.MODIF Parameter Modification

Syntax

.MODIF <<*param1* < + | - | * | *initval* > = *rhs* > <*param2_spfc* ... >
\+ <LOOP <=*nreps*> > <STOP *lname* | *lval* LE *rname* | *rval*>
\+ <PROFF> <PRTBL> <PRMC>
\+ <MODIF *parset2* > <MODIF *parset3*> ...
\+ <AUTOSTOP *outvar1*=(RISE | FALL=*numrep*) *pname1*+| - | * | / *num1* <*cond2*> . . .>
\+ <MODIF DATA *rhs1* *rhs2* . . . >>

This statement allows the user to investigate a circuit for some sets of parameters as they are modified over a user-specified range of values. It simultaneously modifies all of the parameters in a set, and then performs all analyses on the input deck for this set. SmartSpice will terminate the parameter modification process for a set of parameters when one of the user-defined stop conditions, specified in this set, is satisfied.

param1, param2	Names of parameters. Each is one of the following types: a name of a device parameter; a name of a model parameter; TEMP (temperature); GMIN (model and computational parameter). RELTOL (relative error tolerance). ABSTOL (absolute current error tolerance) VNTOL (absolute voltage error tolerance) It is also possible to specify a parameter for all devices of the same type. The syntax for this is: `ALL@`*devtype* <*parname*> SmartSpice will modify the parameter *parname* of all devices of the type *devtype*.
+, -, *, /	The arithmetic operators.
rhs	Parameter value, increment, or multiplier. It is one of the following types: <(*initval*)>*val*; For Monte Carlo analysis: AUNIF (*nomval absvar* <*mult*>); UNIF (*nomval relvar* <*mult*>); AGAUSS (*nomval absvar sigma* <*mult*>); GAUSS (*nomval relvar sigma* <*mult*>); ALIMIT (*nomval absvar*); LIMIT (*nomval relvar*); AWEIBULL (*nomval absvar* <*csign*> *cval rlim*) WEIBULL (*nomval relvar* <*csign*> *cval rlim*);
initval	The initial value of the parameter *param1*.

val	The number for *param1* modification. May be a number or the name of a measure calculated in analysis. This name must be specified in the .MEASURE statement.
AUNIF	Uniform distribution function with absolute variation.
UNIF	Uniform distribution function with relative variation.
AGAUSS	Gaussian distribution function with absolute variation.
GAUSS	Gaussian distribution function with relative variation.
ALIMIT	Random limit distribution function with absolute variation, +/- *absvar* is added to nomval based on whether the random outcome of a -1 to 1 distribution is greater or less than 0.
LIMIT	Random limit distribution function with relative variation.
AWEIBULL	Weibull distribution function with absolute variation.
WEIBULL	Weibull distribution function with relative variation.
nomval	Nominal value for Monte Carlo analysis.
absvar	The absolute variation. The AUNIF, AGAUSS, and AWEIBULL vary the *nomval* by +/- *absvar*.
relvar	The relative variation. The UNIF, GAUSS, and WEIBULL vary the *nomval* by +/- (*nomval* · *relvar*).
mult	The multiplier will repeat function calculation *mult* times, saving only the largest deviation. The default value for *mult* is 1.
sigma	The parameter for Gaussian distribution. The effective standard deviation of a random sample will be 3·*absvar*/sigma.
csign	Plus or minus. This sign specifies the orientation of the non-symmetrical Weibull distribution.
cval	The parameter c of the standard Weibull density function $p_x(x) = cx^{c-1} \exp(-x^c)$, $(x>0)$.
rlim	The parameter for Weibull distribution.
param2_spfc	The specifications for *param2*. These specifications have the same style as the specifications for *param1*.
LOOP	The keyword, followed by the desired number of repetitions.
nreps	The number of steps to be repeated in the parameter set. SmartSpice stops the parameter modification process for this set when the number of repetitions reaches *nreps*. The default value for *nreps* is 5.
STOP	The keyword, followed by a stop condition.
lname, rname	The names of measures calculated in analyses.
lval, rval	The number for the stop condition. The .MODIF statement may contain only one of them (*lval* or *rval*).
LE	The mandatory relational keyword between the left and right sides of the stop condition. SmartSpice terminates the

	parameter modification process if: *lname ≤ rval,* *or lname ≤ rname,* *or lval ≤ rname.*
PROFF	Suppresses on-line printing of measures specified and calculated by means of .MEASURE statements for the current set of parameters.
PRTBL	Causes SmartSpice to print the final table of all parameters and measures, calculated for the current set of parameters.
PRMC	Causes SmartSpice to print Mean, Variance, Sigma, and Average Deviation for each measure calculated during Monte Carlo analysis.
MODIF	This keyword, followed by a set of parameters, begins the next set of parameters. This keyword is not preceded by a period (.).
MODIF *parset2*	The specifications for the second set of parameters.
AUTOSTOP	Keyword followed by the list of stop conditions for basic output variables. SmartSpice will stop the transient analysis when all AUTOSTOP conditions are satisfied. The AUTOSTOP specification is placed at the end of the .MODIF statement.
outvar1	Name of a basic output variable.
numrep	Number of rises or falls.
pname1	Name of a device or a model parameter.
num1	Number for stop condition.
cond2	Stop condition for the basic variable *outvar2*.
DATA	Keyword followed by a number of *rhs* values. This keyword can be used to repeat the previous measurement of *rhs* on multiple sets of the same parameters with new values. The value of *rhs* can only be a number preceded by optional initial conditions.

The .MODIF statement cannot be run with the .ST statement in the same input deck. .MODIF, .TEMP and .DC statements must not attempt to set the same parameters.

NOTE: The parameter modification process is cumulative. Each set of parameters begins with the parameter values generated by the previous iteration.

Any device and model parameter may be specified in the .MODIF statement in one of two ways: using the full-path name, or using parameter labels.

The following forms can be used to specify parameters using the full-path name:

device (parname)= rhs

devname	User-specified device name.
parname	Name of a parameter. Must be in parentheses.
rhs	Value of the parameter.

or

modname (parname)= rhs

modname	User-specified model name.
parname	Name of a parameter. Must be in parentheses.
rhs	Value of the parameter.

A device or model parameter can also be specified by a parameter label defined in the .PARAM statement and referred to in the model or device specification

Each of these methods offers certain advantages.

The advantages of the full-path name are:

- It allows the use of an input netlist without any changes in the circuit description
- It provides access to any model or device when subcircuits are used

Transient analysis voltage and current source parameters have notations but do not have names. For example, V1, V2, *td*, *tr*, *tf*, *pw*, and *per* are notations for initial value, pulsed value, delay time, rise time, fall time, pulse width, and period of a pulse waveform. These notations may not be used in full-path names. The names PAR1, PAR2, ..., PAR7 can be used in place of these notations. For example, the name `vin(PAR3)` can have any of the following meanings:

- Rise delay time for an exponential waveform
- Delay time for a pulse waveform
- Carrier frequency for a single-frequency waveform
- Frequency for a sinusoidal waveform

The advantages of the parameter label method are:

- The same value can be set or updated for different models and devices
- Arbitrary dependencies between different device and model parameters can be introduced by parameter expressions in the .PARAM statement
- Labels can be used in the .MEASURE statement as a right hand side (rhs) value for most parameters

If the value of a model or device parameter is loaded by means of a parameter label defined in the .PARAM statement, then the full-path name of this parameter may not be used in a .MODIF statement.

A parameter label may not be used in a .MODIF statement if:

- The label is not defined in a global .PARAM statement
- The label is defined as a function of another parameter label
- The label is referred to in a Polynomial, Piecewise Linear, or Behavioral type controlled voltage (E and H) or current (F and G) source

Examples

The following examples are consistent with Example 2 in Chapter 13.

```
.MODIF RC1(RES)=2K CLOAD(CAP)=3PF
```

In this example, SmartSpice first sets the parameter RES of the resistor RC1 to a value of 2K, and the parameter CAP of the capacitor CLOAD to a value of 3PF, and then performs all of the analyses of the input deck.

```
.MODIF LOOP=10 RC1(RES)+=0.1K CLOAD(CAP)+=0.2PF
```

In this example, SmartSpice simultaneously modifies the parameters RES and CAP ten times. The increments are 0.1K and 0.2PF, respectively. On the first iteration SmartSpice adds these increments to the original parameter values of the input file.

```
.MODIF RC1(RES)=2K CLOAD(CAP)=0.5PF
+ MODIF LOOP=9 STOP DEL_V3_V1 LE 1.3NS RC1(RES)*=0.9
+ CLOAD(CAP)*=0.95
```

In this example, SmartSpice performs 9 or fewer iterations. On the first iteration, it sets the parameters RES to a value of 2K, and CAP to a value of 0.5PF. On each of the following iterations, SmartSpice multiplies the parameter RES by the coefficient 0.9, and the parameter CAP by the coefficient 0.95. It will terminate the parameter modification process when DEL_V3_V1 ≤ 1.3NS.

```
.MODIF LOOP=9 STOP DEL_V3_V1 LE 1.3NS
+ RC1(RES)*=(2K)0.9 CLOAD(CAP)*=(0.5PF)0.95 PRTBL
```

This example is similar to the previous one. After the parameter modification process has finished, SmartSpice prints the final table of parameters and measures.

```
.MODIF MODIF STOP 2.6 LE MAX_TR_V3 RBIAS(RES)*=1.2
```

In this example, SmartSpice performs 6 or fewer iterations. On the first iteration it performs all analyses of an input file without changing parameters. On each following iterations, SmartSpice multiplies the parameter RES of the resistor RBIAS by the coefficient 1.2. It will terminate the parameter modification process when MAX_TR_V3 ≥ 2.6 .

```
.MODIF LOOP=30 RC1(RES)=UNIF(2K 0.1)
+ CLOAD(CAP)=AGAUSS(1PF 0.15PF 3 10)PRMC
```

In this example, SmartSpice performs 30 iterations of the Monte Carlo analysis using relative uniform distribution for the parameter RES of the resistor RC1 and absolute bimodal gaussian distribution variation +/- 0.15pf at 3 sigma for the parameter CAP. After the analysis is finished, SmartSpice prints the final table with Mean, Variance, Sigma and Average Deviation for each calculated measure.

```
.MODIF ALL@R(RES)*=1.2 QNL(RB)=120
```

In this example, for all resistors, SmartSpice multiplies the parameter RES by coefficient 1.2, and sets the parameter RB of the transistor model QNL to a value of 120.

```
.MODIF VIN(PAR4)=1NS VIN(PAR5)=1NS
```

In this example, SmartSpice sets the rise and fall times of the pulse waveform of voltage source VIN to a value of 1NS.

```
.MODIF LOOP=10 TEMP=   -50 VCC(DC)+=   (10)0.2
+ MODIF DATA           27              (10)0.2
+                      100             (10)0.2
```

In this example, the .MODIF statement is equivalent to:

```
.MODIF   LOOP=10 TEMP=-50 VCC(DC)+=(10)0.2
+ MODIF LOOP=10 TEMP=27   VCC(DC)+=(10)0.2
+ MODIF LOOP=10 TEMP=100 VCC(DC)+=(10)0.2
```

In the following input file, both the parameter labels and the full-path parameter names are used. The parameter labels defined in the .PARAM statement are:

- *vccdc* - referred to in the voltage source *vcc* specification, in the first .MEASURE statement and in the .MODIF statement
- *trtf* - referred to in the pulse waveform rise and fall time specification
- *wp* - referred to in the transistor m1 width specification
- The full-path name m2(w) is specified in the .MODIF statement

```
Parametric analysis
***** Circuit description
vcc vss 0  DC vccDC
vin inp 0  PULSE(0 3 0 trtf trtf  10ns 40ns)
m1 2  inp  vss  vss  pm  w=wp    l=1.6u
m2 2  inp  0    0    nm  w=30u   l=2.0u
cout  2  0  50ff
.MODEL  pm  PMOS ( level=3 tox=.02e-6)
.MODEL  nm  NMOS ( level=3 tox=.02e-6)
***** Analysis statement
```

```
.TRAN 0.1ns 25ns
***** Measure statements
.MEASURE TRAN delrise DELAY v(inp) RISE=1 VAL=1.5
+       TARG=v(2) FALL=1 VAL='0.5*vccDC'
.MEASURE TRAN maxv2 MAX  v(2)
.MEASURE TRAN delfall DELAY v(inp) FALL=1 VAL=1.5
+       TARG=v(2) RISE=1 VAL='0.5*maxv2'
***** Parameter labels
.PARAM vccDC= 5V wp=4.9u  trtf=1ns
***** Parametric analysis  specification
.MODIF proff  prtbl m2(w)=32.6u   LOOP=6 vccDC+=(4.5)0.1v
+MODIF proff  prtbl      TEMP=75  LOOP=6 vccDC+=(4.5)0.1v
.END
```

The .MODIF statement performs parametric analysis for two sets of parameters. For the first set, the width `w` of the transistor `m2` is set to a value of 32.6 microns, and then parameter `vccDC` is swept for six iterations from 4.5 to 5.

For the second set, the temperature is set to a value of 75 deg C, and the parameter `vccDC` is swept. The flag `PROFF` suppresses on-line printing of the .MEASURE statement results. The flag `PRTBL` causes SmartSpice to print the final table of results.

On each iteration, SmartSpice performs a transient analysis and calculates measures. The parameter `VAL` of the first .MEASURE statement is defined as a function of the power supply voltage `vccDC`. The .MEASURE statement that calculates the delay `delfall` uses the previously calculated .MEASURE statement `maxv2`. The results of parametric analysis for the first set of parameters are shown below:

```
TABLE OF RESULTS FOR SET #1
PARAMETERS and MEASUREMENTS
 m2(w)         vccdc        delrise       maxv2        delfall
--------   ----------   ----------   ----------   ----------
3.260e-05   4.500e+00   1.691e-11   4.507e+00   1.859e-10
3.260e-05   4.600e+00   2.946e-11   4.607e+00   1.809e-10
3.260e-05   4.700e+00   3.916e-11   4.707e+00   1.780e-10
3.260e-05   4.800e+00   4.750e-11   4.807e+00   1.710e-10
3.260e-05   4.900e+00   5.474e-11   4.907e+00   1.437e-10
3.260e-05   5.000e+00   5.972e-11   5.007e+00   1.374e-10
```