are activated by simply listing the option name.

The .OPTIONS command is cumulative. That is, if there are two (or more) of the .OPTIONS command, the effect is the same as if all the options were listed together in one .OPTIONS command. If the same option is listed more than once, only its last value is used.

[Table 1-2](#) lists the flag options. The default for any flag option is “off” or “no” (i.e., the opposite of specifying the option). Flag options affect the output file unless otherwise specified.

**Table 1-2** *Flag Options*

Flag Option	Meaning
ACCT	Summary and accounting information is printed at the end of all the analyses (refer to your PSpice user's guide for further information on ACCT).
EXPAND	Lists devices created by subcircuit expansion and lists contents of the bias point file.
LIBRARY	Lists lines used from library files.
LIST	Lists a summary of the circuit elements (devices).
NODE	Lists a summary of the connections (node table).
NOECHO	Suppresses a listing of the input file(s).
NOMOD	Suppresses listing of model parameters and temperature updated values.
NOPAGE	Suppresses paging and the banner for each major section of output.
OPTS	Lists values for all options.
WIDTH	Same as ".WIDTH OUT=" statement (can be set to either 80 or 132).

The following option has a name as its value.

**Table 1-3** *Option With a Name as its Value*

Option	Meaning	Default
DISTRIBUTION	default distribution for <i>Monte Carlo</i> deviations	UNIFORM

## 1-28 Command

The table below lists the options containing values, with their default values:

Options With Values	Meaning	Units	Default
ABSTOL	Best accuracy of currents.	amp	1pA
CHGTOL	Best accuracy of charges.	coulomb	.01pC
CPTIME	CPU time allowed for this run.	sec	1E6
DEFAD	MOSFET default drain area (AD).	meter <sup>2</sup>	0
DEFAS	MOSFET default source area (AS).	meter <sup>2</sup>	0
DEFL	MOSFET default length (L).	meter	100u
DEFW	MOSFET default width (W).	meter	100u
DIGFREQ	Minimum digital time step is 1/DIGFREQ.	hertz	10GHz
DIGSTRF	Max. resistance for F strength.	ohm	10
DIGSTRD	Max. resistance for D strength.	ohm	100
DIGSTRW	Max. resistance for W strength.	ohm	10K
DIGMNTYMX	Default delay selector: 1=min, 2=typical, 3=max.		2
GMIN	Minimum conductance used for any branch.	ohm <sup>-1</sup>	1E-12
ITL1	DC and bias point “blind” repeating limit.		40
ITL2	DC and bias point “educated guess” repeating limit.		20
ITL4	The limit at any repeating point in transient analysis.		10
ITL5	Total repeating limit for all points for transient analysis (ITL5=0 means ITL5=infinity).		5000
LIMPTS	Maximum points allowed for any print table or plot (LIMPTS=0 means LIMPTS=infinity).		infinite
NUMDGT	Number of digits output in print tables (maximum 8 useful digits).		4
PIVREL	Relative magnitude required for pivot in matrix solution.		1E-3
PIVTOL	Absolute magnitude required for pivot in matrix solution.		1E-13
RELTOL	Relative accuracy of V's and I's.		.001
TNOM	Default nominal temperature (also the temperature at which model parameters are assumed to have been measured).	°C	27
TRTOL	Transient analysis accuracy adjustment.		7.0
VNTOL	Best accuracy of voltages.	volt	1uV

## .PARAM (Parameter)

### Purpose

The .PARAM statement defines the value of a parameter. A parameter name can be used in place of most numeric values in the circuit description. Parameters can be constants, or expressions involving constants.

### General Form

```
.PARAM < <name> = <value> >*
```

```
.PARAM < <name> = { <expression> } >*
```

### Example

```
.PARAM VSUPPLY = 5V
.PARAM VCC = 12V, VEE = -12V
.PARAM BANDWIDTH = {100kHz/3}
.PARAM PI = 3.14159, TWO_PI = {2*3.14159}
```

### <name>

Cannot begin with a number, and it cannot be one of the following predefined parameters, TIME, or .TEXT names.

There are several predefined parameters:

Predefined Parameter	Meaning
TEMP	temperature ( <i>works using digital models only</i> )
VT	thermal voltage ( <i>reserved</i> )
GMIN	shunt conductance for semiconductor <i>p-n</i> junctions

The parameter values must be either constants or expressions.

### <value>

Constants (<value>) do not need "{" and "}".

### <expression>

Must contain only constants.

The .PARAM statements are order independent. They cannot be used inside a subcircuit definition.

Once defined, a parameter can be used in place of most numeric values in the circuit description. For example:

- All model parameters.
- All device parameters, such as AREA, L, NRD, ZO. This includes IC=values, but **not** the transmission-line parameters NL and F.
- All independent voltage and current source (V and I device) parameters **except** for PWL values.
- **Not** the E, F, G, and H device polynomial coefficient values.
- Value on .IC and .NODESET statements.

Parameters **cannot** be used in place of node numbers, nor can the values on an analysis command (e.g., TRAN and AC) be parameterized.

# .PLOT (Plot)

**Purpose** The .PLOT command allows results from DC, AC, noise, and transient analyses to be an output in the form of “line printer” plots in the “out” file.

**General Form** .PLOT *<analysis type>* [output variable]\*  
+ ( [*<lower limit value>* , *<upper limit value>*] )\*

**Example** .PLOT DC V(3) V(2,3) V(R1) I(VIN) I(R2) IB(Q13) VBE(Q13)  
.PLOT AC VM(2) VP(2) VM(3,4) VG(5) VDB(5) IR(D4)  
.PLOT NOISE INOISE ONOISE DB(INOISE) DB(ONOISE)  
.PLOT TRAN V(3) V(2,3) (0,5V) ID(M2) I(VCC) (-50mA,50mA)  
.PLOT TRAN D(QA) D(QB) V(3) V(2,3)

Plots are made by using text characters to draw the plot, therefore, they work using any kind of printer.

*<analysis type>* Can be one of DC, AC, NOISE, or TRAN. Exactly one analysis type must be specified.

*<output variable>* Following the analysis type is a list of the output variables and (possibly) Y axis scales. A maximum of 8 output variables are allowed on one .PLOT command. However, an analysis can have any number of a .PLOT command. See the [Output Variables](#) section on [1-51](#) for the syntax of the output variables.

The range and increment of the X axis is fixed by the analysis being plotted. The Y axis defaults to a “nice” range determined by the ranges of the output variables.

**Note** *The Y axis of frequency response plots (AC) is always logarithmic.*

If the different output variables differ considerably in their output ranges, then the plot is given more than one Y axis using ranges corresponding to the different output variables.

(*<lower limit value>*, *<upper limit value>*)

The range of the Y axis can be set by including the lower and upper limit values at the end of the .PLOT command.

This forces all output variables on the same Y axis to use the specified range. The same form, (*<lower limit value>*, *<upper limit value>*), can also be inserted one or more times in the middle of a set of output variables. Each occurrence defines one Y axis that has the specified range. All the output variables which come between it and the next range to the left in the .PLOT command are put on its corresponding Y axis. In the fourth example, the two voltage outputs go on the Y axis using the range (0,5V) and the two current outputs go on the Y axis using the range (-5mA, 50mA).



# .PRINT (Print)

**Purpose** The .PRINT command allows results from DC, AC, noise, and transient analyses to be an output in the form of tables, referred to as print tables in the output file.

**General Form** .PRINT[/DGTLCGH] <analysis type> [output variable]\*

**Example**

```
.PRINT DC V(3) V(2,3) V(R1) I(VIN) I(R2) IB(Q13) VBE(Q13)
.PRINT AC VM(2) VP(2) VM(3,4) VG(5) VDB(5) IR(6) II(7)
.PRINT NOISE INOISE ONOISE DB(INOISE) DB(ONOISE)
.PRINT TRAN V(3) V(2,3) ID(M2) I(VCC)
.PRINT TRAN D(QA) D(QB) V(3) V(2,3)
.PRINT/DGTLCGH TRAN QA QB RESET
```

**[/DGTLCGH]** This is for digital output variables only. Values are printed for each output variable whenever one of the variables changes.

**<analysis type>** Can be one of DC, AC, NOISE, or TRAN. Exactly one analysis type must be specified for each .PRINT command.

**<output variable>** Following the analysis type is a list of the output variables. There is no limit to the number of output variables: the printout is split up depending on the width of the data columns (set using NUMDGT option) and the output width (set using WIDTH option). See the [Output Variables](#) section on [1-51](#) for the syntax of output variables.

The values of the output variables are printed as a table having each column correspond to one output variable. The number of digits which are printed for analog values can be changed by NUMDGT on the .OPTIONS command.

## **.PROBE** (Probe)

**Purpose** The .PROBE command writes the results from DC, AC, and transient analyses to a data file named PROBE.DAT that is used by the Probe waveform analyzer.

**General Form** .PROBE[/CSDF] [output variable]\*

**Example**

```
.PROBE
.PROBE V(3) V(2,3) V(R1) VM(2) VP(2) I(VIN) I(R2) IB(Q13)
+      VBE(Q13) VDB(5)
.PROBE/CSDF
```

The first example (with no output variables) writes all the node voltages and all the device currents to the data file. The list of device currents written is the same as the device currents allowed as output variables as described in the [Output Variables](#) section on [1-51](#).

The second example writes only those output variables specified to the data file.

The third example creates a data file in a text format using the Common Simulation Data File (CSDF) format, not a binary format. This format is primarily used for transfers between different computer families.

**Comments** Refer to your PSpice user's guide for a description of Probe and for information about using the Probe data file.

**Note** *Unlike the .PRINT command and .PLOT command, there are no analysis names before the output variables. Also, the number of output variables is unlimited, is not restricted to 8.*

## **.SENS** (Sensitivity Analysis)

**Purpose** The .SENS command causes a DC sensitivity analysis to be performed.

**General Form** .SENS <output variable>\*

**Example** .SENS V(9) V(4,3) V(17) I(VCC)

By linearizing the circuit about the bias point, the sensitivities of each of the output variables to all the device values and model parameters is calculated and output data generated. This can easily generate huge amounts of output data.

<output variable> Same format and meaning as in the .PRINT command for DC and transient analyses. However, when <output variable> is a current, it is restricted to be the current through a voltage source.

**Note** *The results of the .SENS command are only available in the output file. They cannot be viewed in Probe.*

## **.STEP** (Parametric Analysis)

**Purpose**                      The .STEP command causes a parametric sweep to be performed for all of the analyses of the circuit.

**General Form**            .STEP LIN <sweep variable name>  
+ <start value> <end value> <increment value>

.STEP [DEC |OCT] <sweep variable name>  
+ <start value> <end value> <points value>

.STEP <sweep variable name> LIST <value>\*

The first General Form is for doing a linear sweep. The second form is for doing a logarithmic sweep. The third form is for using a list of values for the sweep variable.

**Example**                    .STEP VCE 0V 10V .5V  
.STEP LIN I2 5mA -2mA 0.1mA  
.STEP RES RMOD(R) 0.9 1.1 .001  
.STEP DEC NPN QFAST(IS) 1E-18 1E-14 5  
.STEP TEMP LIST 0 20 27 50 80 100

The first three examples, are for doing a linear sweep. The fourth example is for doing a logarithmic sweep. The fifth example is for using a list of values for the sweep variable.

The .STEP command is at the same “level” as the .TEMP command: all of the ordinary analyses (e.g., .DC, .AC, and .TRAN) are performed for each step. Once all the runs have finished, the specified .PRINT table or .PLOT plot for each value of the sweep is an output, just as for the .TEMP or .MC command. (Probe allows nested sweeps to be displayed as a family of curves.)

*Sweep type*

The sweep can be linear, logarithmic, or a list of values. For [*linear sweep type*], the keyword LIN is optional, but either OCT or DEC must be specified for the *<logarithmic sweep type>*. The sweep types are as follows.

Sweep Types	Meaning
LIN	Linear sweep. The sweep variable is swept linearly from the starting to the ending value. The <i>&lt;increment value&gt;</i> is the step size
OCT	Sweep by octaves. The sweep variable is swept logarithmically by octaves. The <i>&lt;points value&gt;</i> is the number of steps per octave.
DEC	Sweep by decades. The sweep variable is swept logarithmically by decades. The <i>&lt;points value&gt;</i> is the number of steps per decade.
LIST	Use a list of values. In this case there are no start and end values. Instead, the numbers that follow the keyword LIST are the values that the sweep variable is set to.

**Note** *The LIST values must be in either ascending or descending order.*

*<sweep variable name>*

The *<sweep variable name>* can be one of the following types.

Sweep Variable Name	Meaning
Source	A name of an independent voltage or current source. During the sweep, the source's voltage or current is set to the sweep value.
Model parameter	A model type and model name followed by a model parameter name in parenthesis. The parameter in the model is set to the sweep value.
Temperature	Use the keyword TEMP for <i>&lt;sweep variable name&gt;</i> . The temperature is set to the sweep value. For each value in the sweep, all the circuit components have their model parameters updated to that temperature.

*<start value>* Can be greater or less than *<end value>*: that is, the sweep can go in either direction.

*<increment value>* and *<points value>*

**Must be greater than zero.**

The .STEP command is similar to the .DC command and immediately raises the question of what happens if both .STEP and .DC try to set the same value. The same question can come up using the *Monte Carlo* analysis. The answer is that **this is not allowed**: no two analyses (.STEP, .TEMP, .MC, .WCASE, and .DC) can try to set the same value. This is flagged as an error during read-in and no analyses are performed.

The .STEP command provides the capability to look at the response of a circuit as a parameter varies. For example, how does the center frequency of a filter shift as a capacitor varies? Using .STEP, that capacitor can be varied, yielding a family of AC waveforms showing the variation. Similar comments apply to looking at, for example, propagation delay in transient analysis.

## .SUBCKT, .ENDS (Subcircuit and End Subcircuit)

### Purpose

The .SUBCKT definition statement starts the definition of a subcircuit by specifying its name, the number and order of its terminals, and the names and default parameters which control its behavior. Subcircuits are instantiated by the X devices in [Chapter 2, Analog Devices](#). The .ENDS command marks the end of a subcircuit definition.

### General Form

```
.SUBCKT <name> [node]* [PARAMS: < <name> = <value> >* ]
```

```
...  
.ENDS
```

### Example

```
.SUBCKT OPAMP 1 2 101 102 17
```

```
...  
.ENDS
```

```
.SUBCKT FILTER INPUT, OUTPUT PARAMS: CENTER=100kHz,  
+ WIDTH=10kHz
```

```
...  
.ENDS
```

The subcircuit definition is ended using a .ENDS command. All of the netlist between .SUBCKT and .ENDS is included in the definition. Whenever the subcircuit is used, by an X device, all of the netlist in the definition replaces the X device.

### <name>

The name is used by an X device to reference the subcircuit. It must start with a letter.

### [node]\*

An optional list of nodes (pins). This is optional because it is possible to specify a subcircuit that has no interface nodes.

There must be the same number of nodes in the subcircuit calling statements as in its definition. When the subcircuit is called, the actual nodes (the ones in the calling statement) replace the argument nodes (the ones in the defining statement).

### Note

*Do not use 0 (“zero”) in this node list: that is reserved for global “ground” node.*

The keyword `PARAMS`: allows values to be passed into subcircuits as arguments and used in expressions inside the subcircuit.

Subcircuits can be nested. That is, an X device can appear between a `.SUBCKT` and a `.ENDS` command.

However, subcircuit definitions **cannot be nested**. That is, a `.SUBCKT` statement cannot appear in the statements between a `.SUBCKT` and a `.ENDS`.

Subcircuit definitions should contain only device statements (statements without a leading `“.”`) and possibly `.MODEL` statements. Models defined within a subcircuit definition are **available only within the subcircuit definition** in which they appear. Also, if a `.MODEL` statement appears in the main circuit, it is available in the main circuit and all subcircuits.

Node, device, and model names are local to the subcircuit in which they are defined. It is acceptable to use a name in a subcircuit which has already been used in the main circuit. When the subcircuit is expanded, all its names are prefixed using the subcircuit instance name: for example, `“Q13”` becomes `“X3.Q13”` and node `“5”` becomes `“X3.5”` after expansion. After expansion all names are unique.



## .TEMP (Temperature)

**Purpose** The .TEMP statement sets the temperature at which all analyses are done.

**General Form** .TEMP <temperature value>\*

**Example** .TEMP 125  
.TEMP 0 27 125

The temperatures are in degrees Centigrade. If more than one temperature is given, then all analyses are performed for each temperature.

It is assumed that the model parameters were measured or derived at the nominal temperature, TNOM (27°C by default). See the [.OPTIONS \(Analysis Options\)](#) command (page [1-26](#)) for setting TNOM.

## .TF (Transfer)

**Purpose** The .TF statement causes the small-signal DC gain to be calculated by linearizing the circuit around the bias point.

**General Form** .TF *<output variable>* *<input source name>*

**Example** .TF V(5) VIN  
.TF I(VDRIV) ICNTRL

The gain from *<input source name>* to *<output variable>* and the input and output resistances are evaluated and written to the output file. This output does not require a .PRINT, .PLOT, or .PROBE statement.

*<output variable>* This has the same format and meaning as in the .PRINT statement. When *<output variable>* is a current, it is restricted to be the current through a voltage source.

**Note** *The results of the .TF command are only available in the output file. They cannot be viewed in Probe.*

## .TRAN (Transient Analysis)

**Purpose** The .TRAN statement causes a transient analysis to be performed on the circuit.

**General Form** .TRAN[/OP] *<print step value>* *<final time value>*  
+ [no-print value [step ceiling value]] [UIC]

**Example** .TRAN 1ns 100ns  
.TRAN/OP 1ns 100ns 20ns UIC  
.TRAN 1ns 100ns 0ns .1ns

Prior to doing the transient analysis, *PSpice* computes a bias point for the circuit separate from the regular bias point. This is performed because the independent sources can have different values at the start of a transient analysis than their DC value.

The transient analysis uses an internal time step which is adjusted as the analysis proceeds. Over intervals where there is little activity, the internal time step is increased and during busy intervals it is decreased.

The default ceiling on the internal time step is *<final time value>/50* (**it is not** *<print step value>*).

[/OP] Normally, only the node voltages are printed for the transient analysis bias point. However, the “/OP” suffix (on .TRAN) has the same detailed printing of the bias point that the .OP command has for the regular bias point.

*<print step value>* The time interval used for printing, plotting (.PRINT or .PLOT) the results of the transient analysis. Since the results are computed at different times than they are printed, a 2nd-order polynomial interpolation is used to obtain the printed values.

*<final time value>* The transient analysis calculates the circuit’s behavior over time, starting at TIME=0 and going to the *<final time value>*.

## 1-44 Commands

---

The transient analysis always starts at TIME=0. However, it is possible to suppress output of a portion of the analysis.

[*no-print value*]      The amount of time from TIME=0 which is not printed, plotted, or given to *Probe*.

[*step ceiling value*]      Overrides the default ceiling on the internal times step with a lower value.

UIC      When the UIC is put at the end of the .TRAN statement, the calculation of the bias point is skipped. This option means that the bias conditions are fully determined by the IC= specifications for capacitors and inductors.

The .PRINT, .PLOT, .FOUR, or .PROBE statements must be used to get the results of the transient analysis.

## .WCASE (Sensitivity/Worst-Case Analysis)

**Purpose** The .WCASE statement causes a sensitivity and worst-case analysis of the circuit to be performed.

**General Form** .WCASE <analysis> <output variable> <function> [option]\*

**Example**

```
.WCASE TRAN V(5) YMAX
.WCASE DC IC(Q7) YMAX VARY DEV
.WCASE AC VP(13,5) YMAX DEVICES RQ OUTPUT ALL
```

Multiple runs of the selected analysis (DC, AC, or transient) are performed while parameters are varied. Unlike .MC, .WCASE varies only one parameter per run. This allows *PSpice* to calculate the sensitivity of the output waveform to each parameter. Once all the sensitivities are known, one final run is performed using all parameters varied so as to produce the worst-case waveform. The sensitivity and worst-case runs are performed using variations on model parameters as specified by the DEV and LOT tolerances on each .MODEL parameter (see [.MODEL \(Model\)](#) command section for details on the DEV and LOT tolerances). Other specifications on the .WCASE command control the output generated by the Monte Carlo analysis.

**Note** *Either .MC or .WCASE can be run, but not both in the same circuit.*

<analysis>	Only one of DC, AC, or TRAN must be specified for <analysis>. This analysis is repeated in subsequent passes of the worst-case analysis. All requested analyses are performed during the nominal pass. Only the selected analysis is performed during subsequent passes.
<output variable>	Identical in format to that of a .PRINT output variable; see <a href="#">1-51</a> for details.
<function>	Specifies the operation to be performed on the values of the <output variable> to reduce these to a single value.

This value is the basis for the comparisons between the nominal and subsequent runs. The *<function>* must be one of the following.

Function	Meaning
YMAX	Find the absolute value of the <b>greatest difference</b> in each waveform from the nominal run.
MAX	Find the <b>maximum value</b> of each waveform.
MIN	Find the <b>minimum value</b> of each waveform.
RISE_EDGE ( <i>&lt;value&gt;</i> )	Find the <b>first occurrence</b> of the waveform crossing <b>above</b> the threshold <i>&lt;value&gt;</i> . The waveform must have one or more points at or below <i>&lt;value&gt;</i> followed by one above; the output value listed is where the waveform increases above <i>&lt;value&gt;</i> .
FALL_EDGE ( <i>&lt;value&gt;</i> )	Find the <b>first occurrence</b> of the waveform crossing <b>below</b> the threshold <i>&lt;value&gt;</i> . The waveform must have one or more points at or above <i>&lt;value&gt;</i> followed by one below; the output value listed is where the waveform decreases below <i>&lt;value&gt;</i> .

**Note** *The <function> and all [option]s do not affect the Probe data saved from the simulation. They are only applicable to the output file.*

[option]\* Could have none or one or more of the following.

[option]	Meaning
OUTPUT ALL	Prints output from the sensitivity runs, after the nominal (first) run. The output from any run is governed by the .PRINT, .PLOT, and .PROBE command in the file. If OUTPUT ALL is omitted, then only the nominal and worst-case runs produce output. OUTPUT ALL ensures that all sensitivity information is saved for Probe.
RANGE * ( <i>&lt;low value&gt;</i> , <i>&lt;high value&gt;</i> )	Commands the <i>Monte Carlo</i> analysis to sweep only the parameters which are in the (inclusive) range for computation of the <i>&lt;function&gt;</i> . An “*” can be used in place of a <i>&lt;value&gt;</i> to show “for all values.” For example see the next two rows.
YMAX RANGE(*,.5)	YMAX is evaluated for values of the sweep variable (e.g., time, and frequency) of .5 or less.
MAX RANGE(-1,*)	The maximum of the output variable is found for values of the sweep variable of -1 or more.

[option]	Meaning
HI or LOW	Specify the direction which <i>&lt;function&gt;</i> should move for the worst-case run is to go (relative to the nominal). If <i>&lt;function&gt;</i> is YMAX or MAX, the default is HI, otherwise the default is LOW.
VARY DEV VARY LOT VARY BOTH	By default, any device which has a model parameter specifying either a DEV tolerance or a LOT tolerance is included in the analysis. The analysis can be limited to only those devices which have DEV or LOT tolerances by specifying the appropriate option. The default is VARY BOTH.
BY RELTOL BY <i>&lt;value&gt;</i>	The model parameters are (by default) varied by RELTOL (set on the .OPTIONS statement). You may use some other number by specifying <i>&lt;value&gt;</i> .
DEVICES ( <i>list of device types</i> )	By default, all devices are included in the sensitivity and worst-case analyses. The devices considered can be limited by listing the device types after the keyword DEVICES. Do not use any spaces or tabs in the devices list. For example, to only perform the analysis on resistors and MOSFETs, enter:  DEVICES RM

---

\* If RANGE is omitted, then *<function>* is evaluated over the whole sweep range. This is equivalent to RANGE(\*, \*).

## **.WIDTH**

**Purpose**

The .WIDTH statement sets the width of the output.

**General Form**

.WIDTH OUT= *<value>*

**Example**

.WIDTH OUT=80  
.WIDTH OUT=132

*<value>*

It is in columns and must be either 80 (the default) or 132.



## \* (Comment)

**Purpose** A statement beginning with “\*” is a comment line and has no effect.

**General Form** \* [*any text*]

**Example** \* This is an example of a comment

The use of comment statements throughout the input is recommended. It is good practice to place a comment just before a subcircuit definition to identify the nodes, for example

```
*
      +IN  -IN  V+  V-  +OUT  -OUT .SUBCKT
OPAMP      100 101 1   2    200   201
```

or to identify major blocks of circuitry.

## **;** (In-line Comment)

**Purpose** A “;” is treated as the end of a line.

**General Form** *circuit file text ;[any text]*

The simulator moves on to the next line in the circuit file. The text on the line after the “;” is a comment and has no effect. The use of comments throughout the input is recommended. This type of comment can also replace comment lines, which must start with “\*” in the first column.

Trailing in-line comments that extend to more than one line can use a semicolon to mark the beginning of the subsequent comment lines, as shown in the example.

**Example**

```
R13  6  8  10 ; R13 is a
                ; feedback resistor
C3   15  0  .1U ; decouple supply
```

# OUTPUT VARIABLES

This section describes the types of output variables allowed in a .PRINT, .PLOT, and .PROBE command. Each .PRINT or .PLOT can have up to 8 output variables. This format is similar to that used when calling up waveforms while running Probe. See the tables below for a description of the possible output variables.

## DC Sweep and Transient Analysis

For DC sweep and transient analysis, these are the available output variables:

General Form	Meaning of Output Variable
D(<name>)	digital value of <name> (a digital node) *
I(<name>)	current through a two terminal device
Ix(<name>)	current into a terminal of a three or four terminal device (x is one of "B", "D", "G", or "S")
Iz(<name>)	current into one end of a transmission line (z is either "A" or "B")
V(<node>)	voltage at a node
V(<+ node>, <- node>)	voltage between two nodes
V(<name>)	voltage across a two-terminal device
Vx(<name>)	voltage at a non-grounded terminal of a device (see Ix)
Vz(<name>)	voltage at one end of a transmission line (z is either "A" or "B")
Vxy(<name>)	voltage across two terminals of a three or four terminal device type

\*These values are available for transient and DC analysis only. For the .PRINT/ DGTLCGH statement the "D( )" is optional.

<b>Examples</b>	<b>Meaning of Output Variable</b>
D(QA)	the value of digital node QA
I(D5)	current through diode D5
IG(J10)	current into gate of J10
V(3)	voltage between node three and ground
V(3,2)	voltage between nodes three and two
V(R1)	voltage across resistor R1
VA(T2)	voltage at port A of T2
VB(Q3)	voltage between base of transistor Q3 and ground
VGS(M13)	gate-source voltage of M13

For the V(<name>) and I(<name>) forms, where <name> must be the name of a two-terminal device, the devices are:

<b>Character ID</b>	<b>Two-Terminal Device</b>
C	capacitor
D	diode
E	voltage-controlled voltage source
F	current-controlled current source
G	voltage-controlled current source
H	current-controlled voltage source)
I	independent current source
L	inductor
R	resistor
V	independent voltage source

For the  $V_x(<name>)$ ,  $V_{xy}(<name>)$ , and  $I_x(<name>)$  forms, where  $<name>$  must be the name of a three or four-terminal device and x and y must each be a terminal abbreviation, the devices and the terminals are:

Three & Four-Terminal Device Type	Terminal Abbreviation
B (GaAs MESFET)	D (drain)
	G (gate)
	S (source)
J (Junction FET)	D (drain)
	G (gate)
	S (source)
M (MOSFET)	D (drain)
	G (gate)
	S (source)
Q (Bipolar transistor)	B (bulk, substrate)
	C (collector)
	B (base)
	E (emitter)
	S (substrate)

---

For the  $V_z(<name>)$  and  $I_z(<name>)$  forms,  $<name>$  must be the name of a transmission line (T device) and z must be "A" or "B". "A" means port A (the first two nodes) and "B" means port B (last two nodes).

### AC Analysis

For AC analysis, the output variables listed in the preceding section are augmented by adding a suffix. These are the available suffixes:

Suffix	Meaning of Output Variables for AC Analysis
none	magnitude
DB	magnitude in decibels
G	group delay (-dPHASE/dFREQUENCY)
I	imaginary part
M	magnitude
P	phase in degrees
R	real part

Examples	Meaning of Output Variables for AC Analysis
II(R13)	imaginary part of current through R13
IGG(M3)	group delay of M3's gate current
IR(VIN)	real part of I through VIN
IAG(T2)	group delay of current at port A of T2
V(2,3)	magnitude of complex voltage across nodes 2 & 3
VDB(R1)	db magnitude of V across R1
VBEP(Q3)	phase of base-emitter V at Q3
VM(2)	magnitude of V at node 2

Not as many type of current output are available as for DC and transient analyses.

Specifically, **currents through these devices are not available:**

- E (voltage-controlled voltage source)
- F (current-controlled current source)
- G (voltage-controlled current source)
- H (current-controlled voltage source)

For these devices, a zero-valued voltage source must be put in series with the device (or terminal) of interest. Then, the current through this voltage source can be printed or plotted .

### Noise Analysis

For noise analysis, the output variables are predefined as follows.

Output Variable	Meaning of Output Variables for Noise Analysis
INOISE	Total RMS summed noise at input node
ONoise INoise	equivalent at output node
DB(INoise)	INOISE in decibels
DB(ONoise)	ONoise in decibels

**Note** *The noise from any one device cannot be .PRINTed or .PLOTed. However, the print interval on the .NOISE command can be used to output this information.*

### Comments

Refer to your PSpice user's guide for more information on the use of text files in Probe. You can also consult Probe Help.